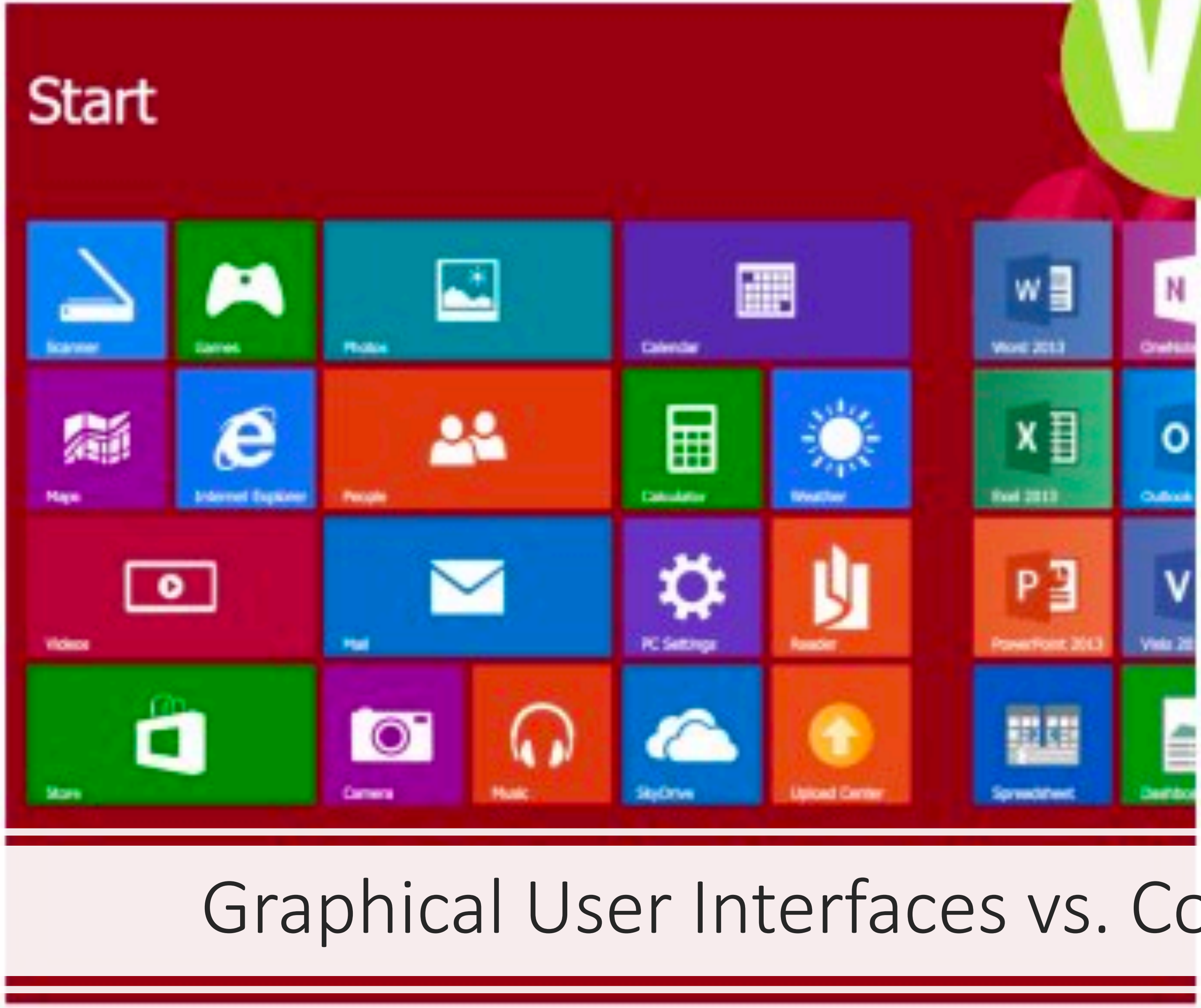
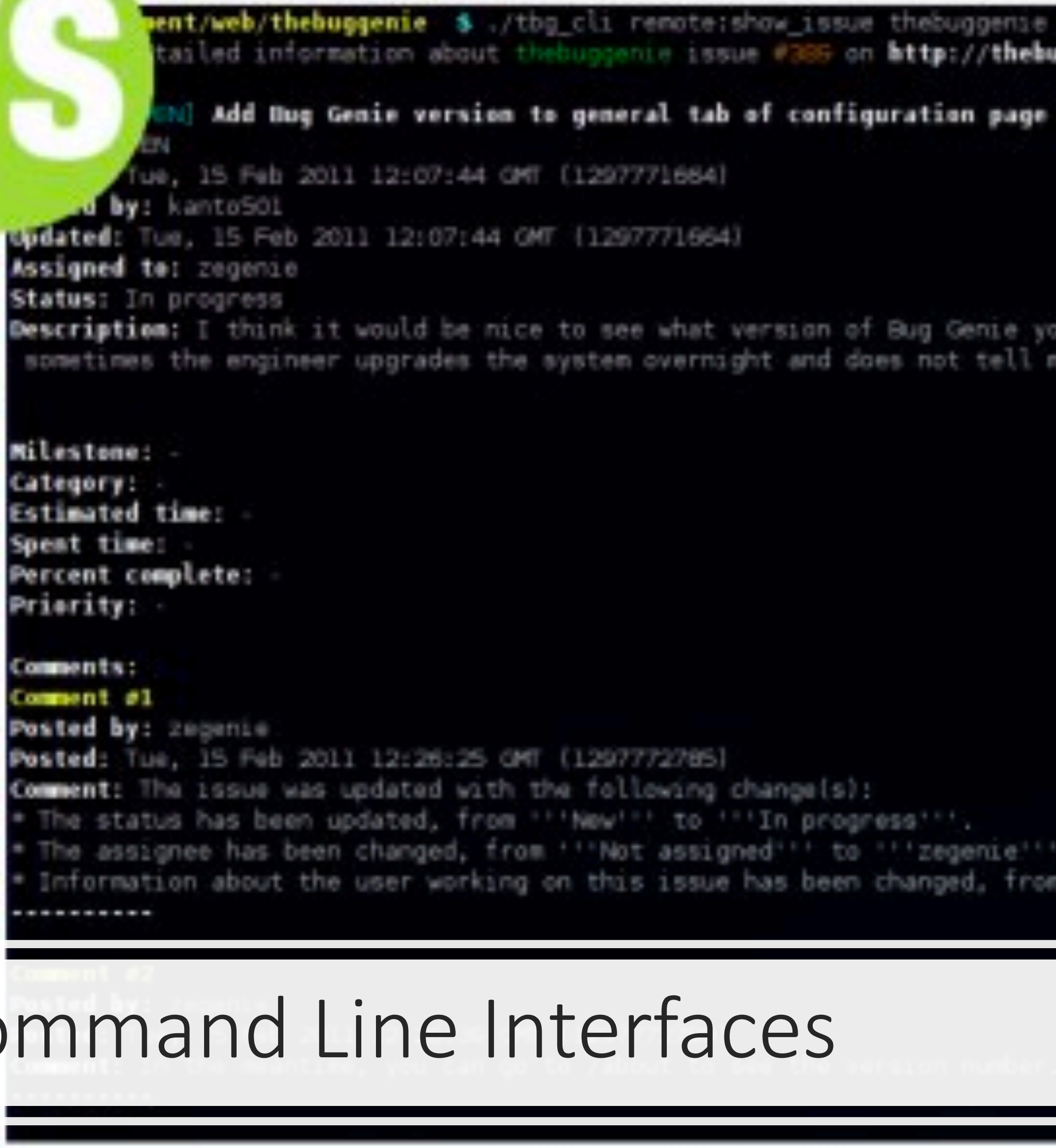
CSCI 1070: Taming Big Data

Finish Setup, Data Types and Lists

Logistics

- Due to Blackboard issues, we will be using Moodle instead
- Dennis is going to be helping us wrap up configuration issues today.
- Assignment 0 will be due Thursday before class.





```
ent/web/thebuggenie  $ ./tbg_cli remote:show_issue thebuggenie
tailed information about thebuggenie issue #386 on http://thebu

N] Add Bug Genie version to general tab of configuration page
EN
Tue, 15 Feb 2011 12:07:44 GMT (1297771664)
d by: kanto501
Updated: Tue, 15 Feb 2011 12:07:44 GMT (1297771664)
Assigned to: zegenie
Status: In progress
Description: I think it would be nice to see what version of Bug Genie yo
 sometimes the engineer upgrades the system overnight and does not tell m


Milestone: -
Category: -
Estimated time: -
Spent time: -
Percent complete: -
Priority: -

Comments:
Comment #1
Posted by: zegenie
Posted: Tue, 15 Feb 2011 12:26:25 GMT (1297772785)
Comment: The issue was updated with the following change(s):
* The status has been updated, from '''New''' to '''In progress'''.
* The assignee has been changed, from '''Not assigned''' to '''zegenie''
* Information about the user working on this issue has been changed, from
----------
```

Graphical User Interfaces vs. Command Line Interfaces

Writing and Running Python Code

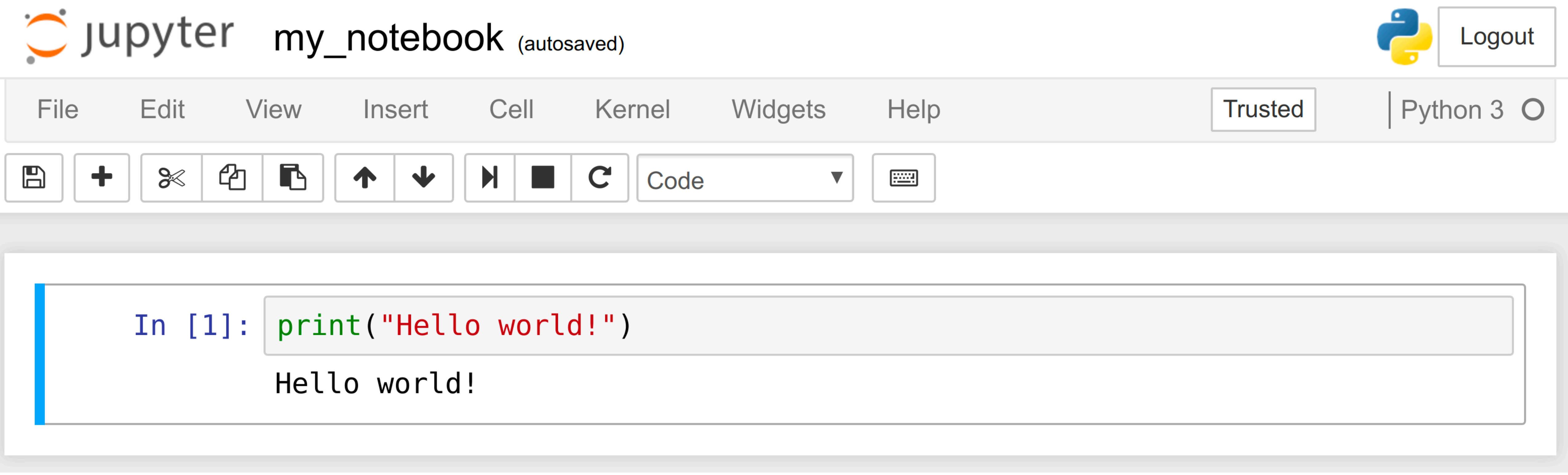
- Old-school pipeline:

1. Write a text file with python code in it
2. Save it to program_name.py
3. Go to your command line interface and run:

```
python program_name.py
```

Jupyter Notebooks

- Write and run Python code in the browser



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

        Hello world!
```

Jupyter Notebooks

- Installed on the lab machines already.
- Open Terminal/Command Prompt and run:

```
pip install jupyter
```

(if that fails, you might need to run pip install --user jupyter)

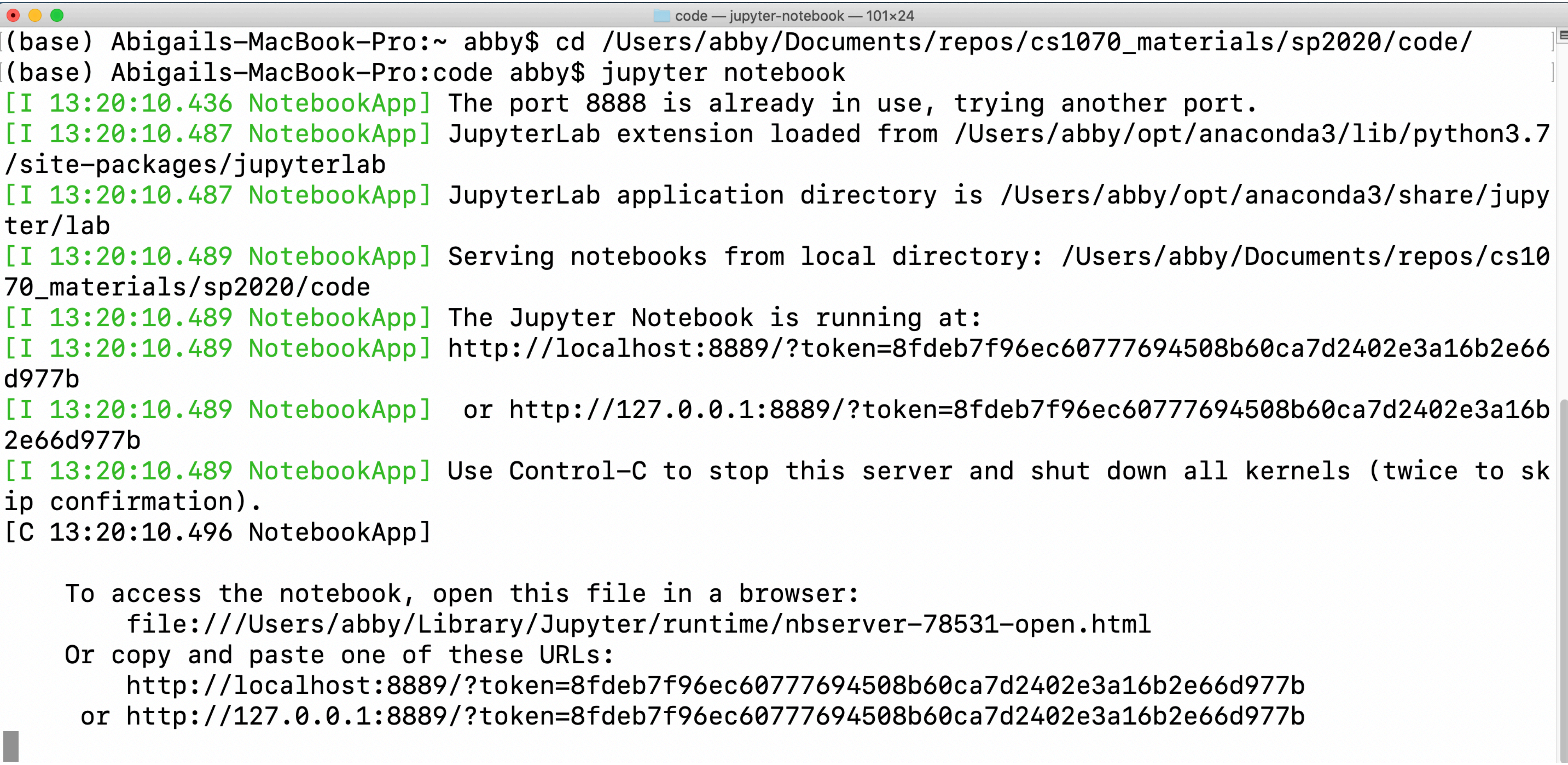
- The lab machines already have it installed
- If your computer doesn’t have pip (Windows machines especially may not, use a lab machine today and come to office hours to get set up)

Jupyter Notebooks

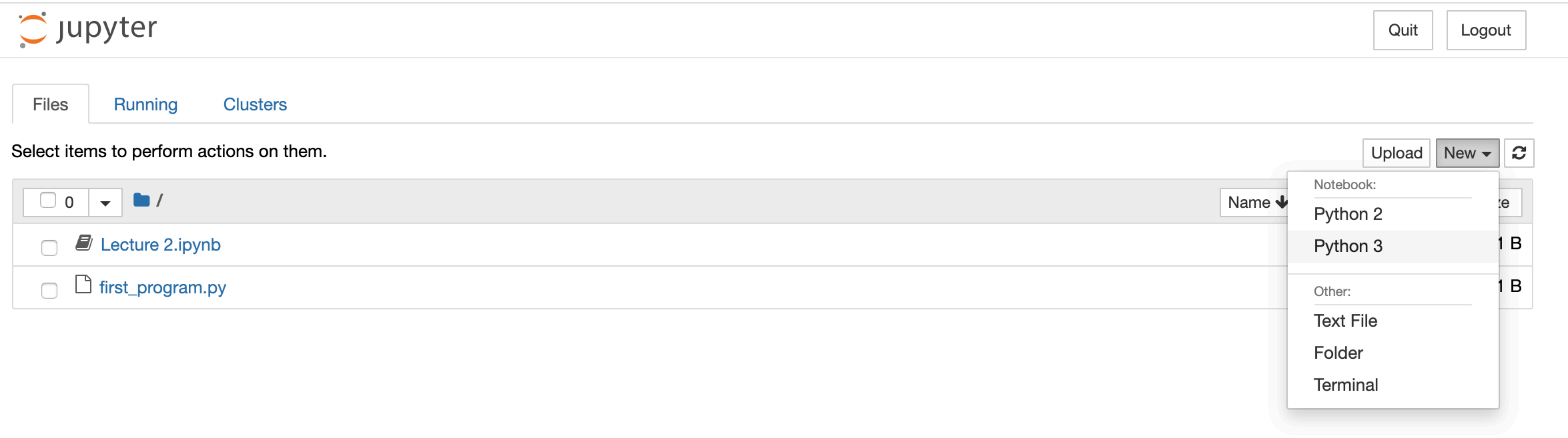
- To start a new notebook, first navigate to the folder you create earlier in Terminal (what was the command?)

- Then, run:

```
jupyter notebook
```

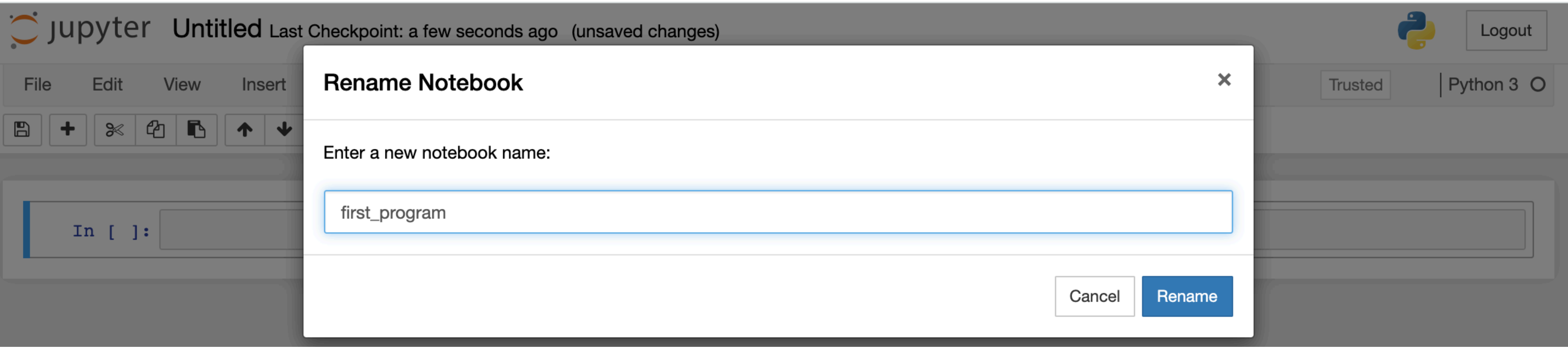


```
(base) Abigails-MacBook-Pro:~ abby$ cd /Users/abby/Documents/repos/cs1070_materials/sp2020/code/
(base) Abigails-MacBook-Pro:code abby$ jupyter notebook
[I 13:20:10.436 NotebookApp] The port 8888 is already in use, trying another port.
[I 13:20:10.487 NotebookApp] JupyterLab extension loaded from /Users/abby/opt/anaconda3/lib/python3.7
/site-packages/jupyterlab
[I 13:20:10.487 NotebookApp] JupyterLab application directory is /Users/abby/opt/anaconda3/share/jupy
ter/lab
[I 13:20:10.489 NotebookApp] Serving notebooks from local directory: /Users/abby/Documents/repos/cs10
70_materials/sp2020/code
[I 13:20:10.489 NotebookApp] The Jupyter Notebook is running at:
[I 13:20:10.489 NotebookApp] http://localhost:8889/?token=8fdeb7f96ec60777694508b60ca7d2402e3a16b2e66
d977b
[I 13:20:10.489 NotebookApp]  or http://127.0.0.1:8889/?token=8fdeb7f96ec60777694508b60ca7d2402e3a16b
2e66d977b
[I 13:20:10.489 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to sk
ip confirmation).
[C 13:20:10.496 NotebookApp]

    To access the notebook, open this file in a browser:
        file:///Users/abby/Library/Jupyter/runtime/nbserver-78531-open.html
    Or copy and paste one of these URLs:
        http://localhost:8889/?token=8fdeb7f96ec60777694508b60ca7d2402e3a16b2e66d977b
     or http://127.0.0.1:8889/?token=8fdeb7f96ec60777694508b60ca7d2402e3a16b2e66d977b
```



Select items to perform actions on them.

Lecture 2.ipynb
first_program.py

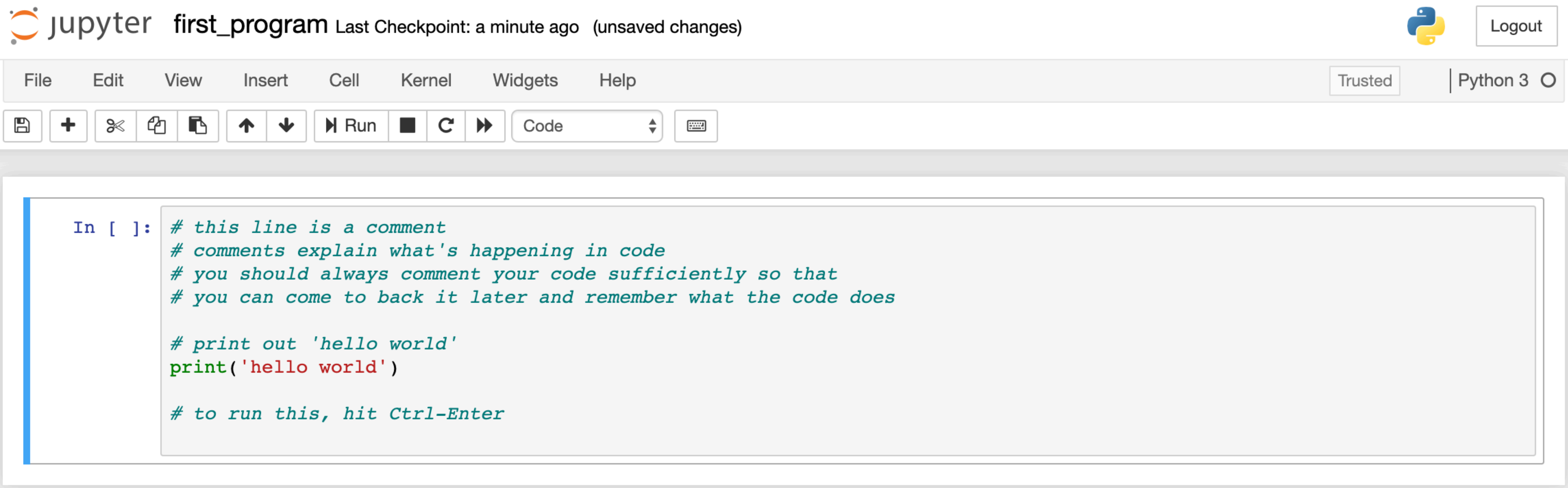

jupyter
Untitled Last Checkpoint: a few seconds ago (unsaved changes)
Logout
File
Edit
View
Insert
Trusted
Python 3
In [ ]:

Rename Notebook

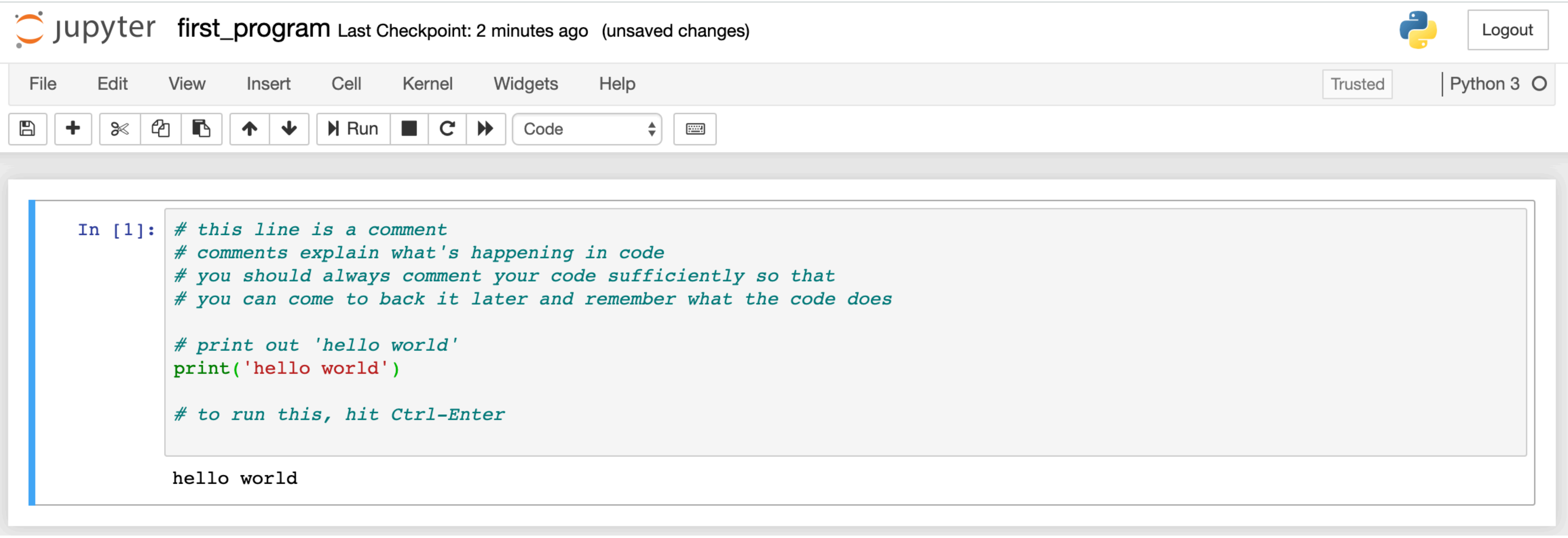
Enter a new notebook name:

first_program

Cancel Rename



```
In [ ]: # this line is a comment
        # comments explain what's happening in code
        # you should always comment your code sufficiently so that
        # you can come to back it later and remember what the code does

        # print out 'hello world'
        print('hello world')

        # to run this, hit Ctrl-Enter
```

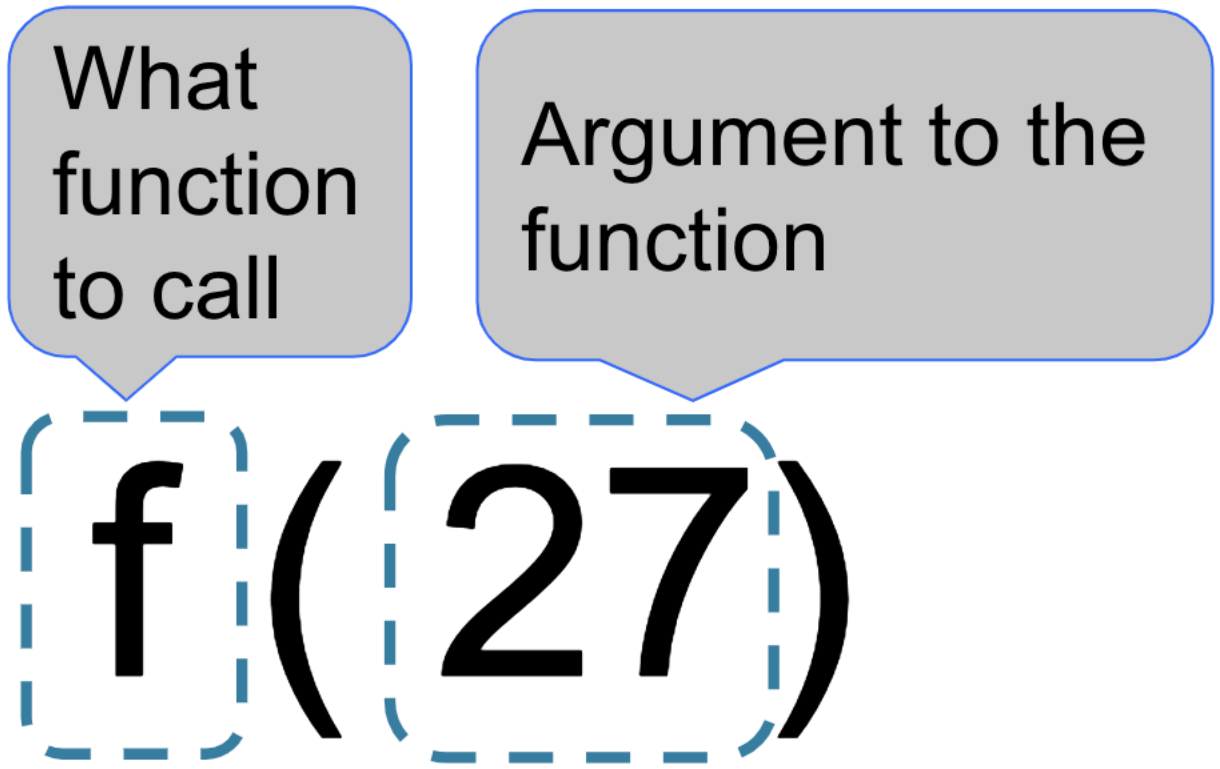

jupyter first_program Last Checkpoint: 2 minutes ago (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code

```
In [1]: # this line is a comment
        # comments explain what's happening in code
        # you should always comment your code sufficiently so that
        # you can come to back it later and remember what the code does

        # print out 'hello world'
        print('hello world')

        # to run this, hit Ctrl-Enter

        hello world
```

Operations

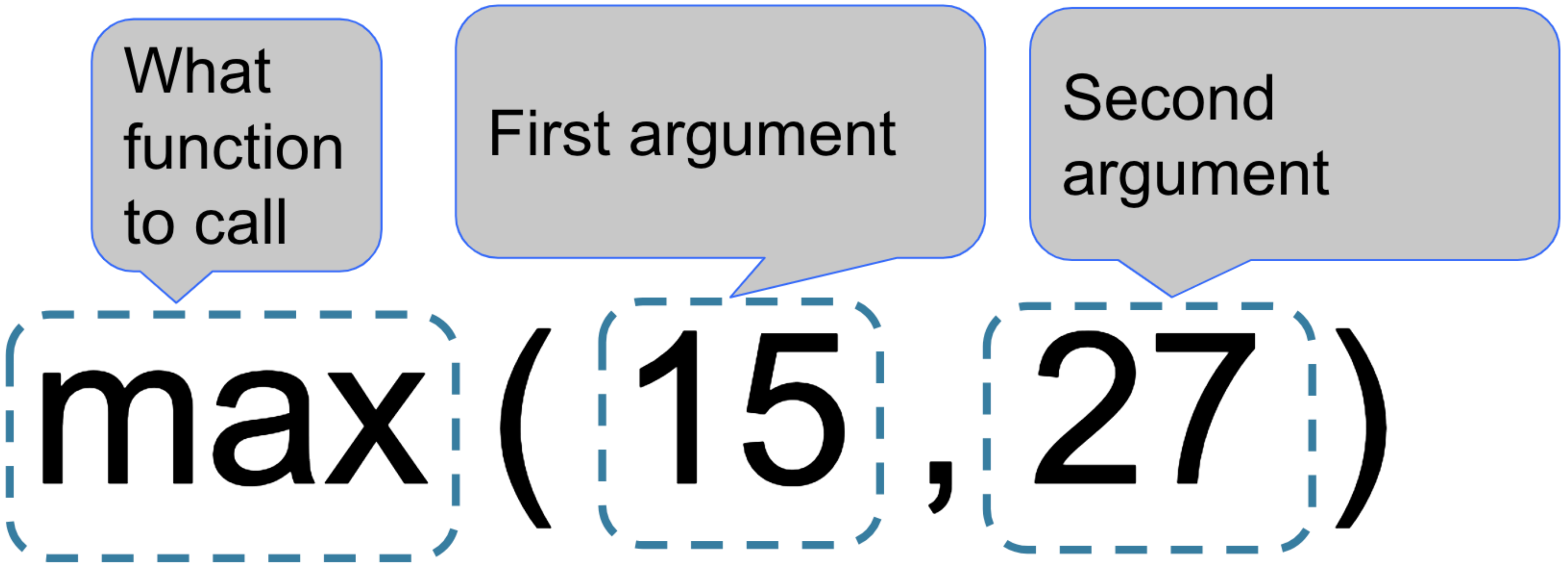
Operation	Operator	Example	Value
Addition	+	2 + 3	5
Subtraction	-	2 - 3	-1
Multiplication	*	2 * 3	6
Division	/	7 / 3	2.66667
Remainder	%	7 % 3	1
Exponentiation	**	2 ** 0.5	1.41421

Functions



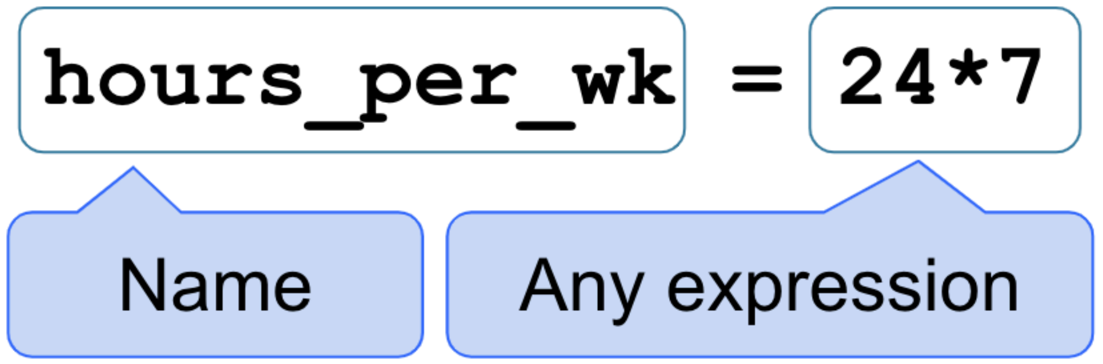
"Call f on 27"

Functions



```
max(15,27)
```

Assignment Statements



- An assignment statement changes the meaning of the name to the left of the = symbol
- The name is bound to a value (not an equation)

Data Types

- `int: 2`
- `float: 2.2`
- `bool: True`
- `str: 'Red fish, blue fish’`
- `Builtin_function_or_method: abs`

Data Types

- `int: 2`
- `float: 2.2`
- `bool: True`
- `str: 'Red fish, blue fish’`
- `Builtin_function_or_method: abs`

The `type` function can tell you the type of a value

- `type(2)`
- `type(2.2)`
- `type(True)`
- `type(’Red fish, blue fish’)`
- `type(abs)`

Conversions

- Strings that contain numbers can be converted to numbers
  - `int(‘12’)`
  - `float(‘1.2’)`

Conversions

- Strings that contain numbers can be converted to numbers
  - `int(‘12’)`
  - `float(‘1.2’)`
- Any value can be converted to a string
  - `str(5)`
  - `str(True)`
  - `str(abs)` ← anyone know what this would return?

Conversions

- Strings that contain numbers can be converted to numbers
  - `int(‘12’)`
  - `float(‘1.2’)`

- Any value can be converted to a string
  - `str(5)`
  - `str(True)`
  - `str(abs)` ← anyone know what this would return?

- Numbers can be converted to other numeric types
  - `float(1)`
  - `int(1.2)`
  - `round(1.2)`

Lists

- Container that holds a number of objects in an order

```
L = ['yellow', 'red', 'blue', 'green', 'black']
```

- Accessing / Indexing

`L[0]`	'yellow'
`L[1:4]`	['red', blue', 'green']
`L[3:]`	['green', 'black']
`L[-1]`	['black']

- Length

`len(L)`	5

Lists

- Built-in methods for adding objects

```
L.append(‘pink’)
print(L)
```

```
[‘yellow’, ‘red’, ‘blue’, ‘green’, ‘black’, ‘pink’]
```

```
L.insert(0,’white’)
print(L)
```

```
[‘white’, ‘yellow’, ‘red’, ‘blue’, ‘green’, ‘black’, ‘pink’]
```

```
L2 = [‘orange’, ‘cyan’, ‘magenta’]
L.extend(L2)
print(L)
```

```
[‘white’, ‘yellow’, ‘red’, ‘blue’, ‘green’, ‘black’, ‘pink’, ‘orange’, ‘cyan’, ‘magenta’]
```

Lists

```
L = [‘white’, ‘yellow’, ‘red’, ‘blue’, ‘green’, ‘black’,
     ‘pink’, ‘orange’, ‘cyan’, ‘magenta’]
```

- Built-in methods for removing objects

```
L.remove(‘white’)
print(L)
```

```
[‘yellow’, ‘red’, ‘blue’, ‘green’, ‘black’, ‘pink’, ‘orange’, ‘cyan’, ‘magenta’]
```

```
del L[0]
print(L)
```

```
[‘red’, ‘blue’, ‘green’, ‘black’, ‘pink’, ‘orange’, ‘cyan’, ‘magenta’]
```

```
L.pop()
```

```
‘magenta’
```

```
print(L)
```

```
[‘yellow’, ‘red’, ‘blue’, ‘green’, ‘black’, ‘pink’, ‘orange’, ‘cyan’]
```

Lists

```
L = [‘yellow’, ‘red’, ‘blue’, ‘green’, ‘black’, ‘pink’,
                    ‘orange’, ‘cyan’]
```

- Other built in methods

```
L.sort()
print(L)
```

```
['black', 'blue', 'cyan', 'green', 'orange', 'pink', 'red', 'yellow’]
```

```
L.count(‘red’)
```

```
1
```

```
L.reverse()
```

```
['yellow', 'red', 'pink', 'orange', 'green', 'cyan', 'blue', 'black']
```