

Chapter 1

Automatic execution of code blocks

Compile this document with

```
doconce format ipynb execute.do.txt --execute
```

```
File "<ipython-input-2-909ae4915a71>", line 1
doconce format ipynb execute.do.txt --execute
~
```

SyntaxError: invalid syntax

This is a normal python block using the pycod environment

```
print('pycod')
```

pycod

Hidden execution cells (**pyhid**, **pycod-e**) can be used to perform operations (e.g. imports, variable initializations) without showing any cell. The **pyhid** environment executes and hides the cell in formats other than .ipynb:

The **pycod-e** environment executes but hides the cell also in .ipynb files:

pycod is a normal cell that should execute automatically when using **-execute**. Note that this cells relies on code executed in a previous hidden cell:

```
print(sys.version)
b = 2
c = a + b
print("The result is {}".format(c))
c
```

```
-----
NameError                                Traceback (most recent call last)
<ipython-input-5-d048aac4e047> in <module>
----> 1 print(sys.version)
      2 b = 2
      3 c = a + b
      4 print("The result is {}".format(c))
      5 c
```

`NameError: name 'sys' is not defined\epycout`

The `\texttt{*~t}` environment (e.g. `\texttt{pycod-t}`) formats a cell to text, and can be used to print `\begin{Verbatim}[numbers=none,fontsize=\fontsize{9pt}{9pt},baselinestretch=0.95]`

```
# This is a for-loop example
for i in [0,10]:
    print(i)

0
10
```

The `*out` (e.g. `pycod-out`) environment can be used to write a cell output:

```
# This is a normal pycod cell
1/0
```

```
-----
ZeroDivisionError                                Traceback (most recent call last)
<ipython-input-7-82521ecac92e> in <module>
      1 # This is a normal pycod cell
----> 2 1/0
```

`ZeroDivisionError: division by zero\epycout`

```
\begin{Verbatim}[numbers=none,fontsize=\fontsize{9pt}{9pt},baselinestretch=0.95]
1/0: You cannot divide by zero
```

The `-h` postfix can be used in the html format to show a Show/Hide button that toggles the code visibility.

```
print('show/hide')

show/hide
```

1.1 Plotting

This is a cell that should plot and output:

```
from pylab import *
x = linspace(0, 10, 100)
plot(x, x*x)
show()
```