

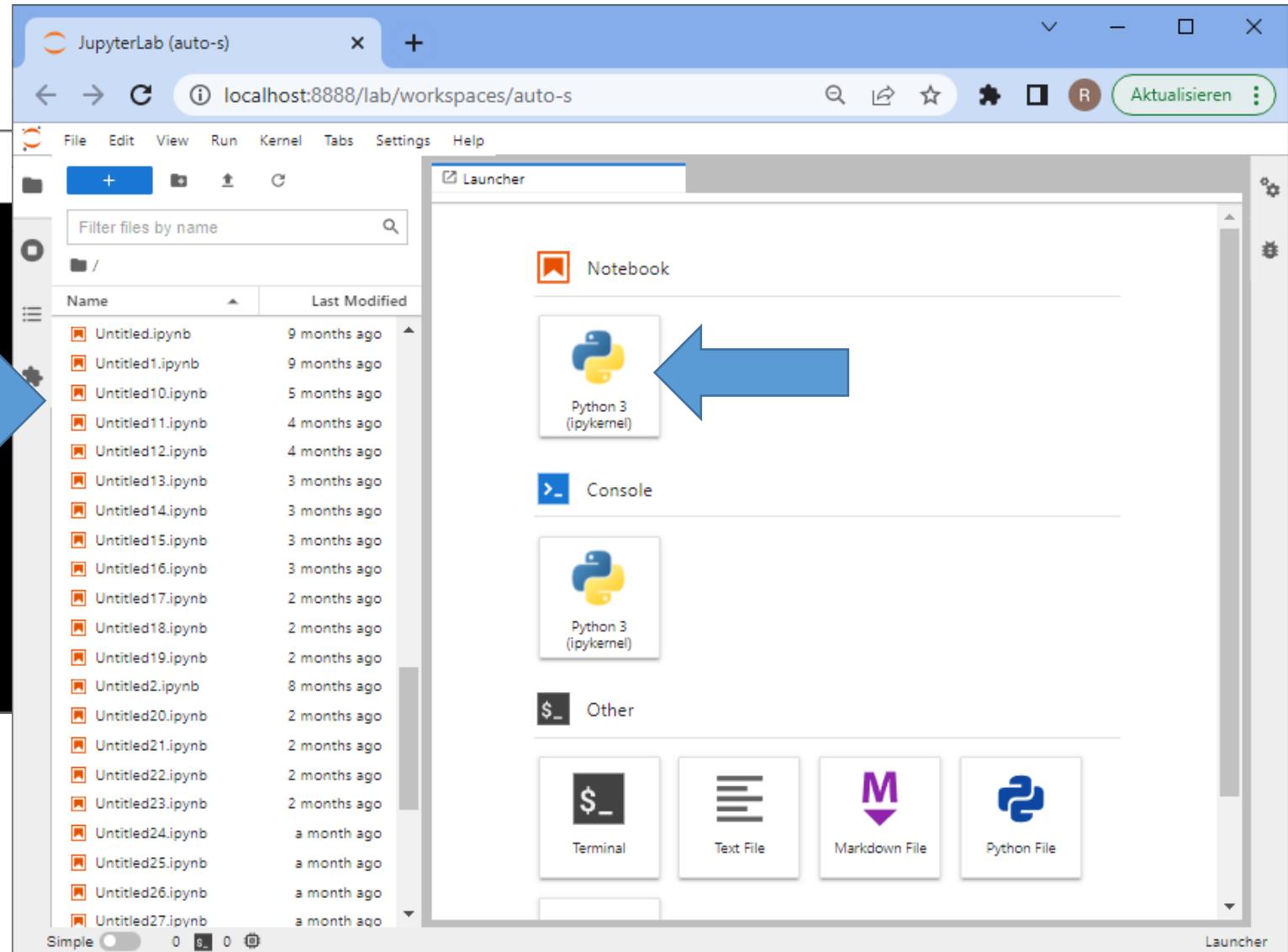


Robert Haase

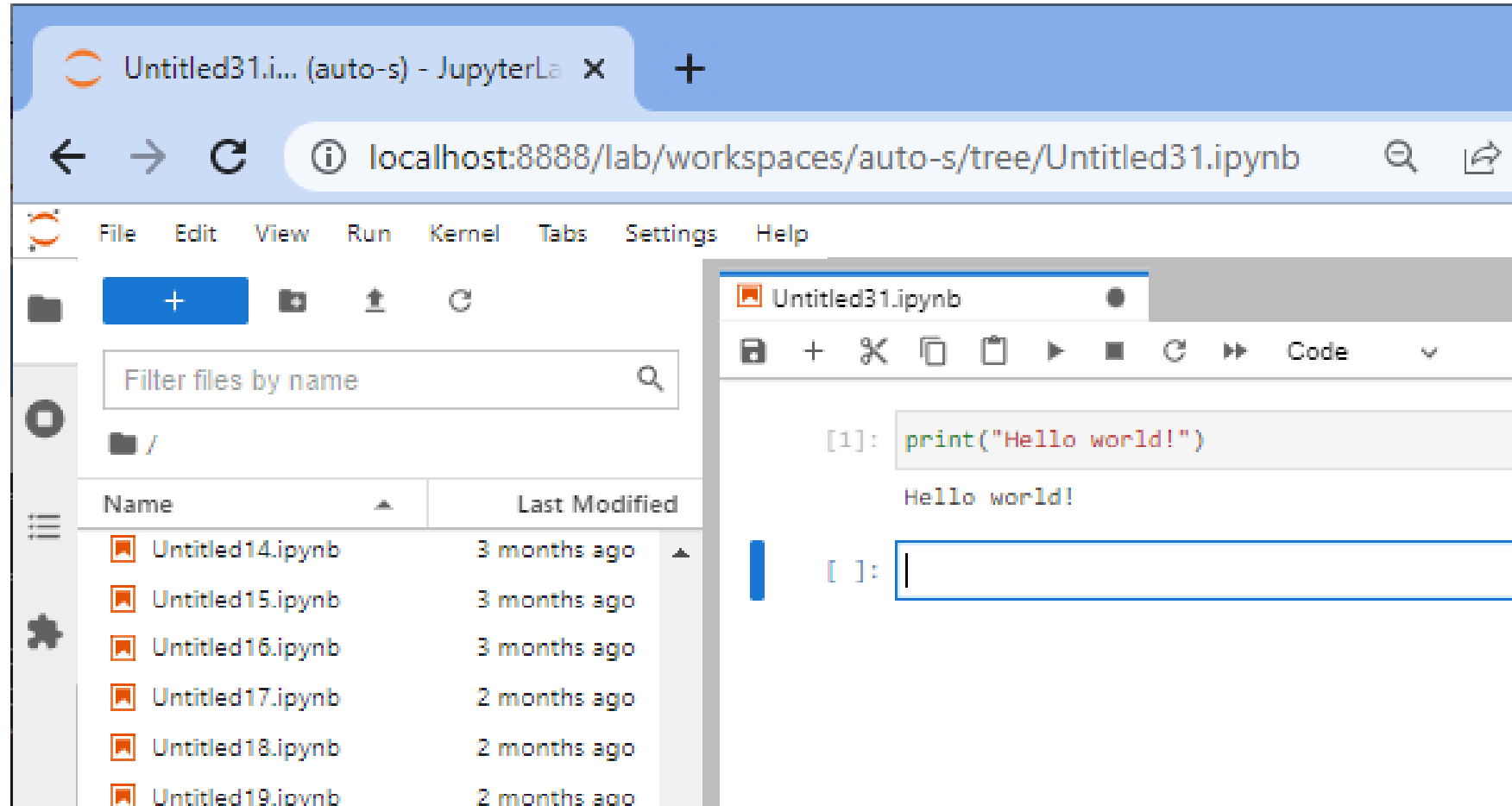
December 2022

- Our programming environment for this course

```
C:\Users\rober>conda deactivate - cond...  
  
c:\Users\rober>conda activate bio_39  
(bio_39) c:\Users\rober>jupyter lab
```



- Execute code cell-by-cell and see results instantaneously



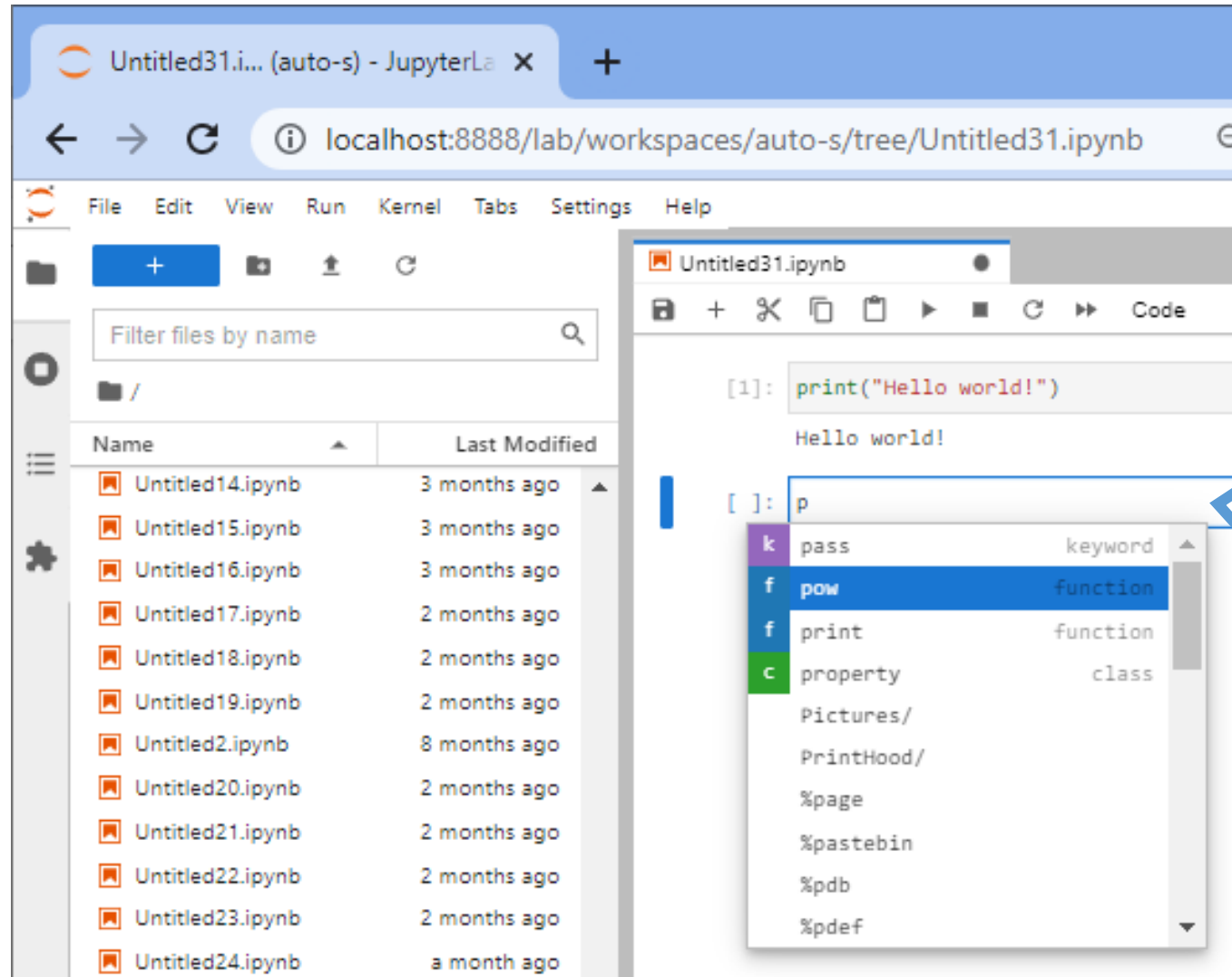
The screenshot shows the JupyterLab web interface. The top bar displays the browser tab 'Untitled31.i... (auto-s) - JupyterLa' and the URL 'localhost:8888/lab/workspaces/auto-s/tree/Untitled31.ipynb'. The left sidebar contains a file browser with a search bar and a list of files: 'Untitled14.ipynb', 'Untitled15.ipynb', 'Untitled16.ipynb', 'Untitled17.ipynb', 'Untitled18.ipynb', and 'Untitled19.ipynb'. The main area shows a code editor for 'Untitled31.ipynb' with a toolbar and a code cell containing the following text:

```
[1]: print("Hello world!")  
Hello world!
```

A blue box highlights the code cell, and a callout bubble points to it with the text:

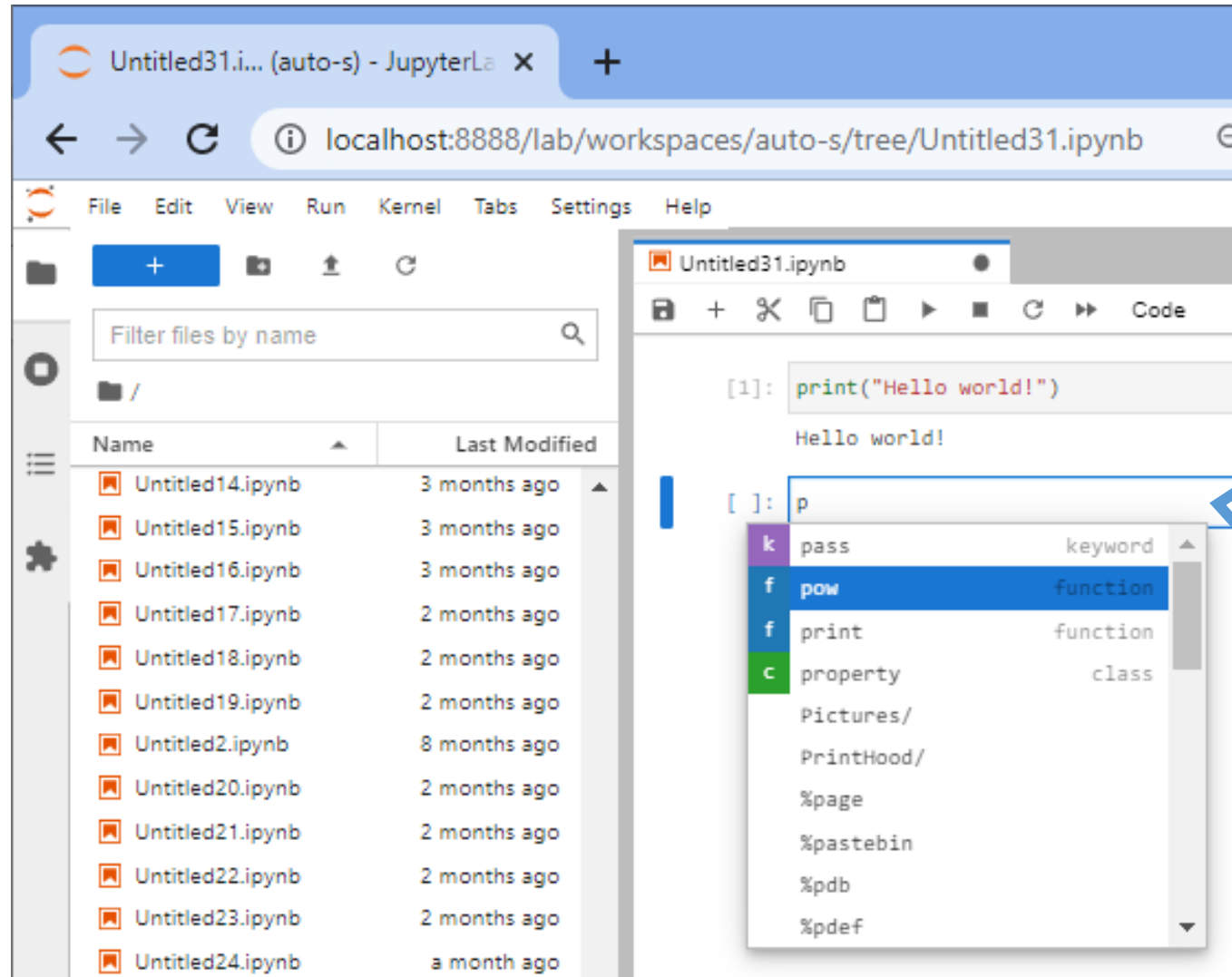
SHIFT + ENTER
to execute a
code cell

- Context-specific help, auto-completion



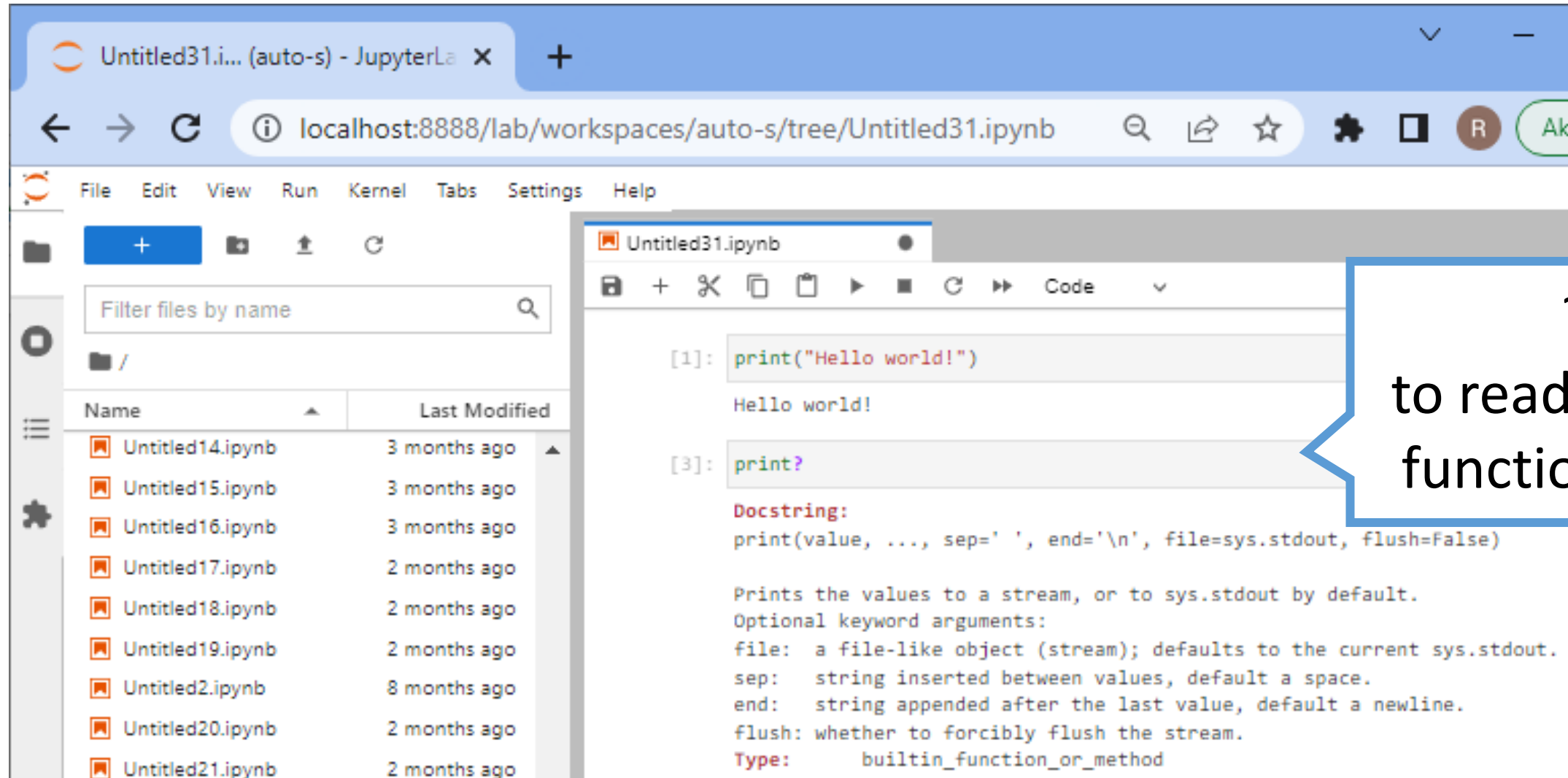
TAB
to open auto-
completion

- Context-specific help, auto-completion



TAB
to open auto-
completion

- Help / “docstrings”



The screenshot shows the JupyterLab interface. On the left is a file browser with a search bar and a list of files. On the right is a code editor for 'Untitled31.ipynb'.

File Browser:

Name	Last Modified
Untitled14.ipynb	3 months ago
Untitled15.ipynb	3 months ago
Untitled16.ipynb	3 months ago
Untitled17.ipynb	2 months ago
Untitled18.ipynb	2 months ago
Untitled19.ipynb	2 months ago
Untitled2.ipynb	8 months ago
Untitled20.ipynb	2 months ago
Untitled21.ipynb	2 months ago

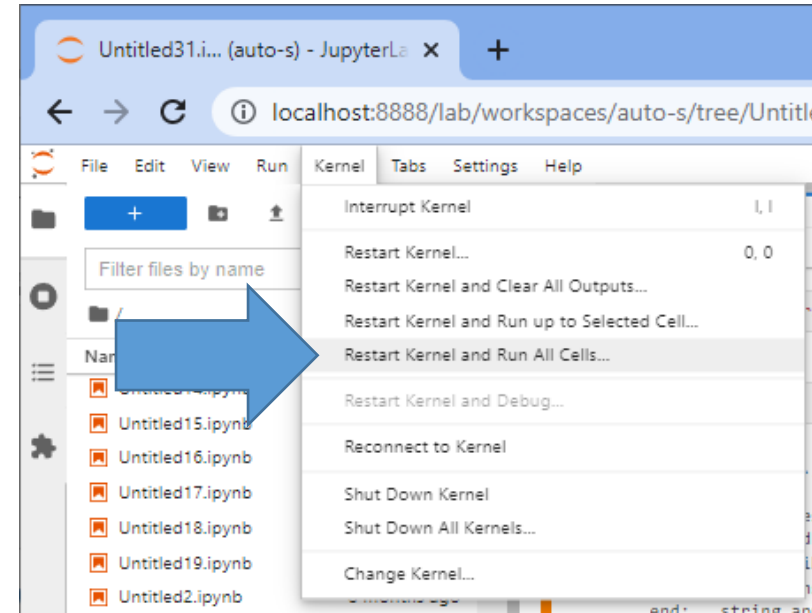
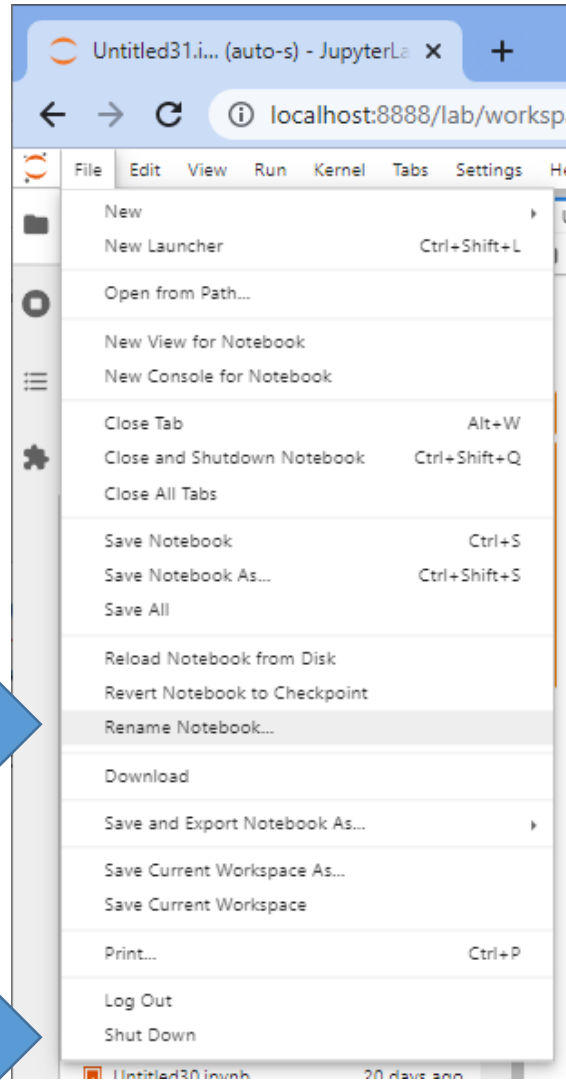
Code Editor:

```
[1]: print("Hello world!")  
Hello world!  
  
[3]: print?  
  
Docstring:  
print(value, ..., sep=' ', end='\n', file=sys.stdout, flush=False)  
  
Prints the values to a stream, or to sys.stdout by default.  
Optional keyword arguments:  
file: a file-like object (stream); defaults to the current sys.stdout.  
sep: string inserted between values, default a space.  
end: string appended after the last value, default a newline.  
flush: whether to forcibly flush the stream.  
Type: builtin_function_or_method
```

?

to read what a
function does

- Saving / renaming / closing



Enforcing a “clean” execution state is important for ensuring reproducibility and repeatability

Python programming basics

Robert Haase

December 2022

- Variables can hold numeric values and you can do math with them

```
▶ # initialize program
a = 5
b = 3

# run algorithm on given parameters
sum = a + b

# print out result
print (sum)
```

8

- Math commands supplement operators to be able to implement any form of calculations

- Power

```
▶ pow(3, 2)
```

```
] : 9
```

- Absolute

```
▶ abs(-8)
```

```
] : 8
```

- Rounding

```
▶ round(4.6)
```

```
] : 5
```

Be careful with
some of them!

```
▶ round(4.5)
```

```
] : 4
```

https://en.wikipedia.org/wiki/Rounding#Round_half_to_even

Comments should contain additional information such as

- User documentation
 - What does the program do?
 - How can this program be used?
- Your name / institute in case a reader has a question
- Comment why things are done.
- Do not comment what is written in the code already!

```
#  
# This program sums up two numbers.  
#  
# Usage:  
# * Run it in Python 3.8  
#  
# Author: Robert Haase, PoL TUD  
#         Robert.haase@tu-dresden.de  
# April 2021  
  
# initialise program  
a = 1  
b = 2.5  
  
# run complicated algorithm  
final_result = a + b  
  
# print the final result  
print( final_result )
```

- Also strings as values for variables are supported

Single and double quotes
allowed

```
▶ firstname = "Robert"  
  lastname = 'Haase'  
  
print("Hello " + firstname + " " + lastname)
```

Hello Robert Haase

- String **f**ormatting is made easy using f-strings.

```
f"This is an f-string. a's value is {a}. Doubling the value of a gives {2*a}."
```

```
"This is an f-string. a's value is 5. Doubling the value of a gives 10."
```

- Using f-strings, you can also call code from within a string. Take care of code readability!

```
f"The first_name variable contains {first_name.lower().count('r')} r letters."
```

```
'The first_name variable contains 2 r letters.'
```