

---

# **Klovve**

***Release 0.9***

**unknown**

**Feb 16, 2025**



# CONTENTS

<b>1</b>	<b>License</b>	<b>3</b>
<b>2</b>	<b>Up-To-Date?</b>	<b>5</b>
<b>3</b>	<b>User Manual</b>	<b>7</b>
3.1	Getting Started . . . . .	7
<b>4</b>	<b>API Reference</b>	<b>9</b>
4.1	klovve package . . . . .	9
	<b>Python Module Index</b>	<b>109</b>
	<b>Index</b>	<b>111</b>



Klovve is a UI lib.



## **LICENSE**

Klovve is distributed under the terms of the AGPL 3 license. This also affects all included files without a license header (non-source files like images), unless they are explicitly mentioned as third-party content. Read the ‘Dependencies’ section for included third-party stuff.





## **UP-TO-DATE?**

Are you currently reading from another source than the homepage? If you are in doubt whether your package is up-to-date, you should visit the project homepage and check that. You are currently reading the documentation for version 0.9.



## USER MANUAL

### 3.1 Getting Started

#### 3.1.1 Foo

This section is about getting ready.



## API REFERENCE

### 4.1 klovve package

Klovve.

Everything needed for a simple Klovve application is in that module. There are more things in various submodules, but you will rarely, if ever, need them.

#### 4.1.1 Subpackages

**klovve.+tests namespace**

**Submodules**

**klovve.+tests.data module**

**klovve.+tests.view module**

**klovve.app package**

*Application* type and some application level utilities.

**class** klovve.app.**Application**(\*, *driver*, *view\_spec*)

Bases: object

A Klovve application. Each instance represents one (potential) running application instance, i.e. one window that actually exists somewhere on the screen as soon as it is started.

#### Parameters

- **driver** (klovve.drivers.Driver) – The driver to use.
- **view\_spec** (Callable[[klovve.view.tree.ViewFactory], klovve.view.tree.ViewTreeNode]) – The view specification for the application's *window*.

**\_\_running\_apps** = []

**\_\_running\_apps\_changed\_event**()

#### Return type

*Event*

**\_\_running\_apps\_changed\_event\_\_internal = None**

**property is\_running: bool**

Whether this application is currently running.

**run()**

Run the application and wait until it is closed.

This is only allowed to be called in the main thread, and when no *asyncio* event loop is currently running in that thread. In particular, it is not allowed to be called when another Klovve application is already running in that process. For running multiple applications, use *start()* and maybe *sleep\_until\_stopped()* instead.

**Return type**

None

**async sleep\_until\_stopped()**

Wait until the application is stopped. Also return if it was not even started yet.

Must be called from inside the Klovve mainloop.

**Return type**

None

**async start()**

Start the application.

Must be called from inside the Klovve mainloop. Returns directly after starting (non-blocking). Afterwards, until it gets closed, *sleep\_until\_stopped()* will sleep and *is\_running* will be *True*.

**Return type**

None

**exception klovve.app.ApplicationUnavailableError**

Bases: *RuntimeError*

Error raised when the application to be started cannot run with any *klovve.drivers.Driver* that is available on this system.

**class klovve.app.MainloopObjectProxy(obj)**

Bases: *object*

Simple proxy object that is wrapped around another one, ensuring that all interactions with it take place inside the mainloop.

**exception klovve.app.ProgramError**

Bases: *Exception*

Internal program errors.

**klovve.app.\_TViewSpec**

View specification type (for type decoration purposes). A view specification defines the application window (what it contains and how it behaves) by a function that gets a view factory as input and returns the view tree node of the application window.

alias of *Callable[[klovve.view.tree.ViewFactory], klovve.view.tree.ViewTreeNode]*

**klovve.app.call\_maybe\_async\_func(func, \*args, \*\*kwargs)**

Start running the input function (may be async or normal) in an async way on the Klovve mainloop.

**Parameters**

- **func** (*Callable*) – The function to call. May be an async function, or a usual, non-async one.
- **args** – Arguments to pass to the function.
- **kwargs** – Keyword arguments to pass to the function.

**Return type***Awaitable*`klovve.app.create_app(view_spec)`

Create and return a Klovve application, described by a view specification, but not start it yet.

**Parameters****view\_spec** (*Callable*[[*ViewFactory*], *ViewTreeNode*]) – The application's view specification.**Return type***Application*`klovve.app.in_mainloop(func)`

Function decorator that dispatches function execution to the mainloop.

**Parameters****func** (*Callable*) – The function to call inside the mainloop.**Return type***Callable*`klovve.app.mainloop()`

Return the mainloop.

**Return type***AbstractEventLoop*`klovve.app.verify_correct_thread()`

Raise an exception if called from another thread than the Klovve thread (i.e. the process main thread). Otherwise, do nothing.

**Return type**

None

## Submodules

### klovve.app.context module

Contexts are objects with additional info and/or functionality that is available to the application developer for particular situations (e.g. user interactions).

**class** `klovve.app.context.ActionContext`(*pieces*)Bases: `Generic[_TGo]`, `ABC`

Action contexts provide a way to interact with the user by means of dialogs in the execution of an action, e.g. after the user has clicked on a button. There is also some more functionality for action execution.

`_abc_impl = <_abc._abc_data object>`

**abstract async \_create\_dialog**(*view*, *done\_future*)

TODO Create a dialog with the given view content, and wait until *done\_future* is done. Implemented by subclasses, but not directly used from outside.

**Parameters**

- **view** ([View](#)) – The view to show in a dialog.
- **done\_future** ([Future](#)) – The future to wait for before closing the dialog again.

**async dialog**(*view*)

Show a dialog to the user and return the answer that the user gave.

The model of this dialog must have a boolean *is\_answered* property that is True once the user has chosen to close and continue (maybe via an OK button, or however the view shall behave), and an *answer* property that contains the value returned by this function then.

**Parameters**

**view** ([ViewTreeNode](#)) – The view node that describes the content to show in the dialog. You usually create it with a [klovve.view.tree.ViewFactory](#).

**Return type**

*Any*

**error\_message\_for\_exceptions**(\**exception\_types*)

If exceptions (of particular types) get raised inside the with-block, a dialog with the exception message will be shown to the user.

**Parameters**

**exception\_types** – The types to catch.

## klovve.data package

Klovve data, models, etc.

## Submodules

### klovve.data.deps module

Dependency tracking for model properties.

Klovve applications typically do not need anything from here.

**class** `klovve.data.deps.DependencySubject`

Bases: `ABC`

Base class for objects that can attend on dependency tracking.

For details, read [klovve.data.model.PropertyBase](#) as well.

**\_abc\_impl** = `<_abc._abc_data object>`

**abstract \_add\_changed\_handler**(*prop*, *func*)

Add a changed handler for a particular property, as a weak reference, which automatically gets removed when garbage-collected.

See also [\\_remove\\_changed\\_handler\(\)](#).

**Parameters**



- **prop** ([PropertyBase](#)) – The property to add a changed handler for.
- **func** ([Callable](#)[[], *None*]) – The changed handler.

**Return type***None***abstract \_available\_version(prop)**

Return which version of *prop* is currently available (i.e. is stored in *\_\_values*).

**Parameters**

**prop** ([PropertyBase](#)) – The property to consider.

**Return type***int***\_detect\_dependencies(prop, \*, append=False)**

Detect dependencies for a code block.

Use it for a `with` statement.

**Parameters**

- **prop** ([PropertyBase](#)) – The property of that object to track dependencies for.
- **append** (*bool*) – If to append dependencies in this block to the known one (instead of replacing them).

**Return type***ContextManager*[*None*]**abstract \_flush\_dependency\_handlers(prop)**

Remove all change handlers on any object and property which handle the dependency of a particular property. Afterward, all change handlers anywhere that would trigger the computation of *prop* are removed. This method is usually called just before the dependencies of *prop* are determined again, i.e. just before its computation.

**Parameters**

**prop** ([PropertyBase](#)) – The property to consider.

**Return type***None***abstract \_get\_dependencies(prop)**

Return all dependencies of a particular property.

**Parameters**

**prop** ([PropertyBase](#)) – The property to consider.

**Return type***Iterable*[*tuple*[*klovve.data.deps.DependencySubject*, *klovve.data.deps.PropertyBase*, *int*]]**\_notify\_dependency(prop)**

Notify the dependency tracking about a new dependency for the current computation (usually after a property read access).

**Parameters**

**prop** ([PropertyBase](#)) – The dependency property in the model.

**Return type***None*

**\_notify\_property\_value\_changed(prop)**

Notify the dependency tracking about the change of a model property and return whether it can proceed computations instantly (instead of stopping here, relying on being triggered again later).

**Parameters**

**prop** ([PropertyBase](#)) – The changed property in the model.

**Return type**

bool

**\_remove\_changed\_handler(prop, func)**

Remove a changed handler for a particular property.

**Parameters**

- **prop** ([PropertyBase](#)) – The property to remove a changed handler from.
- **func** ([Callable](#)[[], None]) – The changed handler.

**Return type**

None

**abstract \_set\_dependencies(prop, new\_dependencies)**

Set the dependencies of a particular property and register change handlers for each one, so *prop* gets re-computed whenever the value of a dependency changed.

**Parameters**

- **prop** ([PropertyBase](#)) – The property to consider.
- **new\_dependencies** ([Iterable](#)[[tuple](#)[[klovve.data.deps.DependencySubject](#), [klovve.data.deps.PropertyBase](#), [int](#)]]) – prop's new dependencies.

**Return type**

None

**class klovve.data.deps.PropertyBase**

Bases: object

Base class for properties on dependency trackable objects. This is mostly a marker, TODO.

**klovve.data.deps.\_TDependencyTuple**

A tuple that refers to a particular property from a particular object at a particular time (by a version number). Internally used for dependency tracking.

alias of [tuple](#)[[DependencySubject](#), [PropertyBase](#), [int](#)]

**klovve.data.deps.\_current\_compute\_dependencies:**

```
ContextVar[Optional[list[tuple[klovve.data.deps.DependencySubject,  
klovve.data.deps.PropertyBase, int]]]] = <ContextVar name='_current_compute_dependencies'  
default=None>
```

Context variable that gets filled with all dependencies for a particular property, while it is computed.

**klovve.data.deps.\_defer\_value\_propagation = 0**

Counts how many `pause_refreshing()` blocks exist on the current stack. If it is larger than 0, refreshing is considered to be paused.

```
klovve.data.deps._defer_value_propagation_for:
Optional[list[tuple[klovve.data.deps.DependencySubject, klovve.data.deps.PropertyBase]]]
= None
```

Context variable that gets filled with all dependencies for a particular property, while it is computed.

```
klovve.data.deps.no_dependency_tracking()
```

Stops dependency tracking of the current computation for a code block.

Use it for a `with` statement. Model property accesses do not count as a dependency in that `with` block.

It will not influence dependency tracking for other computations inside yours (i.e. if you access a computed property, the dependency tracking of this one will not break).

#### Return type

`ContextManager[None]`

```
klovve.data.deps.pause_refreshing()
```

Defers the automatic refreshes on model property changes for a code block and applies them afterward.

Use it for a `with` statement. Model property modifications do not instantly lead to updates on depending properties in that `with` block.

Computed properties will always return the right value on access nevertheless, and not an outdated one.

#### Return type

`ContextManager[None]`

## klovve.data.lists module

Klovve lists.

```
class klovve.data.lists.List(content=())
```

Bases: `MutableSequence`, `CommonObservableValue`

A list implementation that is observable (with `ListObserver` or via `klovve.data.observable_value.ObservableValueProtocol`) and has some convenience methods.

#### Parameters

**content** (`Iterable`) –

```
__correct_index(i)
```

#### Parameters

**i** (`int`) –

#### Return type

`int`

```
_abc_impl = <_abc._abc_data object>
```

```
_add_observer(observer, *, initialize=True)
```

Attach a `ListObserver` to this list.

For a model's list property, you have to use `klovve.data.model.Model.add_list_observer()` instead.

**Parameters**

- **observer** ([ListObserver](#)) – The observer to attach.
- **initialize** (*bool*) – Whether to initialize the observer with the current content.

**Return type**

None

**\_is\_protocol** = **False****\_remove\_observer**(*observer*)Detach a [ListObserver](#) from this list.For a model's list property, you have to use `TODO` instead.**Parameters**

- **observer** ([ListObserver](#)) – The observer to detach.

**Return type**

None

**insert**(*i*, *v*)`S.insert(index, value)` – insert value before index**move**(*target\_index*, *source\_index*)

Move an element.

**Parameters**

- **target\_index** (*int*) – The new index.
- **source\_index** (*int*) – The old index.

**Return type**

None

**update**(*source\_list*, \*, *create\_target\_object\_func*=<function `_update__default_create_target_object`>, *update\_target\_object\_func*=<function `_update__default_update_target_object`>, *is\_matching\_target\_object\_func*=<function `_update__default_is_matching_target_object`>)

Update this list with the content of another one.

**Parameters**

- **source\_list** (*Iterable[Any]*) – The content to apply to this list.
- **create\_target\_object\_func** (*Callable[[Any], Any]*) – Function that creates a representation for a source object in the target list. Advanced feature.
- **update\_target\_object\_func** (*Callable[[Any, Any], None]*) – Function that updates the representation for a source object in the target list. Advanced feature.
- **is\_matching\_target\_object\_func** (*Callable[[Any, Any], bool]*) – Function that checks if a particular object is the representation for a particular source object in the target list. Advanced feature.

**class** `klovve.data.lists.ListObserver`Bases: `ABC`Abstract base class for objects that can listen to changes on a [List](#).**\_abc\_impl** = <\_abc.\_abc\_data object>

**abstract item\_added(index, item)**

Called when an item was added to the observed list.

**Parameters**

- **index** (*int*) – The position where the new item was inserted.
- **item** (*object*) – The new item.

**Return type**

None

**abstract item\_moved(from\_index, to\_index)**

Called when an item was moved inside the observed list.

**Parameters**

- **from\_index** (*int*) – The position of the element that was moved.
- **to\_index** (*int*) – The new position of the moved element.

**Return type**

None

**abstract item\_removed(index)**

Called when an item was removed from the observed list.

**Parameters**

**index** (*int*) – The position of the item that was removed.

**Return type**

None

**item\_replaced(index, item)**

Called when an item was replaced by another one in the observed list.

**Parameters**

- **index** – The position where the item was replaced.
- **item** – The new item.

**Return type**

None

**klovve.data.lists.\_update\_\_default\_create\_target\_object(source\_object)**

Default function for target object creation in [List.update\(\)](#).

It just returns the source object.

**Parameters**

**source\_object** (*Any*) – The source object.

**Return type**

*Any*

**klovve.data.lists.\_update\_\_default\_is\_matching\_target\_object(target\_object, source\_object)**

Default function for target/source object comparison in [List.update\(\)](#).

It compares the objects with the == operator.

**Parameters**

- **target\_object** (*Any*) – The target object.
- **source\_object** (*Any*) – The source object.

**Return type**

bool

`klovve.data.lists._update__default_update_target_object(target_object, source_object)`Default function for target object updating in [List.update\(\)](#).

It does nothing.

**Parameters**

- **target\_object** (*Any*) – The target object.
- **source\_object** (*Any*) – The source object.

**Return type**

None

**klovve.data.model module**

Klovve data models.

`klovve.data.model.ComputedProperty(func=None, *, initially=<function _none>,  
use_last_value_during_recompute=True)`

Decorate a method to be a computed property.

**Parameters**

- **func** (*Optional[Callable[[], Any]]*) – The method (of a [Model](#) subclass) to consider as computed property.
- **initially** (*Callable[[], Any]*) – Function that returns the initial value, which this property holds before any computation. This is only relevant if *func* is *async*.
- **use\_last\_value\_during\_recompute** (*bool*) – Whether this property should keep the old value during computation (instead of falling back to the initial value meanwhile). This is only relevant if *func* is *async*.

**Return type***Callable[[Callable[[], Any]], [\\_ComputedProperty](#)]*`class klovve.data.model.ListProperty(*, default=<function ListProperty.<lambda>>)`Bases: [Property](#)

A variable-like property, but containing a list. The list itself cannot be replaced (i.e. not settable), but you can modify its content (in the same way as for Python lists and some additional ones; see [klovve.data.lists.List](#)).

**Parameters****default** (*Callable[[], Iterable[Any]]*) –**\_fset**(*obj, value*)

Set the property to a new value.

**Parameters**

- **obj** – The object to set the value for.
- **value** – The new value.

---

```
class klovve.data.model.Model(**kwargs)
```

Bases: [DependencySubject](#)

Base class for Klovve data models.

As a [klovve.data.deps.DependencySubject](#) it is part of Klovve data dependency tracking, with special properties defined on it (see [Property](#), [ComputedProperty](#), and others).

```
__compute(prop)
```

Refresh the value for a particular property. This either makes the fresh value directly available, or at least triggers async computation to make a fresh value appear as soon as possible.

**Parameters**

**prop** ([PropertyBase](#)) – The property to refresh.

**Return type**

None

```
__head_version(prop)
```

Return which version of a particular property is the most recent one. This will automatically be higher than the available version if any dependency has changed its value since the last computation of *prop*.

**Parameters**

**prop** ([PropertyBase](#)) – The property to consider.

```
__invalidate(prop)
```

Mark the current value of a particular property as outdated. This does not directly execute any re-computation (but will enforce it on next access).

**Parameters**

**prop** ([PropertyBase](#)) – The property to mark.

```
_abc_impl = <_abc._abc_data object>
```

```
_add_changed_handler(prop, func)
```

Add a changed handler for a particular property, as a weak reference, which automatically gets removed when garbage-collected.

See also [\\_remove\\_changed\\_handler\(\)](#).

**Parameters**

- **prop** – The property to add a changed handler for.
- **func** – The changed handler.

```
_available_version(prop)
```

Return which version of *prop* is currently available (i.e. is stored in `__values`).

**Parameters**

**prop** – The property to consider.

```
_flush_dependency_handlers(prop)
```

Remove all change handlers on any object and property which handle the dependency of a particular property. Afterward, all change handlers anywhere that would trigger the computation of *prop* are removed. This method is usually called just before the dependencies of *prop* are determined again, i.e. just before its computation.

**Parameters**

**prop** – The property to consider.

**\_get\_dependencies**(*prop*)

Return all dependencies of a particular property.

**Parameters**

**prop** – The property to consider.

**\_remove\_changed\_handler**(*prop, func*)

Remove a changed handler for a particular property.

**Parameters**

- **prop** – The property to remove a changed handler from.
- **func** – The changed handler.

**\_set\_dependencies**(*prop, new\_dependencies*)

Set the dependencies of a particular property and register change handlers for each one, so *prop* gets re-computed whenever the value of a dependency changed.

**Parameters**

- **prop** – The property to consider.
- **new\_dependencies** – *prop*'s new dependencies.

**\_set\_value**(*prop, value*)

Set a property to a new value and trigger automatic re-computation of dependent values.

**Parameters**

- **prop** ([PropertyBase](#)) – The property to set.
- **value** (*Any*) – The new value.

**Return type**

None

**\_trigger\_changed**(*prop*)

React to a changed value for a particular property, refresh all dependent computed properties, and so on.

Inside a `klovve.deps.pause_refreshing()` block, it will do nothing, but just ensure to be called again after that block.

**Parameters**

**prop** ([PropertyBase](#)) – The property which value has been changed.

**Return type**

None

**\_value**(*prop*)

Return the value for a property.

It always returns the recent version, i.e. no (synchronous) computation or reaction is going on for this property anymore (even during `klovve.deps.pause_refreshing()`). There can be async computations going on, though.

**Parameters**

**prop** ([PropertyBase](#)) – The property to fetch the value for.

**Return type**

*Any*



```
class klovve.data.model.Property(*, default=<function _none>)
```

Bases: [PropertyBase](#)

A variable-like property (i.e. gettable and settable). If you want to store a list, you should use [ListProperty](#) instead (otherwise performance may degrade and the user interface may behave in undesired ways).

**Parameters**

**default** (*Union[Any, Callable[[], Any]]*) –

**\_compute(obj)**

Compute and return the property value. This will execute the value computation if there is any.

**Parameters**

**obj** – The object to compute the value for.

```
class klovve.data.model.PropertyBase
```

Bases: [PropertyBase](#), `property`

Base class for Klovve properties. They can be like variables (i.e. you can read and write them), or computed, like Python properties (i.e. you can read them, but they contain computed values). See subclasses.

Properties are defined in [Model](#) subclasses. For each instance of those models, a property will carry its value, very much like the most basic usage of variables and properties in usual classes.

On top of bare Python variables and properties, you can directly bind Klovve properties to properties of user interface elements, keeping everything up-to-date without any manual plumbing of change handlers or similar.

**\_compute(obj)**

Compute and return the property value. This will execute the value computation if there is any.

**Parameters**

**obj** – The object to compute the value for.

**Return type**

*Any*

**\_fget(obj)**

Return the property value.

**Parameters**

**obj** – The object to get the value from.

**\_fset(obj, value)**

Set the property to a new value.

**Parameters**

- **obj** – The object to set the value for.
- **value** – The new value.

**\_initially()**

**Return type**

*Any*

**\_use\_last\_value\_during\_recompute()**

**Return type**

`bool`

**class** klovve.data.model.\_**ComputedProperty**(*func*, \*, *initially*, *use\_last\_value\_during\_recompute*)

Bases: [PropertyBase](#)

A computed property. You will usually define it by means of [ComputedProperty\(\)](#).

**Parameters**

- **initially** (*Callable*[[*Any*], *Any*]) –
- **use\_last\_value\_during\_recompute** (*bool*) –

**\_compute**(*obj*)

Compute and return the property value. This will execute the value computation if there is any.

**Parameters**

**obj** – The object to compute the value for.

**\_fset**(*obj*, *value*)

Set the property to a new value.

**Parameters**

- **obj** – The object to set the value for.
- **value** – The new value.

**\_initially**()

**\_use\_last\_value\_during\_recompute**()

klovve.data.model.\_**TGetValueFunc**

Python type for a callable without parameters that returns some value (or not, which is as returning *None*).

alias of *Callable*[[*Any*], *Any*]

klovve.data.model.\_**none**()

Return *None*.

klovve.data.model.**observe\_list**(*model*, *prop\_name*, *observer*, \*, *initialize=True*)

Start observing a list (i.e. a value of a [ListProperty](#)).

**Parameters**

- **model** ([Model](#)) – The model object to observe.
- **prop\_name** (*str*) – The name of the list property.
- **observer** ([ListObserver](#)) – The observer.
- **initialize** (*bool*) – Whether to push the current list content to the observer now.

**Return type**

*None*

**klovve.data.observable\_value module**

**class** klovve.data.observable\_value.CommonObservableValue(\*args, \*\*kwargs)

Bases: [ObservableValueProtocol](#)

Basic implementation for the observable value protocol.

**\_abc\_impl** = <\_abc.\_abc\_data object>

**\_is\_protocol** = False

**\_trigger\_changed()**

Process a change (i.e. maybe recompute other property values, ...).

Call this from inside your observable class whenever the content has been changed.

**add\_changed\_handler(func)**

Add a change handler. It will be called for any subsequent changes, until [remove\\_changed\\_handler\(\)](#) is used.

**Parameters**

**func** – The change handler to add.

**remove\_changed\_handler(func)**

Remove a change handler.

**Parameters**

**func** – The change handler to remove.

**class** klovve.data.observable\_value.ObservableValueProtocol(\*args, \*\*kwargs)

Bases: Protocol

Protocol for an observable value. Observable values can be monitored by the Klovve model infrastructure, so it can refresh models (e.g. computed properties) not only when property values get reassigned, but also when it points to a mutable object (which implements this protocol) and the object `_content` changes.

This is used for lists (see list properties), but could be used for any kind of mutable objects that you want to assign to Klovve properties (although, whenever you can, you should define it as a Klovve model class instead, which gives you all that for free).

**\_abc\_impl** = <\_abc.\_abc\_data object>

**\_is\_protocol** = True

**\_is\_runtime\_protocol** = True

**\_trigger\_changed()**

Process a change (i.e. maybe recompute other property values, ...).

Call this from inside your observable class whenever the content has been changed.

**Return type**

None

**add\_changed\_handler(func)**

Add a change handler. It will be called for any subsequent changes, until [remove\\_changed\\_handler\(\)](#) is used.

**Parameters**

**func** (*Callable[[], None]*) – The change handler to add.

**Return type**

None

**remove\_changed\_handler**(*func*)

Remove a change handler.

**Parameters****func** (*Callable*[[*Any*], *Any*]) – The change handler to remove.**Return type**

None

**klovve.data.reaction module**

Reactions are a way to define a computation that is repeated automatically whenever depending model values change. See [reaction\(\)](#).

**exception** klovve.data.reaction.DeleteBases: *Exception***class** klovve.data.reaction.\_ReactionModel(*func*)Bases: *Model*Artificial model, internally used by [reaction\(\)](#).**\_abc\_impl** = <\_abc.\_abc\_data object>**property value**

Return the property value.

**Parameters****obj** – The object to get the value from.

klovve.data.reaction.**reaction**(\*, *owner*, *initially*=<function \_none>, *use\_last\_value\_during\_recompute*=*True*)

Define a function to be a reaction. It will automatically get executed when defined, and whenever any depending model values change.

Use it as a function decorator.

It is similar to a [klovve.data.model.ComputedProperty](#) in a model, but can be defined anywhere, which makes them very convenient in some situations.

**Parameters**

- **owner** (*Any*) – The object which owns the reaction. The reaction will usually stay alive as long as the owner still exists.
- **initially** (*Callable*[[*Any*], *Any*]) – Function which returns the initial value. Only relevant for *async* reactions.
- **use\_last\_value\_during\_recompute** (*bool*) – Whether to use the last computed value during re-computation (instead of setting it back to the initial value meanwhile).

**Return type***Callable*[[*Callable*[[*Any*], *Any*], *ReactionModel*]

**klovve.data.value\_holder module**

```
class klovve.data.value_holder.BindFactory(model, twoway=True, converter_in=None,  
                                           converter_out=None)
```

Bases: `object`

Bind factories return value holders like bindings on attribute access. The *props* object that you often see in view implementations is a bind factory.

**fuh**(*s*)

```
class klovve.data.value_holder.ComputedValue(func, model, props, pieces)
```

Bases: `ValueHolder`

**\_abc\_impl** = `<_abc._abc_data object>`

**get\_value**()

**is\_settable**()

**set\_value**(*value*)

```
class klovve.data.value_holder.OneWayBinding(model, prop_name, converter_in=None,  
                                              converter_out=None)
```

Bases: `ValueHolder`

Connection of a Klovve model object's property, with data flowing only away from it.

**\_abc\_impl** = `<_abc._abc_data object>`

**get\_value**()

**is\_settable**()

**set\_value**(*value*)

```
class klovve.data.value_holder.TwoWayBinding(model, prop_name, converter_in=None,  
                                              converter_out=None)
```

Bases: `OneWayBinding`

Connection of a Klovve model object's property, with data flowing in both ways.

**\_abc\_impl** = `<_abc._abc_data object>`

**is\_settable**()

```
class klovve.data.value_holder.ValueHolder
```

Bases: `ABC`

Abstract base class for a single-value container that can be connected to properties of user interface elements; either unidirectional or bidirectional.

**\_abc\_impl** = `<_abc._abc_data object>`

**abstract get\_value**()

**Return type**

*Any*

**abstract is\_settable()**

**Return type**

bool

**abstract set\_value(*value*)**

**Parameters**

**value** (*Any*) –

**Return type**

None

## klovve.drivers package

Klovve drivers. Part of internal infrastructure. Klovve applications do not need it.

**class klovve.drivers.Driver**

Bases: ABC

Internal infrastructure functionality that depends on a particular UI provider implementation.

**\_\_driver** = None

**\_\_mainloop** = None

**\_abc\_impl** = <\_abc.\_abc\_data object>

**abstract \_create\_mainloop()**

Create and return a mainloop.

**Return type**

*AbstractEventLoop*

**static get()**

Return the current driver. It will automatically be created and initialized internally if needed.

**Return type**

*Driver*

**static mainloop()**

Return Klovve's main loop.

May only be called after the driver is created.

**Return type**

*AbstractEventLoop*

**model\_type(*name\_tuple*)**

Return a model type by name.

**Parameters**

**name\_tuple** (*Iterable[str]*) – The piece name as tuple.

**Return type**

type['klovve.View']

**abstract static name()**

Return the driver name.

**Return type**

str

**abstract static rank()**

Return the rank number. Drivers with lower rank will be tried to load first.

**Return type**

float

**abstract show\_window(window)**

Show a window.

**Parameters**

**window** ([BaseView](#)) – The window to show.

**Return type**

None

**view\_type(name\_tuple)**

Return a view type by name.

**Parameters**

**name\_tuple** ([Iterable\[str\]](#)) – The piece name as tuple.

**Return type**

type['klovve.View']

`klovve.drivers._find_modules(name_tuple, kind)`

**Parameters**

- **name\_tuple** ([Iterable\[str\]](#)) –
- **kind** (*str*) –

`klovve.drivers._try_find_type_in_module(module_name, typename)`

**Parameters**

- **module\_name** (*str*) –
- **typename** (*str*) –

`klovve.drivers._try_find_type_per_name_tuple(name_tuple, kind, typename)`

**Parameters**

- **name\_tuple** ([Iterable\[str\]](#)) –
- **kind** (*str*) –
- **typename** (*str*) –

## Subpackages

### klovve.drivers.bare\_terminal package

Klovve bare terminal driver.

**class** klovve.drivers.bare\_terminal.Driver

Bases: *Driver*

**\_abc\_impl** = <\_abc.\_abc\_data object>

**\_create\_mainloop()**

Create and return a mainloop.

**static name()**

Return the driver name.

**static rank()**

Return the rank number. Drivers with lower rank will be tried to load first.

**show\_window(window)**

Show a window.

**Parameters**

**window** – The window to show.

## Subpackages

### klovve.drivers.bare\_terminal.views package

## Subpackages

### klovve.drivers.bare\_terminal.views.interact package

## Submodules

### klovve.drivers.bare\_terminal.views.interact.abstract module

**class** klovve.drivers.bare\_terminal.views.interact.abstract.View(model, view\_factory)

Bases: *View*, *BaseView*

**Parameters**

- **model** (*\_TGo*) –
- **view\_factory** (*klovve.view.tree.ViewFactory*) –

**\_abc\_impl** = <\_abc.\_abc\_data object>

**create\_native()**

Create a native representation for this view.

This must be implemented in native views, but is only called internally. See also *native()*.



## klovve.drivers.bare\_terminal.views.viewer package

### Submodules

#### klovve.drivers.bare\_terminal.views.viewer.image module

**class** klovve.drivers.bare\_terminal.views.viewer.image.**View**(*model*, *view\_factory*)

Bases: *View*, *BaseView*

#### Parameters

- **model** (*\_TGo*) –
- **view\_factory** (*klovve.view.tree.ViewFactory*) –

**\_abc\_impl** = <\_abc.\_abc\_data object>

**create\_native()**

Create a native representation for this view.

This must be implemented in native views, but is only called internally. See also `native()`.

### Submodules

#### klovve.drivers.bare\_terminal.views.checkable module

**class** klovve.drivers.bare\_terminal.views.checkable.**View**(*model*, *view\_factory*)

Bases: *View*, *BaseView*

#### Parameters

- **model** (*\_TGo*) –
- **view\_factory** (*klovve.view.tree.ViewFactory*) –

**\_abc\_impl** = <\_abc.\_abc\_data object>

**create\_native()**

Create a native representation for this view.

This must be implemented in native views, but is only called internally. See also `native()`.

#### klovve.drivers.bare\_terminal.views.headered\_panel module

**class** klovve.drivers.bare\_terminal.views.headered\_panel.**View**(*model*, *view\_factory*)

Bases: *View*, *BaseView*

#### Parameters

- **model** (*\_TGo*) –
- **view\_factory** (*klovve.view.tree.ViewFactory*) –

**\_abc\_impl** = <\_abc.\_abc\_data object>

**create\_native()**

Create a native representation for this view.

This must be implemented in native views, but is only called internally. See also `native()`.

**klovve.drivers.bare\_terminal.views.label module**

```
class klovve.drivers.bare_terminal.views.label.View(model, view_factory)
```

Bases: [View](#), [BaseView](#)

**Parameters**

- **model** (`_TGo`) –
- **view\_factory** (`klovve.view.tree.ViewFactory`) –

`_abc_impl = <_abc._abc_data object>`

**create\_native()**

Create a native representation for this view.

This must be implemented in native views, but is only called internally. See also `native()`.

**klovve.drivers.bare\_terminal.views.log\_pager module**

```
class klovve.drivers.bare_terminal.views.log_pager.View(model, view_factory)
```

Bases: [View](#), [BaseView](#)

**Parameters**

- **model** (`_TGo`) –
- **view\_factory** (`klovve.view.tree.ViewFactory`) –

`_abc_impl = <_abc._abc_data object>`

**create\_native()**

Create a native representation for this view.

This must be implemented in native views, but is only called internally. See also `native()`.

**klovve.drivers.bare\_terminal.views.placeholder module**

```
class klovve.drivers.bare_terminal.views.placeholder.View(model, view_factory)
```

Bases: [View](#), [BaseView](#)

**Parameters**

- **model** (`_TGo`) –
- **view\_factory** (`klovve.view.tree.ViewFactory`) –

`_abc_impl = <_abc._abc_data object>`

**create\_native()**

Create a native representation for this view.

This must be implemented in native views, but is only called internally. See also `native()`.

**klovve.drivers.bare\_terminal.views.split module**

**class** klovve.drivers.bare\_terminal.views.split.**View**(*model*, *view\_factory*)

Bases: *View*, *BaseView*

**Parameters**

- **model** (*\_TGo*) –
- **view\_factory** (*klovve.view.tree.ViewFactory*) –

**\_abc\_impl** = <\_abc.\_abc\_data object>

**create\_native()**

Create a native representation for this view.

This must be implemented in native views, but is only called internally. See also *native()*.

**klovve.drivers.bare\_terminal.views.vertical\_box module**

**class** klovve.drivers.bare\_terminal.views.vertical\_box.**View**(*model*, *view\_factory*)

Bases: *View*, *BaseView*

**Parameters**

- **model** (*\_TGo*) –
- **view\_factory** (*klovve.view.tree.ViewFactory*) –

**\_abc\_impl** = <\_abc.\_abc\_data object>

**create\_native()**

Create a native representation for this view.

This must be implemented in native views, but is only called internally. See also *native()*.

**klovve.drivers.bare\_terminal.views.window module**

**class** klovve.drivers.bare\_terminal.views.window.**View**(*model*, *view\_factory*)

Bases: *View*, *BaseView*

**Parameters**

- **model** (*\_TGo*) –
- **view\_factory** (*klovve.view.tree.ViewFactory*) –

**\_abc\_impl** = <\_abc.\_abc\_data object>

**create\_native()**

Create a native representation for this view.

This must be implemented in native views, but is only called internally. See also *native()*.

**klovve.drivers.gtk package**

Klovve Gtk driver.

**class** klovve.drivers.gtk.**Driver**

Bases: *Driver*

**\_abc\_impl** = <\_abc.\_abc\_data object>

**\_create\_mainloop()**

Create and return a mainloop.

**static name()**

Return the driver name.

**static rank()**

Return the rank number. Drivers with lower rank will be tried to load first.

**show\_window(window)**

Show a window.

**Parameters**

**window** – The window to show.

**class** klovve.drivers.gtk.**View(model, view\_factory)**

Bases: *BaseView*

**Parameters**

- **model** (*\_TGo*) –

- **view\_factory** (*klovve.view.tree.ViewFactory*) –

**\_\_bind(\*, gprop\_name, value\_holder)**

**Parameters**

**value\_holder** (*ValueHolder*) –

**\_abc\_impl** = <\_abc.\_abc\_data object>

**static bind(gobj, gprop\_name, value)**

**Parameters**

**value** (*Any*) –

**static gtk\_children(widget)**

**static gtk\_new(viewtype, \*, to\_parent=None, \*\*kwargs)**

**class** klovve.drivers.gtk.**\_EventLoop**

Bases: *\_UnixSelectorEventLoop*

**\_call\_soon(callback, args, context)**

**call\_at(when, callback, \*args, context=None)**

Like `call_later()`, but uses an absolute time.

Absolute time corresponds to the event loop's `time()` method.

**create\_task**(*coro*, \*, *name=None*)

Schedule a coroutine object.

Return a task object.

**class** klovve.drivers.gtk.\_Selector(*context*)

Bases: \_BaseSelectorImpl

**\_abc\_impl** = <\_abc.\_abc\_data object>

**close**()

Close the selector.

This must be called to make sure that any underlying resource is freed.

**interrupt\_glib\_mainloop**()

**register**(*fileobj*, *events*, *data=None*)

Register a file object.

Parameters: *fileobj* – file object or file descriptor *events* – events to monitor (bitwise mask of EVENT\_READ|EVENT\_WRITE) *data* – attached data

Returns: SelectorKey instance

Raises: ValueError if *events* is invalid KeyError if *fileobj* is already registered OSError if *fileobj* is closed or otherwise is unacceptable to

the underlying system call (if a system call is made)

Note: OSError may or may not be raised

**select**(*timeout=None*)

Perform the actual selection, until some monitored file objects are ready or a timeout expires.

Parameters: *timeout* – if *timeout* > 0, this specifies the maximum wait time, in

seconds if *timeout* <= 0, the select() call won't block, and will report the currently ready file objects if *timeout* is None, select() will block until a monitored file object becomes ready

Returns: list of (key, events) for ready file objects *events* is a bitwise mask of EVENT\_READ|EVENT\_WRITE

**unregister**(*fileobj*)

Unregister a file object.

Parameters: *fileobj* – file object or file descriptor

Returns: SelectorKey instance

Raises: KeyError if *fileobj* is not registered

Note: If *fileobj* is registered but has since been closed this does *not* raise OSError (even if the wrapped syscall does)

**class** klovve.drivers.gtk.\_Source(\**args*, \*\**kwargs*)

Bases: Source

### Constructors

Source()

new(source\_funcs:GLib.SourceFuncs, struct\_size:int) -> GLib.Source

```
EVENT_TO_GLIB = dict_items([(1, <flags G_IO_IN of type GLib.IOCondition>), (2,  
<flags G_IO_OUT of type GLib.IOCondition>)])
```

```
add_fd(fd, events)
```

```
check()
```

```
dispatch(callback, args)
```

```
get_events(fd)
```

```
prepare()
```

```
remove_fd(fd)
```

```
klovve.drivers.gtk.children(widget)
```

## Subpackages

**klovve.drivers.gtk.views package**

## Subpackages

**klovve.drivers.gtk.views.\_gtk\_private package**

## Submodules

**klovve.drivers.gtk.views.\_gtk\_private.box module**

```
class klovve.drivers.gtk.views._gtk_private.box.View(model, view_factory)
```

Bases: [View](#)

### Parameters

- **model** ([\\_TGo](#)) –
- **view\_factory** ([klovve.view.tree.ViewFactory](#)) –

```
_abc_impl = <_abc._abc_data object>
```

```
_orientation()
```

```
create_native()
```

Create a native representation for this view.

This must be implemented in native views, but is only called internally. See also [native\(\)](#).

## klovve.drivers.gtk.views.form package

**class** klovve.drivers.gtk.views.form.**View**(*model*, *view\_factory*)

Bases: [View](#), [View](#)

### Parameters

- **model** (*\_TGo*) –
- **view\_factory** ([klovve.view.tree.ViewFactory](#)) –

**\_abc\_impl** = <\_abc.\_abc\_data object>

**create\_native**()

Create a native representation for this view.

This must be implemented in native views, but is only called internally. See also [native\(\)](#).

## Submodules

### klovve.drivers.gtk.views.form.section module

**class** klovve.drivers.gtk.views.form.section.**View**(*model*, *view\_factory*)

Bases: [View](#), [View](#)

### Parameters

- **model** (*\_TGo*) –
- **view\_factory** ([klovve.view.tree.ViewFactory](#)) –

**\_abc\_impl** = <\_abc.\_abc\_data object>

**create\_native**()

Create a native representation for this view.

This must be implemented in native views, but is only called internally. See also [native\(\)](#).

## klovve.drivers.gtk.views.interact package

## Submodules

### klovve.drivers.gtk.views.interact.abstract module

**class** klovve.drivers.gtk.views.interact.abstract.**View**(*model*, *view\_factory*)

Bases: [View](#), [View](#)

### Parameters

- **model** (*\_TGo*) –
- **view\_factory** ([klovve.view.tree.ViewFactory](#)) –

**\_abc\_impl** = <\_abc.\_abc\_data object>

**create\_native()**

Create a native representation for this view.

This must be implemented in native views, but is only called internally. See also `native()`.

**klovve.drivers.gtk.views.property\_panel package**

```
class klovve.drivers.gtk.views.property_panel.View(model, view_factory)
```

Bases: *View*, *View*

**Parameters**

- **model** (*\_TGoO*) –
- **view\_factory** (*klovve.view.tree.ViewFactory*) –

**\_abc\_impl** = <\_abc.\_abc\_data object>

**create\_native()**

Create a native representation for this view.

This must be implemented in native views, but is only called internally. See also `native()`.

**klovve.drivers.gtk.views.tabbed package**

```
class klovve.drivers.gtk.views.tabbed.View(model, view_factory)
```

Bases: *View*, *View*

**Parameters**

- **model** (*\_TGoO*) –
- **view\_factory** (*klovve.view.tree.ViewFactory*) –

**\_abc\_impl** = <\_abc.\_abc\_data object>

**create\_native()**

Create a native representation for this view.

This must be implemented in native views, but is only called internally. See also `native()`.

**Submodules****klovve.drivers.gtk.views.tabbed.tab module**

```
class klovve.drivers.gtk.views.tabbed.tab.View(*args, **kwargs)
```

Bases: *View*, *ComposedView*

**\_abc\_impl** = <\_abc.\_abc\_data object>

**compose()**



**klovve.drivers.gtk.views.viewer package****Submodules****klovve.drivers.gtk.views.viewer.image module**

```
class klovve.drivers.gtk.views.viewer.image.Image(**kwargs)
```

Bases: Overlay

```
__ensure_config_correct()
```

```
__refresh(widget=None, *_ ,force=False)
```

```
_refresch()
```

```
display_coordinates_to_image_coordinates(x, y)
```

**Parameters**

- **x** (*float*) –
- **y** (*float*) –

**Return type**

tuple[float, float]

```
property display_height: int
```

```
property display_width: int
```

```
do_get_property(pspec)
```

```
do_set_property(pspec, value)
```

```
image_coordinates_to_display_coordinates(x, y)
```

**Parameters**

- **x** (*float*) –
- **y** (*float*) –

**Return type**

tuple[float, float]

```
image_offset_x
```

```
image_offset_y
```

```
property original_image_height: int
```

```
property original_image_width: int
```

```
pixbuf
```

```
zoom_factor
```

**zoom\_in**(*by=0.1, \*, center\_x=0.5, center\_y=0.5*)

**Parameters**

- **by** (*float*) –
- **center\_x** (*float*) –
- **center\_y** (*float*) –

**Return type**

None

**zoom\_out**(*by=0.1, \*, center\_x=0.5, center\_y=0.5*)

**Parameters**

- **by** (*float*) –
- **center\_x** (*float*) –
- **center\_y** (*float*) –

**Return type**

None

**class** klovve.drivers.gtk.views.viewer.image.**Outer**(\*\**kwargs*)

Bases: `Widget`

**\_\_on\_mouse\_button\_pressed**(*\_, n\_press, x, y*)

**\_\_on\_mouse\_button\_released**(*\_, n\_press, x, y*)

**\_\_on\_mouse\_moved**(*\_, x, y*)

**\_\_on\_mouse\_wheel**(*\_, dx, dy*)

**do\_get\_property**(*pspec*)

**do\_measure**(*orientation, for\_size*)

measure(*self, orientation:Gtk.Orientation, for\_size:int*) -> *minimum:int, natural:int, minimum\_baseline:int, natural\_baseline:int*

**do\_set\_property**(*pspec, value*)

**do\_size\_allocate**(*width, height, \_*)

size\_allocate(*self, width:int, height:int, baseline:int*)

**pixbuf**

**set\_source**(*source*)

**Parameters**

- **source** (*str*) –

**source**

**class** klovve.drivers.gtk.views.viewer.image.**View**(*model, view\_factory*)

Bases: `View`, `View`

**Parameters**

- **model** (*\_TGoO*) –

- **view\_factory** (`klovve.view.tree.ViewFactory`) –

**\_abc\_impl** = <\_abc.\_abc\_data object>

**create\_native()**

Create a native representation for this view.

This must be implemented in native views, but is only called internally. See also `native()`.

### klovve.drivers.gtk.views.viewer.pdf module

**class** `klovve.drivers.gtk.views.viewer.pdf.View(model, view_factory)`

Bases: `View`, `BaseView`

#### Parameters

- **model** (`_TGo`) –
- **view\_factory** (`klovve.view.tree.ViewFactory`) –

**\_abc\_impl** = <\_abc.\_abc\_data object>

**create\_native()**

Create a native representation for this view.

This must be implemented in native views, but is only called internally. See also `native()`.

### Submodules

#### klovve.drivers.gtk.views.busy\_animation module

**class** `klovve.drivers.gtk.views.busy_animation.View(model, view_factory)`

Bases: `View`, `View`

#### Parameters

- **model** (`_TGo`) –
- **view\_factory** (`klovve.view.tree.ViewFactory`) –

**\_abc\_impl** = <\_abc.\_abc\_data object>

**create\_native()**

Create a native representation for this view.

This must be implemented in native views, but is only called internally. See also `native()`.

**property** `gtk_orientation`

Return the property value.

#### Parameters

**obj** – The object to get the value from.

**property** `spinner_size`

Return the property value.

#### Parameters

**obj** – The object to get the value from.

### klovve.drivers.gtk.views.button module

**class** klovve.drivers.gtk.views.button.**ActionContext**(*button*, \*args)

Bases: [ActionContext](#)

**\_abc\_impl** = <\_abc.\_abc\_data object>

**async \_create\_dialog**(*view*, *done\_future*)

TODO Create a dialog with the given view content, and wait until *done\_future* is done. Implemented by subclasses, but not directly used from outside.

#### Parameters

- **view** – The view to show in a dialog.
- **done\_future** – The future to wait for before closing the dialog again.

**class** klovve.drivers.gtk.views.button.**View**(*model*, *view\_factory*)

Bases: [View](#), [View](#)

#### Parameters

- **model** ([\\_TGo](#)) –
- **view\_factory** ([klovve.view.tree.ViewFactory](#)) –

**\_abc\_impl** = <\_abc.\_abc\_data object>

**create\_native**()

Create a native representation for this view.

This must be implemented in native views, but is only called internally. See also [native\(\)](#).

### klovve.drivers.gtk.views.checkable module

**class** klovve.drivers.gtk.views.checkable.**View**(*model*, *view\_factory*)

Bases: [View](#), [View](#)

#### Parameters

- **model** ([\\_TGo](#)) –
- **view\_factory** ([klovve.view.tree.ViewFactory](#)) –

**\_abc\_impl** = <\_abc.\_abc\_data object>

**create\_native**()

Create a native representation for this view.

This must be implemented in native views, but is only called internally. See also [native\(\)](#).

**klovve.drivers.gtk.views.disableable module**

**class** klovve.drivers.gtk.views.disableable.**View**(*model*, *view\_factory*)

Bases: *View*, *View*

**Parameters**

- **model** (*\_TGoO*) –
- **view\_factory** (*klovve.view.tree.ViewFactory*) –

**\_abc\_impl** = <**\_abc.\_abc\_data** object>

**create\_native()**

Create a native representation for this view.

This must be implemented in native views, but is only called internally. See also **native()**.

**property gtk\_sensitive**

Return the property value.

**Parameters**

- obj** – The object to get the value from.

**klovve.drivers.gtk.views.dropdown module**

**class** klovve.drivers.gtk.views.dropdown.**View**(*model*, *view\_factory*)

Bases: *View*, *View*

**Parameters**

- **model** (*\_TGoO*) –
- **view\_factory** (*klovve.view.tree.ViewFactory*) –

**\_abc\_impl** = <**\_abc.\_abc\_data** object>

**create\_native()**

Create a native representation for this view.

This must be implemented in native views, but is only called internally. See also **native()**.

**klovve.drivers.gtk.views.dual module**

**class** klovve.drivers.gtk.views.dual.**View**(*model*, *view\_factory*)

Bases: *View*, *View*

**Parameters**

- **model** (*\_TGoO*) –
- **view\_factory** (*klovve.view.tree.ViewFactory*) –

**\_abc\_impl** = <**\_abc.\_abc\_data** object>

**create\_native()**

Create a native representation for this view.

This must be implemented in native views, but is only called internally. See also **native()**.

**klovve.drivers.gtk.views.formatted\_text module****class** klovve.drivers.gtk.views.formatted\_text.**View**(*model*, *view\_factory*)Bases: *View*, *View***Parameters**

- **model** (*\_TGo*) –
- **view\_factory** (*klovve.view.tree.ViewFactory*) –

**\_abc\_impl** = <\_abc.\_abc\_data object>**create\_native()**

Create a native representation for this view.

This must be implemented in native views, but is only called internally. See also *native()*.**klovve.drivers.gtk.views.header module****class** klovve.drivers.gtk.views.header.**View**(*model*, *view\_factory*)Bases: *View*, *View***Parameters**

- **model** (*\_TGo*) –
- **view\_factory** (*klovve.view.tree.ViewFactory*) –

**\_abc\_impl** = <\_abc.\_abc\_data object>**create\_native()**

Create a native representation for this view.

This must be implemented in native views, but is only called internally. See also *native()*.**klovve.drivers.gtk.views.headered\_panel module****class** klovve.drivers.gtk.views.headered\_panel.**View**(*model*, *view\_factory*)Bases: *View*, *View***Parameters**

- **model** (*\_TGo*) –
- **view\_factory** (*klovve.view.tree.ViewFactory*) –

**\_abc\_impl** = <\_abc.\_abc\_data object>**create\_native()**

Create a native representation for this view.

This must be implemented in native views, but is only called internally. See also *native()*.**property title\_label\_css\_classes**

Return the property value.

**Parameters****obj** – The object to get the value from.

**klovve.drivers.gtk.views.horizontal\_box module**

```
class klovve.drivers.gtk.views.horizontal_box.View(model, view_factory)
```

Bases: *View*, *View*

**Parameters**

- **model** (*\_TGo*) –
- **view\_factory** (*klovve.view.tree.ViewFactory*) –

**\_abc\_impl** = <\_abc.\_abc\_data object>

**\_orientation()**

**klovve.drivers.gtk.views.label module**

```
class klovve.drivers.gtk.views.label.View(model, view_factory)
```

Bases: *View*, *View*

**Parameters**

- **model** (*\_TGo*) –
- **view\_factory** (*klovve.view.tree.ViewFactory*) –

**\_abc\_impl** = <\_abc.\_abc\_data object>

**create\_native()**

Create a native representation for this view.

This must be implemented in native views, but is only called internally. See also *native()*.

**klovve.drivers.gtk.views.list module**

```
class klovve.drivers.gtk.views.list.View(model, view_factory)
```

Bases: *View*, *View*

**Parameters**

- **model** (*\_TGo*) –
- **view\_factory** (*klovve.view.tree.ViewFactory*) –

**\_abc\_impl** = <\_abc.\_abc\_data object>

**create\_native()**

Create a native representation for this view.

This must be implemented in native views, but is only called internally. See also *native()*.

**klovve.drivers.gtk.views.log\_pager module**

```
class klovve.drivers.gtk.views.log_pager.View(*args, **kwargs)
```

Bases: [View](#), [View](#)

```
__inner_size_changed(widget, *_)
```

```
_abc_impl = <_abc._abc_data object>
```

```
create_native()
```

Create a native representation for this view.

This must be implemented in native views, but is only called internally. See also `native()`.

**klovve.drivers.gtk.views.placeholder module**

```
class klovve.drivers.gtk.views.placeholder.View(model, view_factory)
```

Bases: [View](#), [View](#)

**Parameters**

- **model** ([\\_TGo](#)) –
- **view\_factory** ([klovve.view.tree.ViewFactory](#)) –

```
_abc_impl = <_abc._abc_data object>
```

```
create_native()
```

Create a native representation for this view.

This must be implemented in native views, but is only called internally. See also `native()`.

**klovve.drivers.gtk.views.progress module**

```
class klovve.drivers.gtk.views.progress.View(model, view_factory)
```

Bases: [View](#), [View](#)

**Parameters**

- **model** ([\\_TGo](#)) –
- **view\_factory** ([klovve.view.tree.ViewFactory](#)) –

```
_abc_impl = <_abc._abc_data object>
```

```
create_native()
```

Create a native representation for this view.

This must be implemented in native views, but is only called internally. See also `native()`.



**klovve.drivers.gtk.views.scrollable module**

**class** klovve.drivers.gtk.views.scrollable.**View**(*model*, *view\_factory*)

Bases: *View*, *View*

**Parameters**

- **model** (*\_TGo*) –
- **view\_factory** (*klovve.view.tree.ViewFactory*) –

**\_abc\_impl** = <\_abc.\_abc\_data object>

**create\_native()**

Create a native representation for this view.

This must be implemented in native views, but is only called internally. See also *native()*.

**klovve.drivers.gtk.views.slider module**

**class** klovve.drivers.gtk.views.slider.**View**(*model*, *view\_factory*)

Bases: *View*, *View*

**Parameters**

- **model** (*\_TGo*) –
- **view\_factory** (*klovve.view.tree.ViewFactory*) –

**\_abc\_impl** = <\_abc.\_abc\_data object>

**create\_native()**

Create a native representation for this view.

This must be implemented in native views, but is only called internally. See also *native()*.

**klovve.drivers.gtk.views.split module**

**class** klovve.drivers.gtk.views.split.**View**(*model*, *view\_factory*)

Bases: *View*, *View*

**Parameters**

- **model** (*\_TGo*) –
- **view\_factory** (*klovve.view.tree.ViewFactory*) –

**\_abc\_impl** = <\_abc.\_abc\_data object>

**create\_native()**

Create a native representation for this view.

This must be implemented in native views, but is only called internally. See also *native()*.

**klovve.drivers.gtk.views.text\_field module****class** klovve.drivers.gtk.views.text\_field.**View**(*model*, *view\_factory*)Bases: *View*, *View***Parameters**

- **model** (*\_TGo*) –
- **view\_factory** (*klovve.view.tree.ViewFactory*) –

**\_abc\_impl** = <\_abc.\_abc\_data object>**create\_native**()

Create a native representation for this view.

This must be implemented in native views, but is only called internally. See also *native()*.**klovve.drivers.gtk.views.vertical\_box module****class** klovve.drivers.gtk.views.vertical\_box.**View**(*model*, *view\_factory*)Bases: *View*, *View***Parameters**

- **model** (*\_TGo*) –
- **view\_factory** (*klovve.view.tree.ViewFactory*) –

**\_abc\_impl** = <\_abc.\_abc\_data object>**\_orientation**()**klovve.drivers.gtk.views.window module****class** klovve.drivers.gtk.views.window.**ActionContext**(*main\_window*, \**args*)Bases: *ActionContext***\_abc\_impl** = <\_abc.\_abc\_data object>**async \_create\_dialog**(*view*, *done\_future*)TODO Create a dialog with the given view content, and wait until *done\_future* is done. Implemented by subclasses, but not directly used from outside.**Parameters**

- **view** – The view to show in a dialog.
- **done\_future** – The future to wait for before closing the dialog again.

**class** klovve.drivers.gtk.views.window.**View**(*model*, *view\_factory*)Bases: *View*, *View***Parameters**

- **model** (*\_TGo*) –
- **view\_factory** (*klovve.view.tree.ViewFactory*) –

```
_abc_impl = <_abc._abc_data object>
```

```
create_native()
```

Create a native representation for this view.

This must be implemented in native views, but is only called internally. See also `native()`.

## klovve.drivers.viwid package

Klovve Viwid driver.

```
class klovve.drivers.viwid.Driver
```

Bases: *Driver*

```
_abc_impl = <_abc._abc_data object>
```

```
_create_mainloop()
```

Create and return a mainloop.

```
static name()
```

Return the driver name.

```
static rank()
```

Return the rank number. Drivers with lower rank will be tried to load first.

```
show_window(window)
```

Show a window.

### Parameters

**window** – The window to show.

```
class klovve.drivers.viwid.View(model, view_factory)
```

Bases: *BaseView*

### Parameters

- **model** (*\_TGo*) –
- **view\_factory** (*klovve.view.tree.ViewFactory*) –

```
__bind(*, aprop_name, value_holder)
```

### Parameters

**value\_holder** (*ValueHolder*) –

```
_abc_impl = <_abc._abc_data object>
```

```
static bind(aobj, aprop_name, value)
```

### Parameters

**value** (*Any*) –

```
static fizzlewid_new(viewtype, **kwargs)
```

```
klovve.drivers.viwid.children(widget)
```

## Subpackages

### klovve.drivers.viwid.views package

## Subpackages

### klovve.drivers.viwid.views.\_curses\_private package

## Submodules

### klovve.drivers.viwid.views.\_curses\_private.box module

**class** klovve.drivers.viwid.views.\_curses\_private.box.**View**(*model*, *view\_factory*)

Bases: [View](#)

#### Parameters

- **model** (*\_TGo*) –
- **view\_factory** ([klovve.view.tree.ViewFactory](#)) –

**\_abc\_impl** = <\_abc.\_abc\_data object>

**\_orientation()**

**create\_native()**

Create a native representation for this view.

This must be implemented in native views, but is only called internally. See also [native\(\)](#).

### klovve.drivers.viwid.views.form package

**class** klovve.drivers.viwid.views.form.**View**(*model*, *view\_factory*)

Bases: [View](#), [View](#)

#### Parameters

- **model** (*\_TGo*) –
- **view\_factory** ([klovve.view.tree.ViewFactory](#)) –

**\_abc\_impl** = <\_abc.\_abc\_data object>

**create\_native()**

Create a native representation for this view.

This must be implemented in native views, but is only called internally. See also [native\(\)](#).

## Submodules

### klovve.drivers.viuid.views.form.section module

**class** klovve.drivers.viuid.views.form.section.**View**(*model*, *view\_factory*)

Bases: *View*, *View*

#### Parameters

- **model** (*\_TGo*) –
- **view\_factory** (*klovve.view.tree.ViewFactory*) –

**\_abc\_impl** = <\_abc.\_abc\_data object>

**create\_native()**

Create a native representation for this view.

This must be implemented in native views, but is only called internally. See also *native()*.

### klovve.drivers.viuid.views.interact package

## Submodules

### klovve.drivers.viuid.views.interact.abstract module

**class** klovve.drivers.viuid.views.interact.abstract.**View**(*model*, *view\_factory*)

Bases: *View*, *View*

#### Parameters

- **model** (*\_TGo*) –
- **view\_factory** (*klovve.view.tree.ViewFactory*) –

**\_abc\_impl** = <\_abc.\_abc\_data object>

**create\_native()**

Create a native representation for this view.

This must be implemented in native views, but is only called internally. See also *native()*.

### klovve.drivers.viuid.views.property\_panel package

**class** klovve.drivers.viuid.views.property\_panel.**View**(*model*, *view\_factory*)

Bases: *View*, *View*

#### Parameters

- **model** (*\_TGo*) –
- **view\_factory** (*klovve.view.tree.ViewFactory*) –

**\_abc\_impl** = <\_abc.\_abc\_data object>

**create\_native()**

Create a native representation for this view.

This must be implemented in native views, but is only called internally. See also `native()`.

**klovve.drivers.viwid.views.tabbed package**

**class** `klovve.drivers.viwid.views.tabbed.View(model, view_factory)`

Bases: `View`, `View`

**Parameters**

- **model** (`_TGo`) –
- **view\_factory** (`klovve.view.tree.ViewFactory`) –

`_abc_impl = <_abc._abc_data object>`

**create\_native()**

Create a native representation for this view.

This must be implemented in native views, but is only called internally. See also `native()`.

**Submodules****klovve.drivers.viwid.views.tabbed.tab module**

**class** `klovve.drivers.viwid.views.tabbed.tab.View(*args, **kwargs)`

Bases: `View`, `ComposedView`

`_abc_impl = <_abc._abc_data object>`

`compose()`

**klovve.drivers.viwid.views.viewer package****Submodules****klovve.drivers.viwid.views.viewer.image module**

**class** `klovve.drivers.viwid.views.viewer.image.View(model, view_factory)`

Bases: `View`, `View`

**Parameters**

- **model** (`_TGo`) –
- **view\_factory** (`klovve.view.tree.ViewFactory`) –

`_abc_impl = <_abc._abc_data object>`

**create\_native()**

Create a native representation for this view.

This must be implemented in native views, but is only called internally. See also `native()`.

**klovve.drivers.viwid.views.viewer.pdf module**

```
class klovve.drivers.viwid.views.viewer.pdf.View(model, view_factory)
```

Bases: *View*, *View*

**Parameters**

- **model** (*\_TGoO*) –
- **view\_factory** (*klovve.view.tree.ViewFactory*) –

```
_abc_impl = <_abc._abc_data object>
```

```
create_native()
```

Create a native representation for this view.

This must be implemented in native views, but is only called internally. See also `native()`.

**Submodules****klovve.drivers.viwid.views.busy\_animation module**

```
class klovve.drivers.viwid.views.busy_animation.Model(**kwargs)
```

Bases: *Model*

```
_abc_impl = <_abc._abc_data object>
```

```
gtk_orientation()
```

```
spinner_size()
```

```
class klovve.drivers.viwid.views.busy_animation.View(model, view_factory)
```

Bases: *View*, *View*

**Parameters**

- **model** (*\_TGoO*) –
- **view\_factory** (*klovve.view.tree.ViewFactory*) –

```
_abc_impl = <_abc._abc_data object>
```

```
create_native()
```

Create a native representation for this view.

This must be implemented in native views, but is only called internally. See also `native()`.

**klovve.drivers.viwid.views.button module**

```
class klovve.drivers.viwid.views.button.ActionContext(button, *args)
```

Bases: *ActionContext*

```
_abc_impl = <_abc._abc_data object>
```

**async \_create\_dialog**(*view*, *done\_future*)

TODO Create a dialog with the given view content, and wait until *done\_future* is done. Implemented by subclasses, but not directly used from outside.

**Parameters**

- **view** – The view to show in a dialog.
- **done\_future** – The future to wait for before closing the dialog again.

**class** klovve.drivers.viwid.views.button.**View**(*model*, *view\_factory*)

Bases: [View](#), [View](#)

**Parameters**

- **model** (*\_TGo*) –
- **view\_factory** ([klovve.view.tree.ViewFactory](#)) –

**\_abc\_impl** = <\_abc.\_abc\_data object>

**create\_native**()

Create a native representation for this view.

This must be implemented in native views, but is only called internally. See also [native\(\)](#).

### **klovve.drivers.viwid.views.checkable module**

**class** klovve.drivers.viwid.views.checkable.**View**(*model*, *view\_factory*)

Bases: [View](#), [View](#)

**Parameters**

- **model** (*\_TGo*) –
- **view\_factory** ([klovve.view.tree.ViewFactory](#)) –

**\_abc\_impl** = <\_abc.\_abc\_data object>

**create\_native**()

Create a native representation for this view.

This must be implemented in native views, but is only called internally. See also [native\(\)](#).

### **klovve.drivers.viwid.views.disableable module**

**class** klovve.drivers.viwid.views.disableable.**View**(*model*, *view\_factory*)

Bases: [View](#), [View](#)

**Parameters**

- **model** (*\_TGo*) –
- **view\_factory** ([klovve.view.tree.ViewFactory](#)) –

**\_abc\_impl** = <\_abc.\_abc\_data object>

**create\_native**()

Create a native representation for this view.

This must be implemented in native views, but is only called internally. See also [native\(\)](#).



**klovve.drivers.viwid.views.dropdown module**

```
class klovve.drivers.viwid.views.dropdown.View(model, view_factory)
```

Bases: *View*, *View*

**Parameters**

- **model** (*\_TGo*) –
- **view\_factory** (*klovve.view.tree.ViewFactory*) –

**\_abc\_impl** = *<\_abc.\_abc\_data object>*

**create\_native()**

Create a native representation for this view.

This must be implemented in native views, but is only called internally. See also *native()*.

**klovve.drivers.viwid.views.dual module**

```
class klovve.drivers.viwid.views.dual.View(model, view_factory)
```

Bases: *View*, *View*

**Parameters**

- **model** (*\_TGo*) –
- **view\_factory** (*klovve.view.tree.ViewFactory*) –

**\_abc\_impl** = *<\_abc.\_abc\_data object>*

**create\_native()**

Create a native representation for this view.

This must be implemented in native views, but is only called internally. See also *native()*.

**klovve.drivers.viwid.views.formatted\_text module**

```
class klovve.drivers.viwid.views.formatted_text.View(model, view_factory)
```

Bases: *View*, *View*

**Parameters**

- **model** (*\_TGo*) –
- **view\_factory** (*klovve.view.tree.ViewFactory*) –

**\_abc\_impl** = *<\_abc.\_abc\_data object>*

**create\_native()**

Create a native representation for this view.

This must be implemented in native views, but is only called internally. See also *native()*.

**klovve.drivers.viuid.views.header module**

```
class klovve.drivers.viuid.views.header.View(model, view_factory)
```

Bases: [View](#), [View](#)

**Parameters**

- **model** ([\\_TGo](#)) –
- **view\_factory** ([klovve.view.tree.ViewFactory](#)) –

```
_abc_impl = <_abc._abc_data object>
```

```
create_native()
```

Create a native representation for this view.

This must be implemented in native views, but is only called internally. See also [native\(\)](#).

**klovve.drivers.viuid.views.headered\_panel module**

```
class klovve.drivers.viuid.views.headered_panel.Model(**kwargs)
```

Bases: [Model](#)

```
_abc_impl = <_abc._abc_data object>
```

```
property title_label_background
```

Return the property value.

**Parameters**

- **obj** – The object to get the value from.

```
class klovve.drivers.viuid.views.headered_panel.View(model, view_factory)
```

Bases: [View](#), [View](#)

**Parameters**

- **model** ([\\_TGo](#)) –
- **view\_factory** ([klovve.view.tree.ViewFactory](#)) –

```
_abc_impl = <_abc._abc_data object>
```

```
create_native()
```

Create a native representation for this view.

This must be implemented in native views, but is only called internally. See also [native\(\)](#).

**klovve.drivers.viuid.views.horizontal\_box module**

```
class klovve.drivers.viuid.views.horizontal_box.View(model, view_factory)
```

Bases: [View](#), [View](#)

**Parameters**

- **model** ([\\_TGo](#)) –
- **view\_factory** ([klovve.view.tree.ViewFactory](#)) –

```

    _abc_impl = <_abc._abc_data object>
    _orientation()

```

#### klovve.drivers.viwid.views.label module

```
class klovve.drivers.viwid.views.label.View(model, view_factory)
```

Bases: [View](#), [View](#)

##### Parameters

- **model** ([\\_TGoO](#)) –
- **view\_factory** ([klovve.view.tree.ViewFactory](#)) –

```

    _abc_impl = <_abc._abc_data object>

```

```
create_native()
```

Create a native representation for this view.

This must be implemented in native views, but is only called internally. See also [native\(\)](#).

#### klovve.drivers.viwid.views.list module

```
class klovve.drivers.viwid.views.list.View(model, view_factory)
```

Bases: [View](#), [View](#)

##### Parameters

- **model** ([\\_TGoO](#)) –
- **view\_factory** ([klovve.view.tree.ViewFactory](#)) –

```

    _abc_impl = <_abc._abc_data object>

```

```
create_native()
```

Create a native representation for this view.

This must be implemented in native views, but is only called internally. See also [native\(\)](#).

#### klovve.drivers.viwid.views.log\_pager module

```
class klovve.drivers.viwid.views.log_pager.View(*args, **kwargs)
```

Bases: [View](#), [View](#)

```
__inner_size_changed(scrollable)
```

##### Parameters

**scrollable** ([Scrollable](#)) –

```

    _abc_impl = <_abc._abc_data object>

```

```
create_native()
```

Create a native representation for this view.

This must be implemented in native views, but is only called internally. See also [native\(\)](#).

```
klovve.drivers.viwid.views.log_pager._time_text(d)
```

**klovve.drivers.viwid.views.placeholder module****class** klovve.drivers.viwid.views.placeholder.**View**(*model*, *view\_factory*)Bases: *View*, *View***Parameters**

- **model** (*\_TGo*) –
- **view\_factory** (*klovve.view.tree.ViewFactory*) –

**\_abc\_impl** = <\_abc.\_abc\_data object>**create\_native()**

Create a native representation for this view.

This must be implemented in native views, but is only called internally. See also *native()*.**klovve.drivers.viwid.views.progress module****class** klovve.drivers.viwid.views.progress.**View**(*model*, *view\_factory*)Bases: *View*, *View***Parameters**

- **model** (*\_TGo*) –
- **view\_factory** (*klovve.view.tree.ViewFactory*) –

**\_abc\_impl** = <\_abc.\_abc\_data object>**create\_native()**

Create a native representation for this view.

This must be implemented in native views, but is only called internally. See also *native()*.**klovve.drivers.viwid.views.scrollable module****class** klovve.drivers.viwid.views.scrollable.**View**(*model*, *view\_factory*)Bases: *View*, *View***Parameters**

- **model** (*\_TGo*) –
- **view\_factory** (*klovve.view.tree.ViewFactory*) –

**\_abc\_impl** = <\_abc.\_abc\_data object>**create\_native()**

Create a native representation for this view.

This must be implemented in native views, but is only called internally. See also *native()*.

**klovve.drivers.viwid.views.slider module**

```
class klovve.drivers.viwid.views.slider.View(model, view_factory)
```

Bases: *View*, *View*

**Parameters**

- **model** (*\_TGo*) –
- **view\_factory** (*klovve.view.tree.ViewFactory*) –

**\_abc\_impl** = *<\_abc.\_abc\_data object>*

**create\_native()**

Create a native representation for this view.

This must be implemented in native views, but is only called internally. See also *native()*.

**klovve.drivers.viwid.views.split module**

```
class klovve.drivers.viwid.views.split.View(model, view_factory)
```

Bases: *View*, *View*

**Parameters**

- **model** (*\_TGo*) –
- **view\_factory** (*klovve.view.tree.ViewFactory*) –

**\_abc\_impl** = *<\_abc.\_abc\_data object>*

**create\_native()**

Create a native representation for this view.

This must be implemented in native views, but is only called internally. See also *native()*.

**klovve.drivers.viwid.views.text\_field module**

```
class klovve.drivers.viwid.views.text_field.View(model, view_factory)
```

Bases: *View*, *View*

**Parameters**

- **model** (*\_TGo*) –
- **view\_factory** (*klovve.view.tree.ViewFactory*) –

**\_abc\_impl** = *<\_abc.\_abc\_data object>*

**create\_native()**

Create a native representation for this view.

This must be implemented in native views, but is only called internally. See also *native()*.

**klovve.drivers.viwid.views.vertical\_box module**

```
class klovve.drivers.viwid.views.vertical_box.View(model, view_factory)
```

Bases: [View](#), [View](#)

**Parameters**

- **model** (*\_TGo*) –
- **view\_factory** ([klovve.view.tree.ViewFactory](#)) –

**\_abc\_impl** = *<\_abc.\_abc\_data object>*

**\_orientation()**

**klovve.drivers.viwid.views.window module**

```
class klovve.drivers.viwid.views.window.ActionContext(main_window, *args)
```

Bases: [ActionContext](#)

**\_abc\_impl** = *<\_abc.\_abc\_data object>*

**async \_create\_dialog**(*view, done\_future*)

TODO Create a dialog with the given view content, and wait until *done\_future* is done. Implemented by subclasses, but not directly used from outside.

**Parameters**

- **view** – The view to show in a dialog.
- **done\_future** – The future to wait for before closing the dialog again.

```
class klovve.drivers.viwid.views.window.View(model, view_factory)
```

Bases: [View](#), [View](#)

**Parameters**

- **model** (*\_TGo*) –
- **view\_factory** ([klovve.view.tree.ViewFactory](#)) –

**\_abc\_impl** = *<\_abc.\_abc\_data object>*

**create\_native()**

Create a native representation for this view.

This must be implemented in native views, but is only called internally. See also [native\(\)](#).

**klovve.drivers.viwid.viwid package**

```
class klovve.drivers.viwid.viwid.Point(x, y)
```

Bases: [object](#)

**Parameters**

- **x** (*int*) –
- **y** (*int*) –

**moved\_by**(*x=0, y=0*)

**Parameters**

- **x** (*int*) –
- **y** (*int*) –

**Return type**

*Point*

**with\_x**(*x*)

**Parameters**

- **x** (*int*) –

**Return type**

*Point*

**with\_y**(*y*)

**Parameters**

- **y** (*int*) –

**Return type**

*Point*

**property x:** *int*

**property y:** *int*

**class** klovve.drivers.viuid.viuid.**Rectangle**(*from\_point, to*)

Bases: *object*

**Parameters**

- **from\_point** (*Point*) –
- **to** (*Union[Point, Size]*) –

**property bottom\_right:** *Point*

**property bottom\_y:** *int*

**property height:** *int*

**property left\_x:** *int*

**property right\_x:** *int*

**property top\_left:** *Point*

**property top\_y:** *int*

**property width:** *int*

**class** klovve.drivers.viuid.viuid.**Size**(*width, height*)

Bases: *object*

**Parameters**

- **width** (*int*) –
- **height** (*int*) –

**extend\_by**(*width=0, height=0*)

**Parameters**

- **width** (*int*) –
- **height** (*int*) –

**Return type**

[Size](#)

**property height:** `int`

**property width:** `int`

**with\_height**(*height*)

**Parameters**

**height** (*int*) –

**Return type**

[Size](#)

**with\_width**(*width*)

**Parameters**

**width** (*int*) –

**Return type**

[Size](#)

## Subpackages

**klovve.drivers.viwid.viwid.+samples namespace**

## Submodules

**klovve.drivers.viwid.viwid.+samples.hello\_world module**

**klovve.drivers.viwid.viwid.+samples.hello\_world2 module**

**klovve.drivers.viwid.viwid.drivers package**

**klovve.drivers.viwid.viwid.drivers.get\_driver**(*event\_loop=None*)

**Return type**

[Driver](#)



## Submodules

### klovve.drivers.viuid.viuid.drivers.curses module

```
class klovve.drivers.viuid.viuid.drivers.curses.Driver(event_loop)
```

Bases: Driver

`__requests_queue()`

`_abc_impl = <_abc._abc_data object>`

`add_widget_screen(widget)`

`canvas(widget)`

property `curses_colors`

property `loop`

`process_curses_input()`

`refresh_curses_window()`

`repaint_widget_soon(widget)`

`resize_widget_soon(widget)`

property `screens`

```
class klovve.drivers.viuid.viuid.drivers.curses._Canvas(driver, window, widget)
```

Bases: Canvas

#### Parameters

- **driver** ([Driver](#)) –
- **window** (*window*) –
- **widget** ([Widget](#)) –

`cursor(point)`

`fill(color, rectangle=None)`

`text(text, rectangle=None)`

```
class klovve.drivers.viuid.viuid.drivers.curses._CursesColors
```

Bases: object

```
class Mode(value, names=None, *, module=None, qualname=None, type=None, start=1, boundary=None)
```

Bases: Enum

`COLORS_256 = 2`

`COLORS_8 = 1`

`_CursesColors__curses_color(color)`

#### Parameters

- color** ([Color](#)) –

`color_pair(fg, bg)`

**Parameters**

- `fg` (`Union[FullColor, Color, str]`) –
- `bg` (`Union[FullColor, Color, str]`) –

`colors_by_pair(pair)`

`curses_colors = [0, 4, 2, 6, 1, 5, 3, 7]`

`class klovve.drivers.viwid.viwid.drivers.curses._Loop(event_loop, driver)`

Bases: `Loop`

**Parameters**

`driver` (`Driver`) –

`async run()`

**Return type**

`None`

`class klovve.drivers.viwid.viwid.drivers.curses._Screen(curses_screen, driver)`

Bases: `Screen`

**Parameters**

- `curses_screen` (`window`) –
- `driver` (`Driver`) –

property `focussed_widget`

property `size`: `Size`

`update_curses_screensize()`

## **klovve.drivers.viwid.viwid.drivers.driver module**

`class klovve.drivers.viwid.viwid.drivers.driver.Driver`

Bases: `ABC`

`_abc_impl = <_abc._abc_data object>`

`abstract add_widget_screen(widget)`

**Parameters**

`widget` (`Widget`) –

**Return type**

`Screen`

`abstract canvas(widget)`

**Parameters**

`widget` (`Widget`) –

**Return type**

`Canvas`

**abstract property loop:** [Loop](#)

**abstract repaint\_widget\_soon**(*widget*)

**Parameters**

**widget** ([Widget](#)) –

**Return type**

None

**abstract resize\_widget\_soon**(*widget*)

**Parameters**

**widget** ([Widget](#)) –

**Return type**

None

**abstract property screens:** [Iterable](#)[[Screen](#)]

**klovve.drivers.viuid.viuid.event package**

**Submodules**

**klovve.drivers.viuid.viuid.event.base module**

**class** [klovve.drivers.viuid.viuid.event.base.OngoingEvent](#)(*name*)

Bases: [object](#)

**Parameters**

**name** ([str](#)) –

**property** [is\\_propagation\\_stopped](#): [bool](#)

**property** [name](#)

**stop\_propagation**()

**Return type**

None

**klovve.drivers.viuid.viuid.event.input module**

**class** [klovve.drivers.viuid.viuid.event.input.OngoingKeyboardEvent](#)(*key*)

Bases: [OngoingEvent](#)

**property** [key](#)

**class** [klovve.drivers.viuid.viuid.event.input.OngoingMouseEvent](#)(*screen\_position*,  
*left\_button*, *right\_button*)

Bases: [OngoingEvent](#)

**Parameters**

- **screen\_position** ([Point](#)) –
- **left\_button** ([bool](#)) –

- `right_button (bool)` –

`property left_button: bool`

`property right_button: bool`

`property screen_position: Point`

`class klovve.drivers.viwid.viwid.event.input.OngoingScrollEvent(screen_position, direction)`  
Bases: `OngoingEvent`

**Parameters**

- `screen_position (Point)` –
- `direction (int)` –

`property direction: int`

`property screen_position: Point`

## `klovve.drivers.viwid.viwid.widgets` package

### Submodules

#### `klovve.drivers.viwid.viwid.widgets.box` module

`class klovve.drivers.viwid.viwid.widgets.box.Box(**kwargs)`  
Bases: `Widget`

`__item_added(index, item)`

**Parameters**

- `index (int)` –
- `item (object)` –

**Return type**

None

`__item_removed(index)`

**Parameters**

`index (int)` –

**Return type**

None

`_partitioning2()`

`property children`

`property orientation`

`class klovve.drivers.viwid.viwid.widgets.box.CBox(**kwargs)`  
Bases: `Box`

`property class_style_name`

---

```
class klovve.drivers.viwid.viwid.widgets.box.Orientation(value, names=None, *, module=None,
                                                         qualname=None, type=None, start=1,
                                                         boundary=None)
```

```
    Bases: Enum
```

```
    HORIZONTAL = 2
```

```
    VERTICAL = 1
```

### klovve.drivers.viwid.viwid.widgets.busy\_animation module

```
class klovve.drivers.viwid.viwid.widgets.busy_animation.BusyAnimation(**kwargs)
```

```
    Bases: Label
```

```
    __next()
```

### klovve.drivers.viwid.viwid.widgets.button module

```
class klovve.drivers.viwid.viwid.widgets.button.Button(**kwargs)
```

```
    Bases: Label
```

```
    _keyboard_event(event)
```

```
    _mouse_click_event(event)
```

```
    property class_style_name
```

```
    property focusable
```

```
    property on_click
```

### klovve.drivers.viwid.viwid.widgets.check\_button module

```
class klovve.drivers.viwid.viwid.widgets.check_button.CheckButton(**kwargs)
```

```
    Bases: Button
```

```
    _keyboard_event(event)
```

```
    _mouse_click_event(event)
```

```
    property checked
```

```
    compute_height(width, minimal)
```

```
    compute_width(minimal)
```

```
    paint(canvas)
```

**klovve.drivers.viuid.viuid.widgets.dropdown module**

```
class klovve.drivers.viuid.viuid.widgets.dropdown.DropDown(**kwargs)
    Bases: Button
    __click()
    _select(i)
    property items
    property selected_index
    property selected_item
```

**klovve.drivers.viuid.viuid.widgets.entry module**

```
class klovve.drivers.viuid.viuid.widgets.entry.Entry(**kwargs)
    Bases: Widget
    _keyboard_event(event)
    property class_style_name
    compute_height(width, minimal)

        Parameters
            width (int) –

        Return type
            int

    compute_width(minimal)

        Return type
            int

    property cursor_position
    property focusable
    paint(canvas)
    property text
```

**klovve.drivers.viuid.viuid.widgets.frame module**

```
class klovve.drivers.viuid.viuid.widgets.frame.Frame(**kwargs)
    Bases: Box
    compute_height(width, minimal)
    compute_width(minimal)
    property item
```

**paint**(*canvas*)

**set\_size**(*v*)

#### klovve.drivers.viwid.viwid.widgets.label module

**class** klovve.drivers.viwid.viwid.widgets.label.**Label**(\*\**kwargs*)

Bases: Widget

**\_\_find\_last\_whitespace**(*i\_from*, *i\_to*)

**\_text\_in\_width**(*width*)

**compute\_height**(*width*, *minimal*)

**Parameters**

**width** (*int*) –

**Return type**

int

**compute\_width**(*minimal*)

**Return type**

int

**paint**(*canvas*)

**property** text

#### klovve.drivers.viwid.viwid.widgets.list module

**class** klovve.drivers.viwid.viwid.widgets.list.**List**(\*\**kwargs*)

Bases: Box

**\_keyboard\_event**(*event*)

**\_mouse\_click\_event**(*event*)

**property** class\_style\_name

**property** focusable

**property** items

**property** on\_selection\_activated

**property** selected\_item\_index

**class** klovve.drivers.viwid.viwid.widgets.list.**ListRow**(\*\**kwargs*)

Bases: Label

**property** class\_style\_name

**property** is\_selected

**klovve.drivers.viuid.viuid.widgets.progress\_bar module**

```
class klovve.drivers.viuid.viuid.widgets.progress_bar.ProgressBar(**kwargs)
    Bases: Label
    property class_style_name
    compute_height(width, minimal)
    compute_width(minimal)
    paint(canvas)
    property value
```

**klovve.drivers.viuid.viuid.widgets.radio\_button module**

```
class klovve.drivers.viuid.viuid.widgets.radio_button.RadioButton(**kwargs)
    Bases: Button
```

**klovve.drivers.viuid.viuid.widgets.scrollable module**

```
class klovve.drivers.viuid.viuid.widgets.scrollable.Scrollable(**kwargs)
    Bases: Box
    __correct_offset()
    __move_offset(x, y)
    _preview_keyboard_event(event)
    _preview_scroll_event(event)
    compute_height(width, minimal)
    compute_width(minimal)
    property focusable
    property horizontally_scrollable
    property item
    set_size(v)
    property vertically_scrollable
```



**klovve.drivers.viuid.viuid.widgets.widget module**

```
class klovve.drivers.viuid.viuid.widgets.widget.Alignment(value, names=None, *, module=None,
qualname=None, type=None, start=1,
boundary=None)
```

Bases: Enum

**CENTER** = 3

**FILL** = 4

**START** = 1

**STOP** = 2

```
klovve.drivers.viuid.viuid.widgets.widget.ListProperty(item_added_func, item_removed_func,
item_replaced_func=None)
```

```
klovve.drivers.viuid.viuid.widgets.widget.Property(setter=None, /, *, default=<function
<lambda>>>)
```

```
class klovve.drivers.viuid.viuid.widgets.widget.Widget(**kwargs)
```

Bases: object

```
ListProperty(item_removed_func, item_replaced_func=None)
```

```
class Parti(widget)
```

Bases: object

**focussed()**

**jumpto**(frm, delta)

```
Property(*, default=<function <lambda>>>)
```

```
__item_added(index, item)
```

**Parameters**

- **index** (int) –
- **item** (object) –

**Return type**

None

```
__item_removed(index)
```

**Parameters**

**index** (int) –

**Return type**

None

**property** \_children

```
_internal_mouse_click_event(event)
```

**Parameters**

**event** (OngoingMouseClickedEvent) –

**Return type**

None

**\_keyboard\_event**(*event*)**Parameters****event** ([OngoingKeyboardEvent](#)) –**Return type**

None

**\_mouse\_click\_event**(*event*)**Parameters****event** ([OngoingMouseClickedEvent](#)) –**Return type**

None

**\_preview\_internal\_mouse\_click\_event**(*event*)**Parameters****event** ([OngoingMouseClickedEvent](#)) –**Return type**

None

**\_preview\_keyboard\_event**(*event*)**Parameters****event** ([OngoingKeyboardEvent](#)) –**Return type**

None

**\_preview\_mouse\_click\_event**(*event*)**Parameters****event** ([OngoingMouseClickedEvent](#)) –**Return type**

None

**\_preview\_scroll\_event**(*event*)**Parameters****event** ([OngoingScrollEvent](#)) –**Return type**

None

**\_scroll\_event**(*event*)**Parameters****event** ([OngoingScrollEvent](#)) –**Return type**

None

**alive**()**property background**

**canvas()**

**child\_at\_position**(*xx*, *yy*)

**property class\_style\_name**

**compute\_height**(*width*, *minimal*)

**Parameters**

- **width** (*int*) –
- **minimal** (*bool*) –

**Return type**

*int*

**compute\_height2**(*width*, *minimal*)

**Parameters**

- **width** (*int*) –
- **minimal** (*bool*) –

**Return type**

*int*

**compute\_width**(*minimal*)

**Parameters**

**minimal** (*bool*) –

**Return type**

*int*

**compute\_width2**(*minimal*)

**Parameters**

**minimal** (*bool*) –

**Return type**

*int*

**property computed\_horizontal\_expand\_greedily**

**property computed\_is\_disabled**

**property computed\_vertical\_expand\_greedily**

**dematerialize()**

**Return type**

*None*

**property focusable**

**property foreground**

**property horizontal\_alignment**

**property horizontal\_expand\_greedily**

**property inner\_size:** *Size*

**property is\_disabled**

**property layout**

**listen\_property**(*key, func*)

**materialize**(*screen*)

**Return type**

None

**property offset:** *Point*

**on\_dematerialized**()

**Return type**

None

**on\_materialized**()

**Return type**

None

**paint**(*canvas*)

**paint2**(*parent\_canvas=None*)

**property parent:** Optional[*Widget*]

**property position:** *Point*

**request\_repaint**()

**Return type**

None

**request\_resize**()

**Return type**

None

**property root\_style**

**property screen:** *Screen*

**set\_inner\_size**(*value*)

**Parameters**

**value** (*Size*) –

**Return type**

None

**set\_layout**(*value*)

**set\_offset**(*value*)

**Parameters**

**value** (*Point*) –

**Return type**

None

**set\_parent**(*value*)**Parameters****value** (*Optional*[*Widget*]) –**Return type**

None

**set\_position**(*value*)**Parameters****value** (*Point*) –**Return type**

None

**set\_root\_style**(*v*)**set\_size**(*value*, *setinner*=*True*)**Parameters****value** (*Size*) –**Return type**

None

**property size:** *Size***property style\_atom****style\_atom\_by\_name**(*name*)**try\_focus**()**property vertical\_alignment****property vertical\_expand\_greedily**

```
class klovve.drivers.viwid.viwid.widgets.widget._ListProperty(setter, item_added_func,
                                                             item_removed_func,
                                                             item_replaced_func=None)
```

Bases: *property***\_fget**(*obj*)**\_fset**(*obj*, *value*)

```
class klovve.drivers.viwid.viwid.widgets.widget._Property(setter, default=<function
                                                             _Property.<lambda>>)
```

Bases: *property***\_fget**(*obj*)**\_fset**(*obj*, *value*)

## Submodules

### klovve.drivers.viuid.viuid.canvas module

**class** klovve.drivers.viuid.viuid.canvas.Canvas

Bases: object

**class** Color(*r: float, g: float, b: float*)

Bases: object

**Parameters**

- *r (float)* –
- *g (float)* –
- *b (float)* –

**property** a

**b:** float

**static** from\_string(*s*)

**Parameters**

*s* (*Union*[Color, str]) –

**Return type**

Color

**g:** float

**r:** float

**class** FullColor(*rich: klovve.drivers.viuid.viuid.canvas.Color, base: klovve.drivers.viuid.viuid.canvas.Color*)

Bases: object

**Parameters**

- *rich (Color)* –
- *base (Color)* –

**base:** Color

**static** from\_string(*s*)

**Parameters**

*s* (*Union*[Color, FullColor, str]) –

**Return type**

FullColor

**rich:** Color

**cursor**(*point*)

**Parameters**

*point* (Point) –

**Return type**

None

**fill**(*color*, *rectangle*=None)

**Parameters**

- **color** (*Union*[[Color](#), [FullColor](#), *str*]) –
- **rectangle** (*Optional*[[Rectangle](#)]) –

**Return type**

None

**text**(*text*, *rectangle*=None)

**Parameters**

- **rectangle** (*Optional*[[Rectangle](#)]) –

**Return type**

None

**class** klovve.drivers.viuid.viuid.canvas.**Color**(*r*: float, *g*: float, *b*: float)

Bases: object

**Parameters**

- **r** (*float*) –
- **g** (*float*) –
- **b** (*float*) –

**property** a

**b**: float

**static** from\_string(*s*)

**Parameters**

- **s** (*Union*[[Color](#), *str*]) –

**Return type**

[Color](#)

**g**: float

**r**: float

**class** klovve.drivers.viuid.viuid.canvas.**FullColor**(*rich*: klovve.drivers.viuid.viuid.canvas.Color, *base*: klovve.drivers.viuid.viuid.canvas.Color)

Bases: object

**Parameters**

- **rich** ([Color](#)) –
- **base** ([Color](#)) –

**base**: [Color](#)

**static** from\_string(*s*)

**Parameters**

- **s** (*Union*[[Color](#), [FullColor](#), *str*]) –

**Return type**

[FullColor](#)

rich: *Color*

### klovve.drivers.viwid.viwid.data module

**class** klovve.drivers.viwid.viwid.data.**List**(v=())

Bases: MutableSequence

**\_\_correct\_index**(i)

Parameters

**i** (int) –

Return type

int

**\_abc\_impl** = <\_abc.\_abc\_data object>

**\_add\_observer**(item\_added\_func, item\_removed\_func, item\_replaced\_func=None, \*, initialize=True)

Parameters

**initialize** (bool) –

Return type

None

**\_remove\_observer**(observer)

Parameters

**observer** (ListObserver) –

Return type

None

**insert**(i, v)

S.insert(index, value) – insert value before index

**update**(other)

**class** klovve.drivers.viwid.viwid.data.**ListObserver**

Bases: object

**item\_added**(index, item)

Parameters

• **index** (int) –

• **item** (object) –

Return type

None

**item\_removed**(index)

Parameters

**index** (int) –

Return type

None



**item\_replaced**(*index*, *item*)

**Return type**

None

## klovve.drivers.viuid.viuid.layout module

**class** klovve.drivers.viuid.viuid.layout.**GridLayout**(*partitioning*)

Bases: *Layout*

**Parameters**

**partitioning** (*Callable*[[*list*[*list*['viuid.widgets.widget.Widget']]] –  
**\_\_horizontally\_process\_partitioning**(*minimal*)

**Parameters**

**partitioning** (*list*[*list*[*viuid.widgets.widget.Widget*]]) –  
**\_\_process\_partitioning**(*size*, *fu*=None)  
**\_\_stretch\_axis\_finally**(*stretch\_axis*, *target\_size*)  
**\_\_stretch\_axis\_minimals**(*axis\_minimal*, *target\_size*)  
**\_\_stretch\_from\_partitioning**()

**Parameters**

**partitioning** (*list*[*list*[*viuid.widgets.widget.Widget*]]) –  
**\_\_trim\_geometry**(*position*, *size*, *total\_size*)  
**\_\_vertically\_process\_partitioning**(*minimal*, *cols*)

**Parameters**

- **partitioning** (*list*[*list*[*viuid.widgets.widget.Widget*]]) –
- **cols** (*list*[*int*]) –

**compute\_height**(*width*, *minimal*)

**Parameters**

- **width** (*int*) –
- **minimal** (*bool*) –

**Return type**

int

**compute\_width**(*minimal*)

**Parameters**

**minimal** (*bool*) –

**Return type**

int

**set\_size**(*value*)

```
class klovve.drivers.viuid.viuid.layout.Layout
```

```
    Bases: object
```

```
    compute_height(width, minimal)
```

```
        Parameters
```

```
            minimal (bool) –
```

```
    compute_width(minimal)
```

```
        Parameters
```

```
            minimal (bool) –
```

```
    set_size(value)
```

```
class klovve.drivers.viuid.viuid.layout.NullLayout
```

```
    Bases: object
```

```
    compute_height(width, minimal)
```

```
    compute_width(minimal)
```

```
    set_size(value)
```

### klovve.drivers.viuid.viuid.loop module

```
class klovve.drivers.viuid.viuid.loop.Loop(event_loop=None)
```

```
    Bases: object
```

```
        Parameters
```

```
            event_loop (Optional[BaseEventLoop]) –
```

```
    property event_loop: BaseEventLoop
```

```
    async run()
```

```
        Return type
```

```
            None
```

### klovve.drivers.viuid.viuid.screen module

```
class klovve.drivers.viuid.viuid.screen.AnchorRootAlignmentPositioning(anchor_widget)
```

```
    Bases: RootAlignmentPositioning
```

```
        Parameters
```

```
            anchor_widget (Widget) –
```

```
    _abc_impl = <_abc._abc_data object>
```

```
    get_x(widget, screen_size)
```

```
    get_y(widget, screen_size)
```

```
class klovve.drivers.viuid.viuid.screen.AutoRootAlignmentSizing(minimal_width=False,  
                                                                minimal_height=False)
```

```
    Bases: RootAlignmentSizing
```

```

    _abc_impl = <_abc._abc_data object>
    get_height(width, widget, screen_size)
    get_width(widget, screen_size)
class klovve.drivers.viwid.viwid.screen.ByScreenRootAlignmentSizing(width_fraction=0.5,
                                                                    height_fraction=0.5)
    Bases: RootAlignmentSizing
    _abc_impl = <_abc._abc_data object>
    get_height(width, widget, screen_size)
    get_width(widget, screen_size)
class klovve.drivers.viwid.viwid.screen.CenterRootAlignmentPositioning
    Bases: RootAlignmentPositioning
    _abc_impl = <_abc._abc_data object>
    get_x(widget, screen_size)
    get_y(widget, screen_size)
class klovve.drivers.viwid.viwid.screen.FixedRootAlignmentPositioning(x=0, y=0)
    Bases: RootAlignmentPositioning
    _abc_impl = <_abc._abc_data object>
    get_x(widget, screen_size)
    get_y(widget, screen_size)
class klovve.drivers.viwid.viwid.screen.FixedRootAlignmentSizing(width=1, height=1)
    Bases: RootAlignmentSizing
    _abc_impl = <_abc._abc_data object>
    get_height(width, widget, screen_size)
    get_width(widget, screen_size)
class klovve.drivers.viwid.viwid.screen.RootAlignment(horizontal_positioning, vertical_positioning,
                                                        horizontal_sizing, vertical_sizing)
    Bases: object
    apply(widget)

    Parameters
        widget (Widget) –
class klovve.drivers.viwid.viwid.screen.RootAlignmentPositioning
    Bases: ABC
    _abc_impl = <_abc._abc_data object>

```

**abstract** `get_x(widget, screen_size)`

**Parameters**

- `widget` (`Widget`) –
- `screen_size` (`Size`) –

**Return type**

`int`

**abstract** `get_y(widget, screen_size)`

**Parameters**

- `widget` (`Widget`) –
- `screen_size` (`Size`) –

**Return type**

`int`

**class** `klovve.drivers.viwid.viwid.screen.RootAlignmentSizing`

Bases: `ABC`

`_abc_impl` = `<_abc._abc_data object>`

**abstract** `get_height(width, widget, screen_size)`

**Parameters**

- `width` (`int`) –
- `widget` (`Widget`) –
- `screen_size` (`Size`) –

**Return type**

`int`

**abstract** `get_width(widget, screen_size)`

**Parameters**

- `widget` (`Widget`) –
- `screen_size` (`Size`) –

**Return type**

`int`

**class** `klovve.drivers.viwid.viwid.screen.Screen(driver)`

Bases: `object`

**Parameters**

`driver` (`viwid.drivers.driver.Driver`) –

`__handle_event__tab_key(reverse)`

**Parameters**

- `widget` (`Widget`) –
- `reverse` (`bool`) –

**Return type**

`None`

**add\_root**(*widget*, *alignment=None*, \*, *root\_style\_name='main'*)

**Parameters**

- **widget** (*Widget*) –
- **alignment** (*Optional[RootAlignment]*) –
- **root\_style\_name** (*str*) –

**Return type**

TODO

**property bottommost:** *Optional[Widget]*

**property driver**

**abstract property focussed\_widget:** *Optional[Widget]*

**static handle\_event**(*widget*, *ongoing\_event*)

**Parameters**

- **widget** (*Widget*) –
- **ongoing\_event** (*OngoingEvent*) –

**Return type**

None

**popup**(*widget*, *alignment=None*, \*, *root\_style\_name='popup'*)

**Parameters**

- **widget** (*Widget*) –
- **alignment** (*Optional[RootAlignment]*) –
- **root\_style\_name** (*str*) –

**Return type**

TODO

**property roots:** *Iterable[Widget]*

**property roots\_and\_alignments:** *Iterable[tuple[viwid.widgets.widget.Widget, klovve.drivers.viwid.viwid.screen.RootAlignment]]*

**abstract property size:** *Size*

**property style:** *Style*

**property topmost:** *Optional[Widget]*

**translate\_coordinates**(*point*, \*, *old\_origin=None*, *new\_origin=None*)

**Parameters**

- **point** (*Point*) –
- **old\_origin** (*Optional[Widget]*) –
- **new\_origin** (*Optional[Widget]*) –

**Return type**

*Point*

**translate\_coordinates\_with\_visibility\_check**(*point*, \*, *old\_origin=None*, *new\_origin=None*)

**Parameters**

- **point** (*Point*) –
- **old\_origin** (*Optional[Widget]*) –
- **new\_origin** (*Optional[Widget]*) –

**Return type**

tuple[*viwid.Point*, bool]

**update()**

**Return type**

None

### klovve.drivers.viwid.viwid.style module

**class** klovve.drivers.viwid.viwid.style.**ClassStyle**(\*, *normal*, *focussed*, *disabled*)

Bases: object

**Parameters**

- **normal** (*StyleAtom*) –
- **focussed** (*StyleAtom*) –
- **disabled** (*StyleAtom*) –

**class** klovve.drivers.viwid.viwid.style.**RootStyle**(\*, *root*, *plain*, *control*, *entry*, *list*, *list\_item*, *selected*, *progress\_done*, *progress\_not\_done*)

Bases: object

**Parameters**

- **root** (*ClassStyle*) –
- **plain** (*ClassStyle*) –
- **control** (*ClassStyle*) –
- **entry** (*ClassStyle*) –
- **list** (*ClassStyle*) –
- **list\_item** (*ClassStyle*) –
- **selected** (*ClassStyle*) –
- **progress\_done** (*ClassStyle*) –
- **progress\_not\_done** (*ClassStyle*) –

**class** klovve.drivers.viwid.viwid.style.**Style**(\*, *main*, *popup*)

Bases: object

**Parameters**

- **main** (*RootStyle*) –
- **popup** (*RootStyle*) –

```
class klovve.drivers.viwid.viwid.style.StyleAtom(*, foreground, background=None)
    Bases: object
```

## klovve.pieces package

### Subpackages

## klovve.pieces.form package

```
class klovve.pieces.form.Model(**kwargs)
```

Bases: [Model](#)

```
_abc_impl = <_abc._abc_data object>
```

#### property items

Return the property value.

#### Parameters

**obj** – The object to get the value from.

```
class klovve.pieces.form.View(model, view_factory)
```

Bases: [BaseView\[Model\]](#)

#### Parameters

- **model** ([\\_TGo](#)) –
- **view\_factory** ([klovve.view.tree.ViewFactory](#)) –

```
_abc_impl = <_abc._abc_data object>
```

### Submodules

## klovve.pieces.form.section module

```
class klovve.pieces.form.section.Model(**kwargs)
```

Bases: [Model](#)

```
_abc_impl = <_abc._abc_data object>
```

#### property item

Return the property value.

#### Parameters

**obj** – The object to get the value from.

#### property label

Return the property value.

#### Parameters

**obj** – The object to get the value from.

```
class klovve.pieces.form.section.View(model, view_factory)
```

Bases: [BaseView\[Model\]](#)

**Parameters**

- **model** (*\_TGo*) –
- **view\_factory** ([klovve.view.tree.ViewFactory](#)) –

**\_abc\_impl** = *<\_abc.\_abc\_data object>*

## **klovve.pieces.interact package**

```
class klovve.pieces.interact.Model(**kwargs)
```

Bases: [Model](#)

**\_abc\_impl** = *<\_abc.\_abc\_data object>*

**property answer**

Return the property value.

**Parameters**

**obj** – The object to get the value from.

**property is\_answered**

Return the property value.

**Parameters**

**obj** – The object to get the value from.

**set\_answer**(*answer*)

## **Submodules**

### **klovve.pieces.interact.abstract module**

```
class klovve.pieces.interact.abstract.Model(**kwargs)
```

Bases: [Model](#)

**\_abc\_impl** = *<\_abc.\_abc\_data object>*

**property get\_answer\_func**

Return the property value.

**Parameters**

**obj** – The object to get the value from.

**property inner\_view**

Return the property value.

**Parameters**

**obj** – The object to get the value from.



**property message**

Return the property value.

**Parameters**

**obj** – The object to get the value from.

**property triggers**

Return the property value.

**Parameters**

**obj** – The object to get the value from.

```
class klovve.pieces.interact.abstract.View(model, view_factory)
```

Bases: [BaseView\[Model\]](#)

**Parameters**

- **model** ([\\_TGo](#)) –
- **view\_factory** ([klovve.view.tree.ViewFactory](#)) –

**\_abc\_impl** = [<\\_abc.\\_abc\\_data object>](#)

**klovve.pieces.interact.message module**

```
class klovve.pieces.interact.message.Model(**kwargs)
```

Bases: [Model](#)

**\_abc\_impl** = [<\\_abc.\\_abc\\_data object>](#)

**property message**

Return the property value.

**Parameters**

**obj** – The object to get the value from.

```
class klovve.pieces.interact.message.View(*args, **kwargs)
```

Bases: [ComposedView\[Model\]](#)

**\_abc\_impl** = [<\\_abc.\\_abc\\_data object>](#)

**compose()**

**klovve.pieces.interact.textline module**

```
class klovve.pieces.interact.textline.Model(**kwargs)
```

Bases: [Model](#)

**\_abc\_impl** = [<\\_abc.\\_abc\\_data object>](#)

**property message**

Return the property value.

**Parameters**

**obj** – The object to get the value from.

**property suggestion**

Return the property value.

**Parameters**

**obj** – The object to get the value from.

```
class klovve.pieces.interact.textline.View(*args, **kwargs)
```

Bases: [ComposedView](#)[\[Model\]](#)

```
class TModel(**kwargs)
```

Bases: [Model](#)

```
_abc_impl = <_abc._abc_data object>
```

**property value**

Return the property value.

**Parameters**

**obj** – The object to get the value from.

```
_abc_impl = <_abc._abc_data object>
```

```
compose()
```

**klovve.pieces.interact.triggers module**

```
class klovve.pieces.interact.triggers.Model(**kwargs)
```

Bases: [Model](#)

```
_abc_impl = <_abc._abc_data object>
```

**property message**

Return the property value.

**Parameters**

**obj** – The object to get the value from.

**property triggers**

Return the property value.

**Parameters**

**obj** – The object to get the value from.

```
class klovve.pieces.interact.triggers.View(*args, **kwargs)
```

Bases: [ComposedView](#)[\[Model\]](#)

```
_abc_impl = <_abc._abc_data object>
```

```
compose()
```

**klovve.pieces.interact.yesno module**

```
class klovve.pieces.interact.yesno.Model(**kwargs)
    Bases: Model
    _abc_impl = <_abc._abc_data object>
    property message
        Return the property value.
    Parameters
        obj – The object to get the value from.
class klovve.pieces.interact.yesno.View(*args, **kwargs)
    Bases: ComposedView[Model]
    _abc_impl = <_abc._abc_data object>
    compose()
```

**klovve.pieces.log\_pager package**

```
class klovve.pieces.log_pager.Model(**kwargs)
    Bases: Model
    _abc_impl = <_abc._abc_data object>
    property boff
        Return the property value.
    Parameters
        obj – The object to get the value from.
    property entries
        Return the property value.
    Parameters
        obj – The object to get the value from.
class klovve.pieces.log_pager.View(model, view_factory)
    Bases: BaseView[Model]
    Parameters
        • model (_TGo) –
        • view_factory (klovve.view.tree.ViewFactory) –
    _abc_impl = <_abc._abc_data object>
```

## Submodules

### klovve.pieces.log\_pager.entry module

**class** klovve.pieces.log\_pager.entry.**Model**(\*\*kwargs)

Bases: *Model*

**\_abc\_impl** = <\_abc.\_abc\_data object>

**property** **began\_at**

Return the property value.

**Parameters**

**obj** – The object to get the value from.

**property** **began\_at\_\_text**

Return the property value.

**Parameters**

**obj** – The object to get the value from.

**property** **ended\_at**

Return the property value.

**Parameters**

**obj** – The object to get the value from.

**property** **ended\_at\_\_text**

Return the property value.

**Parameters**

**obj** – The object to get the value from.

**property** **entries**

Return the property value.

**Parameters**

**obj** – The object to get the value from.

**property** **message**

Return the property value.

**Parameters**

**obj** – The object to get the value from.

**property** **only\_single\_time**

Return the property value.

**Parameters**

**obj** – The object to get the value from.

**property** **only\_verbose**

Return the property value.

**Parameters**

**obj** – The object to get the value from.

```
class klovve.pieces.log_pager.entry.View(model, view_factory)
```

Bases: [BaseView\[Model\]](#)

#### Parameters

- **model** (*\_TGo*) –
- **view\_factory** (*klovve.view.tree.ViewFactory*) –

**\_abc\_impl** = *<\_abc.\_abc\_data object>*

```
klovve.pieces.log_pager.entry._time_text(d)
```

### klovve.pieces.property\_panel package

```
class klovve.pieces.property_panel.Model(**kwargs)
```

Bases: [Model](#)

**\_abc\_impl** = *<\_abc.\_abc\_data object>*

#### property entries

Return the property value.

#### Parameters

**obj** – The object to get the value from.

#### property values

Return the property value.

#### Parameters

**obj** – The object to get the value from.

```
class klovve.pieces.property_panel.View(model, view_factory)
```

Bases: [BaseView\[Model\]](#)

#### Parameters

- **model** (*\_TGo*) –
- **view\_factory** (*klovve.view.tree.ViewFactory*) –

**\_abc\_impl** = *<\_abc.\_abc\_data object>*

### Submodules

#### klovve.pieces.property\_panel.entry module

```
class klovve.pieces.property_panel.entry.Model(**kwargs)
```

Bases: [Model](#)

**\_abc\_impl** = *<\_abc.\_abc\_data object>*

#### property label

Return the property value.

#### Parameters

**obj** – The object to get the value from.

**property name**

Return the property value.

**Parameters**

**obj** – The object to get the value from.

**class** klovve.pieces.property\_panel.entry.**View**(*model*, *view\_factory*)

Bases: [BaseView](#)[\[Model\]](#)

**Parameters**

- **model** (*\_TGo*) –
- **view\_factory** ([klovve.view.tree.ViewFactory](#)) –

**\_abc\_impl** = **<\_abc.\_abc\_data object>**

**klovve.pieces.tabbed package**

**class** klovve.pieces.tabbed.**Model**(\*\**kwargs*)

Bases: [Model](#)

**\_abc\_impl** = **<\_abc.\_abc\_data object>**

**property items**

Return the property value.

**Parameters**

**obj** – The object to get the value from.

**class** klovve.pieces.tabbed.**View**(*model*, *view\_factory*)

Bases: [BaseView](#)[\[Model\]](#)

**Parameters**

- **model** (*\_TGo*) –
- **view\_factory** ([klovve.view.tree.ViewFactory](#)) –

**\_abc\_impl** = **<\_abc.\_abc\_data object>**

**Submodules****klovve.pieces.tabbed.tab module**

**class** klovve.pieces.tabbed.tab.**Model**(\*\**kwargs*)

Bases: [Model](#)

**\_abc\_impl** = **<\_abc.\_abc\_data object>**

**property item**

Return the property value.

**Parameters**

**obj** – The object to get the value from.

**property label**

Return the property value.

**Parameters**

**obj** – The object to get the value from.

**class** klovve.pieces.tabbed.tab.**View**(*model*, *view\_factory*)

Bases: [BaseView](#)[\[Model\]](#)

**Parameters**

- **model** (*\_TGo*) –
- **view\_factory** ([klovve.view.tree.ViewFactory](#)) –

**\_abc\_impl** = <\_abc.\_abc\_data object>

**klovve.pieces.viewer package****Submodules****klovve.pieces.viewer.image module**

**class** klovve.pieces.viewer.image.**Model**(\*\**kwargs*)

Bases: [Model](#)

**\_abc\_impl** = <\_abc.\_abc\_data object>

**property source**

Return the property value.

**Parameters**

**obj** – The object to get the value from.

**class** klovve.pieces.viewer.image.**View**(*model*, *view\_factory*)

Bases: [BaseView](#)[\[Model\]](#)

**Parameters**

- **model** (*\_TGo*) –
- **view\_factory** ([klovve.view.tree.ViewFactory](#)) –

**\_abc\_impl** = <\_abc.\_abc\_data object>

**klovve.pieces.viewer.pdf module**

**class** klovve.pieces.viewer.pdf.**Model**(\*\**kwargs*)

Bases: [Model](#)

**\_abc\_impl** = <\_abc.\_abc\_data object>

**property path**

Return the property value.

**Parameters**

**obj** – The object to get the value from.

```
class klovve.pieces.viewer.pdf.View(model, view_factory)
```

Bases: [BaseView](#)[\[Model\]](#)

**Parameters**

- **model** ([\\_TGo](#)) –
- **view\_factory** ([klovve.view.tree.ViewFactory](#)) –

**\_abc\_impl** = [<\\_abc.\\_abc\\_data object>](#)

## Submodules

### [klovve.pieces.busy\\_animation module](#)

```
class klovve.pieces.busy_animation.Model(**kwargs)
```

Bases: [Model](#)

```
class Orientation(value, names=None, *, module=None, qualname=None, type=None, start=1,
                  boundary=None)
```

Bases: Enum

**HORIZONTAL** = 1

**VERTICAL** = 2

```
class Size(value, names=None, *, module=None, qualname=None, type=None, start=1, boundary=None)
```

Bases: Enum

**EXTRA\_LARGE** = 5

**LARGE** = 4

**MEDIUM** = 3

**SMALL** = 2

**TINY** = 1

**\_abc\_impl** = [<\\_abc.\\_abc\\_data object>](#)

#### **property orientation**

Return the property value.

**Parameters**

**obj** – The object to get the value from.

#### **property size**

Return the property value.

**Parameters**

**obj** – The object to get the value from.

#### **property text**

Return the property value.

**Parameters**

**obj** – The object to get the value from.



```
class klovve.pieces.busy_animation.View(model, view_factory)
    Bases: BaseView[Model]

    Parameters
        • model (_TGo) –
        • view_factory (klovve.view.tree.ViewFactory) –

    _abc_impl = <_abc._abc_data object>
```

### klovve.pieces.button module

```
class klovve.pieces.button.Model(**kwargs)
    Bases: Model

    _abc_impl = <_abc._abc_data object>

    property action
        Return the property value.

    Parameters
        obj – The object to get the value from.

    property text
        Return the property value.

    Parameters
        obj – The object to get the value from.

class klovve.pieces.button.View(model, view_factory)
    Bases: BaseView[Model]

    Parameters
        • model (_TGo) –
        • view_factory (klovve.view.tree.ViewFactory) –

    _abc_impl = <_abc._abc_data object>
```

### klovve.pieces.checkable module

```
class klovve.pieces.checkable.Model(**kwargs)
    Bases: Model

    _abc_impl = <_abc._abc_data object>

    property is_checked
        Return the property value.

    Parameters
        obj – The object to get the value from.

    property text
        Return the property value.

    Parameters
        obj – The object to get the value from.
```

```
class klovve.pieces.checkable.View(model, view_factory)
```

Bases: *BaseView*

**Parameters**

- **model** (*\_TGo*) –
- **view\_factory** (*klovve.view.tree.ViewFactory*) –

**\_abc\_impl** = *<\_abc.\_abc\_data object>*

### klovve.pieces.disableable module

```
class klovve.pieces.disableable.Model(**kwargs)
```

Bases: *Model*

**\_abc\_impl** = *<\_abc.\_abc\_data object>*

**property is\_disabled**

Return the property value.

**Parameters**

**obj** – The object to get the value from.

**property item**

Return the property value.

**Parameters**

**obj** – The object to get the value from.

```
class klovve.pieces.disableable.View(model, view_factory)
```

Bases: *BaseView*

**Parameters**

- **model** (*\_TGo*) –
- **view\_factory** (*klovve.view.tree.ViewFactory*) –

**\_abc\_impl** = *<\_abc.\_abc\_data object>*

### klovve.pieces.dropdown module

```
class klovve.pieces.dropdown.Model(**kwargs)
```

Bases: *Model*

**\_abc\_impl** = *<\_abc.\_abc\_data object>*

**property item\_label\_func**

Return the property value.

**Parameters**

**obj** – The object to get the value from.

**property items**

Return the property value.

**Parameters**

**obj** – The object to get the value from.

**property selected\_item**

Return the property value.

**Parameters**

**obj** – The object to get the value from.

**class** klovve.pieces.dropdown.**View**(*model*, *view\_factory*)

Bases: *BaseView*

**Parameters**

- **model** (*\_TGo*) –
- **view\_factory** (*klovve.view.tree.ViewFactory*) –

**\_abc\_impl** = <\_abc.\_abc\_data object>

**klovve.pieces.dual module**

**class** klovve.pieces.dual.**Model**(\*\**kwargs*)

Bases: *Model*

**\_abc\_impl** = <\_abc.\_abc\_data object>

**property main\_item**

Return the property value.

**Parameters**

**obj** – The object to get the value from.

**property side\_item**

Return the property value.

**Parameters**

**obj** – The object to get the value from.

**class** klovve.pieces.dual.**View**(*model*, *view\_factory*)

Bases: *BaseView*

**Parameters**

- **model** (*\_TGo*) –
- **view\_factory** (*klovve.view.tree.ViewFactory*) –

**\_abc\_impl** = <\_abc.\_abc\_data object>

**klovve.pieces.formatted\_text module**

**class** klovve.pieces.formatted\_text.**Model**(\*\**kwargs*)

Bases: *Model*

**\_abc\_impl** = <\_abc.\_abc\_data object>

**property text**

Return the property value.

**Parameters**

**obj** – The object to get the value from.

```
class klovve.pieces.formatted_text.View(model, view_factory)
    Bases: BaseView

    Parameters
        • model (_TGo) –
        • view_factory (klovve.view.tree.ViewFactory) –
    _abc_impl = <_abc._abc_data object>
```

### klovve.pieces.header module

```
class klovve.pieces.header.Model(**kwargs)
    Bases: Model
    _abc_impl = <_abc._abc_data object>

    property text
        Return the property value.

    Parameters
        obj – The object to get the value from.
```

```
class klovve.pieces.header.View(model, view_factory)
    Bases: BaseView

    Parameters
        • model (_TGo) –
        • view_factory (klovve.view.tree.ViewFactory) –
    _abc_impl = <_abc._abc_data object>
```

### klovve.pieces.headered\_panel module

```
class klovve.pieces.headered_panel.Model(**kwargs)
    Bases: Model

    class State(value, names=None, *, module=None, qualname=None, type=None, start=1,
        boundary=None)
        Bases: Enum
        BUSY = 1
        FAILED = 4
        SUCCESSFUL = 2
        SUCCESSFUL_WITH_WARNING = 3
    _abc_impl = <_abc._abc_data object>
```

**property actions**

Return the property value.

**Parameters**

**obj** – The object to get the value from.

**property body**

Return the property value.

**Parameters**

**obj** – The object to get the value from.

**property progress**

Return the property value.

**Parameters**

**obj** – The object to get the value from.

**property state**

Return the property value.

**Parameters**

**obj** – The object to get the value from.

**property title**

Return the property value.

**Parameters**

**obj** – The object to get the value from.

**property title\_secondary\_items**

Return the property value.

**Parameters**

**obj** – The object to get the value from.

```
class klovve.pieces.headered_panel.View(model, view_factory)
```

Bases: [BaseView\[Model\]](#)

**Parameters**

- **model** (*\_TGo*) –
- **view\_factory** (*klovve.view.tree.ViewFactory*) –

```
_abc_impl = <_abc._abc_data object>
```

**klovve.pieces.horizontal\_box module**

```
class klovve.pieces.horizontal_box.Model(**kwargs)
```

Bases: [Model](#)

```
_abc_impl = <_abc._abc_data object>
```

**property items**

Return the property value.

**Parameters**

**obj** – The object to get the value from.

```
class klovve.pieces.horizontal_box.View(model, view_factory)
    Bases: BaseView

    Parameters
        • model (_TGo) –
        • view_factory (klovve.view.tree.ViewFactory) –
    _abc_impl = <_abc._abc_data object>
```

### klovve.pieces.label module

```
class klovve.pieces.label.Model(**kwargs)
    Bases: Model
    _abc_impl = <_abc._abc_data object>

    property text
        Return the property value.

    Parameters
        obj – The object to get the value from.

class klovve.pieces.label.View(model, view_factory)
    Bases: BaseView

    Parameters
        • model (_TGo) –
        • view_factory (klovve.view.tree.ViewFactory) –
    _abc_impl = <_abc._abc_data object>
```

### klovve.pieces.list module

```
class klovve.pieces.list.Model(**kwargs)
    Bases: Model
    _abc_impl = <_abc._abc_data object>

    property allows_multiselect
        Return the property value.

    Parameters
        obj – The object to get the value from.

    property aux_controls
        @item_label_func.default def item_label_func(self):
            return lambda x: str(x)

    property item_actions
        Return the property value.

    Parameters
        obj – The object to get the value from.
```

**property item\_label\_func**

Return the property value.

**Parameters**

**obj** – The object to get the value from.

**property items**

Return the property value.

**Parameters**

**obj** – The object to get the value from.

**property list\_actions**

Return the property value.

**Parameters**

**obj** – The object to get the value from.

**property selected\_item**

Return the property value.

**Parameters**

**obj** – The object to get the value from.

**property selected\_items**

Return the property value.

**Parameters**

**obj** – The object to get the value from.

**class** klovve.pieces.list.**View**(*model*, *view\_factory*)

Bases: [BaseView](#)

**Parameters**

- **model** (*\_TGo*) –
- **view\_factory** ([klovve.view.tree.ViewFactory](#)) –

**\_abc\_impl** = <\_abc.\_abc\_data object>

**klovve.pieces.list\_panel module**

**class** klovve.pieces.list\_panel.**Model**(\*\**kwargs*)

Bases: [Model](#)

**\_abc\_impl** = <\_abc.\_abc\_data object>

**property aux\_controls**

Return the property value.

**Parameters**

**obj** – The object to get the value from.

**property body**

Return the property value.

**Parameters**

**obj** – The object to get the value from.

**property item\_actions**

Return the property value.

**Parameters**

**obj** – The object to get the value from.

**property item\_label\_func**

Return the property value.

**Parameters**

**obj** – The object to get the value from.

**property items**

Return the property value.

**Parameters**

**obj** – The object to get the value from.

**property list\_actions**

Return the property value.

**Parameters**

**obj** – The object to get the value from.

**property selected\_item**

Return the property value.

**Parameters**

**obj** – The object to get the value from.

```
class klovve.pieces.list_panel.View(*args, **kwargs)
```

Bases: [ComposedView\[Model\]](#)

```
_abc_impl = <_abc._abc_data object>
```

```
compose()
```

**klovve.pieces.placeholder module**

```
class klovve.pieces.placeholder.Model(**kwargs)
```

Bases: [Model](#)

```
_abc_impl = <_abc._abc_data object>
```

**property item**

Return the property value.

**Parameters**

**obj** – The object to get the value from.

```
class klovve.pieces.placeholder.View(model, view_factory)
```

Bases: [BaseView](#)

**Parameters**

- **model** ([\\_TGo](#)) –
- **view\_factory** ([klovve.view.tree.ViewFactory](#)) –

```
_abc_impl = <_abc._abc_data object>
```



**klovve.pieces.progress module**

```
class klovve.pieces.progress.Model(**kwargs)
```

Bases: [Model](#)

```
_abc_impl = <_abc._abc_data object>
```

**property value**

Return the property value.

**Parameters**

**obj** – The object to get the value from.

```
class klovve.pieces.progress.View(model, view_factory)
```

Bases: [BaseView](#)

**Parameters**

- **model** ([\\_TGo](#)) –
- **view\_factory** ([klovve.view.tree.ViewFactory](#)) –

```
_abc_impl = <_abc._abc_data object>
```

**klovve.pieces.scrollable module**

```
class klovve.pieces.scrollable.Model(**kwargs)
```

Bases: [Model](#)

```
_abc_impl = <_abc._abc_data object>
```

**property item**

Return the property value.

**Parameters**

**obj** – The object to get the value from.

```
class klovve.pieces.scrollable.View(model, view_factory)
```

Bases: [BaseView](#)

**Parameters**

- **model** ([\\_TGo](#)) –
- **view\_factory** ([klovve.view.tree.ViewFactory](#)) –

```
_abc_impl = <_abc._abc_data object>
```

**klovve.pieces.slider module**

```
class klovve.pieces.slider.Model(**kwargs)
```

Bases: [Model](#)

```
_abc_impl = <_abc._abc_data object>
```

**property max\_value**

Return the property value.

**Parameters**

**obj** – The object to get the value from.

**property min\_value**

Return the property value.

**Parameters**

**obj** – The object to get the value from.

**property step\_value**

Return the property value.

**Parameters**

**obj** – The object to get the value from.

**property value**

Return the property value.

**Parameters**

**obj** – The object to get the value from.

**class** klovve.pieces.slider.**View**(*model*, *view\_factory*)

Bases: [BaseView](#)

**Parameters**

- **model** (*\_TGo*) –
- **view\_factory** ([klovve.view.tree.ViewFactory](#)) –

**\_abc\_impl** = <**\_abc.\_abc\_data** object>

**klovve.pieces.split module**

**class** klovve.pieces.split.**Model**(\*\**kwargs*)

Bases: [Model](#)

**\_abc\_impl** = <**\_abc.\_abc\_data** object>

**property item1**

Return the property value.

**Parameters**

**obj** – The object to get the value from.

**property item2**

Return the property value.

**Parameters**

**obj** – The object to get the value from.

**class** klovve.pieces.split.**View**(*model*, *view\_factory*)

Bases: [BaseView](#)

**Parameters**

- **model** (*\_TGo*) –

- **view\_factory** (`klovve.view.tree.ViewFactory`) –

**\_abc\_impl** = <\_abc.\_abc\_data object>

### klovve.pieces.text\_field module

**class** `klovve.pieces.text_field.Model`(\*\*kwargs)

Bases: `Model`

**\_abc\_impl** = <\_abc.\_abc\_data object>

**property** `hint_text`

Return the property value.

**Parameters**

**obj** – The object to get the value from.

**property** `text`

Return the property value.

**Parameters**

**obj** – The object to get the value from.

**class** `klovve.pieces.text_field.View`(*model*, *view\_factory*)

Bases: `BaseView`

**Parameters**

- **model** (`_TGo`) –

- **view\_factory** (`klovve.view.tree.ViewFactory`) –

**\_abc\_impl** = <\_abc.\_abc\_data object>

### klovve.pieces.vertical\_box module

**class** `klovve.pieces.vertical_box.Model`(\*\*kwargs)

Bases: `Model`

**\_abc\_impl** = <\_abc.\_abc\_data object>

**property** `items`

Return the property value.

**Parameters**

**obj** – The object to get the value from.

**class** `klovve.pieces.vertical_box.View`(*model*, *view\_factory*)

Bases: `BaseView`

**Parameters**

- **model** (`_TGo`) –

- **view\_factory** (`klovve.view.tree.ViewFactory`) –

**\_abc\_impl** = <\_abc.\_abc\_data object>

**klovve.pieces.window module**

```
class klovve.pieces.window.Model(**kwargs)
```

Bases: [Model](#)

```
_abc_impl = <_abc._abc_data object>
```

**property actions**

Return the property value.

**Parameters**

**obj** – The object to get the value from.

**property body**

Return the property value.

**Parameters**

**obj** – The object to get the value from.

**property close\_func**

Return the property value.

**Parameters**

**obj** – The object to get the value from.

**property closing**

Return the property value.

**Parameters**

**obj** – The object to get the value from.

**property icon**

Return the property value.

**Parameters**

**obj** – The object to get the value from.

**property is\_closed**

Return the property value.

**Parameters**

**obj** – The object to get the value from.

**property name**

Return the property value.

**Parameters**

**obj** – The object to get the value from.

**property title**

Return the property value.

**Parameters**

**obj** – The object to get the value from.

**property title2**

Return the property value.

**Parameters**

**obj** – The object to get the value from.

```
class klovve.pieces.window.View(model, view_factory)
```

Bases: [BaseView](#)

#### Parameters

- **model** ([\\_TGo](#)) –
- **view\_factory** ([klovve.view.tree.ViewFactory](#)) –

```
_abc_impl = <_abc._abc_data object>
```

## klovve.view package

```
class klovve.view.BaseView(model, view_factory)
```

Bases: [Generic\[\\_TGo\]](#), [Model](#), [View](#)

Base class for a view piece.

#### Parameters

- **model** ([\\_TGo](#)) –
- **view\_factory** ([klovve.view.tree.ViewFactory](#)) –

```
class Rambo(a, b)
```

Bases: [object](#)

```
__setattr__(key, value)
```

```
_abc_impl = <_abc._abc_data object>
```

```
create_native()
```

Create a native representation for this view.

This must be implemented in native views, but is only called internally. See also [native\(\)](#).

#### Return type

[object](#)

```
make_view()
```

#### Return type

[tuple](#)[[klovve.view.tree.ViewFactory](#), [~\\_TGo](#)]

```
property model: _TGo
```

```
native()
```

Return the native representation for this view. Internally uses [create\\_native\(\)](#) for creating it, but then keeps that result for later calls.

```
class klovve.view.ComposedView(*args, **kwargs)
```

Bases: [BaseView](#), [Generic\[\\_TGo\]](#)

Base class for a composed view piece, i.e. a view that is not implemented using native UI toolkits, but by composing other Klovve view pieces together.

```
_abc_impl = <_abc._abc_data object>
```

```
compose()
```

#### Return type

[ViewTreeNode](#)

**create\_native()**

Create a native representation for this view.

This must be implemented in native views, but is only called internally. See also `native()`.

**class klovve.view.View**

Bases: ABC

A view piece.

**\_abc\_impl** = <\_abc.\_abc\_data object>

**abstract native()**

Return the native representation for this view. Internally uses `create_native()` for creating it, but then keeps that result for later calls.

**Return type**  
object

**Submodules****klovve.view.tree module****exception klovve.view.tree.ViewError**

Bases: Exception

**class klovve.view.tree.ViewFactory(driver)**

Bases: object

View factories return view piece factories on attribute access. These create a [ViewTreeNode](#) when called. The *pieces* object that you often see in view implementations is a view factory.

**Parameters**

**driver** ([klovve.drivers.Driver](#)) –

**class \_Node(view\_factory, view\_name)**

Bases: object

**Parameters**

• **view\_factory** ([ViewFactory](#)) –

• **view\_name** ([Iterable\[str\]](#)) –

**create\_view**(*name\_tuple*, /, \*, *model*=None, \*\**kwargs*)

**Parameters**

**name\_tuple** ([Iterable\[str\]](#)) –

**Return type**

[View](#)

**class klovve.view.tree.ViewTreeNode(view\_factory, name\_tuple, model, data)**

Bases: object

Definition for one view item in a view, including the view type, the model, and data TODO. As the TODO potentially contains other view nodes (for a view that is composed of smaller pieces), this is a hierarchical structure, with the root node specifying your entire user interface.

**Parameters**

**data** (*Iterable[tuple[str, Any]]*) –

**\_\_create\_or\_update\_view**(*old\_tree, new\_tree, model, parent\_view, parent\_view\_prop\_name*)

**Parameters**

- **old\_tree** (*ViewTreeNode*) –
- **new\_tree** (*ViewTreeNode*) –
- **model** (*Model*) –
- **parent\_view** (*View*) –
- **parent\_view\_prop\_name** (*str*) –

**create\_or\_update\_view**(*model, placeholder, old\_tree*)

Create the view defined by this node, or update it.

**Parameters**

- **model** (*Model*) –
- **placeholder** –
- **old\_tree** (*ViewTreeNode*) –

**property data:** *dict[str, Any]*

The data specified for this view piece. Values will contain fixed values as well as value holders. This potentially includes other view nodes (or lists, ... of them).

**property model:** *Optional[Model]*

A model, if this view pieces was specified by a model as argument (which is quite unusual).

**property name\_tuple:** *Iterable[str]*

**view()**

Return the view specified by this node. This is created on-demand but then cached and re-used.

**Return type**

*View*

**klovve.view.tree.\_connect\_value\_holder**(*target\_model, target\_prop\_name, value\_holder*)

**Parameters**

- **target\_model** (*Model*) –
- **target\_prop\_name** (*str*) –
- **value\_holder** (*ValueHolder*) –

**Return type**

object

## 4.1.2 Submodules

### 4.1.3 klovve.debug module

**class** klovve.debug.\_Memory

Bases: object

**\_Memory\_\_change\_counter**(*kind*, *by*)

Parameters

- **kind** (*str*) –
- **by** (*int*) –

Return type

None

**\_Memory\_\_decrement\_counter**(*kind*)

Parameters

**kind** (*str*) –

Return type

None

**\_Memory\_\_increment\_counter**(*kind*)

Parameters

**kind** (*str*) –

Return type

None

**property** current

**new\_object\_created**(*obj*, *kind*)

Parameters

**kind** (*str*) –

**property** since\_startup

klovve.debug.\_print\_loop()



## PYTHON MODULE INDEX

### k

- klovve, 9
- klovve.+tests, 9
- klovve.app, 9
- klovve.app.context, 11
- klovve.data, 12
- klovve.data.deps, 12
- klovve.data.lists, 15
- klovve.data.model, 18
- klovve.data.observable\_value, 23
- klovve.data.reaction, 24
- klovve.data.value\_holder, 25
- klovve.debug, 108
- klovve.drivers, 26
- klovve.drivers.bare\_terminal, 28
- klovve.drivers.bare\_terminal.views, 28
- klovve.drivers.bare\_terminal.views.checkable, 29
- klovve.drivers.bare\_terminal.views.headered\_panel, 29
- klovve.drivers.bare\_terminal.views.interact, 28
- klovve.drivers.bare\_terminal.views.interact.abstract, 28
- klovve.drivers.bare\_terminal.views.label, 30
- klovve.drivers.bare\_terminal.views.log\_pager, 30
- klovve.drivers.bare\_terminal.views.placeholder, 30
- klovve.drivers.bare\_terminal.views.split, 31
- klovve.drivers.bare\_terminal.views.vertical\_box, 31
- klovve.drivers.bare\_terminal.views.viewer, 29
- klovve.drivers.bare\_terminal.views.viewer.image, 29
- klovve.drivers.bare\_terminal.views.window, 31
- klovve.drivers.gtk, 32
- klovve.drivers.gtk.views, 34
- klovve.drivers.gtk.views.\_gtk\_private, 34
- klovve.drivers.gtk.views.\_gtk\_private.box, 34
- klovve.drivers.gtk.views.busy\_animation, 39
- klovve.drivers.gtk.views.button, 40
- klovve.drivers.gtk.views.checkable, 40
- klovve.drivers.gtk.views.disableable, 41
- klovve.drivers.gtk.views.dropdown, 41
- klovve.drivers.gtk.views.dual, 41
- klovve.drivers.gtk.views.form, 35
- klovve.drivers.gtk.views.form.section, 35
- klovve.drivers.gtk.views.formatted\_text, 42
- klovve.drivers.gtk.views.header, 42
- klovve.drivers.gtk.views.headered\_panel, 42
- klovve.drivers.gtk.views.horizontal\_box, 43
- klovve.drivers.gtk.views.interact, 35
- klovve.drivers.gtk.views.interact.abstract, 35
- klovve.drivers.gtk.views.label, 43
- klovve.drivers.gtk.views.list, 43
- klovve.drivers.gtk.views.log\_pager, 44
- klovve.drivers.gtk.views.placeholder, 44
- klovve.drivers.gtk.views.progress, 44
- klovve.drivers.gtk.views.property\_panel, 36
- klovve.drivers.gtk.views.scrollable, 45
- klovve.drivers.gtk.views.slider, 45
- klovve.drivers.gtk.views.split, 45
- klovve.drivers.gtk.views.tabbed, 36
- klovve.drivers.gtk.views.tabbed.tab, 36
- klovve.drivers.gtk.views.text\_field, 46
- klovve.drivers.gtk.views.vertical\_box, 46
- klovve.drivers.gtk.views.viewer, 37
- klovve.drivers.gtk.views.viewer.image, 37
- klovve.drivers.gtk.views.viewer.pdf, 39
- klovve.drivers.gtk.views.window, 46
- klovve.drivers.viwid, 47
- klovve.drivers.viwid.views, 48
- klovve.drivers.viwid.views.\_curses\_private, 48
- klovve.drivers.viwid.views.\_curses\_private.box, 48
- klovve.drivers.viwid.views.busy\_animation, 51
- klovve.drivers.viwid.views.button, 51
- klovve.drivers.viwid.views.checkable, 52
- klovve.drivers.viwid.views.disableable, 52
- klovve.drivers.viwid.views.dropdown, 53
- klovve.drivers.viwid.views.dual, 53

klovve.drivers.viuid.views.form, 48  
klovve.drivers.viuid.views.form.section, 49  
klovve.drivers.viuid.views.formatted\_text, 53  
klovve.drivers.viuid.views.header, 54  
klovve.drivers.viuid.views.headered\_panel, 54  
klovve.drivers.viuid.views.horizontal\_box, 54  
klovve.drivers.viuid.views.interact, 49  
klovve.drivers.viuid.views.interact.abstract, 49  
klovve.drivers.viuid.views.label, 55  
klovve.drivers.viuid.views.list, 55  
klovve.drivers.viuid.views.log\_pager, 55  
klovve.drivers.viuid.views.placeholder, 56  
klovve.drivers.viuid.views.progress, 56  
klovve.drivers.viuid.views.property\_panel, 49  
klovve.drivers.viuid.views.scrollable, 56  
klovve.drivers.viuid.views.slider, 57  
klovve.drivers.viuid.views.split, 57  
klovve.drivers.viuid.views.tabbed, 50  
klovve.drivers.viuid.views.tabbed.tab, 50  
klovve.drivers.viuid.views.text\_field, 57  
klovve.drivers.viuid.views.vertical\_box, 58  
klovve.drivers.viuid.views.viewer, 50  
klovve.drivers.viuid.views.viewer.image, 50  
klovve.drivers.viuid.views.viewer.pdf, 51  
klovve.drivers.viuid.views.window, 58  
klovve.drivers.viuid.viuid, 58  
klovve.drivers.viuid.viuid.+samples, 60  
klovve.drivers.viuid.viuid.canvas, 74  
klovve.drivers.viuid.viuid.data, 76  
klovve.drivers.viuid.viuid.drivers, 60  
klovve.drivers.viuid.viuid.drivers.curses, 61  
klovve.drivers.viuid.viuid.drivers.driver, 62  
klovve.drivers.viuid.viuid.event, 63  
klovve.drivers.viuid.viuid.event.base, 63  
klovve.drivers.viuid.viuid.event.input, 63  
klovve.drivers.viuid.viuid.layout, 77  
klovve.drivers.viuid.viuid.loop, 78  
klovve.drivers.viuid.viuid.screen, 78  
klovve.drivers.viuid.viuid.style, 82  
klovve.drivers.viuid.viuid.widgets, 64  
klovve.drivers.viuid.viuid.widgets.box, 64  
klovve.drivers.viuid.viuid.widgets.busy\_animation, 65  
klovve.drivers.viuid.viuid.widgets.button, 65  
klovve.drivers.viuid.viuid.widgets.check\_button, 65  
klovve.drivers.viuid.viuid.widgets.dropdown, 66  
klovve.drivers.viuid.viuid.widgets.entry, 66  
klovve.drivers.viuid.viuid.widgets.frame, 66  
klovve.drivers.viuid.viuid.widgets.label, 67  
klovve.drivers.viuid.viuid.widgets.list, 67  
klovve.drivers.viuid.viuid.widgets.progress\_bar, 68  
klovve.drivers.viuid.viuid.widgets.radio\_button, 68  
klovve.drivers.viuid.viuid.widgets.scrollable, 68  
klovve.drivers.viuid.viuid.widgets.widget, 69  
klovve.pieces, 83  
klovve.pieces.busy\_animation, 92  
klovve.pieces.button, 93  
klovve.pieces.checkable, 93  
klovve.pieces.disableable, 94  
klovve.pieces.dropdown, 94  
klovve.pieces.dual, 95  
klovve.pieces.form, 83  
klovve.pieces.form.section, 83  
klovve.pieces.formatted\_text, 95  
klovve.pieces.header, 96  
klovve.pieces.headered\_panel, 96  
klovve.pieces.horizontal\_box, 97  
klovve.pieces.interact, 84  
klovve.pieces.interact.abstract, 84  
klovve.pieces.interact.message, 85  
klovve.pieces.interact.textline, 85  
klovve.pieces.interact.triggers, 86  
klovve.pieces.interact.yesno, 87  
klovve.pieces.label, 98  
klovve.pieces.list, 98  
klovve.pieces.list\_panel, 99  
klovve.pieces.log\_pager, 87  
klovve.pieces.log\_pager.entry, 88  
klovve.pieces.placeholder, 100  
klovve.pieces.progress, 101  
klovve.pieces.property\_panel, 89  
klovve.pieces.property\_panel.entry, 89  
klovve.pieces.scrollable, 101  
klovve.pieces.slider, 101  
klovve.pieces.split, 102  
klovve.pieces.tabbed, 90  
klovve.pieces.tabbed.tab, 90  
klovve.pieces.text\_field, 103  
klovve.pieces.vertical\_box, 103  
klovve.pieces.viewer, 91  
klovve.pieces.viewer.image, 91  
klovve.pieces.viewer.pdf, 91  
klovve.pieces.window, 104  
klovve.view, 105  
klovve.view.tree, 106

## INDEX

### Symbols

\_Canvas (class in klovve.drivers.viwid.viwid.drivers.curses), 61  
 \_ComputedProperty (class in klovve.data.model), 21  
 \_CursesColors (class in klovve.drivers.viwid.viwid.drivers.curses), 61  
 \_CursesColors.Mode (class in klovve.drivers.viwid.viwid.drivers.curses), 61  
 \_CursesColors\_\_curses\_color() (klovve.drivers.viwid.viwid.drivers.curses.\_CursesColors method), 61  
 \_EventLoop (class in klovve.drivers.gtk), 32  
 \_ListProperty (class in klovve.drivers.viwid.viwid.widgets.widget), 73  
 \_Loop (class in klovve.drivers.viwid.viwid.drivers.curses), 62  
 \_Memory (class in klovve.debug), 108  
 \_Memory\_\_change\_counter() (klovve.debug.\_Memory method), 108  
 \_Memory\_\_decrement\_counter() (klovve.debug.\_Memory method), 108  
 \_Memory\_\_increment\_counter() (klovve.debug.\_Memory method), 108  
 \_Property (class in klovve.drivers.viwid.viwid.widgets.widget), 73  
 \_ReactionModel (class in klovve.data.reaction), 24  
 \_Screen (class in klovve.drivers.viwid.viwid.drivers.curses), 62  
 \_Selector (class in klovve.drivers.gtk), 33  
 \_Source (class in klovve.drivers.gtk), 33  
 \_TDependencyTuple (in module klovve.data.deps), 14  
 \_TGetValueFunc (in module klovve.data.model), 22  
 \_TVViewSpec (in module klovve.app), 10  
 \_\_bind() (klovve.drivers.gtk.View method), 32  
 \_\_bind() (klovve.drivers.viwid.View method), 47  
 \_\_click() (klovve.drivers.viwid.viwid.widgets.dropdown.Dropdown method), 66  
 \_\_compute() (klovve.data.model.Model method), 19  
 \_\_correct\_index() (klovve.data.lists.List method), 15  
 \_\_correct\_index() (klovve.drivers.viwid.viwid.data.List method), 76  
 \_\_correct\_offset() (klovve.drivers.viwid.viwid.widgets.scrollable.Scrollable method), 68  
 \_\_create\_or\_update\_view() (klovve.view.tree.ViewTreeNode method), 107  
 \_\_driver (klovve.drivers.Driver attribute), 26  
 \_\_ensure\_config\_correct() (klovve.drivers.gtk.views.viewer.image.Image method), 37  
 \_\_find\_last\_whitespace() (klovve.drivers.viwid.viwid.widgets.label.Label method), 67  
 \_\_handle\_event\_\_tab\_key() (klovve.drivers.viwid.viwid.screen.Screen method), 80  
 \_\_head\_version() (klovve.data.model.Model method), 19  
 \_\_horizontally\_process\_partitioning() (klovve.drivers.viwid.viwid.layout.GridLayout method), 77  
 \_\_inner\_size\_changed() (klovve.drivers.gtk.views.log\_pager.View method), 44  
 \_\_inner\_size\_changed() (klovve.drivers.viwid.views.log\_pager.View method), 55  
 \_\_invalidate() (klovve.data.model.Model method), 19  
 \_\_item\_added() (klovve.drivers.viwid.viwid.widgets.box.Box method), 64  
 \_\_item\_added() (klovve.drivers.viwid.viwid.widgets.widget.Widget method), 69  
 \_\_item\_removed() (klovve.drivers.viwid.viwid.widgets.box.Box method), 64  
 \_\_item\_removed() (klovve.drivers.viwid.viwid.widgets.widget.Widget method), 69  
 \_\_mainloop (klovve.drivers.Driver attribute), 26  
 \_\_move\_offset() (klovve.drivers.viwid.viwid.widgets.scrollable.Scrollable method), 68  
 \_\_next() (klovve.drivers.viwid.viwid.widgets.busy\_animation.BusyAnimation method), 65

\_\_on\_mouse\_button\_pressed()  
     (klovve.drivers.gtk.views.viewer.image.Outer  
     method), 38  
 \_\_on\_mouse\_button\_released()  
     (klovve.drivers.gtk.views.viewer.image.Outer  
     method), 38  
 \_\_on\_mouse\_moved() (klovve.drivers.gtk.views.viewer.image.Outer  
     method), 38  
 \_\_on\_mouse\_wheel() (klovve.drivers.gtk.views.viewer.image.Outer  
     method), 38  
 \_\_process\_partitioning()  
     (klovve.drivers.viwid.viwid.layout.GridLayout  
     method), 77  
 \_\_refresh() (klovve.drivers.gtk.views.viewer.image.Image  
     method), 37  
 \_\_requests\_queue() (klovve.drivers.viwid.viwid.drivers.curses.Driver  
     method), 61  
 \_\_running\_apps (klovve.app.Application attribute), 9  
 \_\_running\_apps\_changed\_event()  
     (klovve.app.Application method), 9  
 \_\_running\_apps\_changed\_event\_\_internal  
     (klovve.app.Application attribute), 9  
 \_\_setattr\_\_X() (klovve.view.BaseView.Rambo  
     method), 105  
 \_\_stretch\_axis\_finally()  
     (klovve.drivers.viwid.viwid.layout.GridLayout  
     method), 77  
 \_\_stretch\_axis\_minimals()  
     (klovve.drivers.viwid.viwid.layout.GridLayout  
     method), 77  
 \_\_stretch\_from\_partitioning()  
     (klovve.drivers.viwid.viwid.layout.GridLayout  
     method), 77  
 \_\_trim\_geometry() (klovve.drivers.viwid.viwid.layout.GridLayout  
     method), 77  
 \_\_vertically\_process\_partitioning()  
     (klovve.drivers.viwid.viwid.layout.GridLayout  
     method), 77  
 \_abc\_impl (klovve.app.context.ActionContext attribute),  
     11  
 \_abc\_impl (klovve.data.deps.DependencySubject at-  
     tribute), 12  
 \_abc\_impl (klovve.data.lists.List attribute), 15  
 \_abc\_impl (klovve.data.lists.ListObserver attribute), 16  
 \_abc\_impl (klovve.data.model.Model attribute), 19  
 \_abc\_impl (klovve.data.observable\_value.CommonObservableValue  
     attribute), 23  
 \_abc\_impl (klovve.data.observable\_value.ObservableValue  
     attribute), 23  
 \_abc\_impl (klovve.data.reaction.\_ReactionModel  
     attribute), 24  
 \_abc\_impl (klovve.data.value\_holder.ComputedValue  
     attribute), 25  
 \_abc\_impl (klovve.data.value\_holder.OneWayBinding  
     attribute), 25  
 \_abc\_impl (klovve.data.value\_holder.TwoWayBinding  
     attribute), 25  
 \_abc\_impl (klovve.data.value\_holder.ValueHolder at-  
     tribute), 25  
 \_abc\_impl (klovve.drivers.Driver attribute), 26  
 \_abc\_impl (klovve.drivers.bare\_terminal.Driver at-  
     tribute), 28  
 \_abc\_impl (klovve.drivers.bare\_terminal.views.checkable.View  
     attribute), 29  
 \_abc\_impl (klovve.drivers.bare\_terminal.views.headered\_panel.View  
     attribute), 29  
 \_abc\_impl (klovve.drivers.bare\_terminal.views.interact.abstract.View  
     attribute), 28  
 \_abc\_impl (klovve.drivers.bare\_terminal.views.label.View  
     attribute), 30  
 \_abc\_impl (klovve.drivers.bare\_terminal.views.log\_pager.View  
     attribute), 30  
 \_abc\_impl (klovve.drivers.bare\_terminal.views.placeholder.View  
     attribute), 30  
 \_abc\_impl (klovve.drivers.bare\_terminal.views.split.View  
     attribute), 31  
 \_abc\_impl (klovve.drivers.bare\_terminal.views.vertical\_box.View  
     attribute), 31  
 \_abc\_impl (klovve.drivers.bare\_terminal.views.viewer.image.View  
     attribute), 29  
 \_abc\_impl (klovve.drivers.bare\_terminal.views.window.View  
     attribute), 31  
 \_abc\_impl (klovve.drivers.gtk.Driver attribute), 32  
 \_abc\_impl (klovve.drivers.gtk.View attribute), 32  
 \_abc\_impl (klovve.drivers.gtk.\_Selector attribute), 33  
 \_abc\_impl (klovve.drivers.gtk.views.\_gtk\_private.box.View  
     attribute), 34  
 \_abc\_impl (klovve.drivers.gtk.views.busy\_animation.View  
     attribute), 39  
 \_abc\_impl (klovve.drivers.gtk.views.button.ActionContext  
     attribute), 40  
 \_abc\_impl (klovve.drivers.gtk.views.button.View at-  
     tribute), 40  
 \_abc\_impl (klovve.drivers.gtk.views.checkable.View at-  
     tribute), 40  
 \_abc\_impl (klovve.drivers.gtk.views.disableable.View  
     attribute), 41  
 \_abc\_impl (klovve.drivers.gtk.views.dropdown.View at-  
     tribute), 41  
 \_abc\_impl (klovve.drivers.gtk.views.dual.View at-  
     tribute), 41  
 \_abc\_impl (klovve.drivers.gtk.views.form.View at-  
     tribute), 35  
 \_abc\_impl (klovve.drivers.gtk.views.form.section.View  
     attribute), 35  
 \_abc\_impl (klovve.drivers.gtk.views.formatted\_text.View  
     attribute), 42  
 \_abc\_impl (klovve.drivers.gtk.views.header.View at-

tribute), 42

`_abc_impl (klovve.drivers.gtk.views.headered_panel.View attribute)`, 42

`_abc_impl (klovve.drivers.gtk.views.horizontal_box.View attribute)`, 43

`_abc_impl (klovve.drivers.gtk.views.interact.abstract.View attribute)`, 35

`_abc_impl (klovve.drivers.gtk.views.label.View attribute)`, 43

`_abc_impl (klovve.drivers.gtk.views.list.View attribute)`, 43

`_abc_impl (klovve.drivers.gtk.views.log_pager.View attribute)`, 44

`_abc_impl (klovve.drivers.gtk.views.placeholder.View attribute)`, 44

`_abc_impl (klovve.drivers.gtk.views.progress.View attribute)`, 44

`_abc_impl (klovve.drivers.gtk.views.property_panel.View attribute)`, 36

`_abc_impl (klovve.drivers.gtk.views.scrollable.View attribute)`, 45

`_abc_impl (klovve.drivers.gtk.views.slider.View attribute)`, 45

`_abc_impl (klovve.drivers.gtk.views.split.View attribute)`, 45

`_abc_impl (klovve.drivers.gtk.views.tabbed.View attribute)`, 36

`_abc_impl (klovve.drivers.gtk.views.tabbed.tab.View attribute)`, 36

`_abc_impl (klovve.drivers.gtk.views.text_field.View attribute)`, 46

`_abc_impl (klovve.drivers.gtk.views.vertical_box.View attribute)`, 46

`_abc_impl (klovve.drivers.gtk.views.viewer.image.View attribute)`, 39

`_abc_impl (klovve.drivers.gtk.views.viewer.pdf.View attribute)`, 39

`_abc_impl (klovve.drivers.gtk.views.window.ActionContext attribute)`, 46

`_abc_impl (klovve.drivers.gtk.views.window.View attribute)`, 46

`_abc_impl (klovve.drivers.viwid.Driver attribute)`, 47

`_abc_impl (klovve.drivers.viwid.View attribute)`, 47

`_abc_impl (klovve.drivers.viwid.views._curses_private.box.View attribute)`, 48

`_abc_impl (klovve.drivers.viwid.views.busy_animation.Model attribute)`, 51

`_abc_impl (klovve.drivers.viwid.views.busy_animation.View attribute)`, 51

`_abc_impl (klovve.drivers.viwid.views.button.ActionContext attribute)`, 51

`_abc_impl (klovve.drivers.viwid.views.button.View attribute)`, 52

`_abc_impl (klovve.drivers.viwid.views.checkable.View attribute)`, 52

`_abc_impl (klovve.drivers.viwid.views.disableable.View attribute)`, 52

`_abc_impl (klovve.drivers.viwid.views.dropdown.View attribute)`, 53

`_abc_impl (klovve.drivers.viwid.views.dual.View attribute)`, 53

`_abc_impl (klovve.drivers.viwid.views.form.View attribute)`, 48

`_abc_impl (klovve.drivers.viwid.views.form.section.View attribute)`, 49

`_abc_impl (klovve.drivers.viwid.views.formatted_text.View attribute)`, 53

`_abc_impl (klovve.drivers.viwid.views.header.View attribute)`, 54

`_abc_impl (klovve.drivers.viwid.views.headered_panel.Model attribute)`, 54

`_abc_impl (klovve.drivers.viwid.views.headered_panel.View attribute)`, 54

`_abc_impl (klovve.drivers.viwid.views.horizontal_box.View attribute)`, 54

`_abc_impl (klovve.drivers.viwid.views.interact.abstract.View attribute)`, 49

`_abc_impl (klovve.drivers.viwid.views.label.View attribute)`, 55

`_abc_impl (klovve.drivers.viwid.views.list.View attribute)`, 55

`_abc_impl (klovve.drivers.viwid.views.log_pager.View attribute)`, 55

`_abc_impl (klovve.drivers.viwid.views.placeholder.View attribute)`, 56

`_abc_impl (klovve.drivers.viwid.views.progress.View attribute)`, 56

`_abc_impl (klovve.drivers.viwid.views.property_panel.View attribute)`, 49

`_abc_impl (klovve.drivers.viwid.views.scrollable.View attribute)`, 56

`_abc_impl (klovve.drivers.viwid.views.slider.View attribute)`, 57

`_abc_impl (klovve.drivers.viwid.views.split.View attribute)`, 57

`_abc_impl (klovve.drivers.viwid.views.tabbed.View attribute)`, 50

`_abc_impl (klovve.drivers.viwid.views.tabbed.tab.View attribute)`, 50

`_abc_impl (klovve.drivers.viwid.views.text_field.View attribute)`, 57

`_abc_impl (klovve.drivers.viwid.views.vertical_box.View attribute)`, 58

`_abc_impl (klovve.drivers.viwid.views.viewer.image.View attribute)`, 50

`_abc_impl (klovve.drivers.viwid.views.viewer.pdf.View attribute)`, 51

`_abc_impl (klovve.drivers.viwid.views.window.ActionContext attribute)`, 51



- attribute*), 58
- `_abc_impl` (*klovve.drivers.viwid.views.window.View attribute*), 58
- `_abc_impl` (*klovve.drivers.viwid.viwid.data.List attribute*), 76
- `_abc_impl` (*klovve.drivers.viwid.viwid.drivers.curses.Driver attribute*), 61
- `_abc_impl` (*klovve.drivers.viwid.viwid.drivers.driver.Driver attribute*), 62
- `_abc_impl` (*klovve.drivers.viwid.viwid.screen.AnchorRootAlignment attribute*), 78
- `_abc_impl` (*klovve.drivers.viwid.viwid.screen.AutoRootAlignment attribute*), 78
- `_abc_impl` (*klovve.drivers.viwid.viwid.screen.ByScreenRootAlignment attribute*), 79
- `_abc_impl` (*klovve.drivers.viwid.viwid.screen.CenterRootAlignment attribute*), 79
- `_abc_impl` (*klovve.drivers.viwid.viwid.screen.FixedRootAlignment attribute*), 79
- `_abc_impl` (*klovve.drivers.viwid.viwid.screen.FixedRootAlignment attribute*), 79
- `_abc_impl` (*klovve.drivers.viwid.viwid.screen.RootAlignment attribute*), 79
- `_abc_impl` (*klovve.drivers.viwid.viwid.screen.RootAlignment attribute*), 80
- `_abc_impl` (*klovve.pieces.busy\_animation.Model attribute*), 92
- `_abc_impl` (*klovve.pieces.busy\_animation.View attribute*), 93
- `_abc_impl` (*klovve.pieces.button.Model attribute*), 93
- `_abc_impl` (*klovve.pieces.button.View attribute*), 93
- `_abc_impl` (*klovve.pieces.checkable.Model attribute*), 93
- `_abc_impl` (*klovve.pieces.checkable.View attribute*), 94
- `_abc_impl` (*klovve.pieces.disableable.Model attribute*), 94
- `_abc_impl` (*klovve.pieces.disableable.View attribute*), 94
- `_abc_impl` (*klovve.pieces.dropdown.Model attribute*), 94
- `_abc_impl` (*klovve.pieces.dropdown.View attribute*), 95
- `_abc_impl` (*klovve.pieces.dual.Model attribute*), 95
- `_abc_impl` (*klovve.pieces.dual.View attribute*), 95
- `_abc_impl` (*klovve.pieces.form.Model attribute*), 83
- `_abc_impl` (*klovve.pieces.form.View attribute*), 83
- `_abc_impl` (*klovve.pieces.form.section.Model attribute*), 83
- `_abc_impl` (*klovve.pieces.form.section.View attribute*), 84
- `_abc_impl` (*klovve.pieces.formatted\_text.Model attribute*), 95
- `_abc_impl` (*klovve.pieces.formatted\_text.View attribute*), 96
- `_abc_impl` (*klovve.pieces.header.Model attribute*), 96
- `_abc_impl` (*klovve.pieces.header.View attribute*), 96
- `_abc_impl` (*klovve.pieces.headered\_panel.Model attribute*), 96
- `_abc_impl` (*klovve.pieces.headered\_panel.View attribute*), 97
- `_abc_impl` (*klovve.pieces.horizontal\_box.Model attribute*), 97
- `_abc_impl` (*klovve.pieces.horizontal\_box.View attribute*), 98
- `_abc_impl` (*klovve.pieces.interact.Model attribute*), 84
- `_abc_impl` (*klovve.pieces.interact.abstract.Model attribute*), 84
- `_abc_impl` (*klovve.pieces.interact.abstract.View attribute*), 85
- `_abc_impl` (*klovve.pieces.interact.message.Model attribute*), 85
- `_abc_impl` (*klovve.pieces.interact.message.View attribute*), 85
- `_abc_impl` (*klovve.pieces.interact.textline.Model attribute*), 85
- `_abc_impl` (*klovve.pieces.interact.textline.View attribute*), 86
- `_abc_impl` (*klovve.pieces.interact.textline.View.TModel attribute*), 86
- `_abc_impl` (*klovve.pieces.interact.triggers.Model attribute*), 86
- `_abc_impl` (*klovve.pieces.interact.triggers.View attribute*), 86
- `_abc_impl` (*klovve.pieces.interact.yesno.Model attribute*), 87
- `_abc_impl` (*klovve.pieces.interact.yesno.View attribute*), 87
- `_abc_impl` (*klovve.pieces.label.Model attribute*), 98
- `_abc_impl` (*klovve.pieces.label.View attribute*), 98
- `_abc_impl` (*klovve.pieces.list.Model attribute*), 98
- `_abc_impl` (*klovve.pieces.list.View attribute*), 99
- `_abc_impl` (*klovve.pieces.list\_panel.Model attribute*), 99
- `_abc_impl` (*klovve.pieces.list\_panel.View attribute*), 100
- `_abc_impl` (*klovve.pieces.log\_pager.Model attribute*), 87
- `_abc_impl` (*klovve.pieces.log\_pager.View attribute*), 87
- `_abc_impl` (*klovve.pieces.log\_pager.entry.Model attribute*), 88
- `_abc_impl` (*klovve.pieces.log\_pager.entry.View attribute*), 89
- `_abc_impl` (*klovve.pieces.placeholder.Model attribute*), 100
- `_abc_impl` (*klovve.pieces.placeholder.View attribute*), 100
- `_abc_impl` (*klovve.pieces.progress.Model attribute*), 101
- `_abc_impl` (*klovve.pieces.progress.View attribute*), 101
- `_abc_impl` (*klovve.pieces.property\_panel.Model attribute*), 89
- `_abc_impl` (*klovve.pieces.property\_panel.View attribute*), 89
- `_abc_impl` (*klovve.pieces.property\_panel.entry.Model attribute*), 89
- `_abc_impl` (*klovve.pieces.property\_panel.entry.View attribute*), 89

tribute), 90

`_abc_impl` (klovve.pieces.scrollable.Model attribute), 101

`_abc_impl` (klovve.pieces.scrollable.View attribute), 101

`_abc_impl` (klovve.pieces.slider.Model attribute), 101

`_abc_impl` (klovve.pieces.slider.View attribute), 102

`_abc_impl` (klovve.pieces.split.Model attribute), 102

`_abc_impl` (klovve.pieces.split.View attribute), 103

`_abc_impl` (klovve.pieces.tabbed.Model attribute), 90

`_abc_impl` (klovve.pieces.tabbed.View attribute), 90

`_abc_impl` (klovve.pieces.tabbed.tab.Model attribute), 90

`_abc_impl` (klovve.pieces.tabbed.tab.View attribute), 91

`_abc_impl` (klovve.pieces.text\_field.Model attribute), 103

`_abc_impl` (klovve.pieces.text\_field.View attribute), 103

`_abc_impl` (klovve.pieces.vertical\_box.Model attribute), 103

`_abc_impl` (klovve.pieces.vertical\_box.View attribute), 103

`_abc_impl` (klovve.pieces.viewer.image.Model attribute), 91

`_abc_impl` (klovve.pieces.viewer.image.View attribute), 91

`_abc_impl` (klovve.pieces.viewer.pdf.Model attribute), 91

`_abc_impl` (klovve.pieces.viewer.pdf.View attribute), 92

`_abc_impl` (klovve.pieces.window.Model attribute), 104

`_abc_impl` (klovve.pieces.window.View attribute), 105

`_abc_impl` (klovve.view.BaseView attribute), 105

`_abc_impl` (klovve.view.ComposedView attribute), 105

`_abc_impl` (klovve.view.View attribute), 106

`_add_changed_handler()` (klovve.data.deps.DependencySubject method), 12

`_add_changed_handler()` (klovve.data.model.Model method), 19

`_add_observer()` (klovve.data.lists.List method), 15

`_add_observer()` (klovve.drivers.viwid.viwid.data.List method), 76

`_available_version()` (klovve.data.deps.DependencySubject method), 13

`_available_version()` (klovve.data.model.Model method), 19

`_call_soon()` (klovve.drivers.gtk.\_EventLoop method), 32

`_children` (klovve.drivers.viwid.viwid.widgets.widget.Widget property), 69

`_compute()` (klovve.data.model.Property method), 21

`_compute()` (klovve.data.model.PropertyBase method), 21

`_compute()` (klovve.data.model.\_ComputedProperty method), 22

`_connect_value_holder()` (in module klovve.view.tree), 107

`_create_dialog()` (klovve.app.context.ActionContext method), 11

`_create_dialog()` (klovve.drivers.gtk.views.button.ActionContext method), 40

`_create_dialog()` (klovve.drivers.gtk.views.window.ActionContext method), 46

`_create_dialog()` (klovve.drivers.viwid.views.button.ActionContext method), 51

`_create_dialog()` (klovve.drivers.viwid.views.window.ActionContext method), 58

`_create_mainloop()` (klovve.drivers.Driver method), 26

`_create_mainloop()` (klovve.drivers.bare\_terminal.Driver method), 28

`_create_mainloop()` (klovve.drivers.gtk.Driver method), 32

`_create_mainloop()` (klovve.drivers.viwid.Driver method), 47

`_current_compute_dependencies` (in module klovve.data.deps), 14

`_defer_value_propagation` (in module klovve.data.deps), 14

`_defer_value_propagation_for` (in module klovve.data.deps), 15

`_detect_dependencies()` (klovve.data.deps.DependencySubject method), 13

`_fget()` (klovve.data.model.PropertyBase method), 21

`_fget()` (klovve.drivers.viwid.viwid.widgets.widget.\_ListProperty method), 73

`_fget()` (klovve.drivers.viwid.viwid.widgets.widget.\_Property method), 73

`_find_modules()` (in module klovve.drivers), 27

`_flush_dependency_handlers()` (klovve.data.deps.DependencySubject method), 13

`_flush_dependency_handlers()` (klovve.data.model.Model method), 19

`_fset()` (klovve.data.model.ListProperty method), 18

`_fset()` (klovve.data.model.PropertyBase method), 21

`_fset()` (klovve.data.model.\_ComputedProperty method), 22

`_fset()` (klovve.drivers.viwid.viwid.widgets.widget.\_ListProperty method), 73

`_fset()` (klovve.drivers.viwid.viwid.widgets.widget.\_Property method), 73

`_get_dependencies()` (klovve.data.deps.DependencySubject method), 13

`_get_dependencies()` (klovve.data.model.Model method), 19

`_initially()` (klovve.data.model.PropertyBase

`method`), 21  
`_initially()` (`klovve.data.model._ComputedProperty` `method`), 22  
`_internal_mouse_click_event()` (`klovve.drivers.viwid.viwid.widgets.widget.Widget` `method`), 69  
`_is_protocol` (`klovve.data.lists.List` `attribute`), 16  
`_is_protocol` (`klovve.data.observable_value.CommonObservableValue` `attribute`), 23  
`_is_protocol` (`klovve.data.observable_value.ObservableValue` `attribute`), 23  
`_is_runtime_protocol` (`klovve.data.observable_value.ObservableValueProperty` `attribute`), 23  
`_keyboard_event()` (`klovve.drivers.viwid.viwid.widgets.button.Button` `method`), 65  
`_keyboard_event()` (`klovve.drivers.viwid.viwid.widgets.check_button.CheckButton` `method`), 65  
`_keyboard_event()` (`klovve.drivers.viwid.viwid.widgets.entry.Entry` `method`), 66  
`_keyboard_event()` (`klovve.drivers.viwid.viwid.widgets.list.List` `method`), 67  
`_keyboard_event()` (`klovve.drivers.viwid.viwid.widgets.widget.Widget` `method`), 70  
`_mouse_click_event()` (`klovve.drivers.viwid.viwid.widgets.button.Button` `method`), 65  
`_mouse_click_event()` (`klovve.drivers.viwid.viwid.widgets.check_button.CheckButton` `method`), 65  
`_mouse_click_event()` (`klovve.drivers.viwid.viwid.widgets.list.List` `method`), 67  
`_mouse_click_event()` (`klovve.drivers.viwid.viwid.widgets.widget.Widget` `method`), 70  
`_none()` (in module `klovve.data.model`), 22  
`_notify_dependency()` (`klovve.data.deps.DependencySubject` `method`), 13  
`_notify_property_value_changed()` (`klovve.data.deps.DependencySubject` `method`), 13  
`_orientation()` (`klovve.drivers.gtk.views._gtk_private.box.View` `method`), 34  
`_orientation()` (`klovve.drivers.gtk.views.horizontal_box.View` `method`), 43  
`_orientation()` (`klovve.drivers.gtk.views.vertical_box.View` `method`), 46  
`_orientation()` (`klovve.drivers.viwid.views._curses_private.box.View` `method`), 48  
`_orientation()` (`klovve.drivers.viwid.views.horizontal_box.View` `method`), 55  
`_orientation()` (`klovve.drivers.viwid.views.vertical_box.View` `method`), 58  
`_partitioning2()` (`klovve.drivers.viwid.viwid.widgets.box.Box` `method`), 64  
`_preview_internal_mouse_click_event()` (`klovve.drivers.viwid.viwid.widgets.widget.Widget` `method`), 70  
`_preview_keyboard_event()` (`klovve.drivers.viwid.viwid.widgets.scrollable.Scrollable` `method`), 68  
`_preview_keyboard_event()` (`klovve.drivers.viwid.viwid.widgets.widget.Widget` `method`), 70  
`_preview_mouse_click_event()` (`klovve.drivers.viwid.viwid.widgets.widget.Widget` `method`), 70  
`_preview_scroll_event()` (`klovve.drivers.viwid.viwid.widgets.scrollable.Scrollable` `method`), 68  
`_preview_scroll_event()` (`klovve.drivers.viwid.viwid.widgets.widget.Widget` `method`), 70  
`_print_loop()` (in module `klovve.debug`), 108  
`_resize()` (`klovve.drivers.gtk.views.viewer.image.Image` `method`), 37  
`_remove_changed_handler()` (`klovve.data.deps.DependencySubject` `method`), 14  
`_remove_changed_handler()` (`klovve.data.model.Model` `method`), 20  
`_remove_observer()` (`klovve.data.lists.List` `method`), 16  
`_remove_observer()` (`klovve.drivers.viwid.viwid.data.List` `method`), 76  
`_scroll_event()` (`klovve.drivers.viwid.viwid.widgets.widget.Widget` `method`), 70  
`_select()` (`klovve.drivers.viwid.viwid.widgets.dropdown.DropDown` `method`), 66  
`_set_dependencies()` (`klovve.data.deps.DependencySubject` `method`), 14  
`_set_dependencies()` (`klovve.data.model.Model` `method`), 20  
`_set_value()` (`klovve.data.model.Model` `method`), 20  
`_text_in_width()` (`klovve.drivers.viwid.viwid.widgets.label.Label` `method`), 67  
`_time_text()` (in module `klovve.drivers.viwid.views.log_pager`), 55  
`_time_text()` (in module `klovve.pieces.log_pager.entry`), 89  
`_trigger_changed()` (`klovve.data.model.Model` `method`), 20  
`_trigger_changed()` (`klovve.data.observable_value.CommonObservableValue` `method`), 23  
`_trigger_changed()` (`klovve.data.observable_value.ObservableValueProperty` `method`), 23



- `_try_find_type_in_module()` (in module `klovve.drivers`), 27
- `_try_find_type_per_name_tuple()` (in module `klovve.drivers`), 27
- `_update__default_create_target_object()` (in module `klovve.data.lists`), 17
- `_update__default_is_matching_target_object()` (in module `klovve.data.lists`), 17
- `_update__default_update_target_object()` (in module `klovve.data.lists`), 18
- `_use_last_value_during_recompute()` (`klovve.data.model.PropertyBase` method), 21
- `_use_last_value_during_recompute()` (`klovve.data.model._ComputedProperty` method), 22
- `_value()` (`klovve.data.model.Model` method), 20
- ## A
- `a` (`klovve.drivers.viwid.viwid.canvas.Canvas.Color` property), 74
- `a` (`klovve.drivers.viwid.viwid.canvas.Color` property), 75
- `action` (`klovve.pieces.button.Model` property), 93
- `ActionContext` (class in `klovve.app.context`), 11
- `ActionContext` (class in `klovve.drivers.gtk.views.button`), 40
- `ActionContext` (class in `klovve.drivers.gtk.views.window`), 46
- `ActionContext` (class in `klovve.drivers.viwid.views.button`), 51
- `ActionContext` (class in `klovve.drivers.viwid.views.window`), 58
- `actions` (`klovve.pieces.headered_panel.Model` property), 96
- `actions` (`klovve.pieces.window.Model` property), 104
- `add_changed_handler()` (`klovve.data.observable_value.CommonObservableValue` method), 23
- `add_changed_handler()` (`klovve.data.observable_value.ObservableValueProtocol` method), 23
- `add_fd()` (`klovve.drivers.gtk._Source` method), 34
- `add_root()` (`klovve.drivers.viwid.viwid.screen.Screen` method), 80
- `add_widget_screen()` (`klovve.drivers.viwid.viwid.drivers.curses.Driver` method), 61
- `add_widget_screen()` (`klovve.drivers.viwid.viwid.drivers.driver.Driver` method), 62
- `Alignment` (class in `klovve.drivers.viwid.viwid.widgets.widget`), 69
- `alive()` (`klovve.drivers.viwid.viwid.widgets.widget.Widget` method), 70
- `allows_multiselect` (`klovve.pieces.list.Model` property), 98
- `AnchorRootAlignmentPositioning` (class in `klovve.drivers.viwid.viwid.screen`), 78
- `answer` (`klovve.pieces.interact.Model` property), 84
- `Application` (class in `klovve.app`), 9
- `ApplicationUnavailableError`, 10
- `apply()` (`klovve.drivers.viwid.viwid.screen.RootAlignment` method), 79
- `AutoRootAlignmentSizing` (class in `klovve.drivers.viwid.viwid.screen`), 78
- `aux_controls` (`klovve.pieces.list.Model` property), 98
- `aux_controls` (`klovve.pieces.list_panel.Model` property), 99
- ## B
- `b` (`klovve.drivers.viwid.viwid.canvas.Canvas.Color` attribute), 74
- `b` (`klovve.drivers.viwid.viwid.canvas.Color` attribute), 75
- `background` (`klovve.drivers.viwid.viwid.widgets.widget.Widget` property), 70
- `base` (`klovve.drivers.viwid.viwid.canvas.Canvas.FullColor` attribute), 74
- `base` (`klovve.drivers.viwid.viwid.canvas.FullColor` attribute), 75
- `BaseView` (class in `klovve.view`), 105
- `BaseView.Rambo` (class in `klovve.view`), 105
- `began_at` (`klovve.pieces.log_pager.entry.Model` property), 88
- `began_at__text` (`klovve.pieces.log_pager.entry.Model` property), 88
- `bind()` (`klovve.drivers.gtk.View` static method), 32
- `bind()` (`klovve.drivers.viwid.View` static method), 47
- `BindFactory` (class in `klovve.data.value_holder`), 25
- `body` (`klovve.pieces.headered_panel.Model` property), 97
- `body` (`klovve.pieces.list_panel.Model` property), 99
- `body` (`klovve.pieces.window.Model` property), 104
- `boff` (`klovve.pieces.log_pager.Model` property), 87
- `bottom_right` (`klovve.drivers.viwid.viwid.Rectangle` property), 59
- `bottom_y` (`klovve.drivers.viwid.viwid.Rectangle` property), 59
- `bottommost` (`klovve.drivers.viwid.viwid.screen.Screen` property), 81
- `Box` (class in `klovve.drivers.viwid.viwid.widgets.box`), 64
- `BUSY` (`klovve.pieces.headered_panel.Model.State` attribute), 96
- `BusyAnimation` (class in `klovve.drivers.viwid.viwid.widgets.busy_animation`), 65
- `Button` (class in `klovve.drivers.viwid.viwid.widgets.button`), 65
- `ByScreenRootAlignmentSizing` (class in `klovve.drivers.viwid.viwid.screen`), 79

## C

`call_at()` (*klovve.drivers.gtk.\_EventLoop* method), 32  
`call_maybe_async_func()` (in module *klovve.app*), 10  
`Canvas` (class in *klovve.drivers.viuid.viuid.canvas*), 74  
`canvas()` (*klovve.drivers.viuid.viuid.drivers.curses.Driver* method), 61  
`canvas()` (*klovve.drivers.viuid.viuid.drivers.driver.Driver* method), 62  
`canvas()` (*klovve.drivers.viuid.viuid.widgets.widget.Widget* method), 70  
`Canvas.Color` (class in *klovve.drivers.viuid.viuid.canvas*), 74  
`Canvas.FullColor` (class in *klovve.drivers.viuid.viuid.canvas*), 74  
`CBox` (class in *klovve.drivers.viuid.viuid.widgets.box*), 64  
`CENTER` (*klovve.drivers.viuid.viuid.widgets.widget.Alignment* attribute), 69  
`CenterRootAlignmentPositioning` (class in *klovve.drivers.viuid.viuid.screen*), 79  
`check()` (*klovve.drivers.gtk.\_Source* method), 34  
`CheckButton` (class in *klovve.drivers.viuid.viuid.widgets.check\_button*), 65  
`checked` (*klovve.drivers.viuid.viuid.widgets.check\_button.CheckButton* property), 65  
`child_at_position()` (*klovve.drivers.viuid.viuid.widgets.widget.Widget* method), 71  
`children` (*klovve.drivers.viuid.viuid.widgets.box.Box* property), 64  
`children()` (in module *klovve.drivers.gtk*), 34  
`children()` (in module *klovve.drivers.viuid*), 47  
`class_style_name` (*klovve.drivers.viuid.viuid.widgets.box.CBox* property), 64  
`class_style_name` (*klovve.drivers.viuid.viuid.widgets.button.Button* property), 65  
`class_style_name` (*klovve.drivers.viuid.viuid.widgets.entry.Entry* property), 66  
`class_style_name` (*klovve.drivers.viuid.viuid.widgets.list.List* property), 67  
`class_style_name` (*klovve.drivers.viuid.viuid.widgets.list.ListRow* property), 67  
`class_style_name` (*klovve.drivers.viuid.viuid.widgets.progress\_bar.ProgressBar* property), 68  
`class_style_name` (*klovve.drivers.viuid.viuid.widgets.widget.Widget* property), 71  
`ClassStyle` (class in *klovve.drivers.viuid.viuid.style*), 82  
`close()` (*klovve.drivers.gtk.\_Selector* method), 33  
`close_func` (*klovve.pieces.window.Model* property), 104  
`closing` (*klovve.pieces.window.Model* property), 104  
`Color` (class in *klovve.drivers.viuid.viuid.canvas*), 75  
`color_pair()` (*klovve.drivers.viuid.viuid.drivers.curses.\_CursesColors* method), 61  
`COLORS_256` (*klovve.drivers.viuid.viuid.drivers.curses.\_CursesColors.Mode* attribute), 61  
`COLORS_8` (*klovve.drivers.viuid.viuid.drivers.curses.\_CursesColors.Mode* attribute), 61  
`colors_by_pair()` (*klovve.drivers.viuid.viuid.drivers.curses.\_CursesColors* method), 62  
`CommonObservableValue` (class in *klovve.data.observable\_value*), 23  
`compose()` (*klovve.drivers.gtk.views.tabbed.tab.View* method), 36  
`compose()` (*klovve.drivers.viuid.viuid.views.tabbed.tab.View* method), 50  
`compose()` (*klovve.pieces.interact.message.View* method), 85  
`compose()` (*klovve.pieces.interact.textline.View* method), 86  
`compose()` (*klovve.pieces.interact.triggers.View* method), 86  
`compose()` (*klovve.pieces.interact.yesno.View* method), 87  
`compose()` (*klovve.pieces.list\_panel.View* method), 100  
`compose()` (*klovve.view.ComposedView* method), 105  
`ComposedView` (class in *klovve.view*), 105  
`compute_height()` (*klovve.drivers.viuid.viuid.layout.GridLayout* method), 77  
`compute_height()` (*klovve.drivers.viuid.viuid.layout.Layout* method), 78  
`compute_height()` (*klovve.drivers.viuid.viuid.layout.NullLayout* method), 78  
`compute_height()` (*klovve.drivers.viuid.viuid.widgets.check\_button.CheckButton* method), 65  
`compute_height()` (*klovve.drivers.viuid.viuid.widgets.entry.Entry* method), 66  
`compute_height()` (*klovve.drivers.viuid.viuid.widgets.frame.Frame* method), 66  
`compute_height()` (*klovve.drivers.viuid.viuid.widgets.label.Label* method), 67  
`compute_height()` (*klovve.drivers.viuid.viuid.widgets.progress\_bar.ProgressBar* method), 68  
`compute_height()` (*klovve.drivers.viuid.viuid.widgets.scrollable.Scrollable* method), 68  
`compute_height()` (*klovve.drivers.viuid.viuid.widgets.widget.Widget* method), 71  
`compute_height2()` (*klovve.drivers.viuid.viuid.widgets.widget.Widget* method), 71  
`compute_width()` (*klovve.drivers.viuid.viuid.layout.GridLayout* method), 77  
`compute_width()` (*klovve.drivers.viuid.viuid.layout.Layout* method), 78  
`compute_width()` (*klovve.drivers.viuid.viuid.layout.NullLayout* method), 78  
`compute_width()` (*klovve.drivers.viuid.viuid.widgets.check\_button.CheckButton* method), 65  
`compute_width()` (*klovve.drivers.viuid.viuid.widgets.entry.Entry* method), 65

`method`), 66  
`compute_width()` (*klovve.drivers.viwid.viwid.widgets.frame.Frame* `method`), 41  
`method`), 66  
`compute_width()` (*klovve.drivers.viwid.viwid.widgets.label.Label* `method`), 41  
`method`), 67  
`compute_width()` (*klovve.drivers.viwid.viwid.widgets.progress\_bar.ProgressBar* `method`), 68  
`method`), 68  
`compute_width()` (*klovve.drivers.viwid.viwid.widgets.scrollable.Scrollable* `method`), 35  
`method`), 68  
`compute_width()` (*klovve.drivers.viwid.viwid.widgets.widget.Widget* `method`), 35  
`method`), 71  
`compute_width2()` (*klovve.drivers.viwid.viwid.widgets.widget.Widget* `method`), 42  
`method`), 71  
`computed_horizontal_expand_greedily` (*klovve.drivers.viwid.viwid.widgets.widget.Widget* `property`), 71  
`computed_is_disabled` (*klovve.drivers.viwid.viwid.widgets.widget.Widget* `property`), 71  
`computed_vertical_expand_greedily` (*klovve.drivers.viwid.viwid.widgets.widget.Widget* `property`), 71  
`ComputedProperty()` (in module *klovve.data.model*), 18  
`ComputedValue` (class in *klovve.data.value\_holder*), 25  
`create_app()` (in module *klovve.app*), 11  
`create_native()` (*klovve.drivers.bare\_terminal.views.checkbox.View* `method`), 29  
`create_native()` (*klovve.drivers.bare\_terminal.views.header\_panel.View* `method`), 29  
`create_native()` (*klovve.drivers.bare\_terminal.views.interaction.View* `method`), 28  
`create_native()` (*klovve.drivers.bare\_terminal.views.label.View* `method`), 30  
`create_native()` (*klovve.drivers.bare\_terminal.views.log\_page.View* `method`), 30  
`create_native()` (*klovve.drivers.bare\_terminal.views.placeholder.View* `method`), 30  
`create_native()` (*klovve.drivers.bare\_terminal.views.split.View* `method`), 31  
`create_native()` (*klovve.drivers.bare\_terminal.views.vertical\_box.View* `method`), 31  
`create_native()` (*klovve.drivers.bare\_terminal.views.viewer.image.View* `method`), 29  
`create_native()` (*klovve.drivers.bare\_terminal.views.viewer.pdf.View* `method`), 29  
`create_native()` (*klovve.drivers.bare\_terminal.views.window.View* `method`), 31  
`create_native()` (*klovve.drivers.gtk.views.\_gtk\_private.box.View* `method`), 34  
`create_native()` (*klovve.drivers.gtk.views.busy\_animation.View* `method`), 39  
`create_native()` (*klovve.drivers.gtk.views.button.View* `method`), 40  
`create_native()` (*klovve.drivers.gtk.views.checkable.View* `method`), 40  
`create_native()` (*klovve.drivers.gtk.views.disableable.View* `method`), 41  
`create_native()` (*klovve.drivers.gtk.views.dropdown.View* `method`), 41  
`create_native()` (*klovve.drivers.gtk.views.dual.View* `method`), 41  
`create_native()` (*klovve.drivers.gtk.views.form.section.View* `method`), 41  
`create_native()` (*klovve.drivers.gtk.views.form.View* `method`), 41  
`create_native()` (*klovve.drivers.gtk.views.formatted\_text.View* `method`), 42  
`create_native()` (*klovve.drivers.gtk.views.header.View* `method`), 42  
`create_native()` (*klovve.drivers.gtk.views.headered\_panel.View* `method`), 42  
`create_native()` (*klovve.drivers.gtk.views.interact.abstract.View* `method`), 35  
`create_native()` (*klovve.drivers.gtk.views.label.View* `method`), 43  
`create_native()` (*klovve.drivers.gtk.views.list.View* `method`), 43  
`create_native()` (*klovve.drivers.gtk.views.log\_pager.View* `method`), 44  
`create_native()` (*klovve.drivers.gtk.views.placeholder.View* `method`), 44  
`create_native()` (*klovve.drivers.gtk.views.progress.View* `method`), 44  
`create_native()` (*klovve.drivers.gtk.views.property\_panel.View* `method`), 36  
`create_native()` (*klovve.drivers.gtk.views.scrollable.View* `method`), 45  
`create_native()` (*klovve.drivers.gtk.views.slider.View* `method`), 45  
`create_native()` (*klovve.drivers.gtk.views.split.View* `method`), 45  
`create_native()` (*klovve.drivers.gtk.views.tabbed.View* `method`), 36  
`create_native()` (*klovve.drivers.gtk.views.text\_field.View* `method`), 46  
`create_native()` (*klovve.drivers.gtk.views.viewer.image.View* `method`), 39  
`create_native()` (*klovve.drivers.gtk.views.viewer.pdf.View* `method`), 39  
`create_native()` (*klovve.drivers.gtk.views.window.View* `method`), 47  
`create_native()` (*klovve.drivers.viwid.views.\_curses\_private.box.View* `method`), 48  
`create_native()` (*klovve.drivers.viwid.views.busy\_animation.View* `method`), 51  
`create_native()` (*klovve.drivers.viwid.views.button.View* `method`), 52  
`create_native()` (*klovve.drivers.viwid.views.checkable.View* `method`), 52

`create_native()` (*klovve.drivers.viuid.views.disableable.View method*), 52  
`create_native()` (*klovve.drivers.viuid.views.dropdown.View method*), 53  
`create_native()` (*klovve.drivers.viuid.views.dual.View method*), 53  
`create_native()` (*klovve.drivers.viuid.views.form.section.View method*), 49  
`create_native()` (*klovve.drivers.viuid.views.form.View method*), 48  
`create_native()` (*klovve.drivers.viuid.views.formatted\_text.View method*), 53  
`create_native()` (*klovve.drivers.viuid.views.header.View method*), 54  
`create_native()` (*klovve.drivers.viuid.views.headered\_panel.View method*), 54  
`create_native()` (*klovve.drivers.viuid.views.interact.abstract\_view.View method*), 49  
`create_native()` (*klovve.drivers.viuid.views.label.View method*), 55  
`create_native()` (*klovve.drivers.viuid.views.list.View method*), 55  
`create_native()` (*klovve.drivers.viuid.views.log\_pager.View method*), 55  
`create_native()` (*klovve.drivers.viuid.views.placeholder.View method*), 56  
`create_native()` (*klovve.drivers.viuid.views.progress.View method*), 56  
`create_native()` (*klovve.drivers.viuid.views.property\_panel.View method*), 49  
`create_native()` (*klovve.drivers.viuid.views.scrollable.View method*), 56  
`create_native()` (*klovve.drivers.viuid.views.slider.View method*), 57  
`create_native()` (*klovve.drivers.viuid.views.split.View method*), 57  
`create_native()` (*klovve.drivers.viuid.views.tabbed.View method*), 50  
`create_native()` (*klovve.drivers.viuid.views.text\_field.View method*), 57  
`create_native()` (*klovve.drivers.viuid.views.viewer.image.View method*), 50  
`create_native()` (*klovve.drivers.viuid.views.viewer.pdf.View method*), 51  
`create_native()` (*klovve.drivers.viuid.views.window.View method*), 58  
`create_native()` (*klovve.view.BaseView method*), 105  
`create_native()` (*klovve.view.ComposedView method*), 105  
`create_or_update_view()` (*klovve.view.tree.ViewTreeNode method*), 107  
`create_task()` (*klovve.drivers.gtk.\_EventLoop method*), 32  
`create_view()` (*klovve.view.tree.ViewFactory method*), 106  
`current` (*klovve.debug.\_Memory property*), 108  
`curses_colors` (*klovve.drivers.viuid.viuid.drivers.curses.\_CursesColors attribute*), 62  
`curses_colors` (*klovve.drivers.viuid.viuid.drivers.curses.Driver property*), 61  
`cursor()` (*klovve.drivers.viuid.viuid.canvas.Canvas method*), 74  
`cursor()` (*klovve.drivers.viuid.viuid.drivers.curses.\_Canvas method*), 61  
`cursor_position` (*klovve.drivers.viuid.viuid.widgets.entry.Entry property*), 66  
**D**  
`data` (*klovve.view.tree.ViewTreeNode property*), 107  
`delete_view()` (*klovve.view.tree.View method*), 24  
`dematerialize()` (*klovve.drivers.viuid.viuid.widgets.widget.Widget method*), 71  
`DependencySubject` (*class in klovve.data.deps*), 12  
`dialog()` (*klovve.app.context.ActionContext method*), 12  
`direction` (*klovve.drivers.viuid.viuid.event.input.OngoingScrollEvent property*), 64  
`dispatch()` (*klovve.drivers.gtk.\_Source method*), 34  
`display_coordinates_to_image_coordinates()` (*klovve.drivers.gtk.views.viewer.image.Image method*), 37  
`display_height` (*klovve.drivers.gtk.views.viewer.image.Image property*), 37  
`display_width` (*klovve.drivers.gtk.views.viewer.image.Image property*), 37  
`do_get_property()` (*klovve.drivers.gtk.views.viewer.image.Image method*), 37  
`do_get_property()` (*klovve.drivers.gtk.views.viewer.image.Outer method*), 38  
`do_measure()` (*klovve.drivers.gtk.views.viewer.image.Outer method*), 38  
`do_set_property()` (*klovve.drivers.gtk.views.viewer.image.Image method*), 37  
`do_set_property()` (*klovve.drivers.gtk.views.viewer.image.Outer method*), 38  
`do_size_allocate()` (*klovve.drivers.gtk.views.viewer.image.Outer method*), 38  
`Driver` (*class in klovve.drivers*), 26  
`Driver` (*class in klovve.drivers.bare\_terminal*), 28  
`Driver` (*class in klovve.drivers.gtk*), 32  
`Driver` (*class in klovve.drivers.viuid*), 47  
`Driver` (*class in klovve.drivers.viuid.viuid.drivers.curses*), 61  
`Driver` (*class in klovve.drivers.viuid.viuid.drivers.driver*), 62  
`driver` (*klovve.drivers.viuid.viuid.screen.Screen property*), 81



**DropDown** (class in klovve.drivers.viuid.viuid.widgets.dropdown), 66

**E**

**ended\_at** (klovve.pieces.log\_pager.entry.Model property), 88

**ended\_at\_\_text** (klovve.pieces.log\_pager.entry.Model property), 88

**entries** (klovve.pieces.log\_pager.entry.Model property), 88

**entries** (klovve.pieces.log\_pager.Model property), 87

**entries** (klovve.pieces.property\_panel.Model property), 89

**Entry** (class in klovve.drivers.viuid.viuid.widgets.entry), 66

**error\_message\_for\_exceptions()** (klovve.app.context.ActionContext method), 12

**event\_loop** (klovve.drivers.viuid.viuid.loop.Loop property), 78

**EVENT\_TO\_GLIB** (klovve.drivers.gtk.\_Source attribute), 33

**extend\_by()** (klovve.drivers.viuid.viuid.Size method), 59

**EXTRA\_LARGE** (klovve.pieces.busy\_animation.Model.Size attribute), 92

**F**

**FAILED** (klovve.pieces.headered\_panel.Model.State attribute), 96

**FILL** (klovve.drivers.viuid.viuid.widgets.widget.Alignment attribute), 69

**fill()** (klovve.drivers.viuid.viuid.canvas.Canvas method), 74

**fill()** (klovve.drivers.viuid.viuid.drivers.curses.\_Canvas method), 61

**FixedRootAlignmentPositioning** (class in klovve.drivers.viuid.viuid.screen), 79

**FixedRootAlignmentSizing** (class in klovve.drivers.viuid.viuid.screen), 79

**fizzwid\_new()** (klovve.drivers.viuid.View static method), 47

**focusable** (klovve.drivers.viuid.viuid.widgets.button.Button property), 65

**focusable** (klovve.drivers.viuid.viuid.widgets.entry.Entry property), 66

**focusable** (klovve.drivers.viuid.viuid.widgets.list.List property), 67

**focusable** (klovve.drivers.viuid.viuid.widgets.scrollable.Scrollable property), 68

**focusable** (klovve.drivers.viuid.viuid.widgets.widget.Widget property), 71

**focussed()** (klovve.drivers.viuid.viuid.widgets.widget.Widget method), 69

**focussed\_widget** (klovve.drivers.viuid.viuid.drivers.curses.\_Screen property), 62

**focussed\_widget** (klovve.drivers.viuid.viuid.screen.Screen property), 81

**foreground** (klovve.drivers.viuid.viuid.widgets.widget.Widget property), 71

**Frame** (class in klovve.drivers.viuid.viuid.widgets.frame), 66

**from\_string()** (klovve.drivers.viuid.viuid.canvas.Canvas.Color static method), 74

**from\_string()** (klovve.drivers.viuid.viuid.canvas.Canvas.FullColor static method), 74

**from\_string()** (klovve.drivers.viuid.viuid.canvas.Color static method), 75

**from\_string()** (klovve.drivers.viuid.viuid.canvas.FullColor static method), 75

**fuh()** (klovve.data.value\_holder.BindFactory method), 25

**FullColor** (class in klovve.drivers.viuid.viuid.canvas), 75

**G**

**g** (klovve.drivers.viuid.viuid.canvas.Canvas.Color attribute), 74

**g** (klovve.drivers.viuid.viuid.canvas.Color attribute), 75

**get()** (klovve.drivers.Driver static method), 26

**get\_answer\_func** (klovve.pieces.interact.abstract.Model property), 84

**get\_driver()** (in module klovve.drivers.viuid.viuid.drivers), 60

**get\_events()** (klovve.drivers.gtk.\_Source method), 34

**get\_height()** (klovve.drivers.viuid.viuid.screen.AutoRootAlignmentSizing method), 79

**get\_height()** (klovve.drivers.viuid.viuid.screen.ByScreenRootAlignmentSizing method), 79

**get\_height()** (klovve.drivers.viuid.viuid.screen.FixedRootAlignmentSizing method), 79

**get\_height()** (klovve.drivers.viuid.viuid.screen.RootAlignmentSizing method), 80

**get\_value()** (klovve.data.value\_holder.ComputedValue method), 25

**get\_value()** (klovve.data.value\_holder.OneWayBinding method), 25

**get\_value()** (klovve.data.value\_holder.ValueHolder method), 25

**get\_width()** (klovve.drivers.viuid.viuid.screen.AutoRootAlignmentSizing method), 79

**get\_width()** (klovve.drivers.viuid.viuid.screen.ByScreenRootAlignmentSizing method), 79

**get\_width()** (klovve.drivers.viuid.viuid.screen.FixedRootAlignmentSizing method), 79

**get\_width()** (klovve.drivers.viuid.viuid.screen.RootAlignmentSizing method), 80

`get_x()` (*klovve.drivers.viwid.viwid.screen.AnchorRootAlignmentPositioning* attribute), 78  
`get_x()` (*klovve.drivers.viwid.viwid.screen.CenterRootAlignmentPositioning* attribute), 79  
`get_x()` (*klovve.drivers.viwid.viwid.screen.FixedRootAlignmentPositioning* attribute), 79  
`get_x()` (*klovve.drivers.viwid.viwid.screen.RootAlignmentPositioning* attribute), 79  
`get_y()` (*klovve.drivers.viwid.viwid.screen.AnchorRootAlignmentPositioning* attribute), 78  
`get_y()` (*klovve.drivers.viwid.viwid.screen.CenterRootAlignmentPositioning* attribute), 79  
`get_y()` (*klovve.drivers.viwid.viwid.screen.FixedRootAlignmentPositioning* attribute), 79  
`get_y()` (*klovve.drivers.viwid.viwid.screen.RootAlignmentPositioning* attribute), 80  
`GridLayout` (class in *klovve.drivers.viwid.viwid.layout*), 77  
`gtk_children()` (*klovve.drivers.gtk.View* static method), 32  
`gtk_new()` (*klovve.drivers.gtk.View* static method), 32  
`gtk_orientation` (*klovve.drivers.gtk.views.busy\_animation.Model* property), 39  
`gtk_orientation()` (*klovve.drivers.viwid.views.busy\_animation.Model* property), 51  
`gtk_sensitive` (*klovve.drivers.gtk.views.disableable.View* property), 41  
**H**  
`handle_event()` (*klovve.drivers.viwid.viwid.screen.Screen* static method), 81  
`height` (*klovve.drivers.viwid.viwid.Rectangle* property), 59  
`height` (*klovve.drivers.viwid.viwid.Size* property), 60  
`hint_text` (*klovve.pieces.text\_field.Model* property), 103  
`HORIZONTAL` (*klovve.drivers.viwid.viwid.widgets.box.Orientation* attribute), 65  
`HORIZONTAL` (*klovve.pieces.busy\_animation.Model.Orientation* attribute), 92  
`horizontal_alignment` (*klovve.drivers.viwid.viwid.widgets.widget.Widget* property), 71  
`horizontal_expand_greedily` (*klovve.drivers.viwid.viwid.widgets.widget.Widget* property), 71  
`horizontally_scrollable` (*klovve.drivers.viwid.viwid.widgets.scrollable.Scrollable* property), 68  
**I**  
`icon` (*klovve.pieces.window.Model* property), 104  
`Image` (class in *klovve.drivers.gtk.views.viewer.image*), 37  
`image_coordinates_to_display_coordinates()` (*klovve.drivers.gtk.views.viewer.image.Image* method), 37  
`image_offset_x` (*klovve.drivers.gtk.views.viewer.image.Image* attribute), 37  
`image_offset_y` (*klovve.drivers.gtk.views.viewer.image.Image* attribute), 37  
`in_mainloop()` (in module *klovve.app*), 11  
`inner_size` (*klovve.drivers.viwid.viwid.widgets.widget.Widget* property), 71  
`inner_view` (*klovve.pieces.interact.abstract.Model* property), 84  
`insert()` (*klovve.data.lists.List* method), 16  
`insert()` (*klovve.drivers.viwid.viwid.data.List* method), 76  
`interrupt_glib_mainloop()` (*klovve.drivers.gtk.\_Selector* method), 33  
`is_answered` (*klovve.pieces.interact.Model* property), 84  
`is_checked` (*klovve.pieces.checkable.Model* property), 93  
`is_closed` (*klovve.pieces.window.Model* property), 104  
`is_disabled` (*klovve.drivers.viwid.viwid.widgets.widget.Widget* property), 72  
`is_disabled` (*klovve.pieces.disableable.Model* property), 94  
`is_propagation_stopped` (*klovve.drivers.viwid.viwid.event.base.OngoingEvent* property), 63  
`is_running` (*klovve.app.Application* property), 10  
`is_selected` (*klovve.drivers.viwid.viwid.widgets.list.ListRow* property), 67  
`is_settable()` (*klovve.data.value\_holder.ComputedValue* method), 25  
`is_settable()` (*klovve.data.value\_holder.OneWayBinding* method), 25  
`is_settable()` (*klovve.data.value\_holder.TwoWayBinding* method), 25  
`is_settable()` (*klovve.data.value\_holder.ValueHolder* method), 25  
`item` (*klovve.drivers.viwid.viwid.widgets.frame.Frame* property), 66  
`item` (*klovve.drivers.viwid.viwid.widgets.scrollable.Scrollable* property), 68  
`item` (*klovve.pieces.disableable.Model* property), 94  
`item` (*klovve.pieces.form.section.Model* property), 83  
`item` (*klovve.pieces.placeholder.Model* property), 100  
`item` (*klovve.pieces.scrollable.Model* property), 101  
`item` (*klovve.pieces.tabbed.tab.Model* property), 90  
`item1` (*klovve.pieces.split.Model* property), 102  
`item2` (*klovve.pieces.split.Model* property), 102  
`item_actions` (*klovve.pieces.list.Model* property), 98  
`item_actions` (*klovve.pieces.list\_panel.Model* property), 99

item\_added() (*klovve.data.lists.ListObserver* method), 16  
 item\_added() (*klovve.drivers.viwid.viwid.data.ListObserver* method), 76  
 item\_label\_func (*klovve.pieces.dropdown.Model* property), 94  
 item\_label\_func (*klovve.pieces.list.Model* property), 98  
 item\_label\_func (*klovve.pieces.list\_panel.Model* property), 100  
 item\_moved() (*klovve.data.lists.ListObserver* method), 17  
 item\_removed() (*klovve.data.lists.ListObserver* method), 17  
 item\_removed() (*klovve.drivers.viwid.viwid.data.ListObserver* method), 76  
 item\_replaced() (*klovve.data.lists.ListObserver* method), 17  
 item\_replaced() (*klovve.drivers.viwid.viwid.data.ListObserver* method), 76  
 items (*klovve.drivers.viwid.viwid.widgets.dropdown.DropDown* property), 66  
 items (*klovve.drivers.viwid.viwid.widgets.list.List* property), 67  
 items (*klovve.pieces.dropdown.Model* property), 94  
 items (*klovve.pieces.form.Model* property), 83  
 items (*klovve.pieces.horizontal\_box.Model* property), 97  
 items (*klovve.pieces.list.Model* property), 99  
 items (*klovve.pieces.list\_panel.Model* property), 100  
 items (*klovve.pieces.tabbed.Model* property), 90  
 items (*klovve.pieces.vertical\_box.Model* property), 103  
**J**  
 jumpto() (*klovve.drivers.viwid.viwid.widgets.widget.Widget* method), 69  
**K**  
 key (*klovve.drivers.viwid.viwid.event.input.OngoingKeyboardEvent* property), 63  
 klovve  
   module, 9  
 klovve.+tests  
   module, 9  
 klovve.app  
   module, 9  
 klovve.app.context  
   module, 11  
 klovve.data  
   module, 12  
 klovve.data.deps  
   module, 12  
 klovve.data.lists  
   module, 15  
 klovve.data.model  
   module, 18  
   klovve.data.observable\_value  
     module, 23  
   klovve.data.reaction  
     module, 24  
   klovve.data.value\_holder  
     module, 25  
   klovve.debug  
     module, 108  
   klovve.drivers  
     module, 26  
   klovve.drivers.bare\_terminal  
     module, 28  
   klovve.drivers.bare\_terminal.views  
     klovve.drivers.bare\_terminal.views.checkable  
       module, 29  
   klovve.drivers.bare\_terminal.views.headered\_panel  
     module, 29  
   klovve.drivers.bare\_terminal.views.interact  
     module, 28  
   klovve.drivers.bare\_terminal.views.interact.abstract  
     module, 28  
   klovve.drivers.bare\_terminal.views.label  
     module, 30  
   klovve.drivers.bare\_terminal.views.log\_pager  
     module, 30  
   klovve.drivers.bare\_terminal.views.placeholder  
     module, 30  
   klovve.drivers.bare\_terminal.views.split  
     module, 31  
   klovve.drivers.bare\_terminal.views.vertical\_box  
     module, 31  
   klovve.drivers.bare\_terminal.views.viewer  
     module, 29  
   klovve.drivers.bare\_terminal.views.viewer.image  
     module, 29  
   klovve.drivers.bare\_terminal.views.window  
     module, 31  
   klovve.drivers.gtk  
     module, 32  
   klovve.drivers.gtk.views  
     module, 34  
   klovve.drivers.gtk.views.\_gtk\_private  
     module, 34  
   klovve.drivers.gtk.views.\_gtk\_private.box  
     module, 34  
   klovve.drivers.gtk.views.busy\_animation  
     module, 39  
   klovve.drivers.gtk.views.button  
     module, 40  
   klovve.drivers.gtk.views.checkable  
     module, 40  
   klovve.drivers.gtk.views.disableable

- module, 41
- klovve.drivers.gtk.views.dropdown
  - module, 41
- klovve.drivers.gtk.views.dual
  - module, 41
- klovve.drivers.gtk.views.form
  - module, 35
- klovve.drivers.gtk.views.form.section
  - module, 35
- klovve.drivers.gtk.views.formatted\_text
  - module, 42
- klovve.drivers.gtk.views.header
  - module, 42
- klovve.drivers.gtk.views.headered\_panel
  - module, 42
- klovve.drivers.gtk.views.horizontal\_box
  - module, 43
- klovve.drivers.gtk.views.interact
  - module, 35
- klovve.drivers.gtk.views.interact.abstract
  - module, 35
- klovve.drivers.gtk.views.label
  - module, 43
- klovve.drivers.gtk.views.list
  - module, 43
- klovve.drivers.gtk.views.log\_pager
  - module, 44
- klovve.drivers.gtk.views.placeholder
  - module, 44
- klovve.drivers.gtk.views.progress
  - module, 44
- klovve.drivers.gtk.views.property\_panel
  - module, 36
- klovve.drivers.gtk.views.scrollable
  - module, 45
- klovve.drivers.gtk.views.slider
  - module, 45
- klovve.drivers.gtk.views.split
  - module, 45
- klovve.drivers.gtk.views.tabbed
  - module, 36
- klovve.drivers.gtk.views.tabbed.tab
  - module, 36
- klovve.drivers.gtk.views.text\_field
  - module, 46
- klovve.drivers.gtk.views.vertical\_box
  - module, 46
- klovve.drivers.gtk.views.viewer
  - module, 37
- klovve.drivers.gtk.views.viewer.image
  - module, 37
- klovve.drivers.gtk.views.viewer.pdf
  - module, 39
- klovve.drivers.gtk.views.window

- module, 46
- klovve.drivers.viwid
  - module, 47
- klovve.drivers.viwid.views
  - module, 48
- klovve.drivers.viwid.views.\_curses\_private
  - module, 48
- klovve.drivers.viwid.views.\_curses\_private.box
  - module, 48
- klovve.drivers.viwid.views.busy\_animation
  - module, 51
- klovve.drivers.viwid.views.button
  - module, 51
- klovve.drivers.viwid.views.checkable
  - module, 52
- klovve.drivers.viwid.views.disableable
  - module, 52
- klovve.drivers.viwid.views.dropdown
  - module, 53
- klovve.drivers.viwid.views.dual
  - module, 53
- klovve.drivers.viwid.views.form
  - module, 48
- klovve.drivers.viwid.views.form.section
  - module, 49
- klovve.drivers.viwid.views.formatted\_text
  - module, 53
- klovve.drivers.viwid.views.header
  - module, 54
- klovve.drivers.viwid.views.headered\_panel
  - module, 54
- klovve.drivers.viwid.views.horizontal\_box
  - module, 54
- klovve.drivers.viwid.views.interact
  - module, 49
- klovve.drivers.viwid.views.interact.abstract
  - module, 49
- klovve.drivers.viwid.views.label
  - module, 55
- klovve.drivers.viwid.views.list
  - module, 55
- klovve.drivers.viwid.views.log\_pager
  - module, 55
- klovve.drivers.viwid.views.placeholder
  - module, 56
- klovve.drivers.viwid.views.progress
  - module, 56
- klovve.drivers.viwid.views.property\_panel
  - module, 49
- klovve.drivers.viwid.views.scrollable
  - module, 56
- klovve.drivers.viwid.views.slider
  - module, 57
- klovve.drivers.viwid.views.split



---

module, 57	module, 65
klovve.drivers.viwid.views.tabbed	klovve.drivers.viwid.viwid.widgets.dropdown
module, 50	module, 66
klovve.drivers.viwid.views.tabbed.tab	klovve.drivers.viwid.viwid.widgets.entry
module, 50	module, 66
klovve.drivers.viwid.views.text_field	klovve.drivers.viwid.viwid.widgets.frame
module, 57	module, 66
klovve.drivers.viwid.views.vertical_box	klovve.drivers.viwid.viwid.widgets.label
module, 58	module, 67
klovve.drivers.viwid.views.viewer	klovve.drivers.viwid.viwid.widgets.list
module, 50	module, 67
klovve.drivers.viwid.views.viewer.image	klovve.drivers.viwid.viwid.widgets.progress_bar
module, 50	module, 68
klovve.drivers.viwid.views.viewer.pdf	klovve.drivers.viwid.viwid.widgets.radio_button
module, 51	module, 68
klovve.drivers.viwid.views.window	klovve.drivers.viwid.viwid.widgets.scrollable
module, 58	module, 68
klovve.drivers.viwid.viwid	klovve.drivers.viwid.viwid.widgets.widget
module, 58	module, 69
klovve.drivers.viwid.viwid.+samples	klovve.pieces
module, 60	module, 83
klovve.drivers.viwid.viwid.canvas	klovve.pieces.busy_animation
module, 74	module, 92
klovve.drivers.viwid.viwid.data	klovve.pieces.button
module, 76	module, 93
klovve.drivers.viwid.viwid.drivers	klovve.pieces.checkable
module, 60	module, 93
klovve.drivers.viwid.viwid.drivers.curses	klovve.pieces.disableable
module, 61	module, 94
klovve.drivers.viwid.viwid.drivers.driver	klovve.pieces.dropdown
module, 62	module, 94
klovve.drivers.viwid.viwid.event	klovve.pieces.dual
module, 63	module, 95
klovve.drivers.viwid.viwid.event.base	klovve.pieces.form
module, 63	module, 83
klovve.drivers.viwid.viwid.event.input	klovve.pieces.form.section
module, 63	module, 83
klovve.drivers.viwid.viwid.layout	klovve.pieces.formatted_text
module, 77	module, 95
klovve.drivers.viwid.viwid.loop	klovve.pieces.header
module, 78	module, 96
klovve.drivers.viwid.viwid.screen	klovve.pieces.headered_panel
module, 78	module, 96
klovve.drivers.viwid.viwid.style	klovve.pieces.horizontal_box
module, 82	module, 97
klovve.drivers.viwid.viwid.widgets	klovve.pieces.interact
module, 64	module, 84
klovve.drivers.viwid.viwid.widgets.box	klovve.pieces.interact.abstract
module, 64	module, 84
klovve.drivers.viwid.viwid.widgets.busy_animation	klovve.pieces.interact.message
module, 65	module, 85
klovve.drivers.viwid.viwid.widgets.button	klovve.pieces.interact.textline
module, 65	module, 85
klovve.drivers.viwid.viwid.widgets.check_button	klovve.pieces.interact.triggers

module, 86  
 klovve.pieces.interact.yesno  
   module, 87  
 klovve.pieces.label  
   module, 98  
 klovve.pieces.list  
   module, 98  
 klovve.pieces.list\_panel  
   module, 99  
 klovve.pieces.log\_pager  
   module, 87  
 klovve.pieces.log\_pager.entry  
   module, 88  
 klovve.pieces.placeholder  
   module, 100  
 klovve.pieces.progress  
   module, 101  
 klovve.pieces.property\_panel  
   module, 89  
 klovve.pieces.property\_panel.entry  
   module, 89  
 klovve.pieces.scrollable  
   module, 101  
 klovve.pieces.slider  
   module, 101  
 klovve.pieces.split  
   module, 102  
 klovve.pieces.tabbed  
   module, 90  
 klovve.pieces.tabbed.tab  
   module, 90  
 klovve.pieces.text\_field  
   module, 103  
 klovve.pieces.vertical\_box  
   module, 103  
 klovve.pieces.viewer  
   module, 91  
 klovve.pieces.viewer.image  
   module, 91  
 klovve.pieces.viewer.pdf  
   module, 91  
 klovve.pieces.window  
   module, 104  
 klovve.view  
   module, 105  
 klovve.view.tree  
   module, 106

## L

Label (class in klovve.drivers.viwid.viwid.widgets.label), 67  
 label (klovve.pieces.form.section.Model property), 83  
 label (klovve.pieces.property\_panel.entry.Model property), 89

label (klovve.pieces.tabbed.tab.Model property), 90  
 LARGE (klovve.pieces.busy\_animation.Model.Size attribute), 92  
 Layout (class in klovve.drivers.viwid.viwid.layout), 77  
 layout (klovve.drivers.viwid.viwid.widgets.widget.Widget property), 72  
 left\_button (klovve.drivers.viwid.viwid.event.input.OngoingMouseClicked property), 64  
 left\_x (klovve.drivers.viwid.viwid.Rectangle property), 59  
 List (class in klovve.data.lists), 15  
 List (class in klovve.drivers.viwid.viwid.data), 76  
 List (class in klovve.drivers.viwid.viwid.widgets.list), 67  
 list\_actions (klovve.pieces.list.Model property), 99  
 list\_actions (klovve.pieces.list\_panel.Model property), 100  
 listen\_property() (klovve.drivers.viwid.viwid.widgets.widget.Widget method), 72  
 ListObserver (class in klovve.data.lists), 16  
 ListObserver (class in klovve.drivers.viwid.viwid.data), 76  
 ListProperty (class in klovve.data.model), 18  
 ListProperty() (in module klovve.drivers.viwid.viwid.widgets.widget), 69  
 ListProperty() (klovve.drivers.viwid.viwid.widgets.widget.Widget method), 69  
 ListRow (class in klovve.drivers.viwid.viwid.widgets.list), 67  
 Loop (class in klovve.drivers.viwid.viwid.loop), 78  
 loop (klovve.drivers.viwid.viwid.drivers.curses.Driver property), 61  
 loop (klovve.drivers.viwid.viwid.drivers.driver.Driver property), 62

## M

main\_item (klovve.pieces.dual.Model property), 95  
 mainloop() (in module klovve.app), 11  
 mainloop() (klovve.drivers.Driver static method), 26  
 MainloopObjectProxy (class in klovve.app), 10  
 make\_view() (klovve.view.BaseView method), 105  
 materialize() (klovve.drivers.viwid.viwid.widgets.widget.Widget method), 72  
 max\_value (klovve.pieces.slider.Model property), 101  
 MEDIUM (klovve.pieces.busy\_animation.Model.Size attribute), 92  
 message (klovve.pieces.interact.abstract.Model property), 84  
 message (klovve.pieces.interact.message.Model property), 85  
 message (klovve.pieces.interact.textline.Model property), 85  
 message (klovve.pieces.interact.triggers.Model property), 86

message (*klovve.pieces.interact.yesno.Model* property), 87  
 message (*klovve.pieces.log\_pager.entry.Model* property), 88  
 min\_value (*klovve.pieces.slider.Model* property), 102  
 Model (class in *klovve.data.model*), 18  
 Model (class in *klovve.drivers.viwid.views.busy\_animation*), 51  
 Model (class in *klovve.drivers.viwid.views.headered\_panel*), 54  
 Model (class in *klovve.pieces.busy\_animation*), 92  
 Model (class in *klovve.pieces.button*), 93  
 Model (class in *klovve.pieces.checkable*), 93  
 Model (class in *klovve.pieces.disableable*), 94  
 Model (class in *klovve.pieces.dropdown*), 94  
 Model (class in *klovve.pieces.dual*), 95  
 Model (class in *klovve.pieces.form*), 83  
 Model (class in *klovve.pieces.form.section*), 83  
 Model (class in *klovve.pieces.formatted\_text*), 95  
 Model (class in *klovve.pieces.header*), 96  
 Model (class in *klovve.pieces.headered\_panel*), 96  
 Model (class in *klovve.pieces.horizontal\_box*), 97  
 Model (class in *klovve.pieces.interact*), 84  
 Model (class in *klovve.pieces.interact.abstract*), 84  
 Model (class in *klovve.pieces.interact.message*), 85  
 Model (class in *klovve.pieces.interact.textline*), 85  
 Model (class in *klovve.pieces.interact.triggers*), 86  
 Model (class in *klovve.pieces.interact.yesno*), 87  
 Model (class in *klovve.pieces.label*), 98  
 Model (class in *klovve.pieces.list*), 98  
 Model (class in *klovve.pieces.list\_panel*), 99  
 Model (class in *klovve.pieces.log\_pager*), 87  
 Model (class in *klovve.pieces.log\_pager.entry*), 88  
 Model (class in *klovve.pieces.placeholder*), 100  
 Model (class in *klovve.pieces.progress*), 101  
 Model (class in *klovve.pieces.property\_panel*), 89  
 Model (class in *klovve.pieces.property\_panel.entry*), 89  
 Model (class in *klovve.pieces.scrollable*), 101  
 Model (class in *klovve.pieces.slider*), 101  
 Model (class in *klovve.pieces.split*), 102  
 Model (class in *klovve.pieces.tabbed*), 90  
 Model (class in *klovve.pieces.tabbed.tab*), 90  
 Model (class in *klovve.pieces.text\_field*), 103  
 Model (class in *klovve.pieces.vertical\_box*), 103  
 Model (class in *klovve.pieces.viewer.image*), 91  
 Model (class in *klovve.pieces.viewer.pdf*), 91  
 Model (class in *klovve.pieces.window*), 104  
 model (*klovve.view.BaseView* property), 105  
 model (*klovve.view.tree.ViewTreeNode* property), 107  
 Model.Orientation (class in *klovve.pieces.busy\_animation*), 92  
 Model.Size (class in *klovve.pieces.busy\_animation*), 92  
 Model.State (class in *klovve.pieces.headered\_panel*), 96  
 model\_type() (*klovve.drivers.Driver* method), 26  
 module  
   klovve, 9  
   klovve.+tests, 9  
   klovve.app, 9  
   klovve.app.context, 11  
   klovve.data, 12  
   klovve.data.deps, 12  
   klovve.data.lists, 15  
   klovve.data.model, 18  
   klovve.data.observable\_value, 23  
   klovve.data.reaction, 24  
   klovve.data.value\_holder, 25  
   klovve.debug, 108  
   klovve.drivers, 26  
   klovve.drivers.bare\_terminal, 28  
   klovve.drivers.bare\_terminal.views, 28  
   klovve.drivers.bare\_terminal.views.checkable, 29  
   klovve.drivers.bare\_terminal.views.headered\_panel, 29  
   klovve.drivers.bare\_terminal.views.interact, 28  
   klovve.drivers.bare\_terminal.views.interact.abstract, 28  
   klovve.drivers.bare\_terminal.views.label, 30  
   klovve.drivers.bare\_terminal.views.log\_pager, 30  
   klovve.drivers.bare\_terminal.views.placeholder, 30  
   klovve.drivers.bare\_terminal.views.split, 31  
   klovve.drivers.bare\_terminal.views.vertical\_box, 31  
   klovve.drivers.bare\_terminal.views.viewer, 29  
   klovve.drivers.bare\_terminal.views.viewer.image, 29  
   klovve.drivers.bare\_terminal.views.window, 31  
   klovve.drivers.gtk, 32  
   klovve.drivers.gtk.views, 34  
   klovve.drivers.gtk.views.\_gtk\_private, 34  
   klovve.drivers.gtk.views.\_gtk\_private.box, 34  
   klovve.drivers.gtk.views.busy\_animation, 39  
   klovve.drivers.gtk.views.button, 40  
   klovve.drivers.gtk.views.checkable, 40  
   klovve.drivers.gtk.views.disableable, 41  
   klovve.drivers.gtk.views.dropdown, 41  
   klovve.drivers.gtk.views.dual, 41  
   klovve.drivers.gtk.views.form, 35

klovve.drivers.gtk.views.form.section, 35  
klovve.drivers.gtk.views.formatted\_text, 42  
klovve.drivers.gtk.views.header, 42  
klovve.drivers.gtk.views.headered\_panel, 42  
klovve.drivers.gtk.views.horizontal\_box, 43  
klovve.drivers.gtk.views.interact, 35  
klovve.drivers.gtk.views.interact.abstract, 35  
klovve.drivers.gtk.views.label, 43  
klovve.drivers.gtk.views.list, 43  
klovve.drivers.gtk.views.log\_pager, 44  
klovve.drivers.gtk.views.placeholder, 44  
klovve.drivers.gtk.views.progress, 44  
klovve.drivers.gtk.views.property\_panel, 36  
klovve.drivers.gtk.views.scrollable, 45  
klovve.drivers.gtk.views.slider, 45  
klovve.drivers.gtk.views.split, 45  
klovve.drivers.gtk.views.tabbed, 36  
klovve.drivers.gtk.views.tabbed.tab, 36  
klovve.drivers.gtk.views.text\_field, 46  
klovve.drivers.gtk.views.vertical\_box, 46  
klovve.drivers.gtk.views.viewer, 37  
klovve.drivers.gtk.views.viewer.image, 37  
klovve.drivers.gtk.views.viewer.pdf, 39  
klovve.drivers.gtk.views.window, 46  
klovve.drivers.viuid, 47  
klovve.drivers.viuid.views, 48  
klovve.drivers.viuid.views.\_curses\_private, 48  
klovve.drivers.viuid.views.\_curses\_private.box, 48  
klovve.drivers.viuid.views.busy\_animation, 51  
klovve.drivers.viuid.views.button, 51  
klovve.drivers.viuid.views.checkable, 52  
klovve.drivers.viuid.views.disableable, 52  
klovve.drivers.viuid.views.dropdown, 53  
klovve.drivers.viuid.views.dual, 53  
klovve.drivers.viuid.views.form, 48  
klovve.drivers.viuid.views.form.section, 49  
klovve.drivers.viuid.views.formatted\_text, 53  
klovve.drivers.viuid.views.header, 54  
klovve.drivers.viuid.views.headered\_panel, 54  
klovve.drivers.viuid.views.horizontal\_box, 54  
klovve.drivers.viuid.views.interact, 49  
klovve.drivers.viuid.views.interact.abstract, 49  
klovve.drivers.viuid.views.label, 55  
klovve.drivers.viuid.views.list, 55  
klovve.drivers.viuid.views.log\_pager, 55  
klovve.drivers.viuid.views.placeholder, 56  
klovve.drivers.viuid.views.progress, 56  
klovve.drivers.viuid.views.property\_panel, 49  
klovve.drivers.viuid.views.scrollable, 56  
klovve.drivers.viuid.views.slider, 57  
klovve.drivers.viuid.views.split, 57  
klovve.drivers.viuid.views.tabbed, 50  
klovve.drivers.viuid.views.tabbed.tab, 50  
klovve.drivers.viuid.views.text\_field, 57  
klovve.drivers.viuid.views.vertical\_box, 58  
klovve.drivers.viuid.views.viewer, 50  
klovve.drivers.viuid.views.viewer.image, 50  
klovve.drivers.viuid.views.viewer.pdf, 51  
klovve.drivers.viuid.views.window, 58  
klovve.drivers.viuid.viuid, 58  
klovve.drivers.viuid.viuid.+samples, 60  
klovve.drivers.viuid.viuid.canvas, 74  
klovve.drivers.viuid.viuid.data, 76  
klovve.drivers.viuid.viuid.drivers, 60  
klovve.drivers.viuid.viuid.drivers.curses, 61  
klovve.drivers.viuid.viuid.drivers.driver, 62  
klovve.drivers.viuid.viuid.event, 63  
klovve.drivers.viuid.viuid.event.base, 63  
klovve.drivers.viuid.viuid.event.input, 63  
klovve.drivers.viuid.viuid.layout, 77  
klovve.drivers.viuid.viuid.loop, 78  
klovve.drivers.viuid.viuid.screen, 78  
klovve.drivers.viuid.viuid.style, 82  
klovve.drivers.viuid.viuid.widgets, 64  
klovve.drivers.viuid.viuid.widgets.box, 64  
klovve.drivers.viuid.viuid.widgets.busy\_animation, 65  
klovve.drivers.viuid.viuid.widgets.button, 65  
klovve.drivers.viuid.viuid.widgets.check\_button, 65  
klovve.drivers.viuid.viuid.widgets.dropdown, 66  
klovve.drivers.viuid.viuid.widgets.entry, 66

klovve.drivers.viwid.viwid.widgets.frame, 66  
 klovve.drivers.viwid.viwid.widgets.label, 67  
 klovve.drivers.viwid.viwid.widgets.list, 67  
 klovve.drivers.viwid.viwid.widgets.progress\_bar, 68  
 klovve.drivers.viwid.viwid.widgets.radio\_button, 68  
 klovve.drivers.viwid.viwid.widgets.scrollable, 68  
 klovve.drivers.viwid.viwid.widgets.widget, 69  
 klovve.pieces, 83  
 klovve.pieces.busy\_animation, 92  
 klovve.pieces.button, 93  
 klovve.pieces.checkable, 93  
 klovve.pieces.disableable, 94  
 klovve.pieces.dropdown, 94  
 klovve.pieces.dual, 95  
 klovve.pieces.form, 83  
 klovve.pieces.form.section, 83  
 klovve.pieces.formatted\_text, 95  
 klovve.pieces.header, 96  
 klovve.pieces.headered\_panel, 96  
 klovve.pieces.horizontal\_box, 97  
 klovve.pieces.interact, 84  
 klovve.pieces.interact.abstract, 84  
 klovve.pieces.interact.message, 85  
 klovve.pieces.interact.textline, 85  
 klovve.pieces.interact.triggers, 86  
 klovve.pieces.interact.yesno, 87  
 klovve.pieces.label, 98  
 klovve.pieces.list, 98  
 klovve.pieces.list\_panel, 99  
 klovve.pieces.log\_pager, 87  
 klovve.pieces.log\_pager.entry, 88  
 klovve.pieces.placeholder, 100  
 klovve.pieces.progress, 101  
 klovve.pieces.property\_panel, 89  
 klovve.pieces.property\_panel.entry, 89  
 klovve.pieces.scrollable, 101  
 klovve.pieces.slider, 101  
 klovve.pieces.split, 102  
 klovve.pieces.tabbed, 90  
 klovve.pieces.tabbed.tab, 90  
 klovve.pieces.text\_field, 103  
 klovve.pieces.vertical\_box, 103  
 klovve.pieces.viewer, 91  
 klovve.pieces.viewer.image, 91  
 klovve.pieces.viewer.pdf, 91  
 klovve.pieces.window, 104  
 klovve.view, 105  
 klovve.view.tree, 106  
 move() (klovve.data.lists.List method), 16  
 moved\_by() (klovve.drivers.viwid.viwid.Point method), 58  
**N**  
 name() (klovve.drivers.viwid.viwid.event.base.OngoingEvent property), 63  
 name() (klovve.pieces.property\_panel.entry.Model property), 89  
 name() (klovve.pieces.window.Model property), 104  
 name() (klovve.drivers.bare\_terminal.Driver static method), 28  
 name() (klovve.drivers.Driver static method), 26  
 name() (klovve.drivers.gtk.Driver static method), 32  
 name() (klovve.drivers.viwid.Driver static method), 47  
 name\_tuple (klovve.view.tree.ViewTreeNode property), 107  
 native() (klovve.view.BaseView method), 105  
 native() (klovve.view.View method), 106  
 new\_object\_created() (klovve.debug.\_Memory method), 108  
 no\_dependency\_tracking() (in module klovve.data.deps), 15  
 NullLayout (class in klovve.drivers.viwid.viwid.layout), 78  
**O**  
 ObservableValueProtocol (class in klovve.data.observable\_value), 23  
 observe\_list() (in module klovve.data.model), 22  
 offset (klovve.drivers.viwid.viwid.widgets.widget.Widget property), 72  
 on\_click (klovve.drivers.viwid.viwid.widgets.button.Button property), 65  
 on\_dematerialized() (klovve.drivers.viwid.viwid.widgets.widget.Widget method), 72  
 on\_materialized() (klovve.drivers.viwid.viwid.widgets.widget.Widget method), 72  
 on\_selection\_activated (klovve.drivers.viwid.viwid.widgets.list.List property), 67  
 OneWayBinding (class in klovve.data.value\_holder), 25  
 OngoingEvent (class in klovve.drivers.viwid.viwid.event.base), 63  
 OngoingKeyboardEvent (class in klovve.drivers.viwid.viwid.event.input), 63  
 OngoingMouseClickedEvent (class in klovve.drivers.viwid.viwid.event.input), 63  
 OngoingScrollEvent (class in klovve.drivers.viwid.viwid.event.input), 64  
 only\_single\_time (klovve.pieces.log\_pager.entry.Model property), 88



- `only_verbose` (*klovve.pieces.log\_pager.entry.Model* property), 88
- `Orientation` (class in *klovve.drivers.viwid.viwid.widgets.box*), 64
- `orientation` (*klovve.drivers.viwid.viwid.widgets.box.Box* property), 64
- `orientation` (*klovve.pieces.busy\_animation.Model* property), 92
- `original_image_height` (*klovve.drivers.gtk.views.viewer.image.Image* property), 37
- `original_image_width` (*klovve.drivers.gtk.views.viewer.image.Image* property), 37
- `Outer` (class in *klovve.drivers.gtk.views.viewer.image*), 38
- ## P
- `paint()` (*klovve.drivers.viwid.viwid.widgets.check\_button.CheckButton* method), 65
- `paint()` (*klovve.drivers.viwid.viwid.widgets.entry.Entry* method), 66
- `paint()` (*klovve.drivers.viwid.viwid.widgets.frame.Frame* method), 66
- `paint()` (*klovve.drivers.viwid.viwid.widgets.label.Label* method), 67
- `paint()` (*klovve.drivers.viwid.viwid.widgets.progress\_bar.ProgressBar* method), 61
- `paint()` (*klovve.drivers.viwid.viwid.widgets.widget.Widget* method), 72
- `paint2()` (*klovve.drivers.viwid.viwid.widgets.widget.Widget* method), 72
- `parent` (*klovve.drivers.viwid.viwid.widgets.widget.Widget* property), 72
- `path` (*klovve.pieces.viewer.pdf.Model* property), 91
- `pause_refreshing()` (in module *klovve.data.deps*), 15
- `pixbuf` (*klovve.drivers.gtk.views.viewer.image.Image* attribute), 37
- `pixbuf` (*klovve.drivers.gtk.views.viewer.image.Outer* attribute), 38
- `Point` (class in *klovve.drivers.viwid.viwid*), 58
- `popup()` (*klovve.drivers.viwid.viwid.screen.Screen* method), 81
- `position` (*klovve.drivers.viwid.viwid.widgets.widget.Widget* property), 72
- `prepare()` (*klovve.drivers.gtk.\_Source* method), 34
- `process_curses_input()` (*klovve.drivers.viwid.viwid.drivers.curses.Driver* method), 61
- `ProgramError`, 10
- `progress` (*klovve.pieces.headered\_panel.Model* property), 97
- `ProgressBar` (class in *klovve.drivers.viwid.viwid.widgets.progress\_bar*), 68
- `Property` (class in *klovve.data.model*), 20
- `Property()` (in module *klovve.drivers.viwid.viwid.widgets.widget*), 69
- `Property()` (*klovve.drivers.viwid.viwid.widgets.widget.Widget* method), 69
- `PropertyBase` (class in *klovve.data.deps*), 14
- `PropertyBase` (class in *klovve.data.model*), 21
- ## R
- `r` (*klovve.drivers.viwid.viwid.canvas.Canvas.Color* attribute), 74
- `r` (*klovve.drivers.viwid.viwid.canvas.Color* attribute), 75
- `RadioButton` (class in *klovve.drivers.viwid.viwid.widgets.radio\_button*), 68
- `rank()` (*klovve.drivers.bare\_terminal.Driver* static method), 28
- `rank()` (*klovve.drivers.Driver* static method), 27
- `rank()` (*klovve.drivers.gtk.Driver* static method), 32
- `rank()` (*klovve.drivers.viwid.Driver* static method), 47
- `reaction()` (in module *klovve.data.reaction*), 24
- `Rectangle` (class in *klovve.drivers.viwid.viwid*), 59
- `refresh_curses_window()` (*klovve.drivers.viwid.viwid.drivers.curses.Driver* method), 61
- `register()` (*klovve.drivers.gtk.\_Selector* method), 33
- `remove_changed_handler()` (*klovve.data.observable\_value.CommonObservableValue* method), 23
- `remove_changed_handler()` (*klovve.data.observable\_value.ObservableValueProtocol* method), 24
- `remove_fd()` (*klovve.drivers.gtk.\_Source* method), 34
- `repaint_widget_soon()` (*klovve.drivers.viwid.viwid.drivers.curses.Driver* method), 61
- `repaint_widget_soon()` (*klovve.drivers.viwid.viwid.drivers.driver.Driver* method), 63
- `request_repaint()` (*klovve.drivers.viwid.viwid.widgets.widget.Widget* method), 72
- `request_resize()` (*klovve.drivers.viwid.viwid.widgets.widget.Widget* method), 72
- `resize_widget_soon()` (*klovve.drivers.viwid.viwid.drivers.curses.Driver* method), 61
- `resize_widget_soon()` (*klovve.drivers.viwid.viwid.drivers.driver.Driver* method), 63
- `rich` (*klovve.drivers.viwid.viwid.canvas.Canvas.FullColor* attribute), 74
- `rich` (*klovve.drivers.viwid.viwid.canvas.FullColor* attribute), 75

`right_button` (*klovve.drivers.viwid.viwid.event.input.OngoingMouseEvent* property), 64  
`right_x` (*klovve.drivers.viwid.viwid.Rectangle* property), 59  
`root_style` (*klovve.drivers.viwid.viwid.widgets.widget.Widget* property), 72  
`RootAlignment` (class in *klovve.drivers.viwid.viwid.screen*), 79  
`RootAlignmentPositioning` (class in *klovve.drivers.viwid.viwid.screen*), 79  
`RootAlignmentSizing` (class in *klovve.drivers.viwid.viwid.screen*), 80  
`roots` (*klovve.drivers.viwid.viwid.screen.Screen* property), 81  
`roots_and_alignments` (*klovve.drivers.viwid.viwid.screen.Screen* property), 81  
`RootStyle` (class in *klovve.drivers.viwid.viwid.style*), 82  
`run()` (*klovve.app.Application* method), 10  
`run()` (*klovve.drivers.viwid.viwid.drivers.curses.\_Loop* method), 62  
`run()` (*klovve.drivers.viwid.viwid.loop.Loop* method), 78  
**S**  
`Screen` (class in *klovve.drivers.viwid.viwid.screen*), 80  
`screen` (*klovve.drivers.viwid.viwid.widgets.widget.Widget* property), 72  
`screen_position` (*klovve.drivers.viwid.viwid.event.input.OngoingMouseEvent* property), 64  
`screen_position` (*klovve.drivers.viwid.viwid.event.input.OngoingMouseEvent* property), 64  
`screens` (*klovve.drivers.viwid.viwid.drivers.curses.Driver* property), 61  
`screens` (*klovve.drivers.viwid.viwid.drivers.driver.Driver* property), 63  
`Scrollable` (class in *klovve.drivers.viwid.viwid.widgets.scrollable*), 68  
`select()` (*klovve.drivers.gtk.\_Selector* method), 33  
`selected_index` (*klovve.drivers.viwid.viwid.widgets.dropdown.Dropdown* property), 66  
`selected_item` (*klovve.drivers.viwid.viwid.widgets.dropdown.Dropdown* property), 66  
`selected_item` (*klovve.pieces.dropdown.Model* property), 94  
`selected_item` (*klovve.pieces.list.Model* property), 99  
`selected_item` (*klovve.pieces.list\_panel.Model* property), 100  
`selected_item_index` (*klovve.drivers.viwid.viwid.widgets.list.List* property), 67  
`selected_items` (*klovve.pieces.list.Model* property), 99  
`set_answer()` (*klovve.pieces.interact.Model* method), 84  
`setSize()` (*klovve.drivers.viwid.viwid.widgets.widget.Widget* method), 72  
`setLayout()` (*klovve.drivers.viwid.viwid.widgets.widget.Widget* method), 72  
`set_offset()` (*klovve.drivers.viwid.viwid.widgets.widget.Widget* method), 72  
`set_parent()` (*klovve.drivers.viwid.viwid.widgets.widget.Widget* method), 73  
`set_position()` (*klovve.drivers.viwid.viwid.widgets.widget.Widget* method), 73  
`set_root_style()` (*klovve.drivers.viwid.viwid.widgets.widget.Widget* method), 73  
`set_size()` (*klovve.drivers.viwid.viwid.layout.GridLayout* method), 77  
`set_size()` (*klovve.drivers.viwid.viwid.layout.Layout* method), 78  
`set_size()` (*klovve.drivers.viwid.viwid.layout.NullLayout* method), 78  
`set_size()` (*klovve.drivers.viwid.viwid.widgets.frame.Frame* method), 67  
`set_size()` (*klovve.drivers.viwid.viwid.widgets.scrollable.Scrollable* method), 68  
`set_size()` (*klovve.drivers.viwid.viwid.widgets.widget.Widget* method), 73  
`set_source()` (*klovve.drivers.gtk.views.viewer.image.Outer* method), 38  
`set_value()` (*klovve.data.value\_holder.ComputedValue* method), 25  
`set_value()` (*klovve.data.value\_holder.OneWayBinding* method), 25  
`set_value()` (*klovve.data.value\_holder.ValueHolder* method), 26  
`show_window()` (*klovve.drivers.bare\_terminal.Driver* method), 28  
`show_window()` (*klovve.drivers.Driver* method), 27  
`show_window()` (*klovve.drivers.gtk.Driver* method), 32  
`show_window()` (*klovve.drivers.viwid.Driver* method), 47  
`sidebar` (*klovve.pieces.dual.Model* property), 95  
`since_startup` (*klovve.debug.\_Memory* property), 108  
`Size` (class in *klovve.drivers.viwid.viwid*), 59  
`size` (*klovve.drivers.viwid.viwid.drivers.curses.\_Screen* property), 62  
`size` (*klovve.drivers.viwid.viwid.screen.Screen* property), 81  
`size` (*klovve.drivers.viwid.viwid.widgets.widget.Widget* property), 73  
`size` (*klovve.pieces.busy\_animation.Model* property), 92  
`sleep_until_stopped()` (*klovve.app.Application* method), 10  
`SMALL` (*klovve.pieces.busy\_animation.Model.Size* attribute), 92  
`source` (*klovve.drivers.gtk.views.viewer.image.Outer* attribute), 38

- source (*klovve.pieces.viewer.image.Model* property), 91  
 spinner\_size (*klovve.drivers.gtk.views.busy\_animation.View* property), 39  
 spinner\_size() (*klovve.drivers.viwid.viwid.views.busy\_animation.Model* method), 51  
 START (*klovve.drivers.viwid.viwid.widgets.widget.Alignment* attribute), 69  
 start() (*klovve.app.Application* method), 10  
 state (*klovve.pieces.headered\_panel.Model* property), 97  
 step\_value (*klovve.pieces.slider.Model* property), 102  
 STOP (*klovve.drivers.viwid.viwid.widgets.widget.Alignment* attribute), 69  
 stop\_propagation() (*klovve.drivers.viwid.viwid.event.base.Event* method), 63  
 Style (class in *klovve.drivers.viwid.viwid.style*), 82  
 style (*klovve.drivers.viwid.viwid.screen.Screen* property), 81  
 style\_atom (*klovve.drivers.viwid.viwid.widgets.widget.Widget* property), 73  
 style\_atom\_by\_name() (*klovve.drivers.viwid.viwid.widgets.widget.Widget* method), 73  
 StyleAtom (class in *klovve.drivers.viwid.viwid.style*), 82  
 SUCCESSFUL (*klovve.pieces.headered\_panel.Model.State* attribute), 96  
 SUCCESSFUL\_WITH\_WARNING (*klovve.pieces.headered\_panel.Model.State* attribute), 96  
 suggestion (*klovve.pieces.interact.textline.Model* property), 85
- ## T
- text (*klovve.drivers.viwid.viwid.widgets.entry.Entry* property), 66  
 text (*klovve.drivers.viwid.viwid.widgets.label.Label* property), 67  
 text (*klovve.pieces.busy\_animation.Model* property), 92  
 text (*klovve.pieces.button.Model* property), 93  
 text (*klovve.pieces.checkable.Model* property), 93  
 text (*klovve.pieces.formatted\_text.Model* property), 95  
 text (*klovve.pieces.header.Model* property), 96  
 text (*klovve.pieces.label.Model* property), 98  
 text (*klovve.pieces.text\_field.Model* property), 103  
 text() (*klovve.drivers.viwid.viwid.canvas.Canvas* method), 75  
 text() (*klovve.drivers.viwid.viwid.drivers.curses.\_Canvas* method), 61  
 TINY (*klovve.pieces.busy\_animation.Model.Size* attribute), 92  
 title (*klovve.pieces.headered\_panel.Model* property), 97  
 title (*klovve.pieces.window.Model* property), 104  
 title2 (*klovve.pieces.window.Model* property), 104
- title\_label\_background (*klovve.drivers.viwid.viwid.views.headered\_panel.Model* property), 54  
 title\_label\_css\_classes (*klovve.drivers.gtk.views.headered\_panel.View* property), 42  
 title\_secondary\_items (*klovve.pieces.headered\_panel.Model* property), 97  
 top\_left (*klovve.drivers.viwid.viwid.Rectangle* property), 59  
 top\_y (*klovve.drivers.viwid.viwid.Rectangle* property), 59  
 top\_right (*klovve.drivers.viwid.viwid.screen.Screen* property), 81  
 translate\_coordinates() (*klovve.drivers.viwid.viwid.screen.Screen* method), 81  
 translate\_coordinates\_with\_visibility\_check() (*klovve.drivers.viwid.viwid.screen.Screen* method), 81  
 triggers (*klovve.pieces.interact.abstract.Model* property), 85  
 triggers (*klovve.pieces.interact.triggers.Model* property), 86  
 try\_focus() (*klovve.drivers.viwid.viwid.widgets.widget.Widget* method), 73  
 TwoWayBinding (class in *klovve.data.value\_holder*), 25
- ## U
- unregister() (*klovve.drivers.gtk.\_Selector* method), 33  
 update() (*klovve.data.lists.List* method), 16  
 update() (*klovve.drivers.viwid.viwid.data.List* method), 76  
 update() (*klovve.drivers.viwid.viwid.screen.Screen* method), 82  
 update\_curses\_screensize() (*klovve.drivers.viwid.viwid.drivers.curses.\_Screen* method), 62
- ## V
- value (*klovve.data.reaction.\_ReactionModel* property), 24  
 value (*klovve.drivers.viwid.viwid.widgets.progress\_bar.ProgressBar* property), 68  
 value (*klovve.pieces.interact.textline.View.TModel* property), 86  
 value (*klovve.pieces.progress.Model* property), 101  
 value (*klovve.pieces.slider.Model* property), 102  
 ValueHolder (class in *klovve.data.value\_holder*), 25  
 values (*klovve.pieces.property\_panel.Model* property), 89  
 verify\_correct\_thread() (in module *klovve.app*), 11



VERTICAL (*klovve.drivers.viwid.viwid.widgets.box.Orientation* attribute), 65  
 VERTICAL (*klovve.pieces.busy\_animation.Model.Orientation* attribute), 92  
 vertical\_alignment (*klovve.drivers.viwid.viwid.widgets.widget.Widget* property), 73  
 vertical\_expand\_greedily (*klovve.drivers.viwid.viwid.widgets.widget.Widget* property), 73  
 vertically\_scrollable (*klovve.drivers.viwid.viwid.widgets.scrollable.Scrollable* property), 68  
 View (class in *klovve.drivers.bare\_terminal.views.checkable\_view*), 29  
 View (class in *klovve.drivers.bare\_terminal.views.headered\_panel*), 29  
 View (class in *klovve.drivers.bare\_terminal.views.interact.abstract*), 28  
 View (class in *klovve.drivers.bare\_terminal.views.label*), 30  
 View (class in *klovve.drivers.bare\_terminal.views.log\_pager*), 30  
 View (class in *klovve.drivers.bare\_terminal.views.placeholder*), 30  
 View (class in *klovve.drivers.bare\_terminal.views.split*), 31  
 View (class in *klovve.drivers.bare\_terminal.views.vertical\_box*), 31  
 View (class in *klovve.drivers.bare\_terminal.views.viewer.image*), 29  
 View (class in *klovve.drivers.bare\_terminal.views.window*), 31  
 View (class in *klovve.drivers.gtk*), 32  
 View (class in *klovve.drivers.gtk.views.\_gtk\_private.box*), 34  
 View (class in *klovve.drivers.gtk.views.busy\_animation*), 39  
 View (class in *klovve.drivers.gtk.views.button*), 40  
 View (class in *klovve.drivers.gtk.views.checkable*), 40  
 View (class in *klovve.drivers.gtk.views.disableable*), 41  
 View (class in *klovve.drivers.gtk.views.dropdown*), 41  
 View (class in *klovve.drivers.gtk.views.dual*), 41  
 View (class in *klovve.drivers.gtk.views.form*), 35  
 View (class in *klovve.drivers.gtk.views.form.section*), 35  
 View (class in *klovve.drivers.gtk.views.formatted\_text*), 42  
 View (class in *klovve.drivers.gtk.views.header*), 42  
 View (class in *klovve.drivers.gtk.views.headered\_panel*), 42  
 View (class in *klovve.drivers.gtk.views.horizontal\_box*), 43  
 View (class in *klovve.drivers.gtk.views.interact.abstract*), 35  
 View (class in *klovve.drivers.gtk.views.label*), 43  
 View (class in *klovve.drivers.gtk.views.list*), 43  
 View (class in *klovve.drivers.gtk.views.log\_pager*), 44  
 View (class in *klovve.drivers.gtk.views.placeholder*), 44  
 View (class in *klovve.drivers.gtk.views.progress*), 44  
 View (class in *klovve.drivers.gtk.views.property\_panel*), 36  
 View (class in *klovve.drivers.gtk.views.scrollable*), 45  
 View (class in *klovve.drivers.gtk.views.slider*), 45  
 View (class in *klovve.drivers.gtk.views.split*), 45  
 View (class in *klovve.drivers.gtk.views.tabbed*), 36  
 View (class in *klovve.drivers.gtk.views.tabbed.tab*), 36  
 View (class in *klovve.drivers.gtk.views.text\_field*), 46  
 View (class in *klovve.drivers.gtk.views.vertical\_box*), 46  
 View (class in *klovve.drivers.gtk.views.viewer.image*), 38  
 View (class in *klovve.drivers.gtk.views.viewer.pdf*), 39  
 View (class in *klovve.drivers.gtk.views.window*), 46  
 View (class in *klovve.drivers.viwid*), 47  
 View (class in *klovve.drivers.viwid.views.\_curses\_private.box*), 48  
 View (class in *klovve.drivers.viwid.views.busy\_animation*), 51  
 View (class in *klovve.drivers.viwid.views.button*), 52  
 View (class in *klovve.drivers.viwid.views.checkable*), 52  
 View (class in *klovve.drivers.viwid.views.disableable*), 52  
 View (class in *klovve.drivers.viwid.views.dropdown*), 53  
 View (class in *klovve.drivers.viwid.views.dual*), 53  
 View (class in *klovve.drivers.viwid.views.form*), 48  
 View (class in *klovve.drivers.viwid.views.form.section*), 49  
 View (class in *klovve.drivers.viwid.views.formatted\_text*), 53  
 View (class in *klovve.drivers.viwid.views.header*), 54  
 View (class in *klovve.drivers.viwid.views.headered\_panel*), 54  
 View (class in *klovve.drivers.viwid.views.horizontal\_box*), 54  
 View (class in *klovve.drivers.viwid.views.interact.abstract*), 49  
 View (class in *klovve.drivers.viwid.views.label*), 55  
 View (class in *klovve.drivers.viwid.views.list*), 55  
 View (class in *klovve.drivers.viwid.views.log\_pager*), 55  
 View (class in *klovve.drivers.viwid.views.placeholder*), 56  
 View (class in *klovve.drivers.viwid.views.progress*), 56  
 View (class in *klovve.drivers.viwid.views.property\_panel*), 49  
 View (class in *klovve.drivers.viwid.views.scrollable*), 56  
 View (class in *klovve.drivers.viwid.views.slider*), 57  
 View (class in *klovve.drivers.viwid.views.split*), 57  
 View (class in *klovve.drivers.viwid.views.tabbed*), 50  
 View (class in *klovve.drivers.viwid.views.tabbed.tab*), 50  
 View (class in *klovve.drivers.viwid.views.text\_field*), 57  
 View (class in *klovve.drivers.viwid.views.vertical\_box*), 58

`View` (class in `klovve.drivers.viwid.views.viewer.image`), 50

`View` (class in `klovve.drivers.viwid.views.viewer.pdf`), 51

`View` (class in `klovve.drivers.viwid.views.window`), 58

`View` (class in `klovve.pieces.busy_animation`), 92

`View` (class in `klovve.pieces.button`), 93

`View` (class in `klovve.pieces.checkable`), 93

`View` (class in `klovve.pieces.disableable`), 94

`View` (class in `klovve.pieces.dropdown`), 95

`View` (class in `klovve.pieces.dual`), 95

`View` (class in `klovve.pieces.form`), 83

`View` (class in `klovve.pieces.form.section`), 83

`View` (class in `klovve.pieces.formatted_text`), 95

`View` (class in `klovve.pieces.header`), 96

`View` (class in `klovve.pieces.headered_panel`), 97

`View` (class in `klovve.pieces.horizontal_box`), 97

`View` (class in `klovve.pieces.interact.abstract`), 85

`View` (class in `klovve.pieces.interact.message`), 85

`View` (class in `klovve.pieces.interact.textline`), 86

`View` (class in `klovve.pieces.interact.triggers`), 86

`View` (class in `klovve.pieces.interact.yesno`), 87

`View` (class in `klovve.pieces.label`), 98

`View` (class in `klovve.pieces.list`), 99

`View` (class in `klovve.pieces.list_panel`), 100

`View` (class in `klovve.pieces.log_pager`), 87

`View` (class in `klovve.pieces.log_pager.entry`), 88

`View` (class in `klovve.pieces.placeholder`), 100

`View` (class in `klovve.pieces.progress`), 101

`View` (class in `klovve.pieces.property_panel`), 89

`View` (class in `klovve.pieces.property_panel.entry`), 90

`View` (class in `klovve.pieces.scrollable`), 101

`View` (class in `klovve.pieces.slider`), 102

`View` (class in `klovve.pieces.split`), 102

`View` (class in `klovve.pieces.tabbed`), 90

`View` (class in `klovve.pieces.tabbed.tab`), 91

`View` (class in `klovve.pieces.text_field`), 103

`View` (class in `klovve.pieces.vertical_box`), 103

`View` (class in `klovve.pieces.viewer.image`), 91

`View` (class in `klovve.pieces.viewer.pdf`), 91

`View` (class in `klovve.pieces.window`), 104

`View` (class in `klovve.view`), 106

`view()` (`klovve.view.tree.ViewTreeNode` method), 107

`View.TModel` (class in `klovve.pieces.interact.textline`), 86

`view_type()` (`klovve.drivers.Driver` method), 27

`ViewError`, 106

`ViewFactory` (class in `klovve.view.tree`), 106

`ViewFactory._Node` (class in `klovve.view.tree`), 106

`ViewTreeNode` (class in `klovve.view.tree`), 106

## W

`Widget` (class in `klovve.drivers.viwid.viwid.widgets.widget`), 69

`Widget.Parti` (class in `klovve.drivers.viwid.viwid.widgets.widget`),

69

`width` (`klovve.drivers.viwid.viwid.Rectangle` property), 59

`width` (`klovve.drivers.viwid.viwid.Size` property), 60

`with_height()` (`klovve.drivers.viwid.viwid.Size` method), 60

`with_width()` (`klovve.drivers.viwid.viwid.Size` method), 60

`with_x()` (`klovve.drivers.viwid.viwid.Point` method), 59

`with_y()` (`klovve.drivers.viwid.viwid.Point` method), 59

## X

`x` (`klovve.drivers.viwid.viwid.Point` property), 59

## Y

`y` (`klovve.drivers.viwid.viwid.Point` property), 59

## Z

`zoom_factor` (`klovve.drivers.gtk.views.viewer.image.Image` attribute), 37

`zoom_in()` (`klovve.drivers.gtk.views.viewer.image.Image` method), 37

`zoom_out()` (`klovve.drivers.gtk.views.viewer.image.Image` method), 38