

# Jupyter Notebooks

Dr. Thomas Robey

Supercomputing Challenge Kickoff

Fall, 2020

## Introduction

Jupyter Notebooks allow computer code, equations, text and images. The computer code can be run right from the notebook. Over 40 computer languages are supported. You can try different things out and document what you are doing so that you can go back and find things later. You can run your model multiple times and put the results in your notebook. Writing a report or presentation is much easier by just going back to your notebook. Notebooks can be shared.

Tip: As you write and update your computer code get in the habit of cutting and pasting it into a notebook. Add the date and the reason for the change. Even if it is a code snippet that cannot be run on its own, having a record of your changes allow going back to previous versions if you run into problems. If you do this properly then it will give you much more freedom to try different things without worrying about breaking things.

# Getting Started

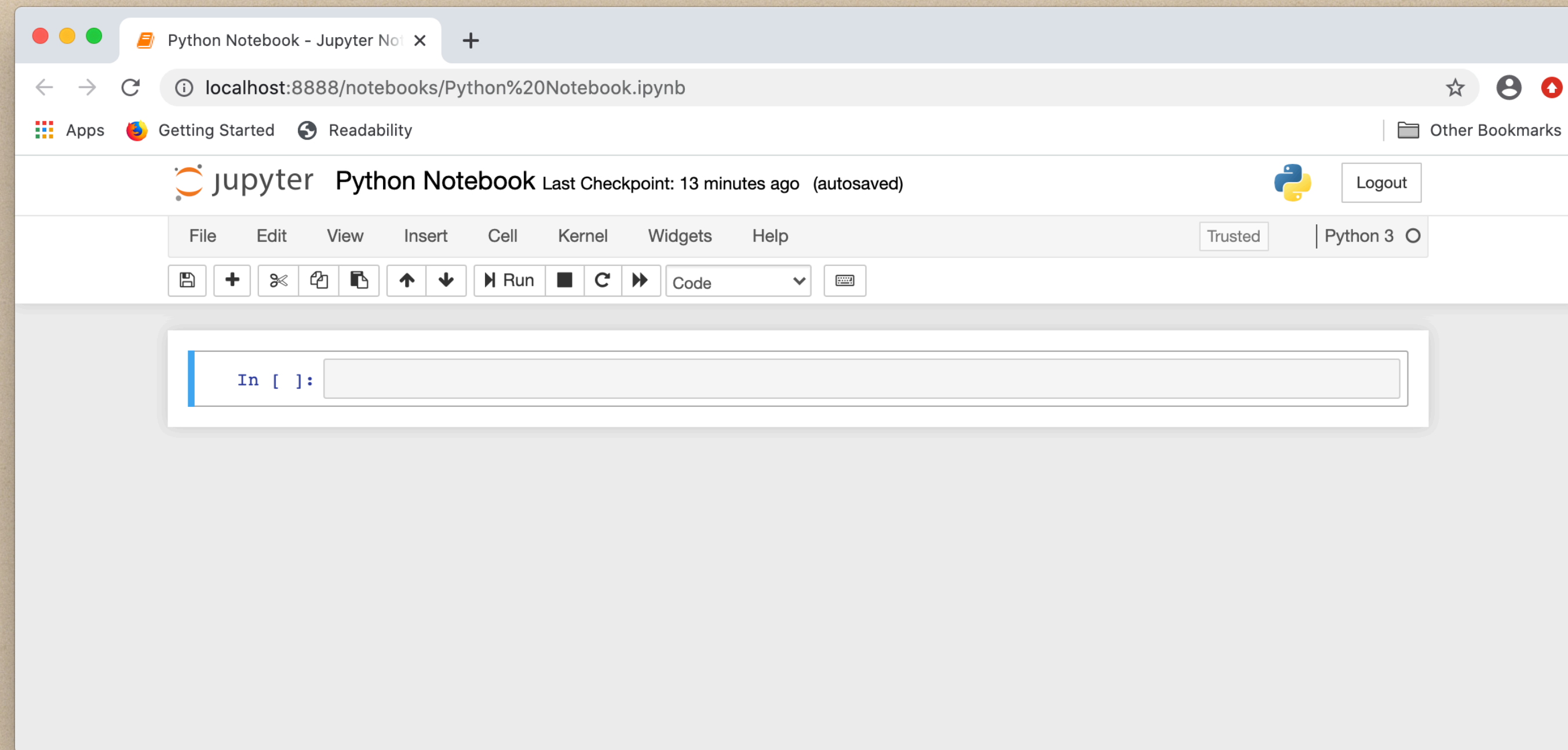
Start by installing Jupyter notebooks. There are instructions in the Appendices. Then from a terminal start Jupyter notebooks.

- jupyter notebook

For Linux run

- jupyter-notebook

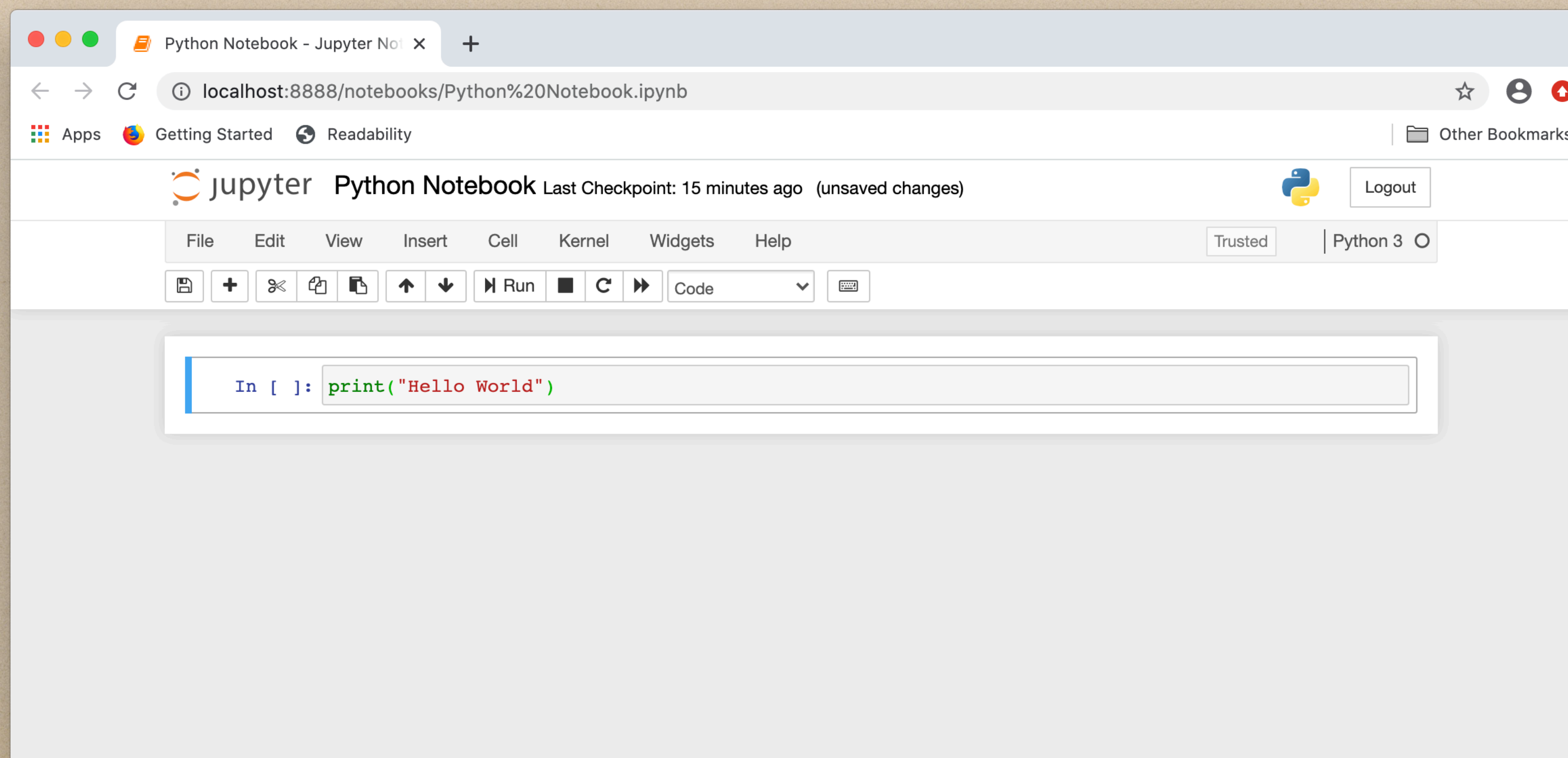
This will open a browser. In the upper right hand corner click on New and select Python 3. Click on File -> Save as.. and select a name for your notebook.



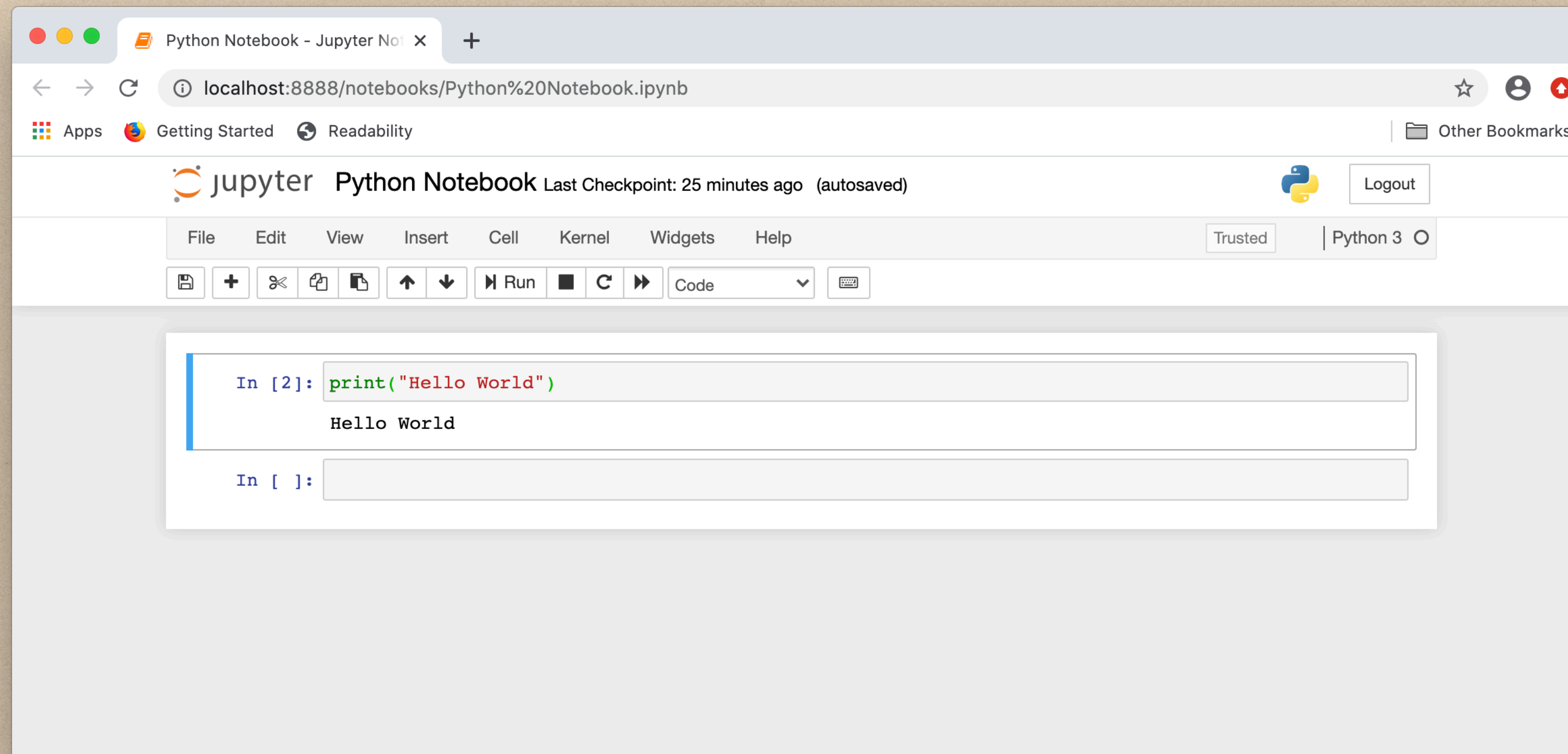
# Computer Code

Use your mouse to click in the box in your browser. Type in

- `print("Hello World")`



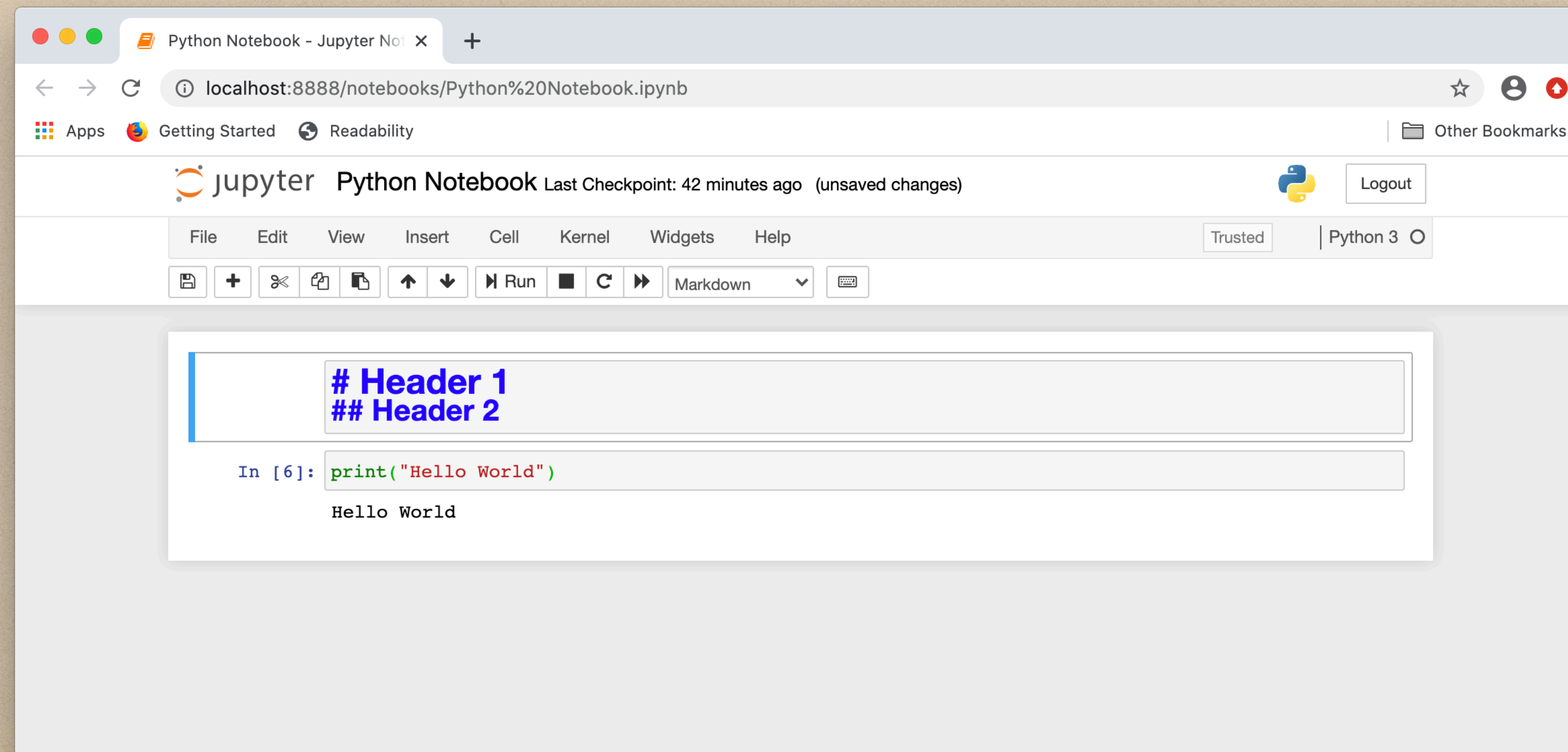
Now go up to the menu and click on run. Your Python code should execute and the output displayed. Congratulations, you have just run your code!



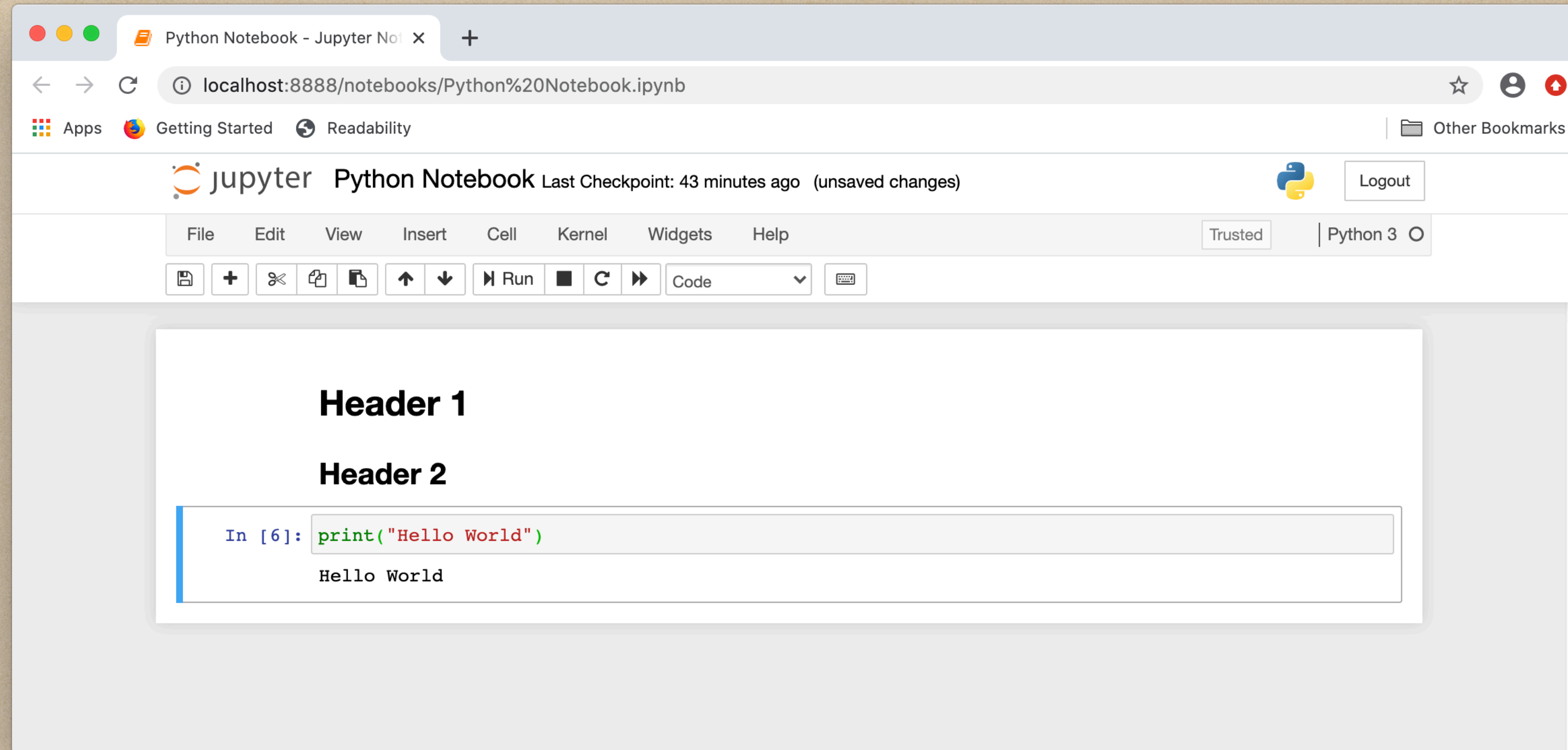
Jupyter Notebooks for learning Python 3: <https://github.com/jerry-git/learn-python3>

# Text

Jupyter Notebooks uses markdown text. If you have your code cell selected, then go to Insert -> Insert Cell Above. Then with your new cell selected change Code to Markdown. Markdown uses # for headers. Repeat to get smaller headers. Make sure to include a space before your header text.

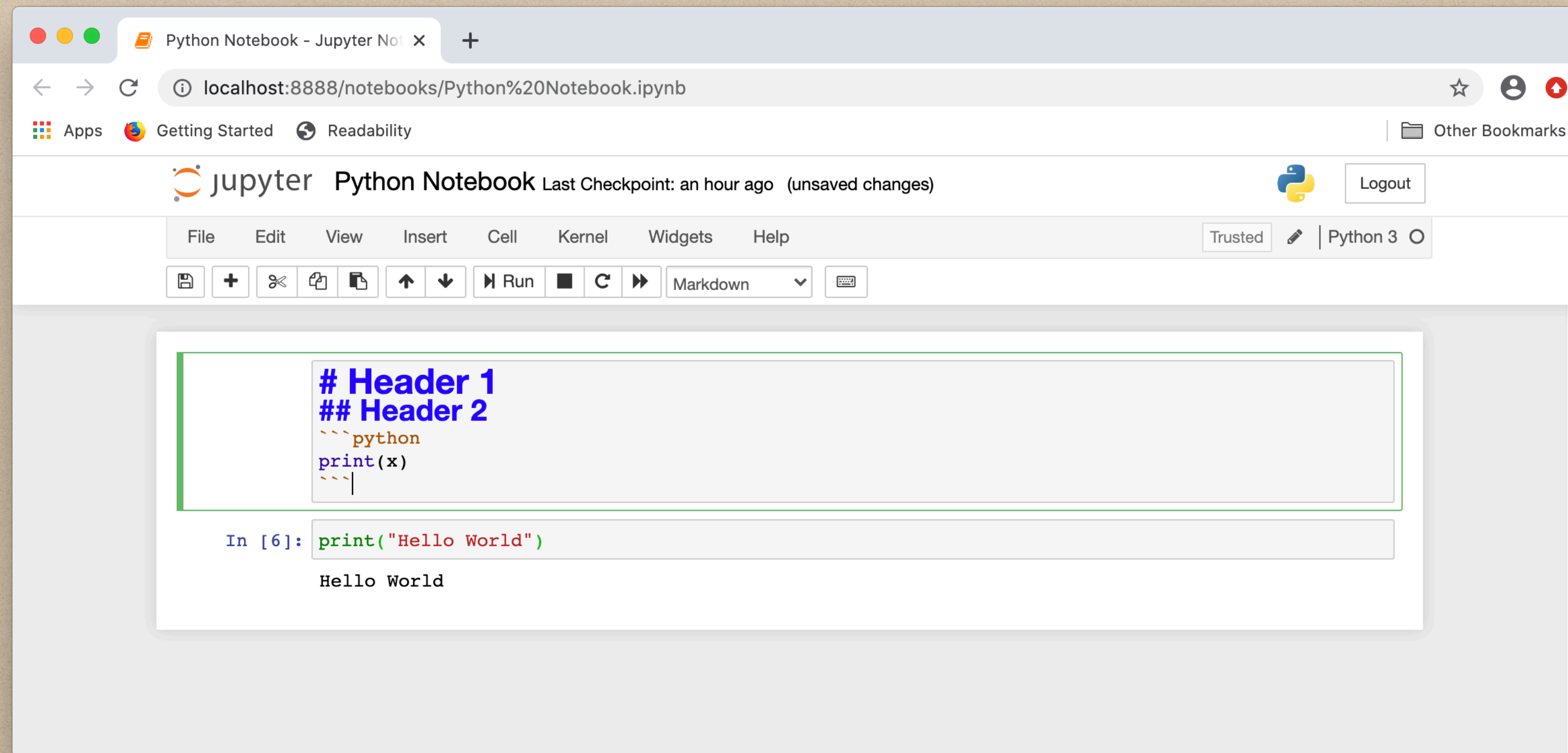


Click on run.



Surround text with single asterisks or underscores to get italics. Use two to make the text bold. Double click your markdown cell and try it! Use a dash, plus sign, or asterisk followed by a space to get a list.

You can enter code in your text (which cannot be run) by surrounding it with backticks. A block of code can be entered with triple backticks (and the computer language to get code highlighting).

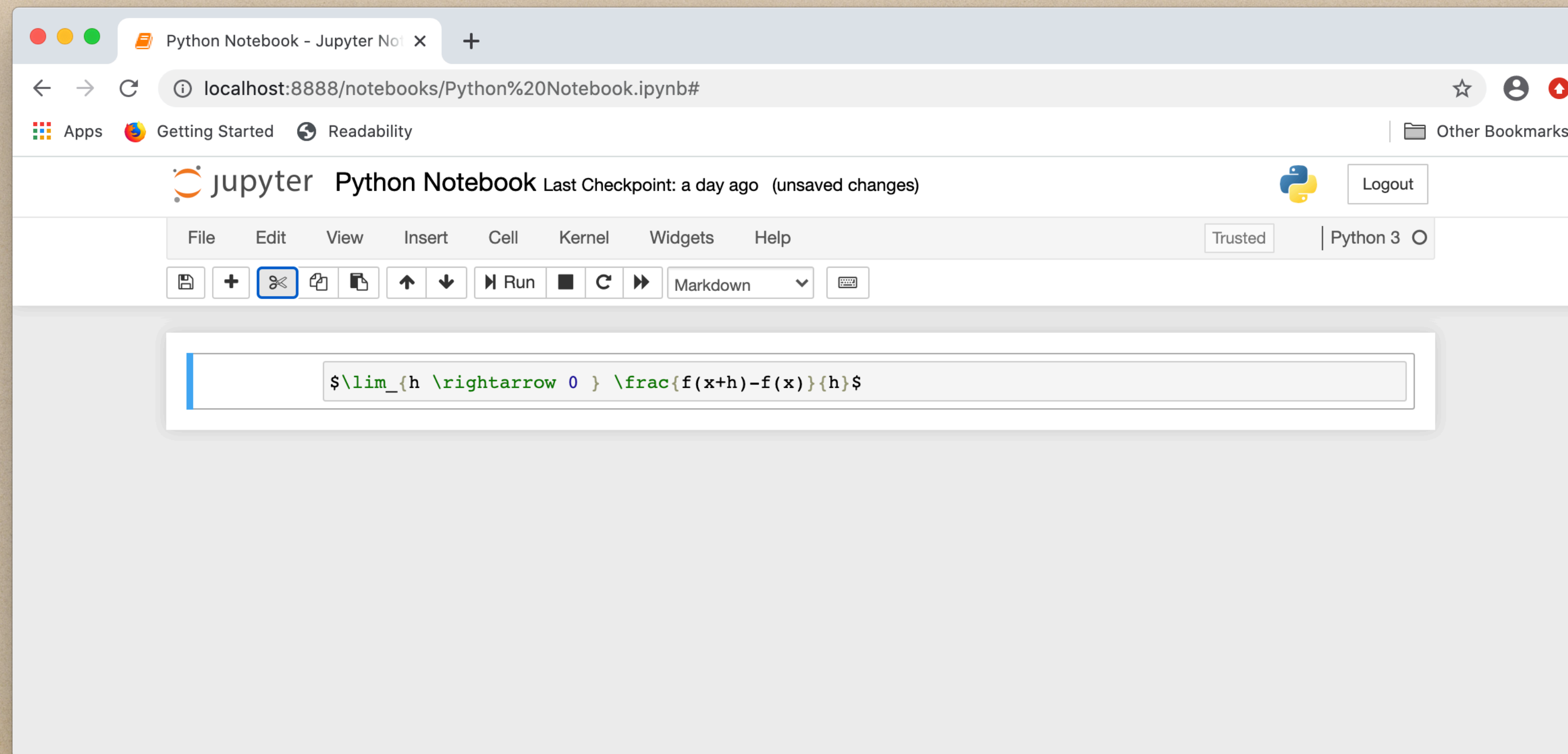


Markdown cheat sheet: <https://www.markdownguide.org/cheat-sheet/>

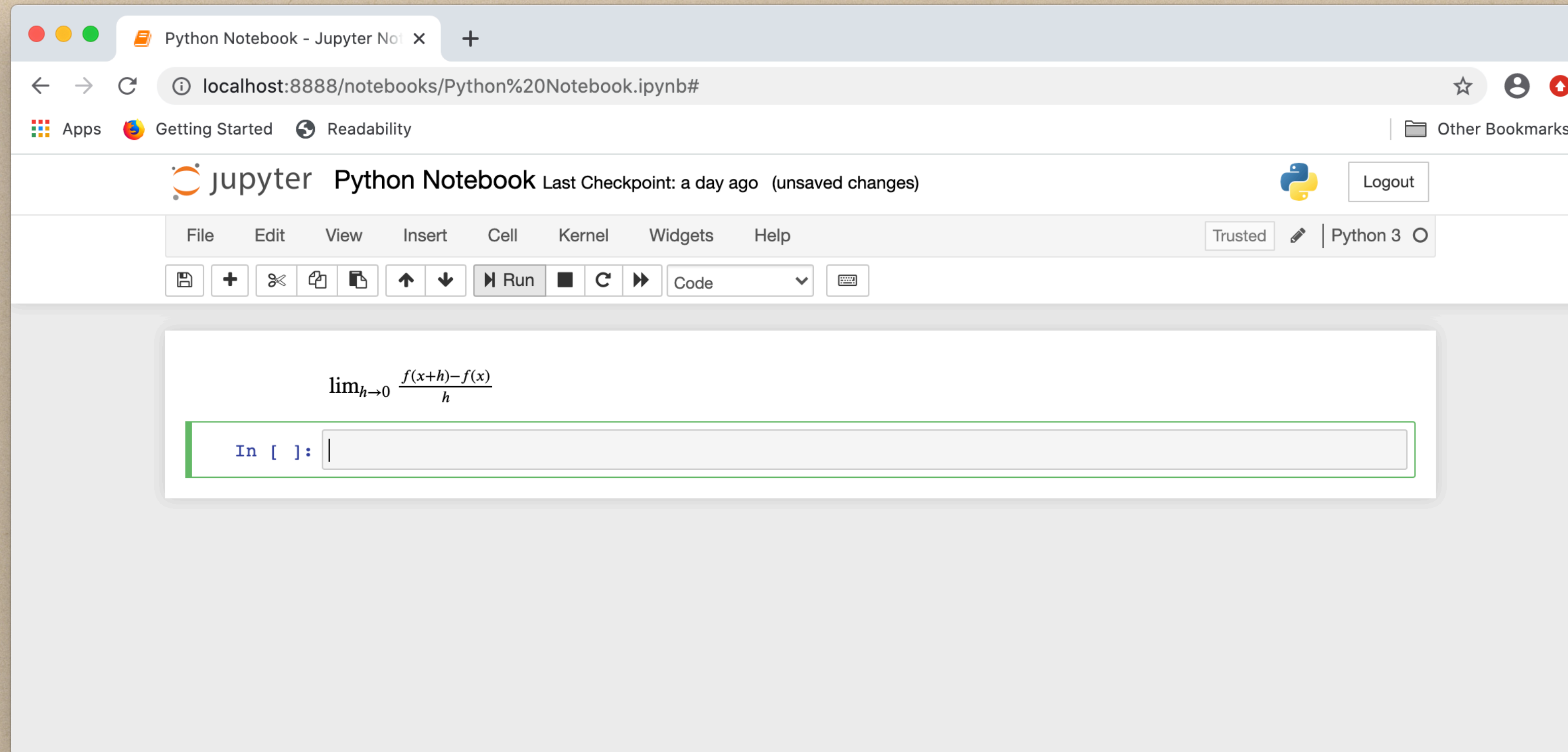


# Mathematical Equations

Jupyter Notebook has LaTeX support built-in which provides the ability to enter mathematical equations. For inline equations enter the equation between single \$. For a block with an equation use double \$\$ instead. For example



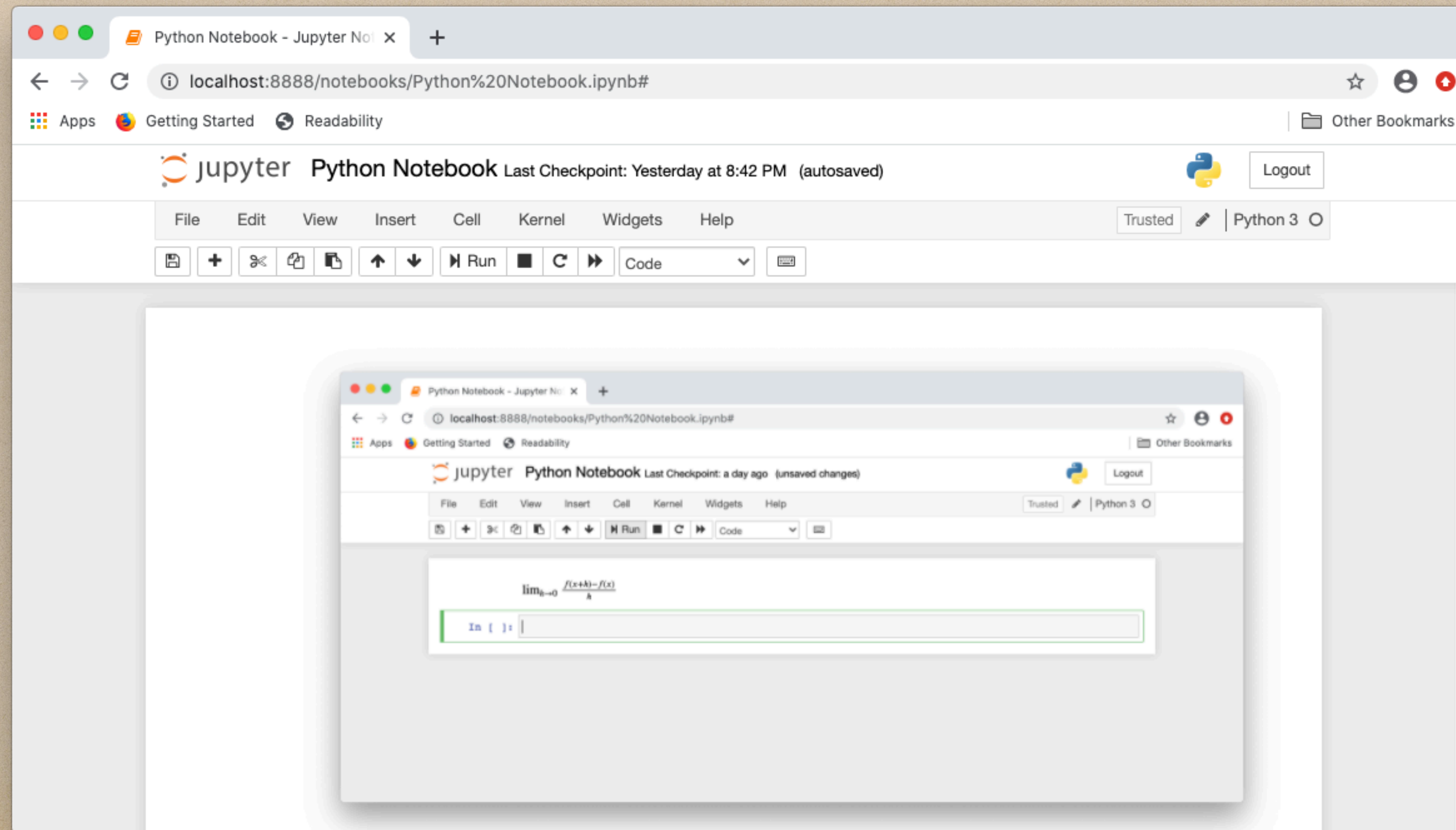
Click on the run button to format the equation



LaTeX math cheat sheet: <http://tug.ctan.org/info/undergradmath/undergradmath.pdf>

# Images

Jupyter Notebooks has support for drag and drop for images since version 5.0.0. Just drag the file into a markdown cell and then run.



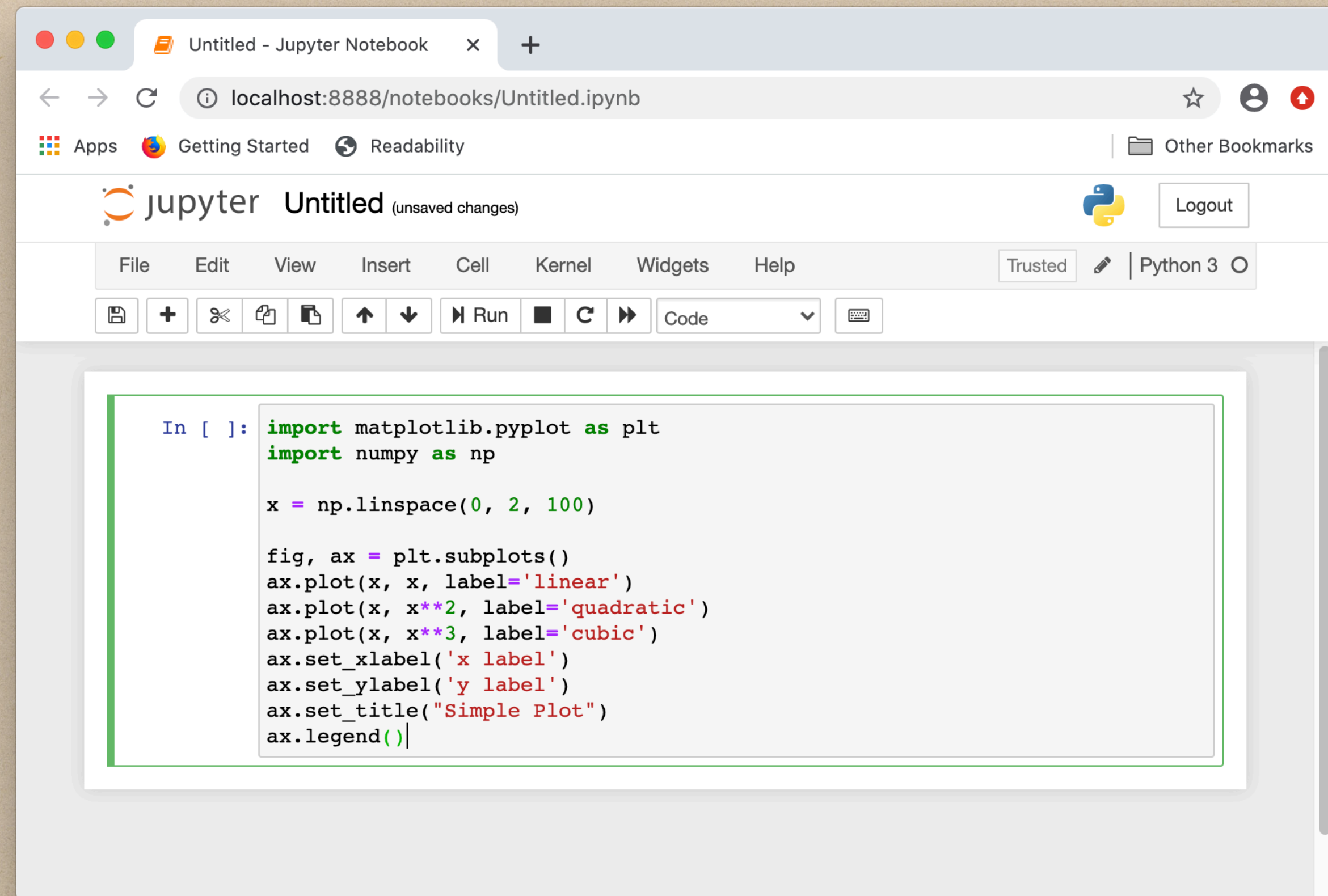
# Charts and Plots

Python uses matplotlib to create graphs. In a code cell:

```
import matplotlib.pyplot as plt  
import numpy as np
```

```
x = np.linspace(0, 2, 100)
```

```
fig, ax = plt.subplots()  
ax.plot(x, x, label='linear')  
ax.plot(x, x**2, label='quadratic')  
ax.plot(x, x**3, label='cubic')  
ax.set_xlabel('x label')  
ax.set_ylabel('y label')  
ax.set_title("Simple Plot")  
ax.legend()
```



The screenshot shows a Jupyter Notebook interface in a browser window. The browser address bar shows 'localhost:8888/notebooks/Untitled.ipynb'. The notebook title is 'Untitled (unsaved changes)'. The code cell contains the following Python code:

```
In [ ]: import matplotlib.pyplot as plt  
import numpy as np  
  
x = np.linspace(0, 2, 100)  
  
fig, ax = plt.subplots()  
ax.plot(x, x, label='linear')  
ax.plot(x, x**2, label='quadratic')  
ax.plot(x, x**3, label='cubic')  
ax.set_xlabel('x label')  
ax.set_ylabel('y label')  
ax.set_title("Simple Plot")  
ax.legend()
```

Untitled - Jupyter Notebook x +

localhost:8888/notebooks/Untitled.ipynb

Apps Getting Started Readability Other Bookmarks

jupyter Untitled (unsaved changes) Python 3 Logout

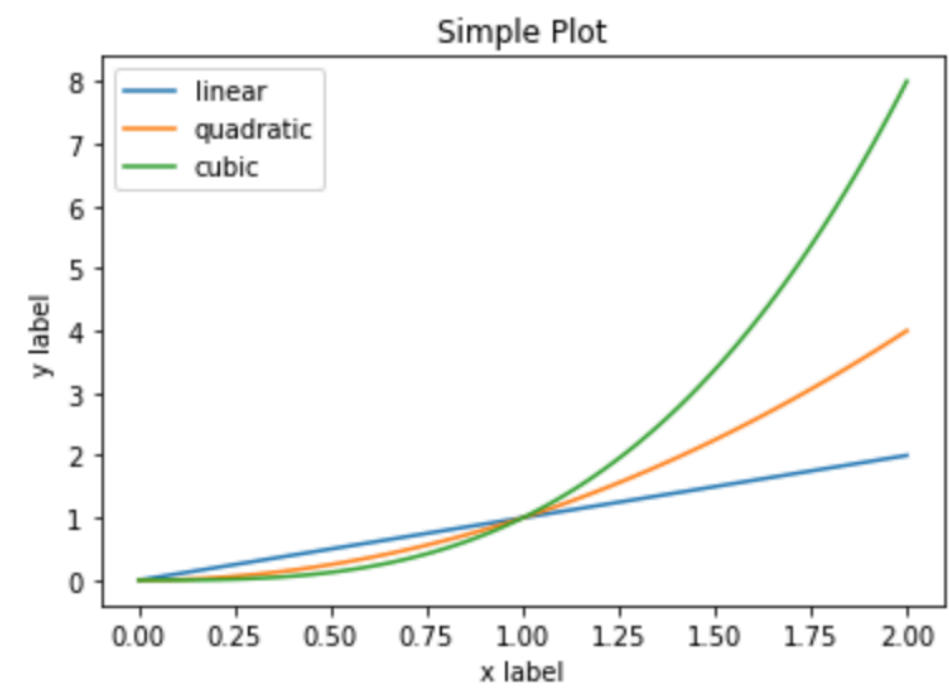
File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

```
In [4]: import matplotlib.pyplot as plt
import numpy as np

x = np.linspace(0, 2, 100)

fig, ax = plt.subplots()
ax.plot(x, x, label='linear')
ax.plot(x, x**2, label='quadratic')
ax.plot(x, x**3, label='cubic')
ax.set_xlabel('x label')
ax.set_ylabel('y label')
ax.set_title("Simple Plot")
ax.legend()
```

Out[4]: <matplotlib.legend.Legend at 0x1078373d0>



```
In [ ]: |
```

An interesting example of using Python for plotting live coronavirus data. This example demonstrates reading data off a web page, creating a table and generating a chart.

<https://www.geeksforgeeks.org/corona-virus-live-updates-for-india-using-python/?ref=leftbar-rightbar>

More information: <https://matplotlib.org/index.html>

# Magic Commands

For running Python code that is outside a notebook a magic command `%run` can be used. This example will use Python Mesa. This can be downloaded from <https://github.com/projectmesa/mesa>. Once that is downloaded, go to `examples/bank_reserves` and get the requirements

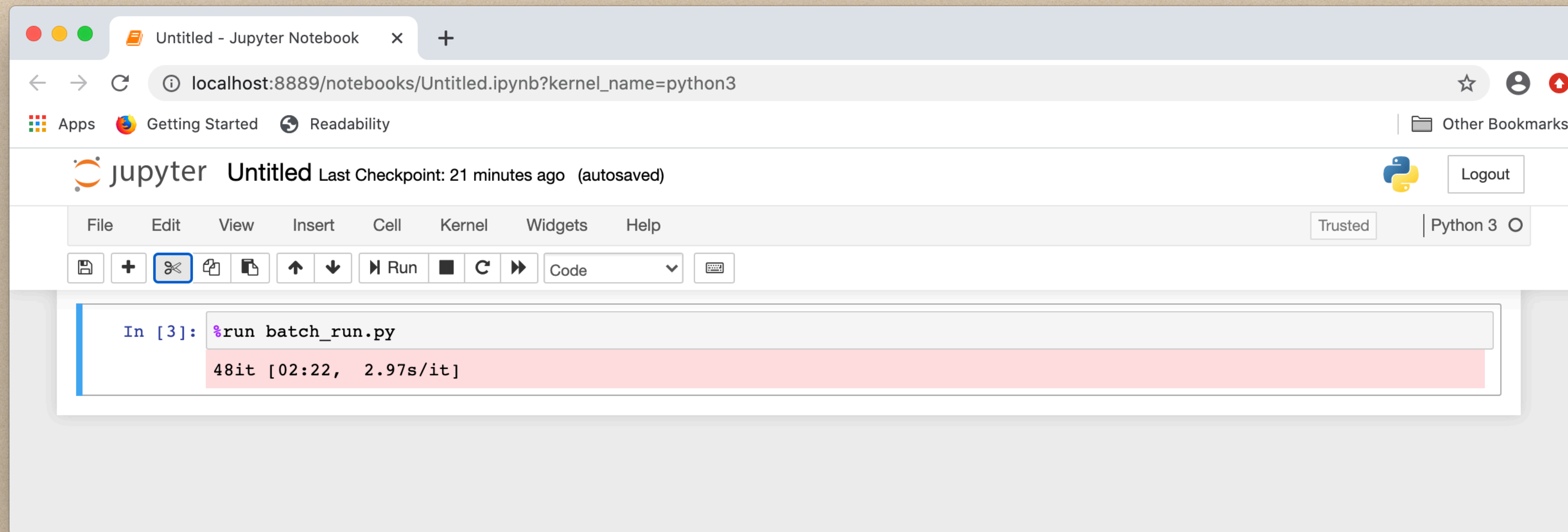
- `pip install -r requirements.txt`

An error about `intertools` may appear but it does not appear to be a problem. Then start up Jupyter Notebooks.

Python Mesa is an agent based model framework. This example has a `batch_run.py` file that can be used to run the model in batch mode instead of interactive mode. In a code cell, type

- `%run batch_run.py`

Click on run and the progress is shown. The output shows up in the BankReservesModel\_Step\_Data.csv file.



The screenshot displays a Jupyter Notebook interface in a web browser. The browser's address bar shows the URL `localhost:8889/notebooks/Untitled.ipynb?kernel_name=python3`. The notebook's title bar indicates it is an "Untitled" notebook with a last checkpoint 21 minutes ago. The interface includes a menu bar with options like File, Edit, View, Insert, Cell, Kernel, Widgets, and Help. Below the menu is a toolbar with icons for file operations and execution. The main area contains a code cell with the following content:

```
In [3]: %run batch_run.py
48it [02:22, 2.97s/it]
```

Magic commands are beyond the scope of this presentation but you can get more information on the web or using the following magic commands

`%magic` prints a complete guide to magic commands

`%quickref` prints a brief guide to magic commands

`%lsmagic` lists all the magic commands

If you want more information about a magic command follow it with a question mark

`%run?`



## NetLogo Integration

There are a few ways for integrating NetLogo into Jupyter Notebooks. This is based on NL4Py (<https://arxiv.org/pdf/1808.03292.pdf>). This connects from Python to Java. The connection does not seem to be closed properly and only one connection is allowed so, at least on some operating systems, rebooting may be required to run the model again. The Java development kit (JDK) 8 or higher is required. For Linux (Debian distributions)

- `sudo apt install default-jdk`

If you do not have privilege to this then install locally. Go to [jdk.java.net](http://jdk.java.net) and download the JDK. Uncompress it in a local directory and then set `JAVA_HOME` to the path of the JDK install and prepend `$JAVA_HOME/bin` to the path

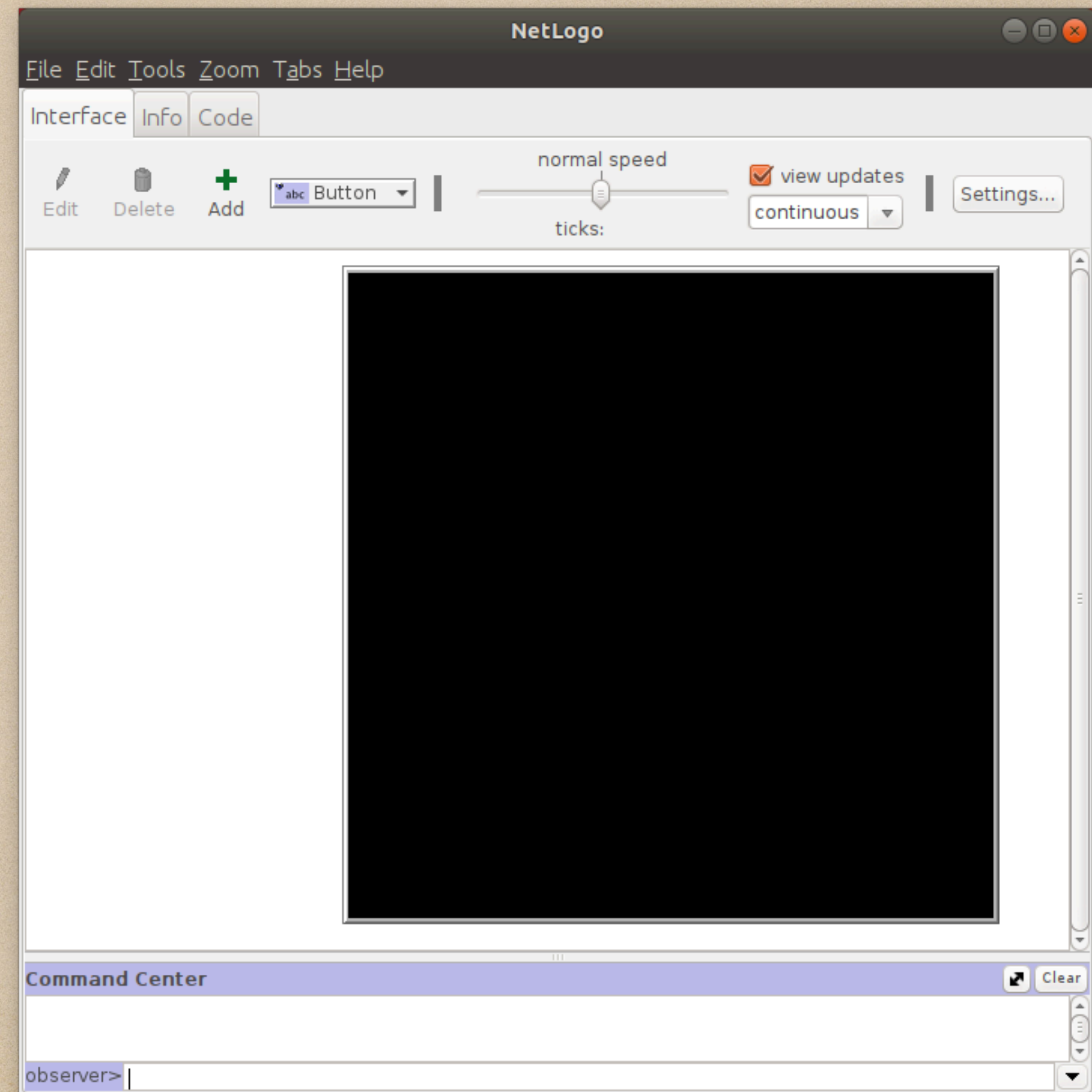
```
export JAVA_HOME=~/.JDK
export PATH="$JAVA_HOME/bin:$PATH"
```

Install NL4Py

- `pip install NL4Py`

Start step by step through a Python shell to get a better understanding what each command does. The first part is to start the server and launch NetLogo. The path to NetLogo will need to be changed for your computer.

```
Terminal
File Edit View Search Terminal Help
trobey@ubuntu ~ $ python3
Python 3.6.9 (default, Apr 18 2020, 01:56:04)
[GCC 8.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import nl4py
>>> nl4py.startServer('/home/trobery/NetLogo')
>>> app = nl4py.NetLogoApp()
Gtk-Message: 12:26:16.533: Failed to load module "canberra-gtk-module"
>>>
```



Python launches NetLogo.

Now, load the Wolf Sheep model. The file with this model should be in the current directory.

```
Terminal
File Edit View Search Terminal Help
trobey@ubuntu ~ $ python3
Python 3.6.9 (default, Apr 18 2020, 01:56:04)
[GCC 8.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import nl4py
>>> nl4py.startServer('/home/trobey/NetLogo')
>>> app = nl4py.NetLogoApp()
Gtk-Message: 12:26:16.533: Failed to load module "canberra-gtk-module"
>>> app.openModel("Wolf Sheep Predation.nlogo")
>>>
```

Wolf Sheep Predation - NetLogo (/home/trobey)

File Edit Tools Zoom Tabs Help

Interface Info Code

Edit Delete Add abc Button | normal speed | view updates | Settings...  
ticks: on ticks

model-version  
sheep-wolves

initial-number-sheep 100 initial-number-wolv... 50  
grass-regrowth-time 30

setup go

Sheep settings Wolf settings

sheep-gain-from-food 4 wolf-gain-from-food 20  
sheep-reproduce 4 % wolf-reproduce 5 %

On show-energy?  
Off

sheep wolves grass  
0 0 N/A

populations

pop. 100 0  
time 0 100

sheep wolves grass / 4

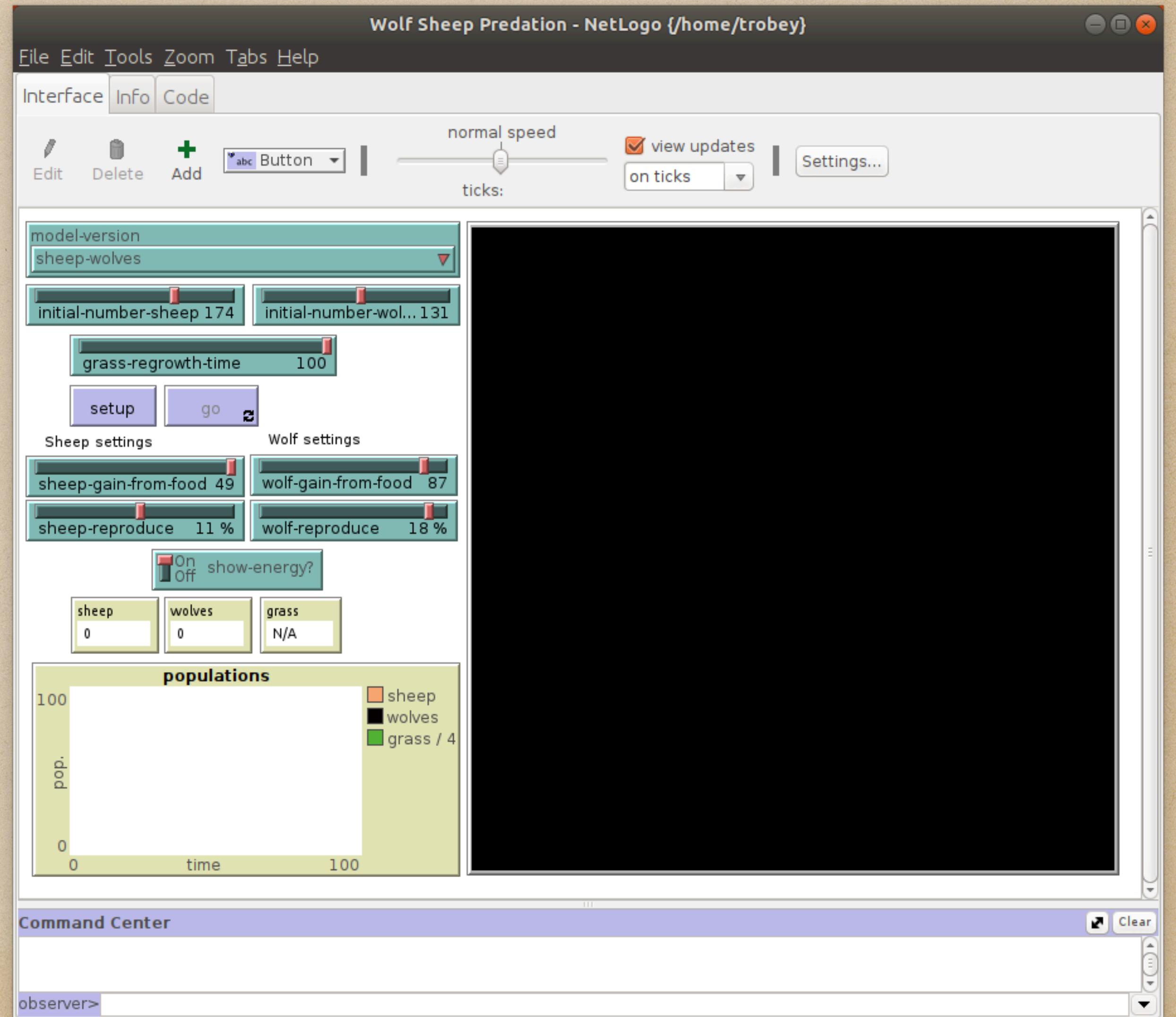
Command Center Clear

observer>

Python loads the requested NetLogo model.

Python can retrieve the available parameters and their ranges and can set the parameter values. Here the parameters are set to random values.

```
Terminal
File Edit View Search Terminal Help
>>> app.getParamNames()
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by py4j.reflection.MethodInvoker$1 (file:/home/trobey/.local/lib/python3.6/site-packages/nl4py/nl4pyServer/NetLogo6.1/py4j0.10.6.jar) to method java.util.ArrayList$Itr.next()
WARNING: Please consider reporting this to the maintainers of py4j.reflection.MethodInvoker$1
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations
WARNING: All illegal access operations will be denied in a future release
['initial-number-sheep', 'sheep-gain-from-food', 'sheep-reproduce', 'initial-number-wolves', 'wolf-gain-from-food', 'wolf-reproduce', 'grass-regrowth-time', 'show-energy?', 'model-version']
>>> app.getParamRanges()
[[0.0, 1.0, 250.0], [0.0, 1.0, 50.0], [1.0, 1.0, 20.0], [0.0, 1.0, 250.0], [0.0, 1.0, 100.0], [0.0, 1.0, 20.0], [0.0, 1.0, 100.0], [True, False], ['sheep-wolves', 'sheep-wolves-grass']]
>>> app.setParamsRandom()
NetLogo command: set initial-number-sheep 174.0
NetLogo command: set sheep-gain-from-food 49.0
NetLogo command: set sheep-reproduce 11.0
NetLogo command: set initial-number-wolves 131.0
NetLogo command: set wolf-gain-from-food 87.0
NetLogo command: set wolf-reproduce 18.0
NetLogo command: set grass-regrowth-time 100.0
NetLogo command: set show-energy? True
NetLogo command: set model-version "sheep-wolves"
>>>
```

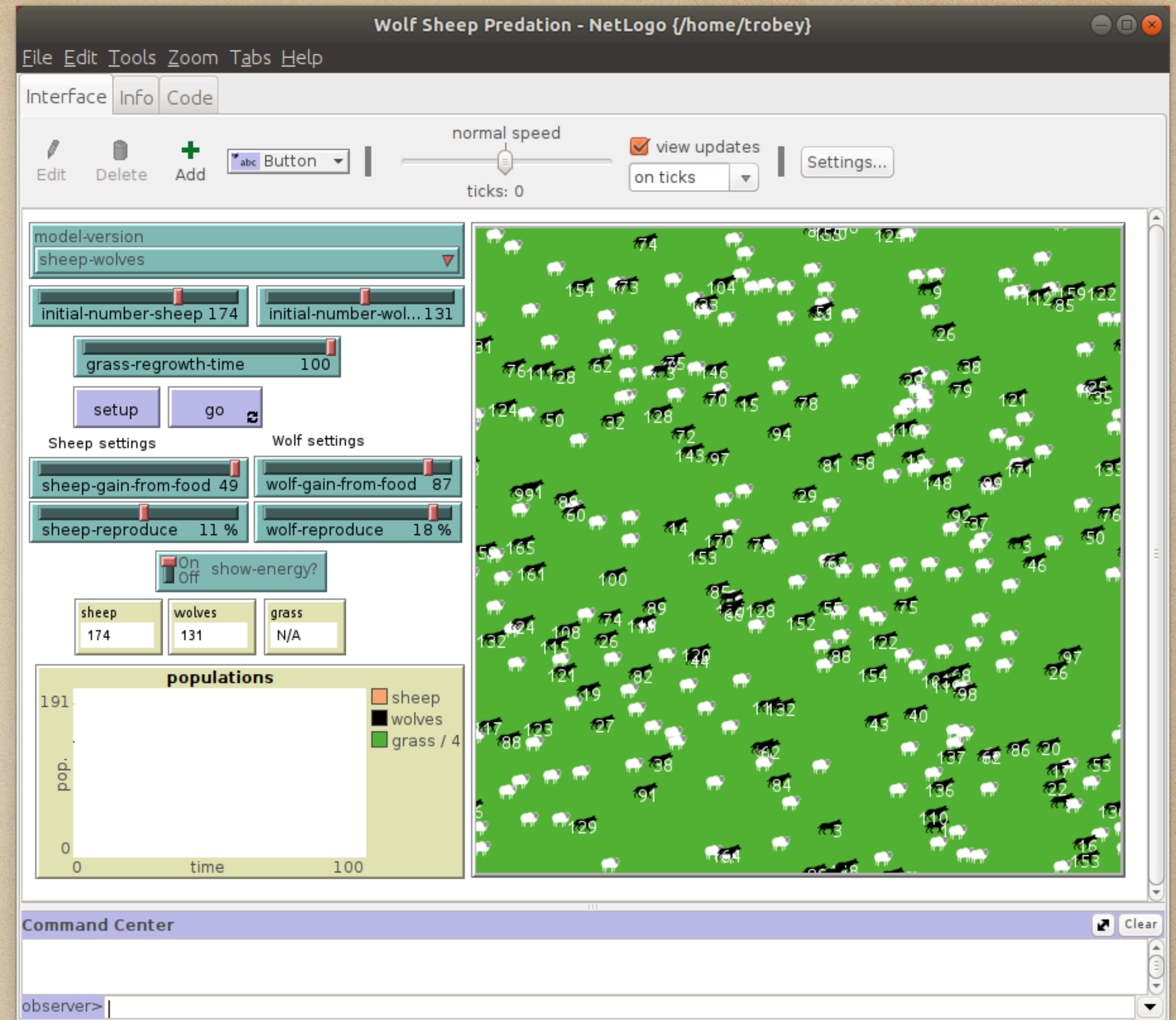


Python changes the parameters.

Now, set up the model.

```
Terminal
File Edit View Search Terminal Help
WARNING: Please consider reporting this to the maintainers of py4j.reflection.Me
thodInvoker$1
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflect
ive access operations
WARNING: All illegal access operations will be denied in a future release
['initial-number-sheep', 'sheep-gain-from-food', 'sheep-reproduce', 'initial-num
ber-wolves', 'wolf-gain-from-food', 'wolf-reproduce', 'grass-regrowth-time', 'sh
ow-energy?', 'model-version']
>>> app.getParamRanges()
[[0.0, 1.0, 250.0], [0.0, 1.0, 50.0], [1.0, 1.0, 20.0], [0.0, 1.0, 250.0], [0.0,
1.0, 100.0], [0.0, 1.0, 20.0], [0.0, 1.0, 100.0], [True, False], ['sheep-wolves
', 'sheep-wolves-grass']]
>>> app.setParamsRandom()
NetLogo command: set initial-number-sheep 174.0
NetLogo command: set sheep-gain-from-food 49.0
NetLogo command: set sheep-reproduce 11.0
NetLogo command: set initial-number-wolves 131.0
NetLogo command: set wolf-gain-from-food 87.0
NetLogo command: set wolf-reproduce 18.0
NetLogo command: set grass-regrowth-time 100.0
NetLogo command: set show-energy? True
NetLogo command: set model-version "sheep-wolves"
>>> app.command('setup')
>>>
```

The wolves and sheep appear in the NetLogo model.



Time to run the model.

```
Terminal
File Edit View Search Terminal Help
thodInvoker$1
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflect
ive access operations
WARNING: All illegal access operations will be denied in a future release
['initial-number-sheep', 'sheep-gain-from-food', 'sheep-reproduce', 'initial-num
ber-wolves', 'wolf-gain-from-food', 'wolf-reproduce', 'grass-regrowth-time', 'sh
ow-energy?', 'model-version']
>>> app.getParamRanges()
[[0.0, 1.0, 250.0], [0.0, 1.0, 50.0], [1.0, 1.0, 20.0], [0.0, 1.0, 250.0], [0.0,
1.0, 100.0], [0.0, 1.0, 20.0], [0.0, 1.0, 100.0], [True, False], ['sheep-wolves
', 'sheep-wolves-grass']]
>>> app.setParamsRandom()
NetLogo command: set initial-number-sheep 174.0
NetLogo command: set sheep-gain-from-food 49.0
NetLogo command: set sheep-reproduce 11.0
NetLogo command: set initial-number-wolves 131.0
NetLogo command: set wolf-gain-from-food 87.0
NetLogo command: set wolf-reproduce 18.0
NetLogo command: set grass-regrowth-time 100.0
NetLogo command: set show-energy? True
NetLogo command: set model-version "sheep-wolves"
>>> app.command('setup')
>>> app.command('repeat 1000 [go]')
>>>
```

The model is run and no sheep or wolves are left.

Wolf Sheep Predation - NetLogo (/home/trobey)

File Edit Tools Zoom Tabs Help

Interface Info Code

Edit Delete Add abc Button | normal speed | view updates | Settings...  
ticks: 80

model-version  
sheep-wolves

initial-number-sheep 174 initial-number-wol... 131

grass-regrowth-time 100

setup go

Sheep settings Wolf settings

sheep-gain-from-food 49 wolf-gain-from-food 87

sheep-reproduce 11 % wolf-reproduce 18 %

On Off show-energy?

sheep	wolves	grass
0	0	N/A

populations

pop. 2610

time 0 100

sheep wolves grass / 4

Command Center Clear

observer>

Clean up the model and server. This closes the NetLogo window and is supposed to clean up the connection between Python and Java.

```
Terminal
File Edit View Search Terminal Help
ow-energy?', 'model-version']
>>> app.getParamRanges()
[[0.0, 1.0, 250.0], [0.0, 1.0, 50.0], [1.0, 1.0, 20.0], [0.0, 1.0, 250.0], [0.0,
1.0, 100.0], [0.0, 1.0, 20.0], [0.0, 1.0, 100.0], [True, False], ['sheep-wolves
', 'sheep-wolves-grass']]
>>> app.setParamsRandom()
NetLogo command: set initial-number-sheep 174.0
NetLogo command: set sheep-gain-from-food 49.0
NetLogo command: set sheep-reproduce 11.0
NetLogo command: set initial-number-wolves 131.0
NetLogo command: set wolf-gain-from-food 87.0
NetLogo command: set wolf-reproduce 18.0
NetLogo command: set grass-regrowth-time 100.0
NetLogo command: set show-energy? True
NetLogo command: set model-version "sheep-wolves"
>>> app.command('setup')
>>> app.command('repeat 1000 [go]')
>>> app.closeModel()
>>> app.stopServer()
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
AttributeError: 'NetLogoGUI' object has no attribute 'stopServer'
>>> nl4py.stopServer()
>>>
```

```
Terminal
File Edit View Search Terminal Help
', 'sheep-wolves-grass']]
>>> app.setParamsRandom()
NetLogo command: set initial-number-sheep 174.0
NetLogo command: set sheep-gain-from-food 49.0
NetLogo command: set sheep-reproduce 11.0
NetLogo command: set initial-number-wolves 131.0
NetLogo command: set wolf-gain-from-food 87.0
NetLogo command: set wolf-reproduce 18.0
NetLogo command: set grass-regrowth-time 100.0
NetLogo command: set show-energy? True
NetLogo command: set model-version "sheep-wolves"
>>> app.command('setup')
>>> app.command('repeat 1000 [go]')
>>> app.closeModel()
>>> app.stopServer()
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
AttributeError: 'NetLogoGUI' object has no attribute 'stopServer'
>>> nl4py.stopServer()
>>> nl4py.startServer('/home/trobey/NetLogo')
Error: Could not find or load main class nl4py.server.NetLogoControllerServer
Caused by: java.lang.ClassNotFoundException: nl4py.server.NetLogoControllerServe
r
>>>
```

Trying to restart the server produces an error so a reboot is required to completely clean up from the run.



In our Jupyter Notebook, the code will be entered in four cells. First, load the model and set parameters.

```
netlogo - Jupyter Notebook - Mozilla Firefox
localhost:8888/notebooks/netlogo.ipynb

jupyter netlogo (unsaved changes)
File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

In [2]: import nl4py
nl4py.startServer('/home/trobey/NetLogo')
app = nl4py.NetLogoApp()
app.openModel("Wolf Sheep Predation.nlogo")
print(app.getParamNames())
print(app.getParamRanges())
app.setParamsRandom()

['initial-number-sheep', 'sheep-gain-from-food', 'sheep-reproduce', 'i
nitial-number-wolves', 'wolf-gain-from-food', 'wolf-reproduce', 'grass
-regrowth-time', 'show-energy?', 'model-version']
[[0.0, 1.0, 250.0], [0.0, 1.0, 50.0], [1.0, 1.0, 20.0], [0.0, 1.0, 25
0.0], [0.0, 1.0, 100.0], [0.0, 1.0, 20.0], [0.0, 1.0, 100.0], [True, F
alse], ['sheep-wolves', 'sheep-wolves-grass']]
NetLogo command: set initial-number-sheep 245.0
NetLogo command: set sheep-gain-from-food 37.0
NetLogo command: set sheep-reproduce 13.0
NetLogo command: set initial-number-wolves 141.0
NetLogo command: set wolf-gain-from-food 90.0
NetLogo command: set wolf-reproduce 14.0
NetLogo command: set grass-regrowth-time 51.0
NetLogo command: set show-energy? False
NetLogo command: set model-version "sheep-wolves"

In [3]: app.command('setup')

In [ ]:
```

Wolf Sheep Predation - NetLogo (/home/trobey)

File Edit Tools Zoom Tabs Help

Interface Info Code

Edit Delete Add abc Button normal speed view updates on ticks Settings... ticks: 0

model-version sheep-wolves

initial-number-sheep 245 initial-number-wol... 141

grass-regrowth-time 51

setup go

Sheep settings Wolf settings

sheep-gain-from-food 37 wolf-gain-from-food 90

sheep-reproduce 13 % wolf-reproduce 14 %

On Off show-energy?

sheep	wolves	grass
245	141	N/A

populations

Command Center

observer>

# Set up the model.

netlogo - Jupyter Notebook - Mozilla Firefox

netlogo - Jupyter Noteb x +

localhost:8888/notebooks/netlogo.ipynb

jupyter netlogo (unsaved changes) Python 3

```
In [2]: import nl4py
nl4py.startServer('/home/trobey/NetLogo')
app = nl4py.NetLogoApp()
app.openModel("Wolf Sheep Predation.nlogo")
print(app.getParamNames())
print(app.getParamRanges())
app.setParamsRandom()

['initial-number-sheep', 'sheep-gain-from-food', 'sheep-reproduce', 'initial-number-wolves', 'wolf-gain-from-food', 'wolf-reproduce', 'grass-regrowth-time', 'show-energy?', 'model-version']
[[0.0, 1.0, 250.0], [0.0, 1.0, 50.0], [1.0, 1.0, 20.0], [0.0, 1.0, 250.0], [0.0, 1.0, 100.0], [0.0, 1.0, 20.0], [0.0, 1.0, 100.0], [True, False], ['sheep-wolves', 'sheep-wolves-grass']]
NetLogo command: set initial-number-sheep 245.0
NetLogo command: set sheep-gain-from-food 37.0
NetLogo command: set sheep-reproduce 13.0
NetLogo command: set initial-number-wolves 141.0
NetLogo command: set wolf-gain-from-food 90.0
NetLogo command: set wolf-reproduce 14.0
NetLogo command: set grass-regrowth-time 51.0
NetLogo command: set show-energy? False
NetLogo command: set model-version "sheep-wolves"
```

```
In [3]: app.command('setup')
```

```
In [ ]:
```

Wolf Sheep Predation - NetLogo (/home/trobey)

File Edit Tools Zoom Tabs Help

Interface Info Code

normal speed view updates on ticks Settings...

ticks: 0

model-version: sheep-wolves

initial-number-sheep 245 initial-number-wolves 141

grass-regrowth-time 51

setup go

Sheep settings Wolf settings

sheep-gain-from-food 37 wolf-gain-from-food 90

sheep-reproduce 13 % wolf-reproduce 14 %

show-energy? On Off

sheep	wolves	grass
245	141	N/A

populations

Command Center

observer> |

# Run the model.

```
app = nlogo.NetLogoApp()
app.openModel("Wolf Sheep Predation.nlogo")
print(app.getParamNames())
print(app.getParamRanges())
app.setParamsRandom()

['initial-number-sheep', 'sheep-gain-from-food', 'sheep-reproduce', 'initial-number-wolves', 'wolf-gain-from-food', 'wolf-reproduce', 'grass-regrowth-time', 'show-energy?', 'model-version']
[[0.0, 1.0, 250.0], [0.0, 1.0, 50.0], [1.0, 1.0, 20.0], [0.0, 1.0, 250.0], [0.0, 1.0, 100.0], [0.0, 1.0, 20.0], [0.0, 1.0, 100.0], [True, False], ['sheep-wolves', 'sheep-wolves-grass']]
NetLogo command: set initial-number-sheep 245.0
NetLogo command: set sheep-gain-from-food 37.0
NetLogo command: set sheep-reproduce 13.0
NetLogo command: set initial-number-wolves 141.0
NetLogo command: set wolf-gain-from-food 90.0
NetLogo command: set wolf-reproduce 14.0
NetLogo command: set grass-regrowth-time 51.0
NetLogo command: set show-energy? False
NetLogo command: set model-version "sheep-wolves"

In [3]: app.command('setup')

In [4]: app.command('repeat 1000 [go]')

In [ ]: |
```

Wolf Sheep Predation - NetLogo {/home/trobey}

File Edit Tools Zoom Tabs Help

Interface Info Code

Edit Delete Add  | normal speed |  view updates | Settings...  
ticks: 120 on ticks

model-version  
sheep-wolves

initial-number-sheep 245 initial-number-wolves 141

grass-regrowth-time 51

setup go

Sheep settings Wolf settings

sheep-gain-from-food 37 wolf-gain-from-food 90

sheep-reproduce 13% wolf-reproduce 14%

On show-energy?  Off

sheep 0 wolves 0 grass N/A

populations

pop. 3850 0

time 0 126

sheep wolves grass / 4

Command Center Clear

observer>

Finally, cleanup the model.

```
print(app.getParameters())
app.setParamsRandom()

['initial-number-sheep', 'sheep-gain-from-food', 'sheep-reproduce', 'initial-number-wolves', 'wolf-gain-from-food', 'wolf-reproduce', 'grass-regrowth-time', 'show-energy?', 'model-version']
[[0.0, 1.0, 250.0], [0.0, 1.0, 50.0], [1.0, 1.0, 20.0], [0.0, 1.0, 250.0], [0.0, 1.0, 100.0], [0.0, 1.0, 20.0], [0.0, 1.0, 100.0], [True, False], ['sheep-wolves', 'sheep-wolves-grass']]
NetLogo command: set initial-number-sheep 245.0
NetLogo command: set sheep-gain-from-food 37.0
NetLogo command: set sheep-reproduce 13.0
NetLogo command: set initial-number-wolves 141.0
NetLogo command: set wolf-gain-from-food 90.0
NetLogo command: set wolf-reproduce 14.0
NetLogo command: set grass-regrowth-time 51.0
NetLogo command: set show-energy? False
NetLogo command: set model-version "sheep-wolves"

In [3]: app.command('setup')

In [4]: app.command('repeat 1000 [go]')

In [5]: app.closeModel()
nl4py.stopServer()

In [ ]:
```

This can also be run in headless mode where the GUI is not displayed.

```
netlogo - Jupyter Notebook - Mozilla Firefox
localhost:8888/notebooks/netlogo.i
jupyter netlogo (unsaved changes)
File Edit View Insert Cell Kernel Widgets Help Trusted Python 3
Run Code

In [1]: import nl4py
nl4py.startServer('/home/trobey/NetLogo')
app = nl4py.newNetLogoHeadlessWorkspace()
app.openModel("Wolf Sheep Predation.nlogo")
print(app.getParamNames())
print(app.getParamRanges())
app.setParamsRandom()
app.command('setup')
app.command('repeat 1000 [go]')

['initial-number-sheep', 'sheep-gain-from-food', 'sheep-reproduce', 'i
nitial-number-wolves', 'wolf-gain-from-food', 'wolf-reproduce', 'grass
-regrowth-time', 'show-energy?', 'model-version']
[[0.0, 1.0, 250.0], [0.0, 1.0, 50.0], [1.0, 1.0, 20.0], [0.0, 1.0, 25
0.0], [0.0, 1.0, 100.0], [0.0, 1.0, 20.0], [0.0, 1.0, 100.0], [True, F
alse], ['sheep-wolves', 'sheep-wolves-grass']]
NetLogo command: set initial-number-sheep 2.0
NetLogo command: set sheep-gain-from-food 28.0
NetLogo command: set sheep-reproduce 16.0
NetLogo command: set initial-number-wolves 226.0
NetLogo command: set wolf-gain-from-food 87.0
NetLogo command: set wolf-reproduce 18.0
NetLogo command: set grass-regrowth-time 78.0
NetLogo command: set show-energy? True
NetLogo command: set model-version "sheep-wolves-grass"

In [ ]:
```

```
netlogo - Jupyter Notebook - Mozilla Firefox
localhost:8888/notebooks/netlogo.i
jupyter netlogo (unsaved changes)
File Edit View Insert Cell Kernel Widgets Help Trusted Python 3
Run Code

In [1]: import nl4py
nl4py.startServer('/home/trobey/NetLogo')
app = nl4py.newNetLogoHeadlessWorkspace()
app.openModel("Wolf Sheep Predation.nlogo")
print(app.getParamNames())
print(app.getParamRanges())
app.setParamsRandom()
app.command('setup')
app.command('repeat 1000 [go]')

['initial-number-sheep', 'sheep-gain-from-food', 'sheep-reproduce', 'i
nitial-number-wolves', 'wolf-gain-from-food', 'wolf-reproduce', 'grass
-regrowth-time', 'show-energy?', 'model-version']
[[0.0, 1.0, 250.0], [0.0, 1.0, 50.0], [1.0, 1.0, 20.0], [0.0, 1.0, 25
0.0], [0.0, 1.0, 100.0], [0.0, 1.0, 20.0], [0.0, 1.0, 100.0], [True, F
alse], ['sheep-wolves', 'sheep-wolves-grass']]
NetLogo command: set initial-number-sheep 2.0
NetLogo command: set sheep-gain-from-food 28.0
NetLogo command: set sheep-reproduce 16.0
NetLogo command: set initial-number-wolves 226.0
NetLogo command: set wolf-gain-from-food 87.0
NetLogo command: set wolf-reproduce 18.0
NetLogo command: set grass-regrowth-time 78.0
NetLogo command: set show-energy? True
NetLogo command: set model-version "sheep-wolves-grass"

In [2]: app.closeModel()
nl4py.stopServer()

In [ ]:
```

An example of setting an individual parameter is

- `app.command('set model-version "sheep-wolves-grass"')`

Extracting data has not been covered in the example but for both headless and GUI modes, data can be recorded using a command like

- `reporters_array = ["ticks", 'count sheep', 'count wolves']`
- `app.scheduleReportersAndRun(reporters_array, startAtTick = 0, intervalTicks = 1, stopAtTick = -1, goCommand = "go")`

Then access the results with

- `app.getScheduledReporterResults()`

This function will return an empty array until the model has finished. This is an example of how this should be used.

- `newResults = []`
- `while(len(newResults) == 0):`
- `time.sleep(2)`
- `newResults = app.getScheduledReporterResults()`
- `if len(newResults) > 0:`
- `...`

Try modifying the code to extract results from the model.

## Appendix 1: Installing Python 3

It is easy to find instructions by searching on the web. Some are more complicated than necessary but if you get stuck just look around for instructions. Usually typing `python` gets the default version and `python3` (or `pip3`) runs just the Python 3 version in case Python 2 is also installed. Python 2 is no longer supported. Note that the long dash `—` is typed as two short dashes.

### Windows

- Go to <https://www.python.org/downloads/windows>. Find the 32 bit or 64 bit executable installer for Python 3 and download it. There is a bug in Python 3.8 for connecting to browsers so choose 3.7.
- Run the installer. Make sure to check to add Python to the path. You may get a message to disable the path length limit. Select this if it appears.
- Type `python —version` to verify install.
- Launch from a terminal by typing `python`.

PIP is included with Python 3.4+. Check by typing `pip —version`.



## Linux

Python comes already installed on Linux. Although Python 2 is no longer supported check by typing

- `python --version`
- `python3 --version`

If the default version is Python 2 then you may need to use `python3` and `pip3` instead of just `python` and `pip`.

## Mac

Python is installed on a Mac but this is for the system to use. Leave this alone to avoid breaking things. Homebrew will be used. It is a package manager for Macs. First, Xcode will need to be installed. It can be found in the app store. It is a large download. Once Xcode is installed, then open up a terminal

- `/usr/bin/ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"`

Then install Python

- `brew install python3`

Check to verify installation

- `python --version`

## Appendix II: Installing Jupyter Notebooks

Start by installing Python 3 and PIP 3 (see Appendix I).

For Windows and Mac in a terminal run

- `python3 -m pip install --upgrade pip`
- `python3 -m pip install jupyter`

For Linux (Debian derivatives) instead use

- `sudo apt install jupyter-notebook`