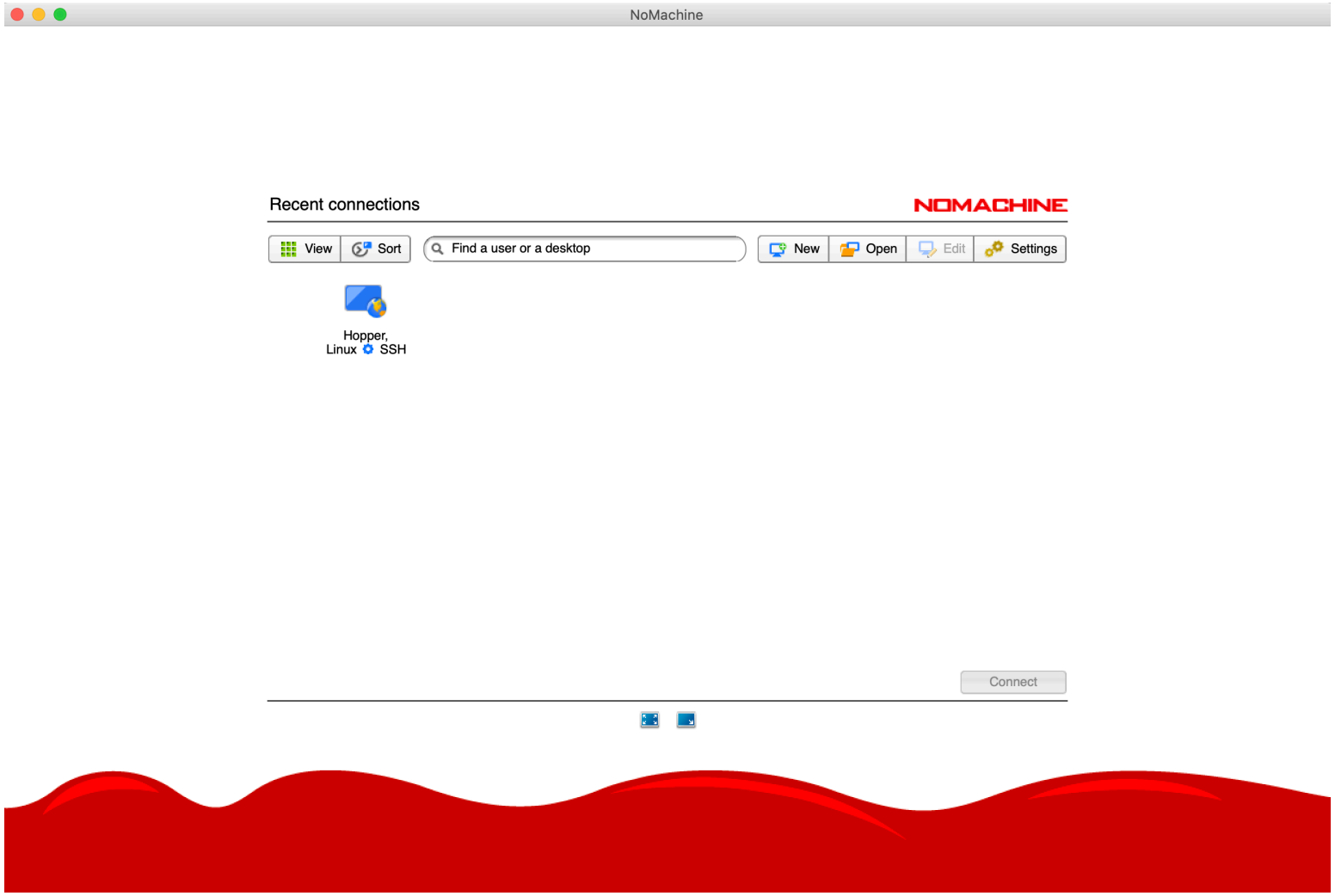
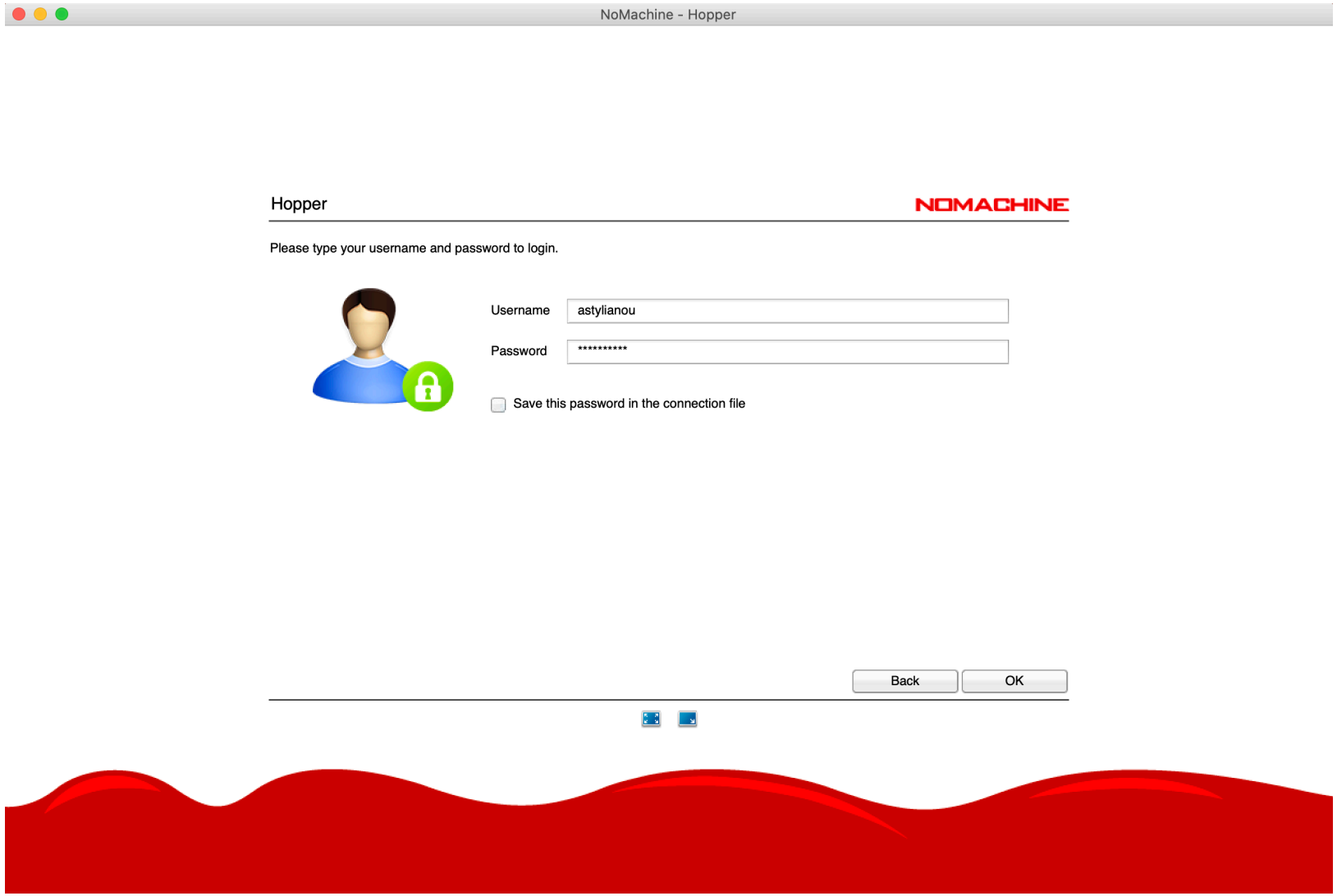
CS1070: Taming Big Data

Lab Time

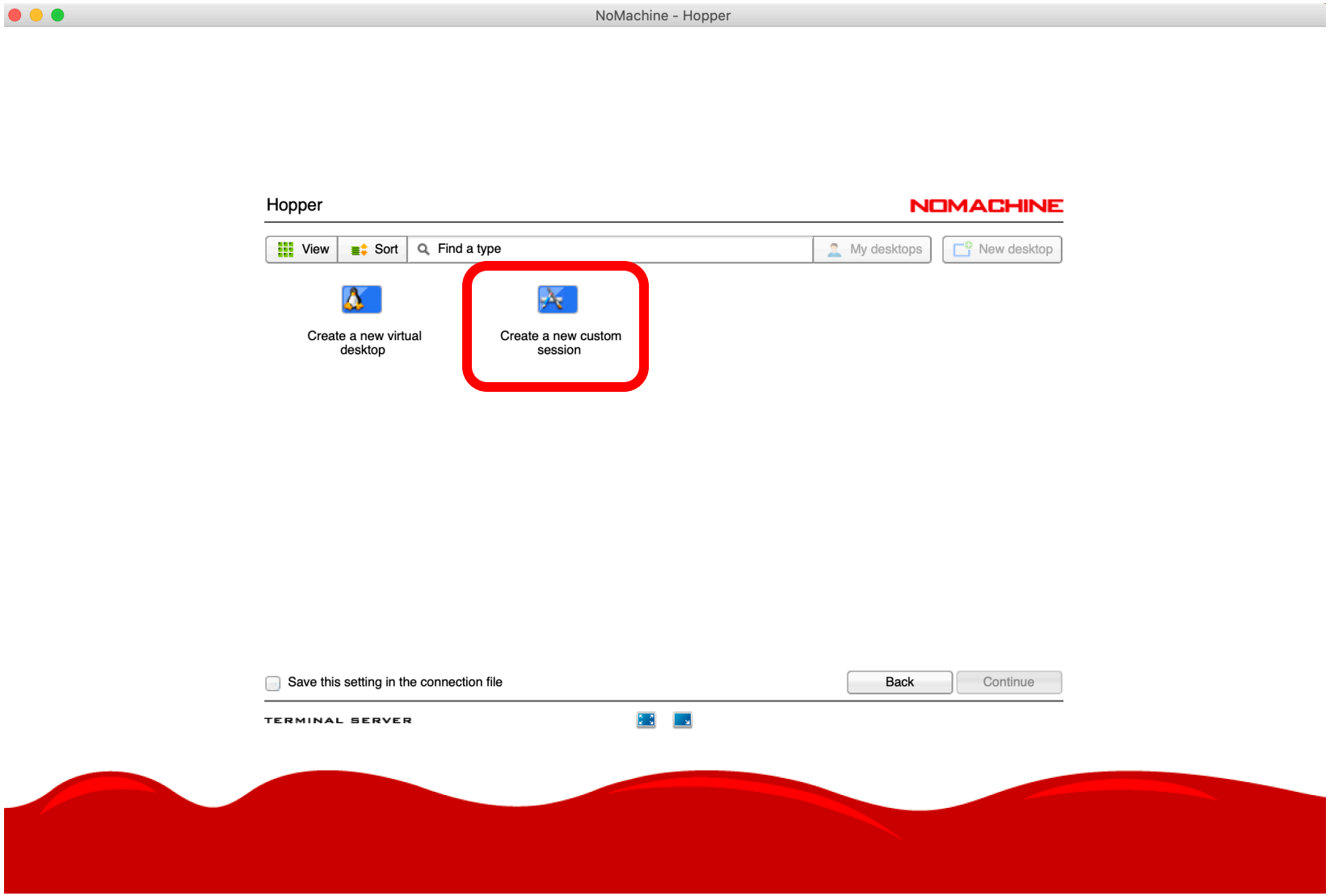
Log on to Hopper



Log on to Hopper



Log on to Hopper



Set Chrome as Default Browser

Run Jupyter Notebooks

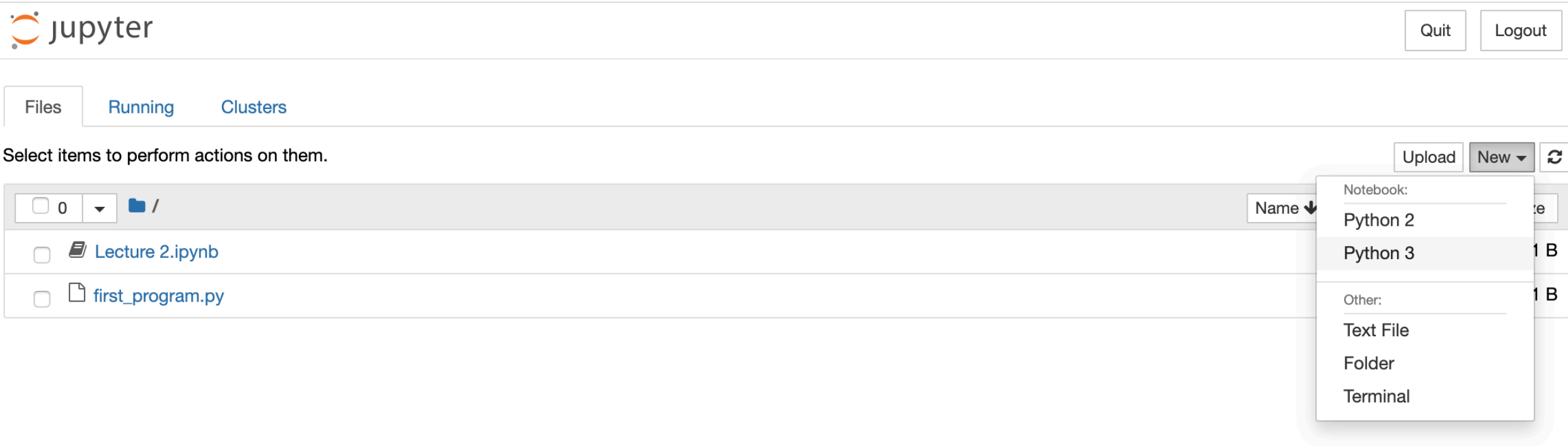
- To start a new notebook, first navigate to your home directory in Terminal (what was the command?)

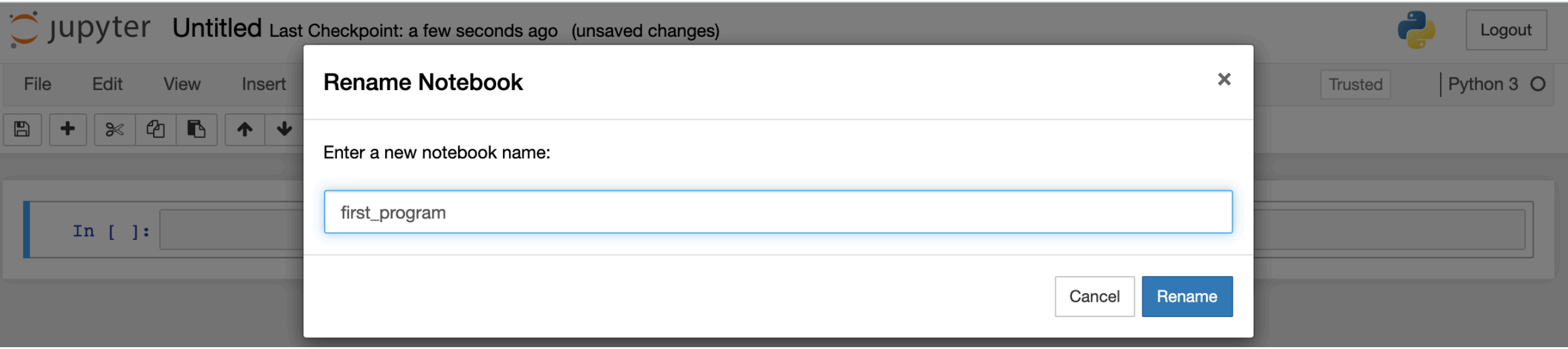
- Then, run:

```
jupyter notebook
```

```
code — jupyter-notebook — 101×24
(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=8fdeb7f96ec6077694508b60ca7d2402e3a16b2e66
d977b
[I 13:20:10.489 NotebookApp]  or http://127.0.0.1:8889/?token=8fdeb7f96ec6077694508b60ca7d2402e3a16b
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=8fdeb7f96ec6077694508b60ca7d2402e3a16b2e66d977b
     or http://127.0.0.1:8889/?token=8fdeb7f96ec6077694508b60ca7d2402e3a16b2e66d977b
```


jupyter
Quit
Logout
Files
Running
Clusters
Select items to perform actions on them.
Upload
New
0
/
Name
Lecture 2.ipynb
first_program.py
Notebook:
Python 2
Python 3
Other:
Text File
Folder
Terminal


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

jupyter first_program Last Checkpoint: a minute ago (unsaved changes)

Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted Python 3

Run Code

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
```

Complete Assignment 0 and Turn In

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']

Control Structures

- Direct the order of execution of statements in a program
- if / else: “If the weather is nice, I will mow the lawn, otherwise I’ll watch tv”
- loops:
  - for loop: for every element in X, do Y
  - while loop: while *condition* is True, do Y

For Loops

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

```
for color in L:
        print color
```

For Loops

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

```
for color in L:
        print color
```

yellow
red
pink
orange
green
cyan
blue
black

For Loops

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

```
for idx in range(len(L)):
        print color[idx]
```

```
yellow
red
pink
orange
green
cyan
blue
black
```

range

Outputs numbers in a range that is inclusive on the lower end and exclusive on the upper end

- range(5): [0, 1, 2, 3, 4]

range

Outputs numbers in a range that is inclusive on the lower end and exclusive on the upper end

- range(5): [0, 1, 2, 3, 4]
- range(1, 5): [1, 2, 3, 4]

range

Outputs numbers in a range that is inclusive on the lower end and exclusive on the upper end

- range(5): [0, 1, 2, 3, 4]
- range(1, 5): [1, 2, 3, 4]
- range(0,15,5): [0, 5, 10]

range

Outputs numbers in a range that is inclusive on the lower end and exclusive on the upper end

- range(5): [0, 1, 2, 3, 4]
- range(1, 5): [1, 2, 3, 4]
- range(0,15,5): [0, 5, 10]

Caveat: if you want to store the range as a list variable, you need to put it in a list – list(range(5)). You don't have to do this if your doing a for loop like on the previous slide.

Fibonacci Sequence

Fibonacci Sequence

- Group activity:
  - Form groups of 2-3
  - Create a new Jupyter notebook called Fibonacci
  - Use your knowledge of python operations, lists and for loops to write code that creates a list with the first 15 elements of the Fibonacci sequence
  - At the end, check that your list is actually 15 elements long
  - *Hints:*
    - The first two elements of your list should be [0,1]
    - The last element should be 377