Machine Learning: Optional: IPython

CPSC 501: Advanced Programming Techniques
Winter 2025

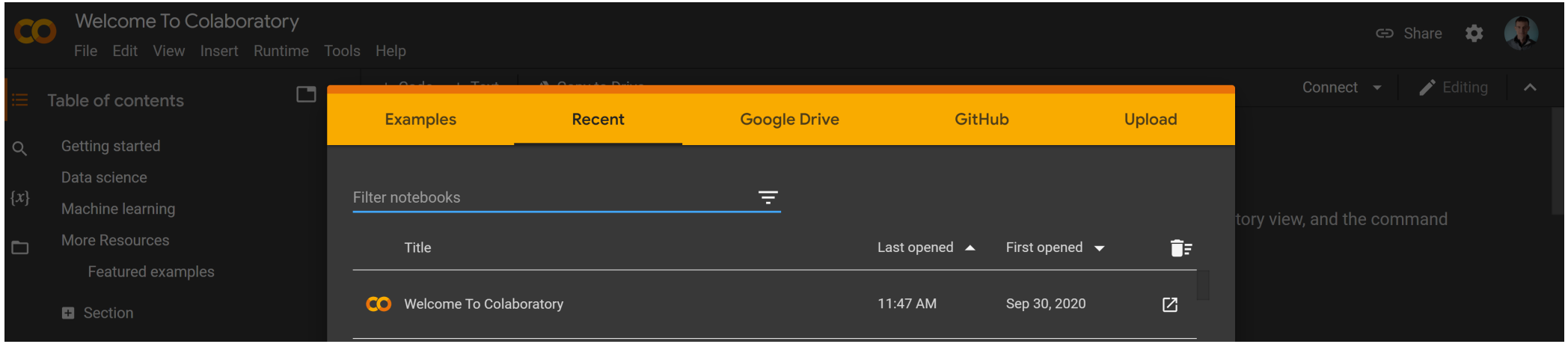
Jonathan Hudson, Ph.D
Assistant Professor (Teaching)
Department of Computer Science
University of Calgary

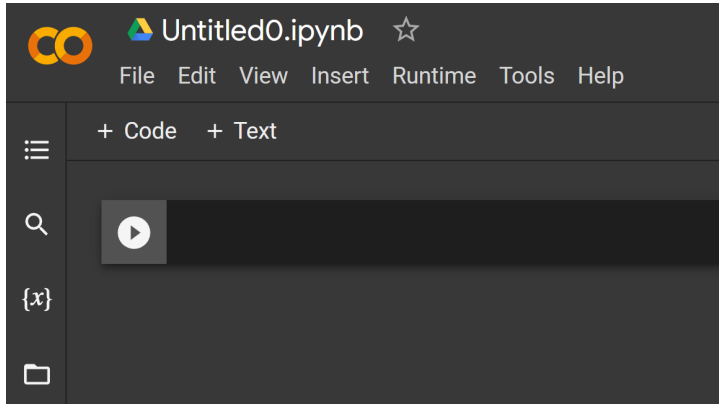
Thursday, February 13, 2025

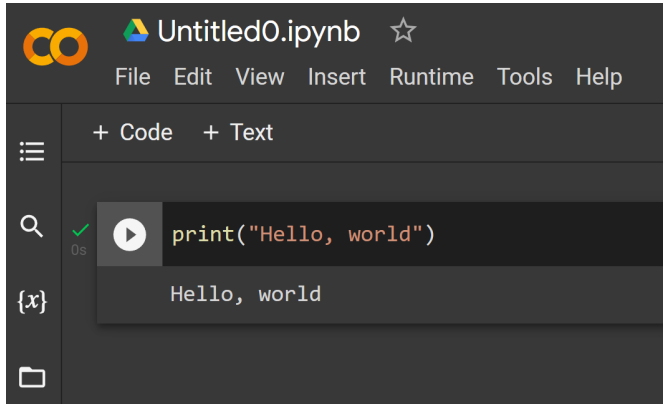
UNIVERSITY OF CALGARY

Beginner?

Just use **Google Colaboratory**! (web-based version of **Python Juypter** notebook)



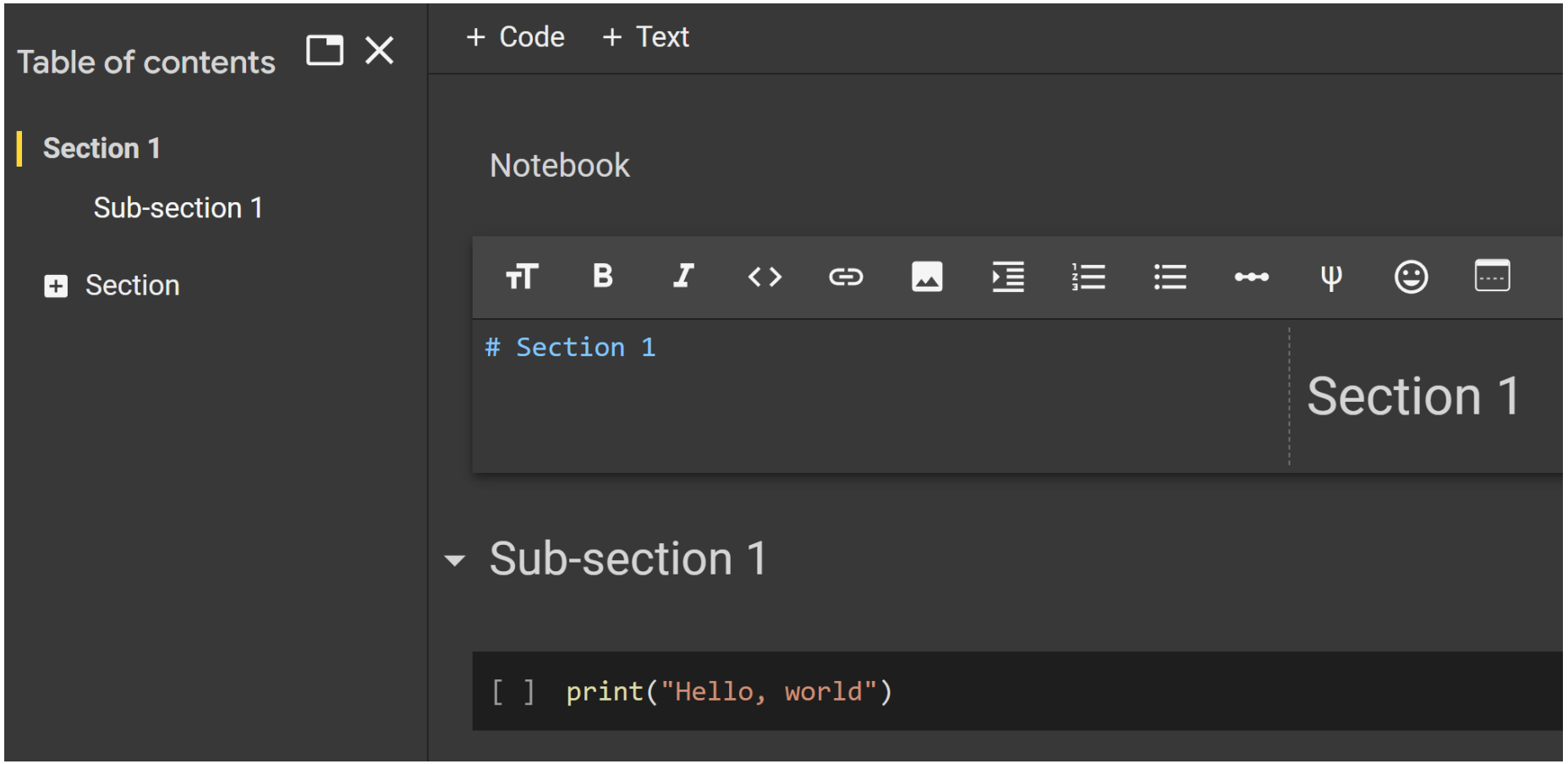




UNIVERSITY OF CALGARY

IPython (markdown language)

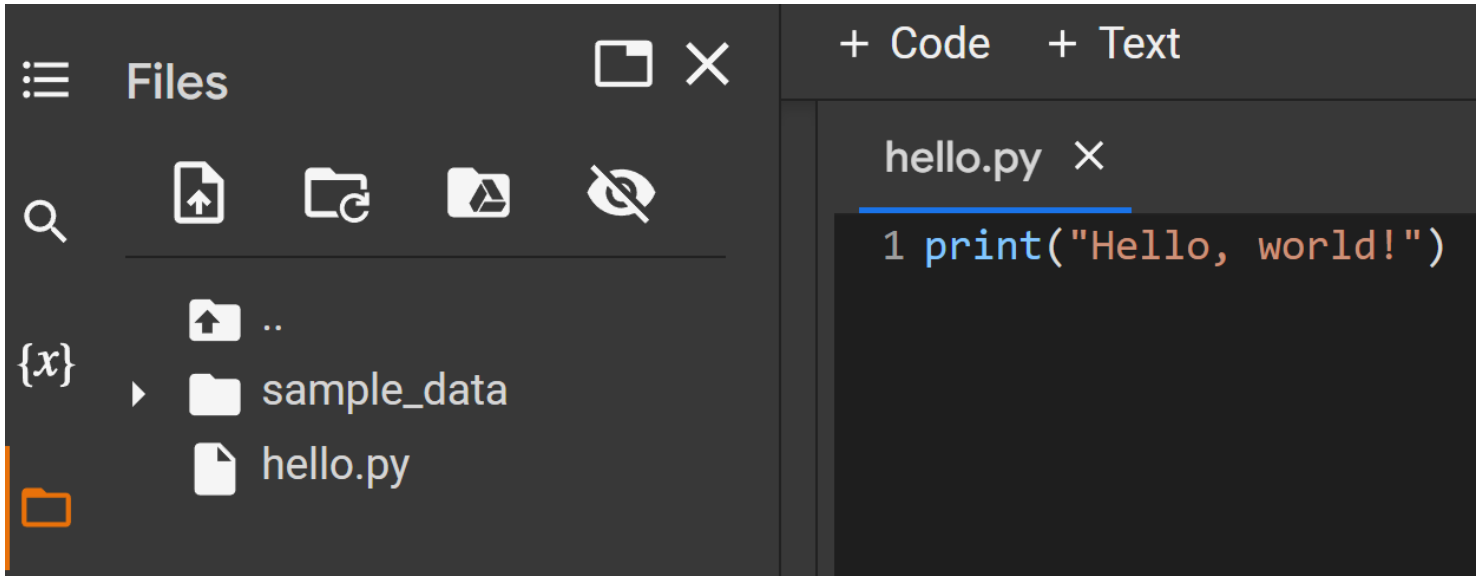
https://www.markdownguide.org/cheat-sheet/



UNIVERSITY OF CALGARY

IPython (files)

- Supports additional file creation and management (also Google Drive linking)

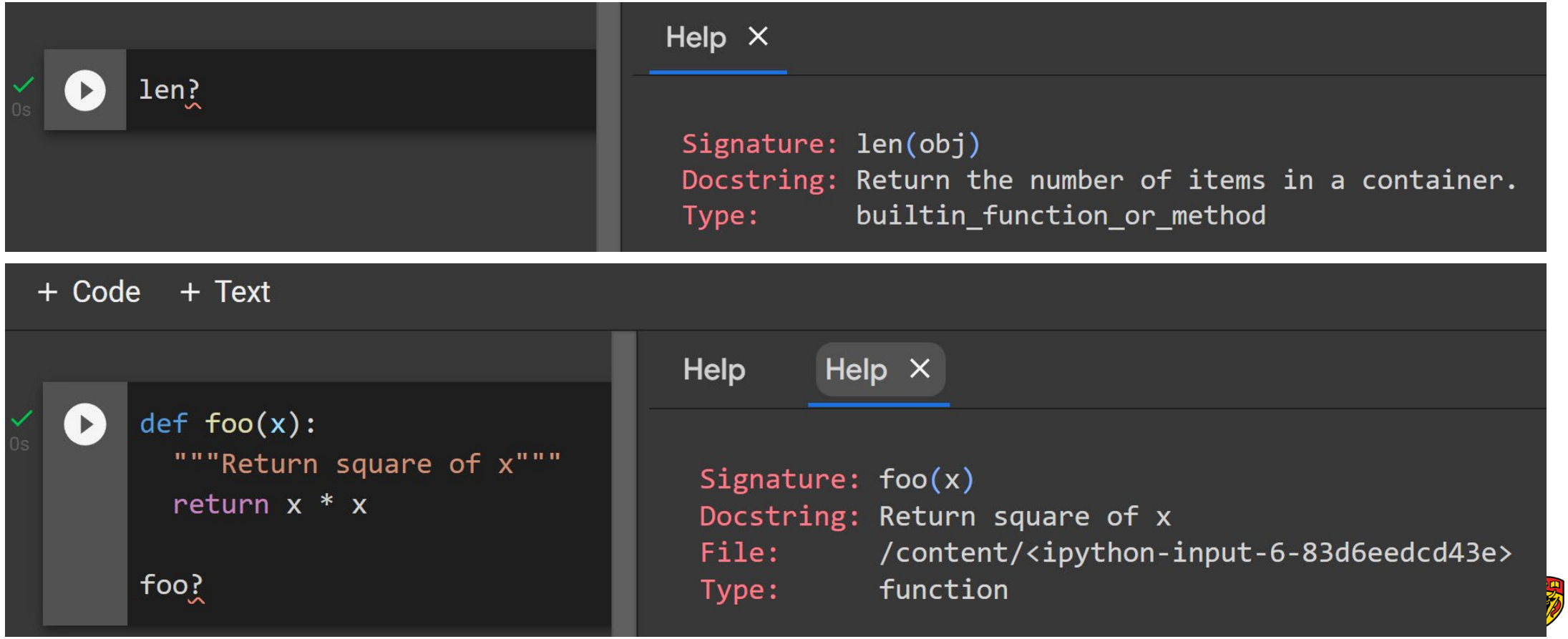


- By default saved in **Google Drive**, can also save notebooks to **Github** or export

IPython (getting information)

help(<item>)

<item>?

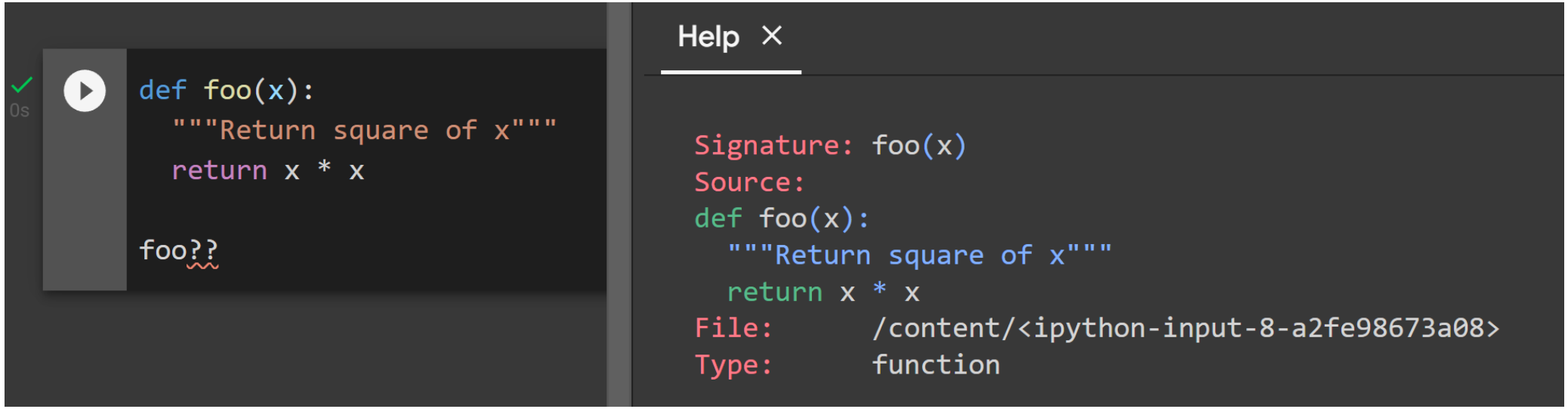


UNIVERSITY OF CALGARY

IPython (getting source code)

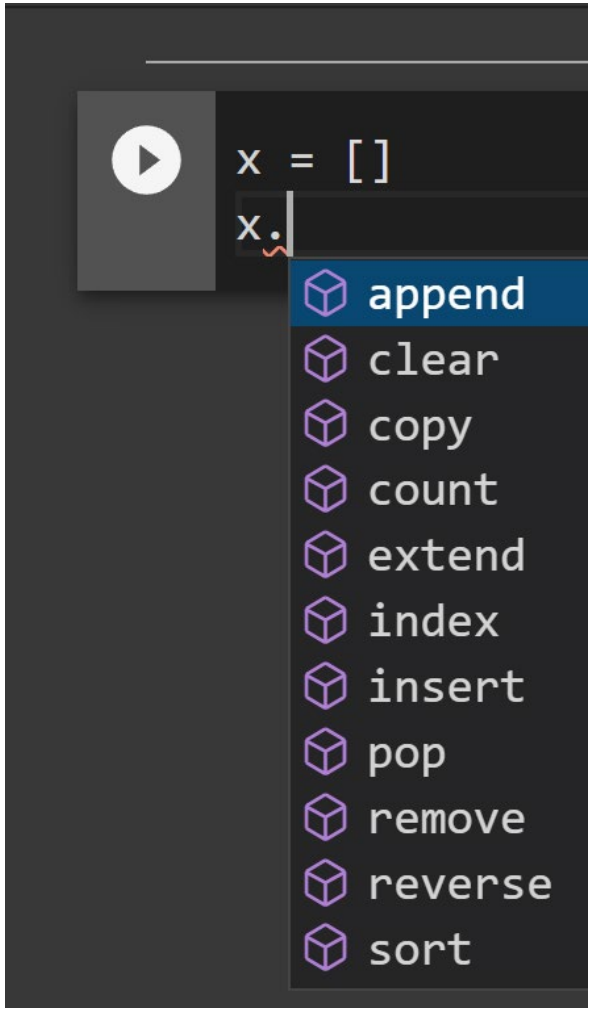
item??

foo??



IPython (getting suggestions – table completion)

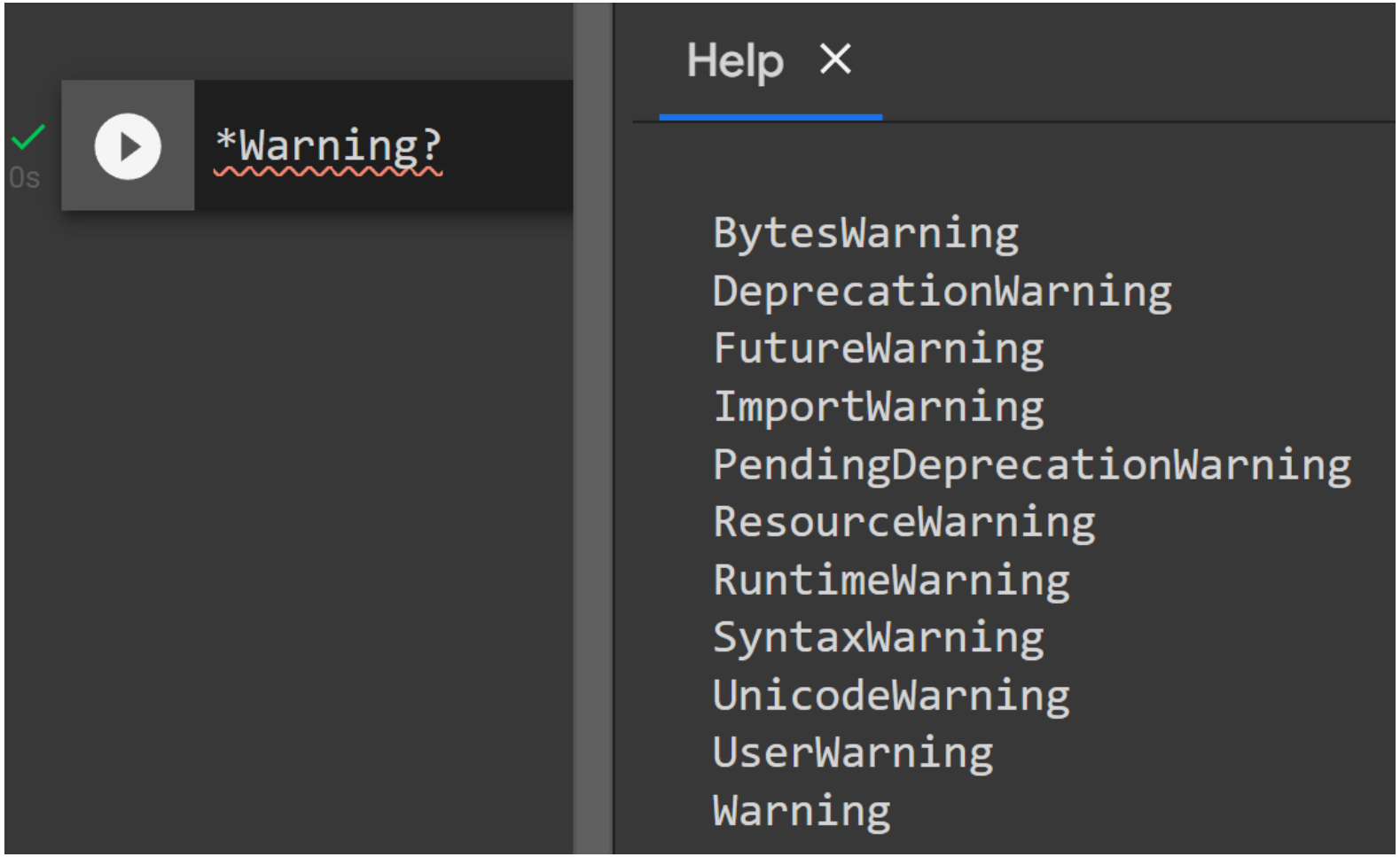
Tab completion (mimics the pop-up you get in most IDEs for completion)



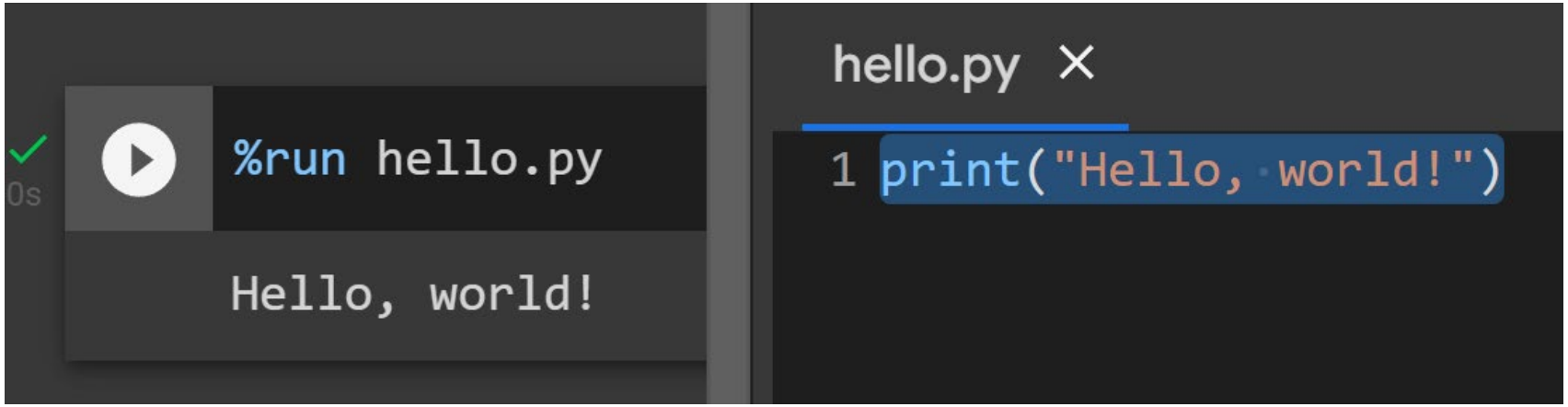

```
In [1]: x = []

In [2]: x._
        append()  count()   insert()  reverse()
        clear()   extend()  pop()     sort()
        copy()    index()   remove()
```

IPython (getting suggestions – wildcard)

*Text?



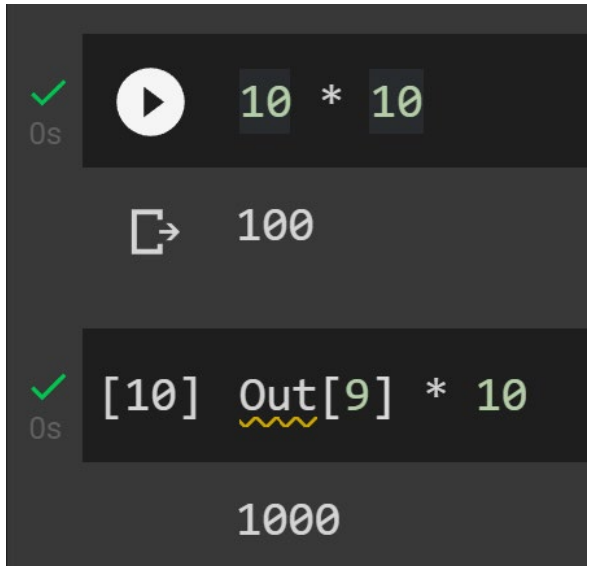
IPython (running external python .py files)

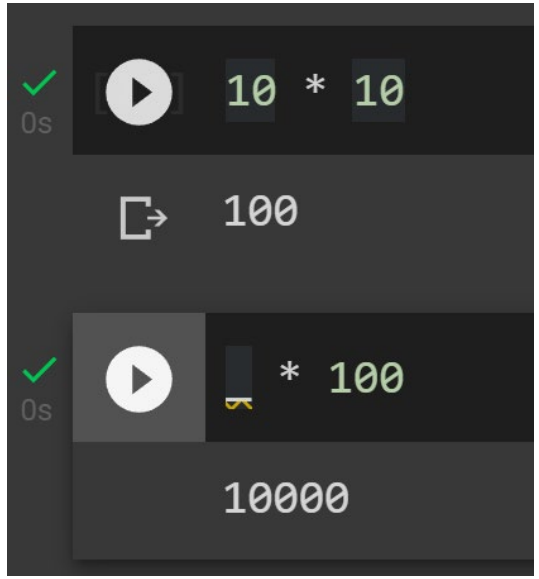


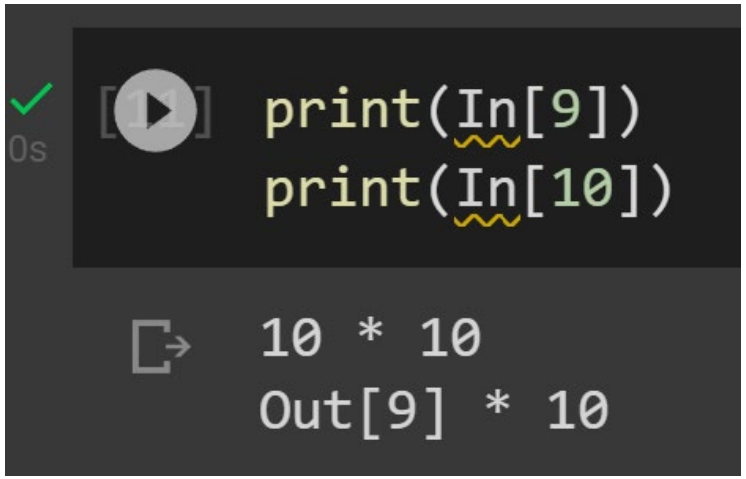
IPython (In/Out)

History stored in array blocks **In** and **Out**

Here **In**[9] produced **Out**[9] = 100, so we used it in **In**[10] as input to make **Out**[10]







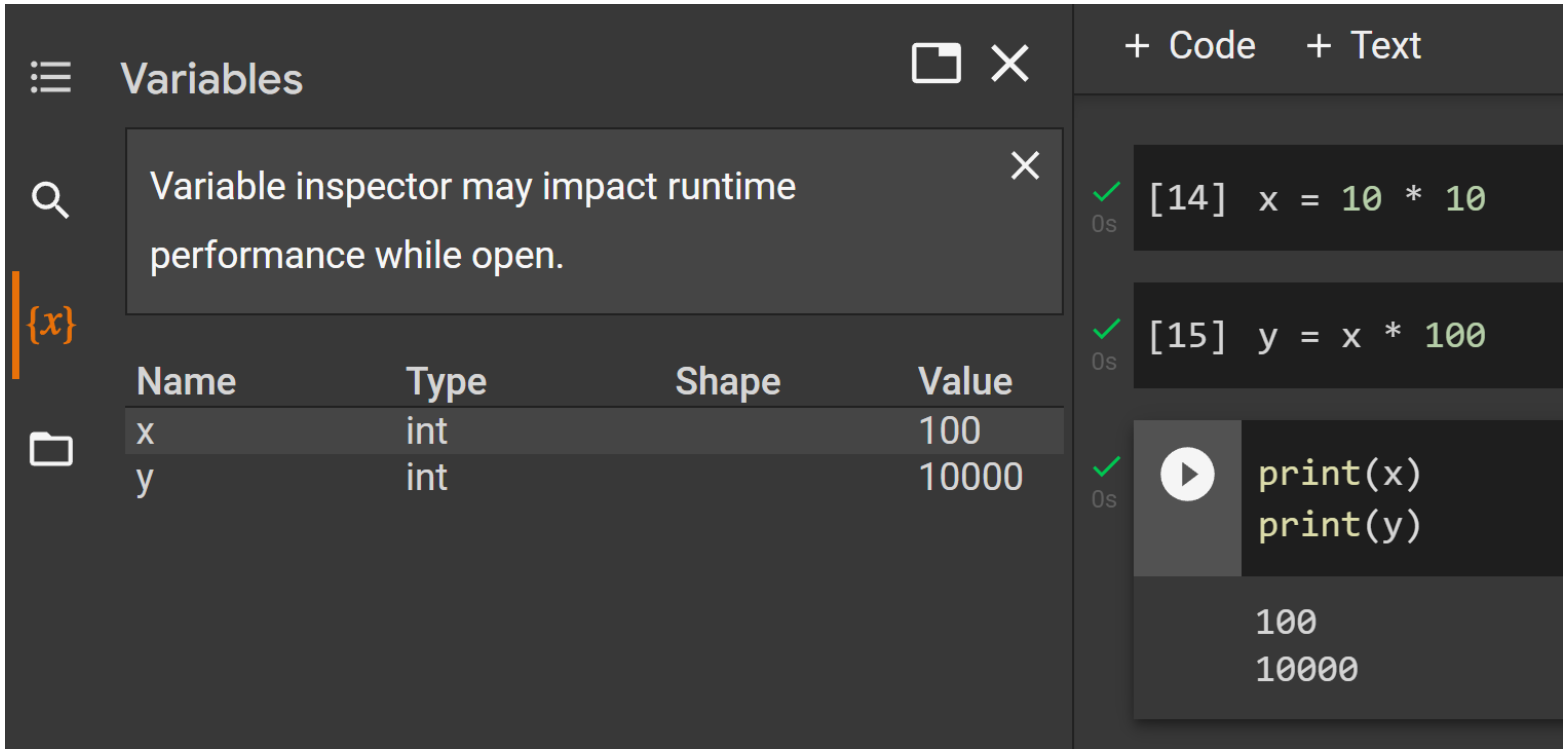
Great when you have a long or large calculation you don't want to re-calculate but didn't store into variable

_ for previous
__ for previous previous
___ for previous previous previous

IPython (variables)

You can also use variables to store data in between blocks which is often a better

But do remember you have to run a previous block before later can access it

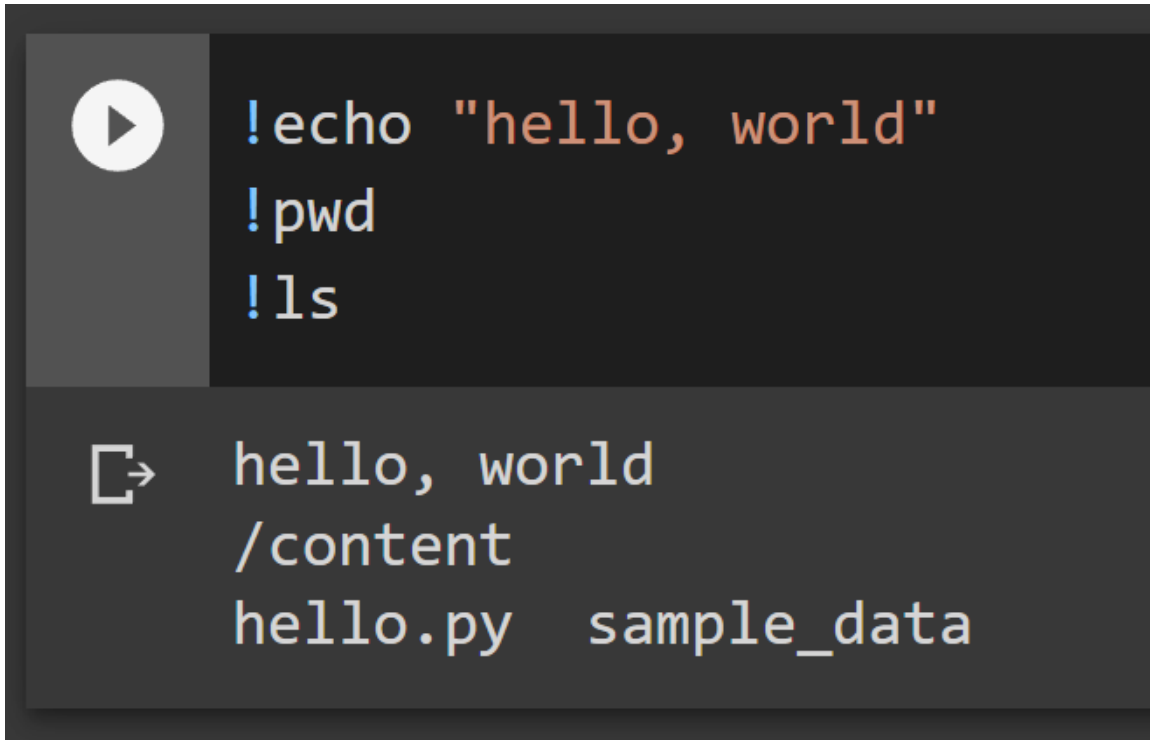


IPython (terminal commands)

Not only can use access terminal commands using !

But you can also get data from them, or send data out of program

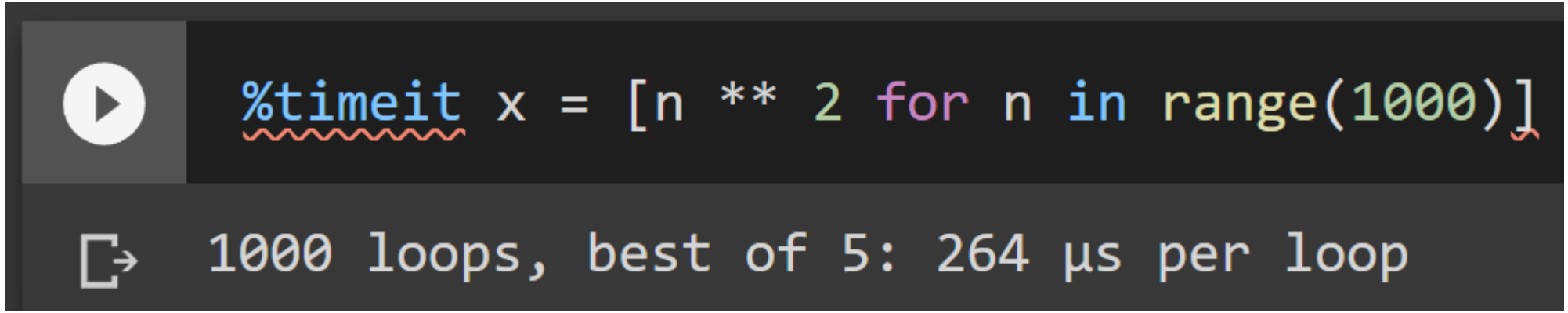
Some commands don't need !, like **run**, cd, mkdir, ls, cp, rm, cat, man, more, mv, …



IPython (timing)

%timeit for one line (%time run once)

%%timeit for multiple lines (%%time run multiple lines once)




```
In [2]: %timeit x = [n ** 2 for n in range(1000)]
230 µs ± 9.36 µs per loop (mean ± std. dev. of 7 runs,
 1,000 loops each)
```

IPython (timing dangers)

Be wary of stored results in **timeit** (could use time to only run once, or make sure we are sorting random each time)

```
import random
L = [random.random() for i in range(100000)]
%timeit L.sort()
```

```
The slowest run took 30.01 times longer than the fastest. This could mean that an intermediate result is being cached.
1000 loops, best of 5: 691 µs per loop
```

You can also profile something using **prun**

```
import random
L = [random.random() for i in range(100000)]
%prun %timeit L.sort()
```

```
         7412 function calls (7312 primitive calls) in 4.308 seconds

   Ordered by: internal time

   ncalls  tottime  percall  cumtime  percall filename:lineno(function)
     6111    4.269    0.001    4.269    0.001 {method 'sort' of 'list' objects}
        9    0.036    0.004    4.305    0.478 <magic-timeit>:1(inner)
        5    0.002    0.000    0.002    0.000 socket.py:543(send)
       12    0.000    0.000    0.000    0.000 {built-in method builtins.compile}
        1    0.000    0.000    4.308    4.308 execution.py:909(timeit)
        9    0.000    0.000    4.305    0.478 execution.py:125(timeit)
    27/1    0.000    0.000    0.000    0.000 ast.py:320(generic_visit)
       71    0.000    0.000    0.000    0.000 ast.py:193(iter_child_nodes)
      138    0.000    0.000    0.000    0.000 {built-in method builtins.getattr}
        1    0.000    0.000    3.472    3.472 timeit.py:183(repeat)
   36/1    0.000    0.000    0.000    0.000 ast.py:153(_fix)
```

*I profiled the **timeit** command*

It ran 6111 samples

Onward to … Optional Advanced Python

Jonathan Hudson
jwhudson@ucalgary.ca
https://pages.cpsc.ucalgary.ca/~jwhudson/

UNIVERSITY OF CALGARY