# Part 2.1:
# Introduction to Python

Printing, Variables and String Slices

Dana L Carper and Travis J Lawrence

Quantitative and Systems Biology

University of California, Merced

# Objectives: Second Half

- Learn Python3

# Objectives: Second Half

- Learn ~~Python~~3 basic programming concepts

# Objectives: Second Half

- Learn ~~Python~~3 basic programming concepts
  - Programming logic applies to all languages

# Objectives: Second Half

- Learn ~~Python~~3 basic programming concepts
  - Programming logic applies to all languages
  - Each programming language has pros and cons

# Objectives: Second Half

- Learn ~~Python3~~ basic programming concepts
  - Programming logic applies to all languages
  - Each programming language has pros and cons
- Why Python3?

# Objectives: Second Half

- Learn ~~Python~~3 basic programming concepts
  - Programming logic applies to all languages
  - Each programming language has pros and cons
- Why Python3?
  - Free

# Objectives: Second Half

- Learn ~~Python~~3 basic programming concepts
  - Programming logic applies to all languages
  - Each programming language has pros and cons
- Why Python3?
  - Free
  - Easy to learn

# Objectives: Second Half

- Learn ~~Python~~3 basic programming concepts
  - Programming logic applies to all languages
  - Each programming language has pros and cons
- Why Python3?
  - Free
  - Easy to learn
  - Runs on almost anything

# Objectives: Second Half

- Learn ~~Python~~3 basic programming concepts
  - Programming logic applies to all languages
  - Each programming language has pros and cons
- Why Python3?
  - Free
  - Easy to learn
  - Runs on almost anything
  - Well documented

# Objectives: Second Half

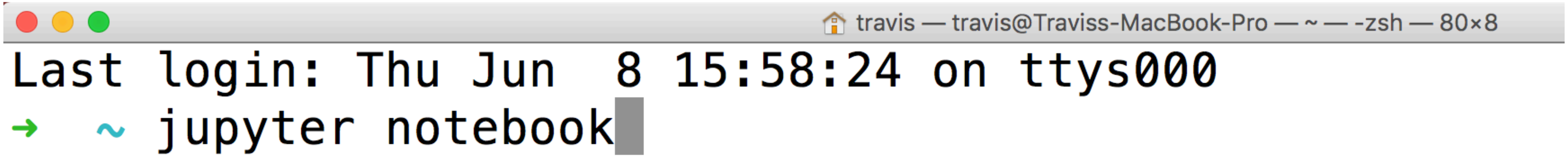
- Learn ~~Python~~3 basic programming concepts
  - Programming logic applies to all languages
  - Each programming language has pros and cons
- Why Python3?
  - Free
  - Easy to learn
  - Runs on almost anything
  - Well documented
  - Increased use among scientists

# Objectives: 2.1

1. Assigning values to variables
2. Printing to terminal
3. Perform operations on variables
4. Indexing and slice strings

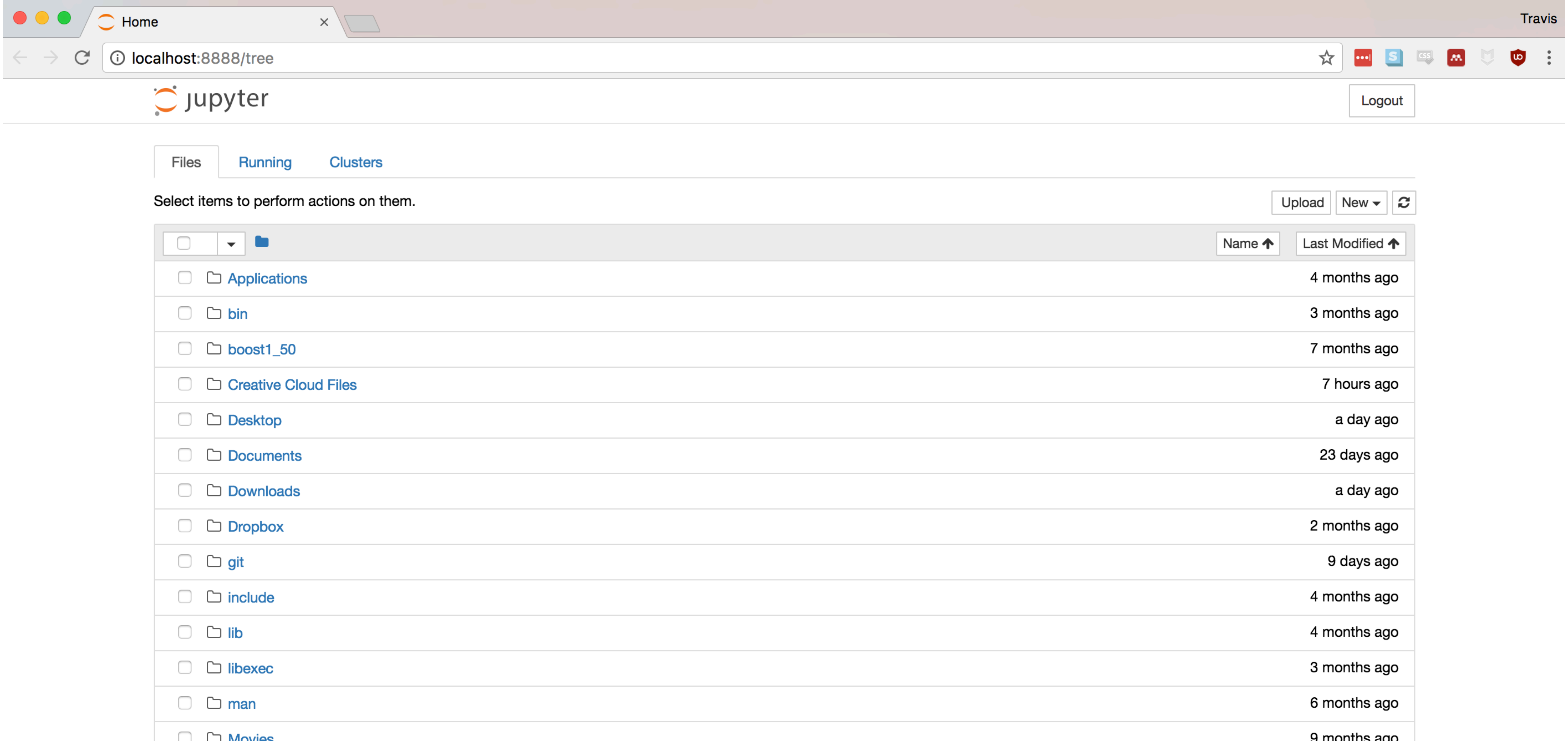
# Jupyter Notebook

- Interactive environment for writing and running code.
  - We installed it early with pip3
- Runs in your browser
- Starting Jupyter Notebook



```
Last login: Thu Jun  8 15:58:24 on ttys000
➜  ~ jupyter notebook
```

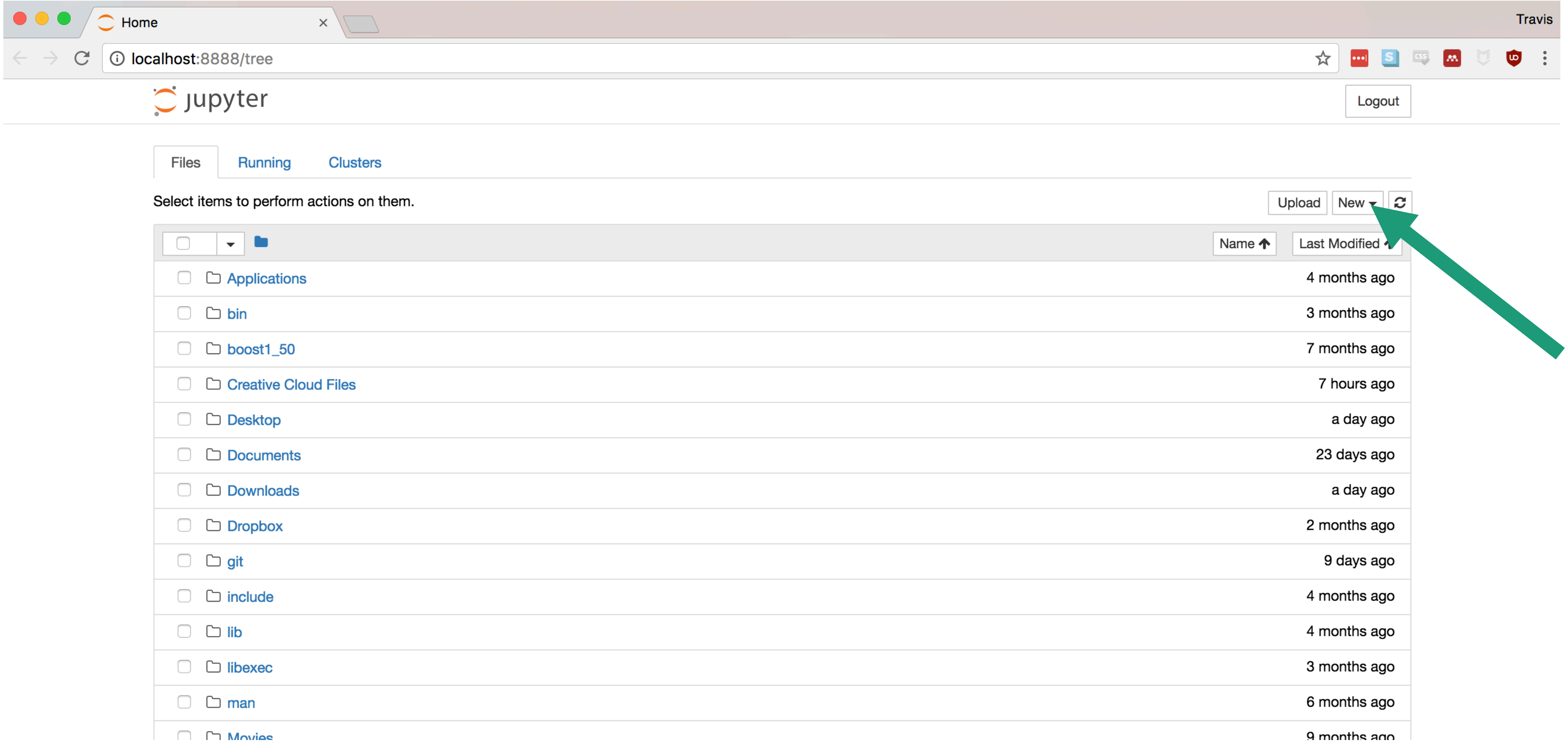
# Jupyter Server



Select items to perform actions on them.

| Name | Last Modified |
|---|---|
| Applications | 4 months ago |
| bin | 3 months ago |
| boost1_50 | 7 months ago |
| Creative Cloud Files | 7 hours ago |
| Desktop | a day ago |
| Documents | 23 days ago |
| Downloads | a day ago |
| Dropbox | 2 months ago |
| git | 9 days ago |
| include | 4 months ago |
| lib | 4 months ago |
| libexec | 3 months ago |
| man | 6 months ago |
| Movies | 9 months ago |

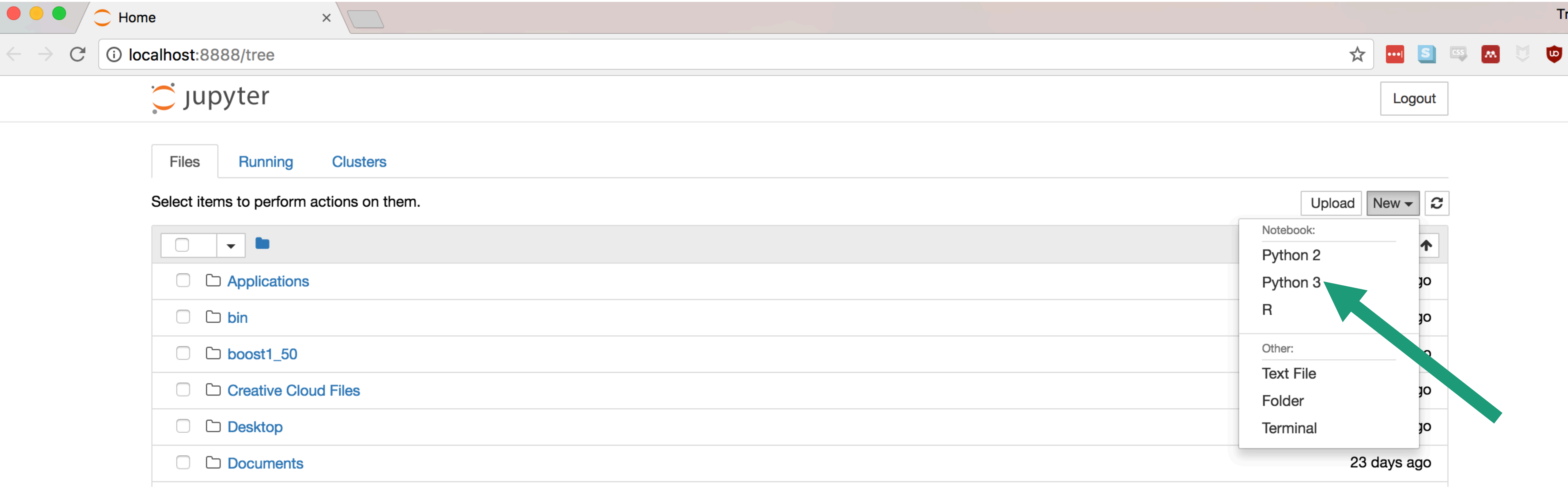
# Jupyter Server



Select items to perform actions on them.

| | |
|---|---|
| Applications | 4 months ago |
| bin | 3 months ago |
| boost1_50 | 7 months ago |
| Creative Cloud Files | 7 hours ago |
| Desktop | a day ago |
| Documents | 23 days ago |
| Downloads | a day ago |
| Dropbox | 2 months ago |
| git | 9 days ago |
| include | 4 months ago |
| lib | 4 months ago |
| libexec | 3 months ago |
| man | 6 months ago |
| Movies | 9 months ago |

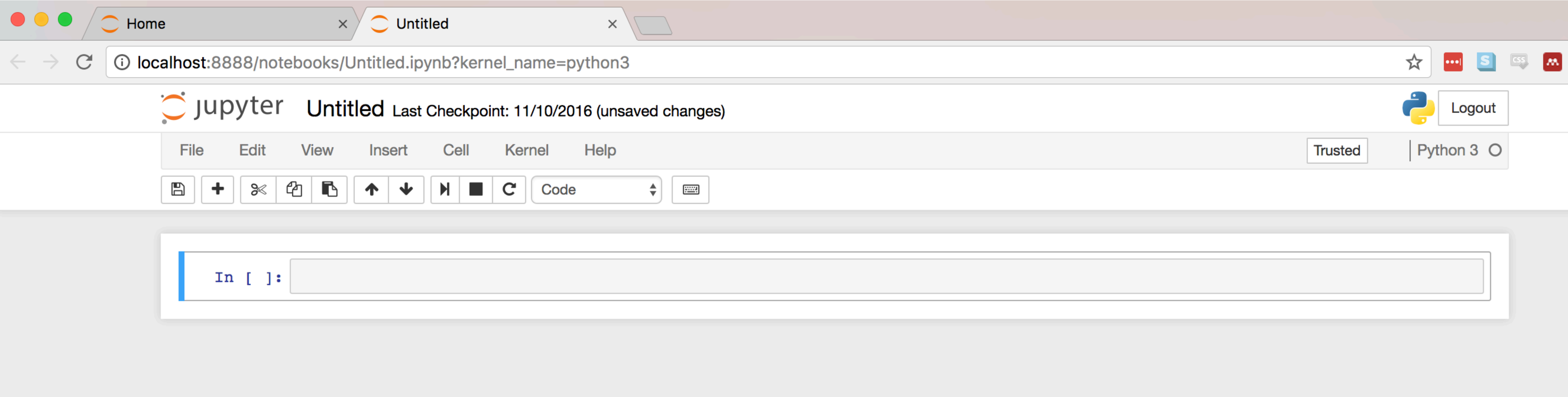
# Jupyter Server



Select items to perform actions on them.

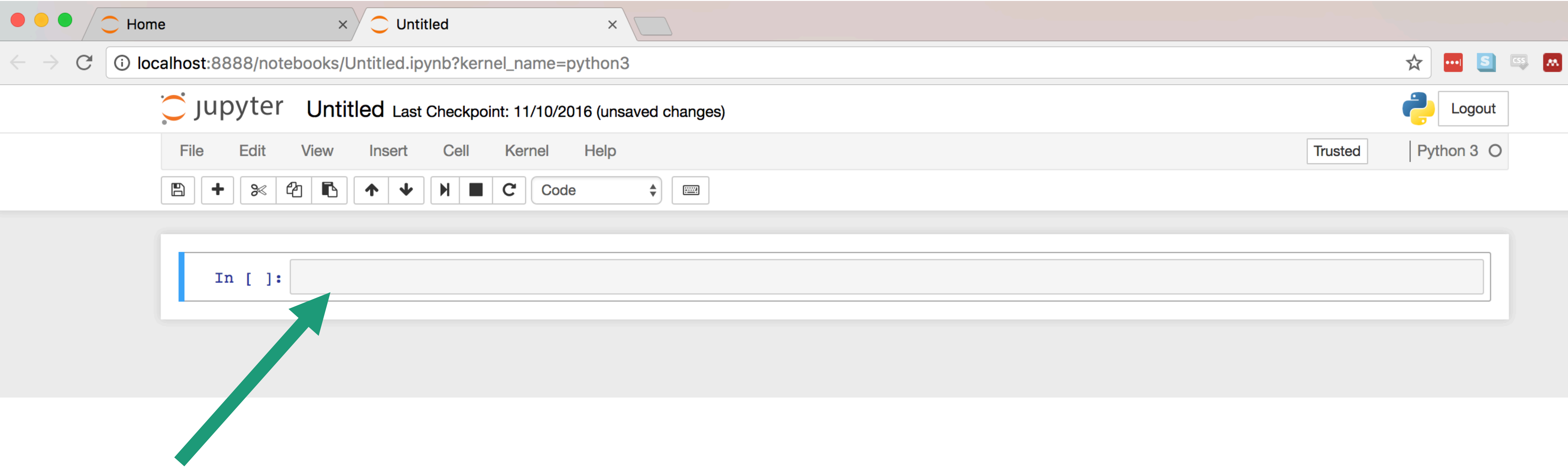
| | |
|---|---|
| □ Applications | |
| □ bin | |
| □ boost1_50 | |
| □ Creative Cloud Files | |
| □ Desktop | |
| □ Documents | 23 days ago |

# Jupyter Notebook



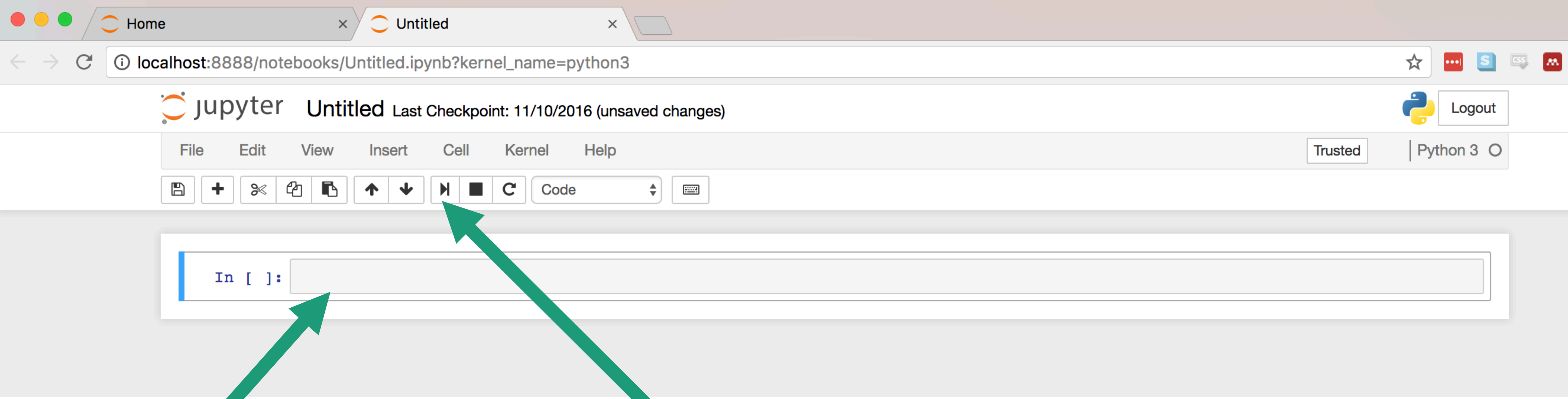
Home
Untitled
localhost:8888/notebooks/Untitled.ipynb?kernel_name=python3
jupyter
Untitled Last Checkpoint: 11/10/2016 (unsaved changes)
Logout
File
Edit
View
Insert
Cell
Kernel
Help
Trusted
Python 3
Code
In [ ]:

# Jupyter Notebook


Home
Untitled
localhost:8888/notebooks/Untitled.ipynb?kernel_name=python3
jupyter
Untitled
Last Checkpoint: 11/10/2016 (unsaved changes)
Logout
File
Edit
View
Insert
Cell
Kernel
Help
Trusted
Python 3
Code
In [ ]:

This is a code cell. You enter code to run in this cell type.

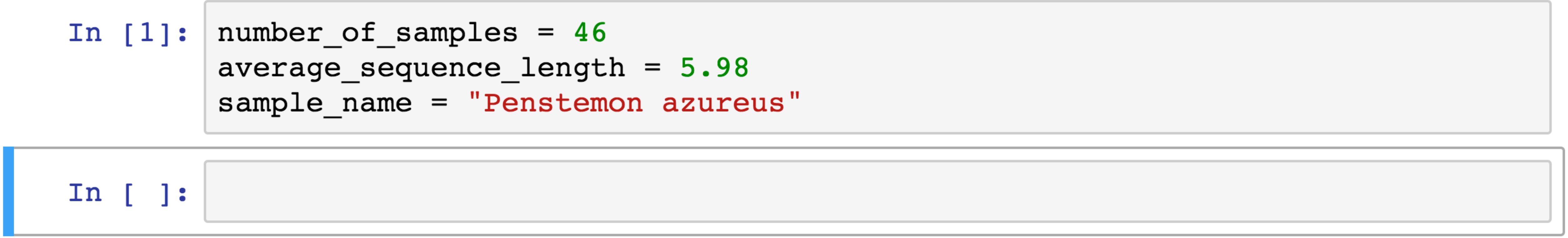
# Jupyter Notebook


Home
Untitled
localhost:8888/notebooks/Untitled.ipynb?kernel_name=python3
jupyter
Untitled
Last Checkpoint: 11/10/2016 (unsaved changes)
Logout
File
Edit
View
Insert
Cell
Kernel
Help
Trusted
Python 3
Code
In [ ]:

This is a code cell. You enter code to run in this cell type.

This button runs the code in the active cell. Any output is placed in an output cell.

# Variables: Assigment

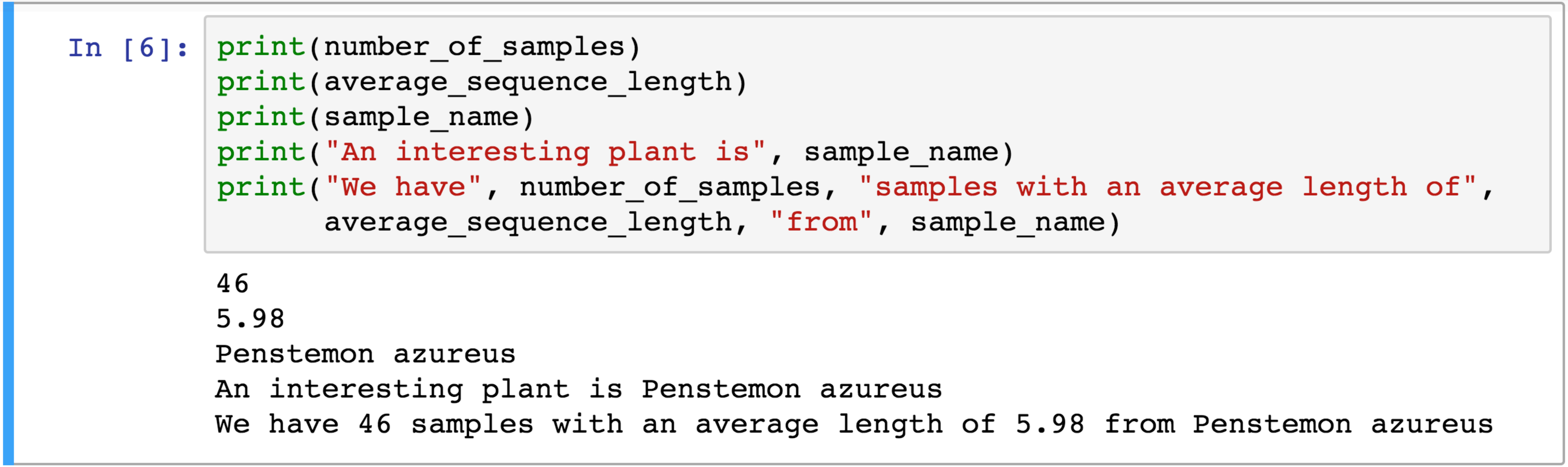


```
In [1]: number_of_samples = 46
        average_sequence_length = 5.98
        sample_name = "Penstemon azureus"

In [ ]:
```

- A variable is just a label for a value
- Variables are created using the **=** symbol
- Python variables must start with a letter
- Variables are case sensitive

# Variables: Printing



In [6]:

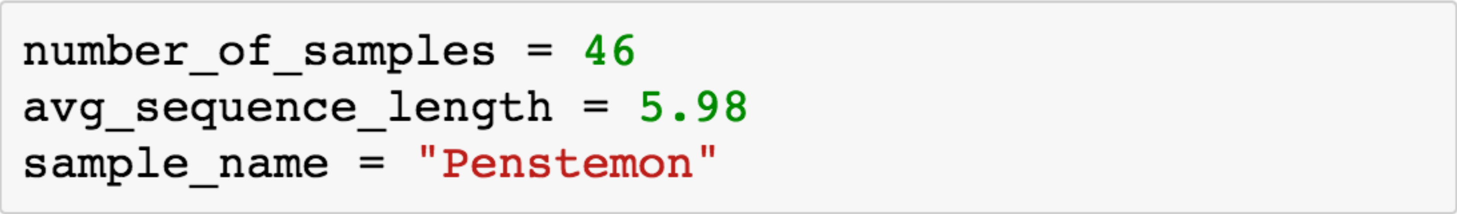
```
print(number_of_samples)
print(average_sequence_length)
print(sample_name)
print("An interesting plant is", sample_name)
print("We have", number_of_samples, "samples with an average length of",
      average_sequence_length, "from", sample_name)
```

```
46
5.98
Penstemon azureus
An interesting plant is Penstemon azureus
We have 46 samples with an average length of 5.98 from Penstemon azureus
```

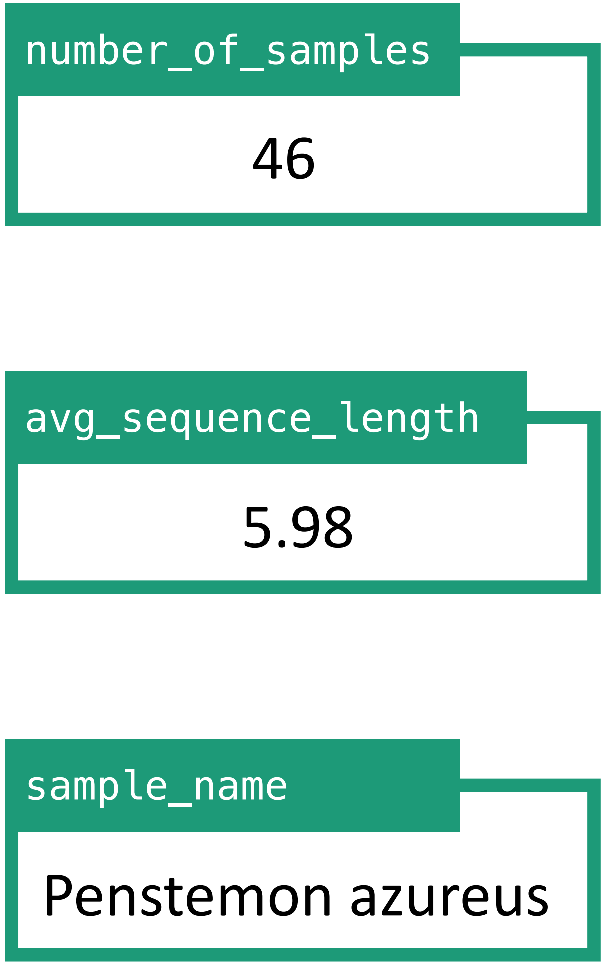
- Variables can be printed to the screen using the Python3 print function
- The print function's input is always surrounded by parenthesis
- A newline character is added to the output of the print function
- Multiple variables can be printed if separated by commas

# Variables

- Think about variables as labeled boxes



```python
number_of_samples = 46
avg_sequence_length = 5.98
sample_name = "Penstemon"
```



number_of_samples

46

avg_sequence_length

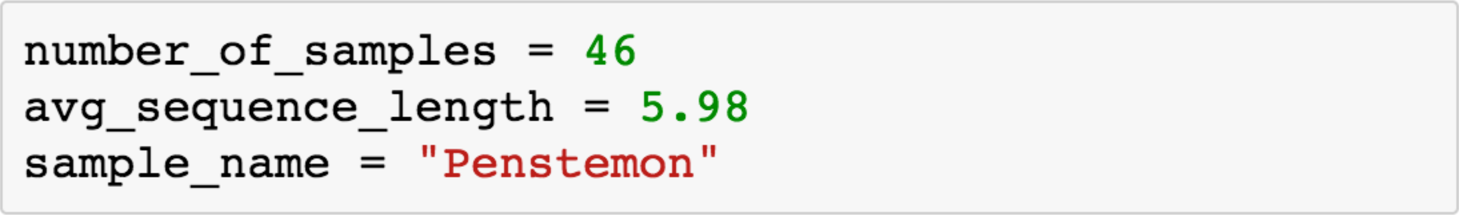
5.98

sample_name

Penstemon azureus

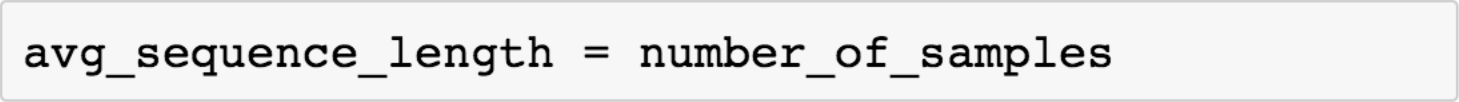
# Variables

- Think about variables as labeled boxes

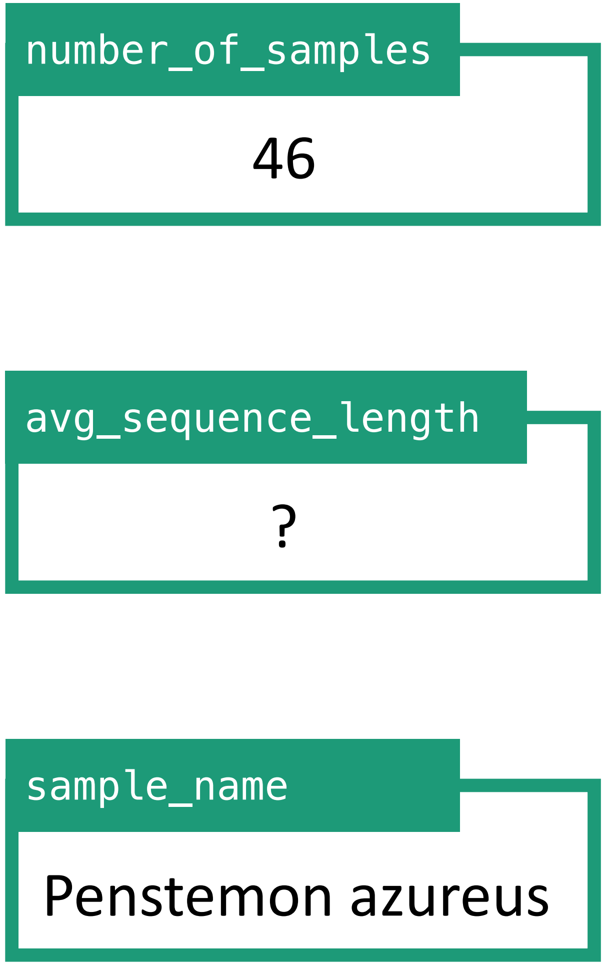


```
number_of_samples = 46
avg_sequence_length = 5.98
sample_name = "Penstemon"
```

- Reassigning values

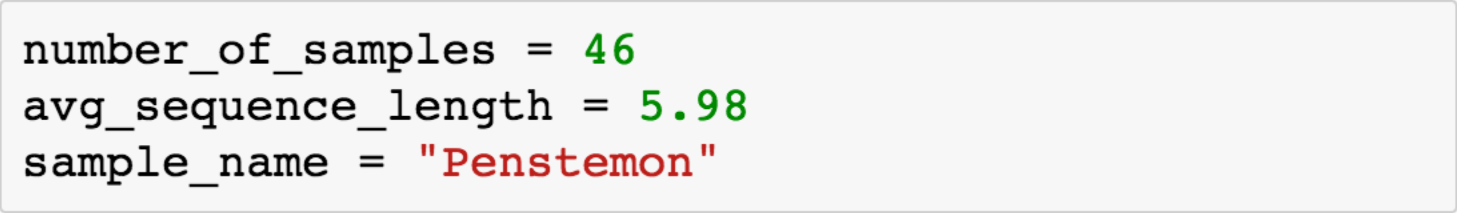


```
avg_sequence_length = number_of_samples
```



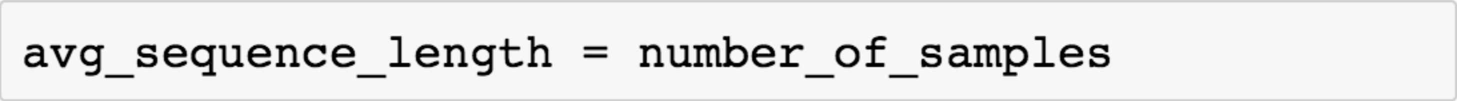
# Variables

- Think about variables as labeled boxes

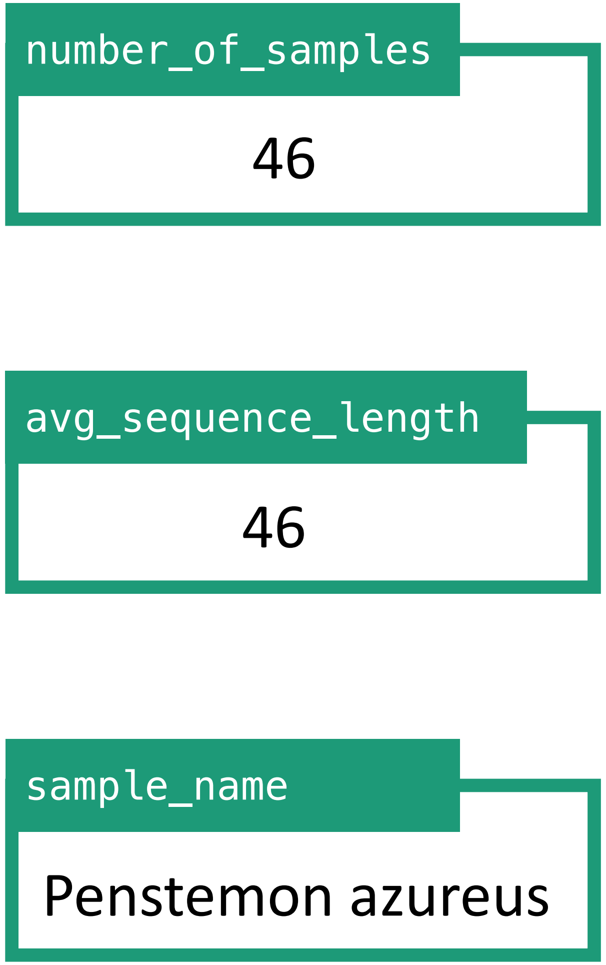


```
number_of_samples = 46
avg_sequence_length = 5.98
sample_name = "Penstemon"
```

- Reassigning values

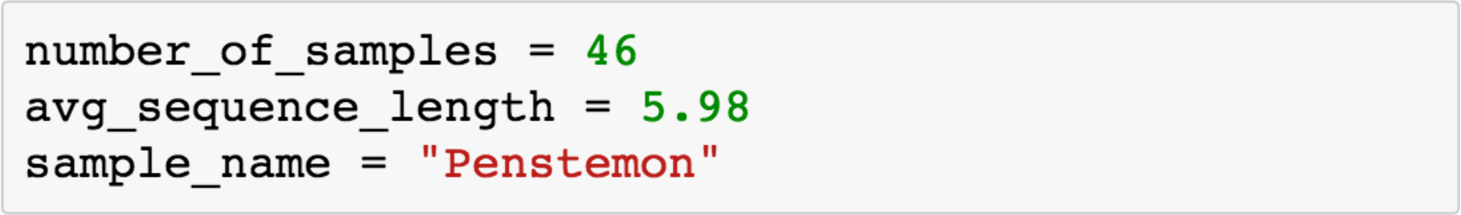


```
avg_sequence_length = number_of_samples
```



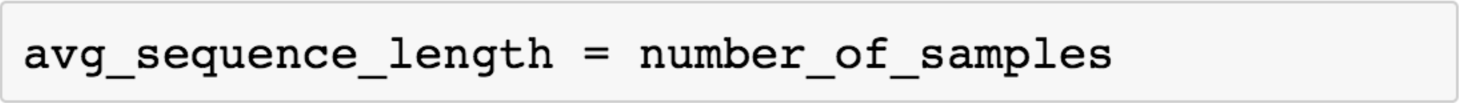
# Variables

- Think about variables as labeled boxes

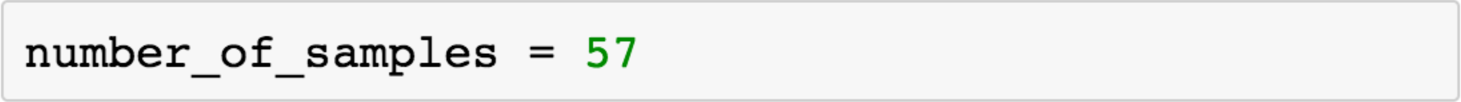


```
number_of_samples = 46
avg_sequence_length = 5.98
sample_name = "Penstemon"
```

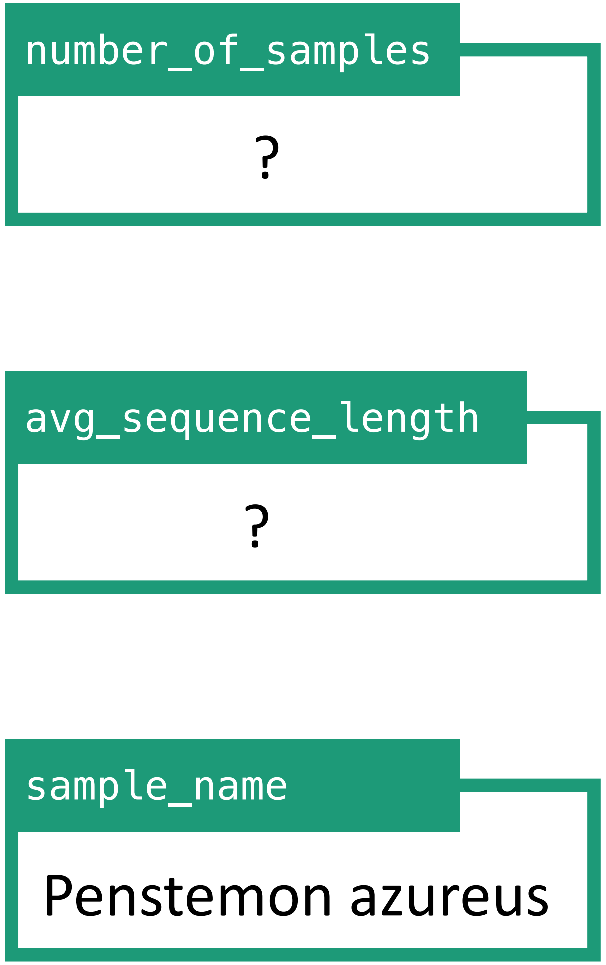
- Reassigning values



```
avg_sequence_length = number_of_samples
```

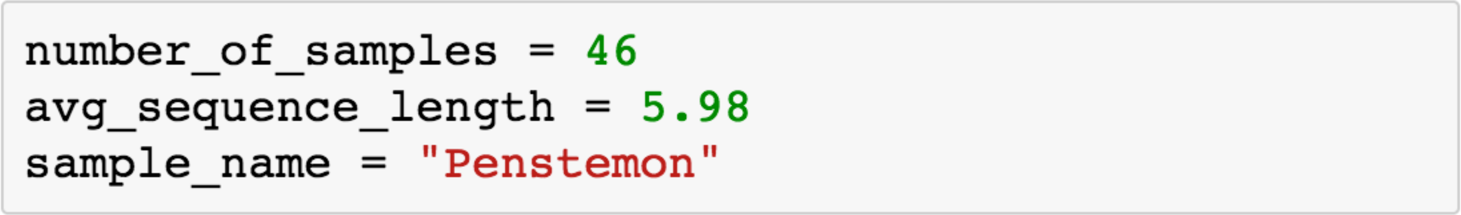


```
number_of_samples = 57
```



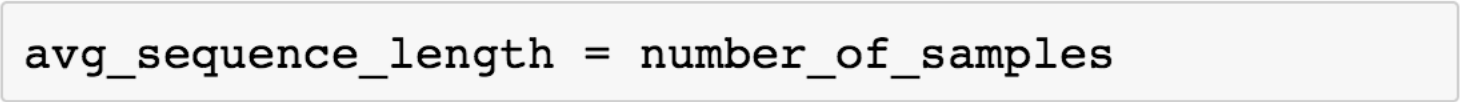
# Variables

- Think about variables as labeled boxes

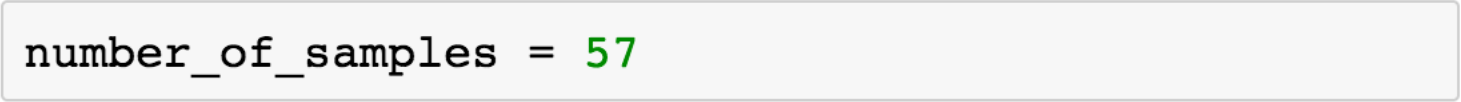


```
number_of_samples = 46
avg_sequence_length = 5.98
sample_name = "Penstemon"
```

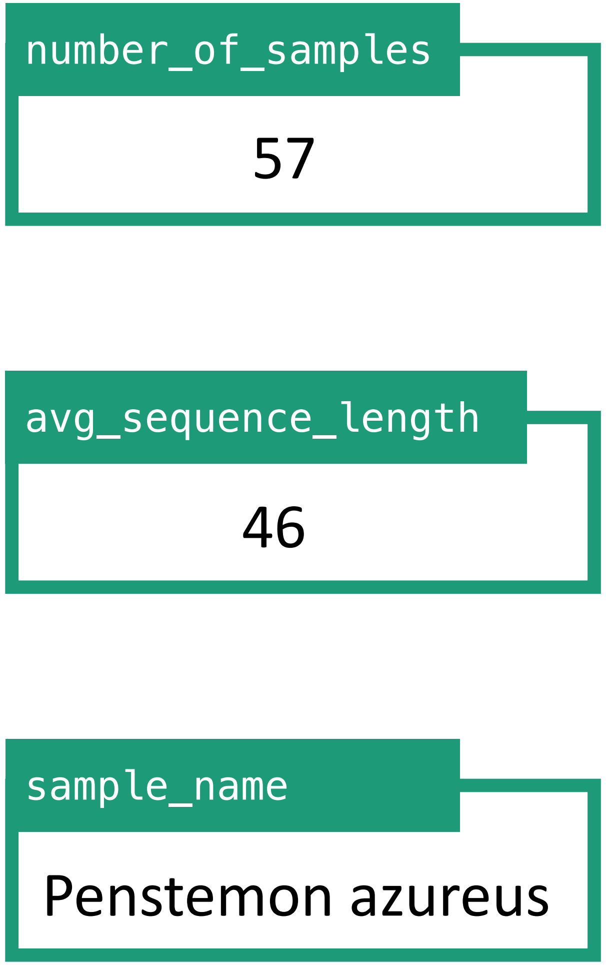
- Reassigning values



```
avg_sequence_length = number_of_samples
```

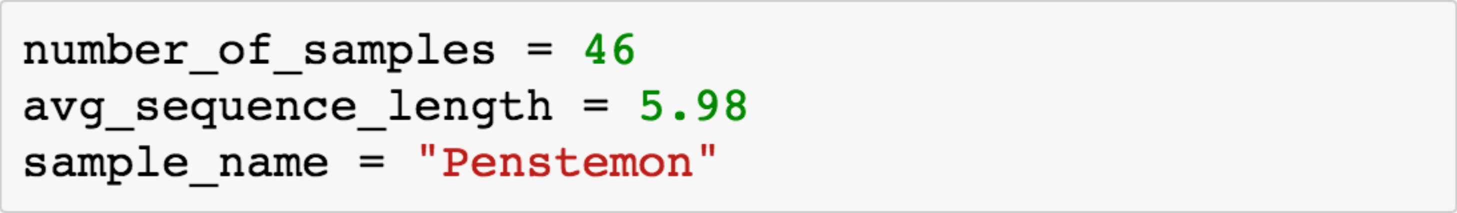


```
number_of_samples = 57
```



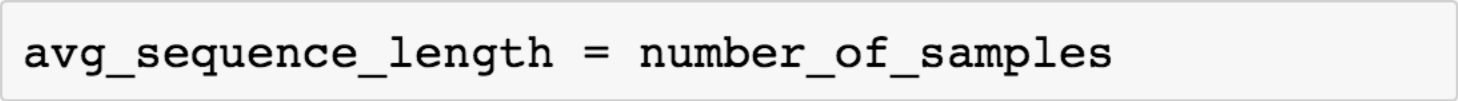
# Variables

- Think about variables as labeled boxes

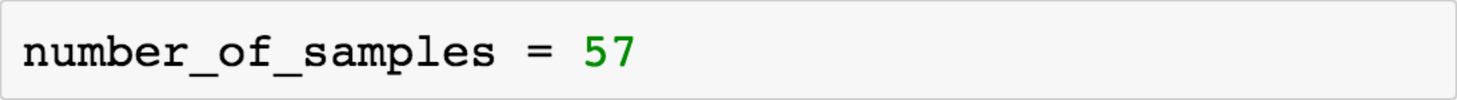


```
number_of_samples = 46
avg_sequence_length = 5.98
sample_name = "Penstemon"
```

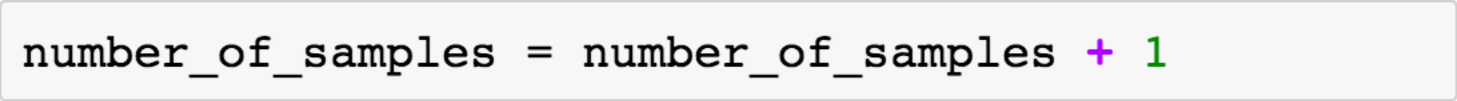
- Reassigning values



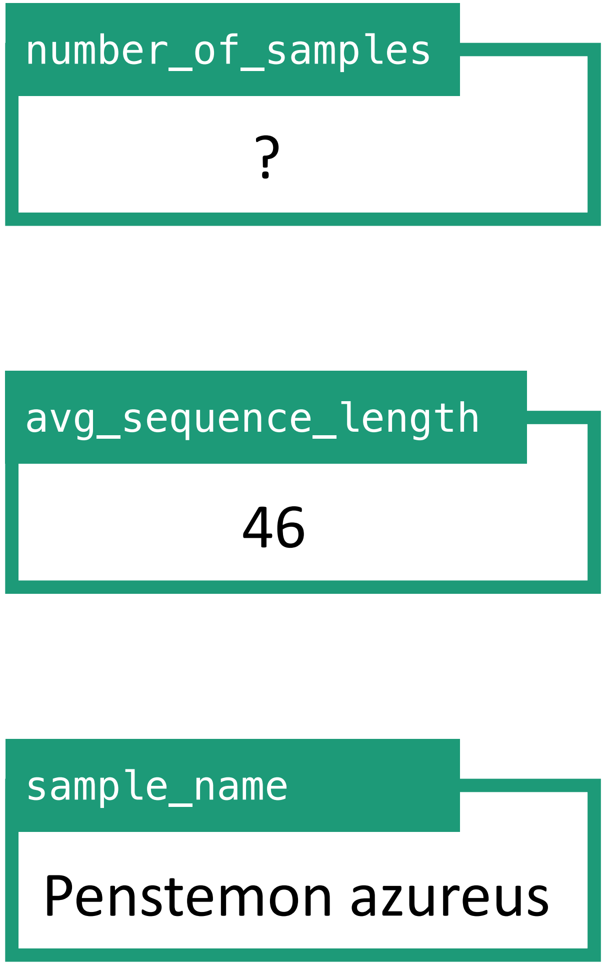
```
avg_sequence_length = number_of_samples
```



```
number_of_samples = 57
```



```
number_of_samples = number_of_samples + 1
```



number_of_samples

?

avg_sequence_length

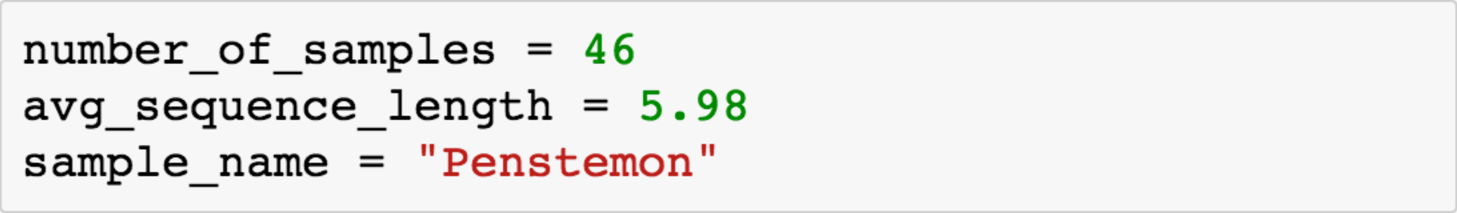
46

sample_name

Penstemon azureus

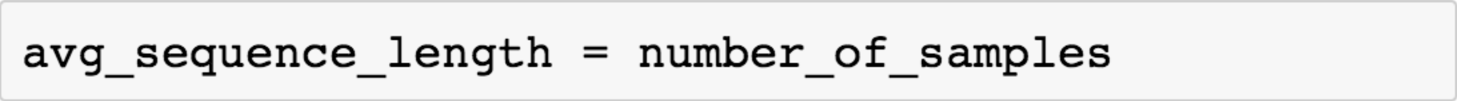
# Variables

- Think about variables as labeled boxes

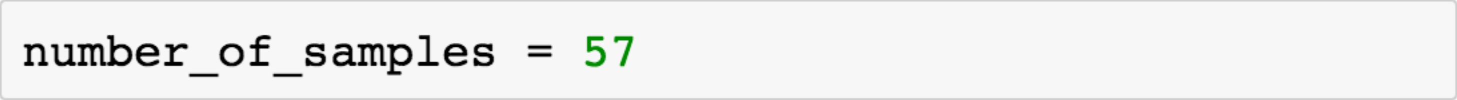


```
number_of_samples = 46
avg_sequence_length = 5.98
sample_name = "Penstemon"
```

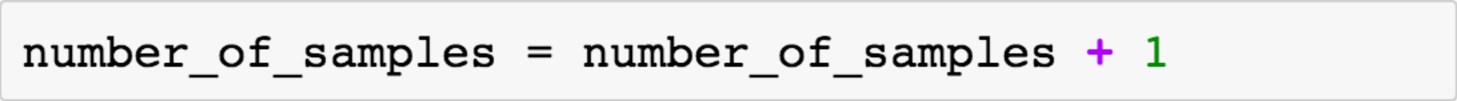
- Reassigning values



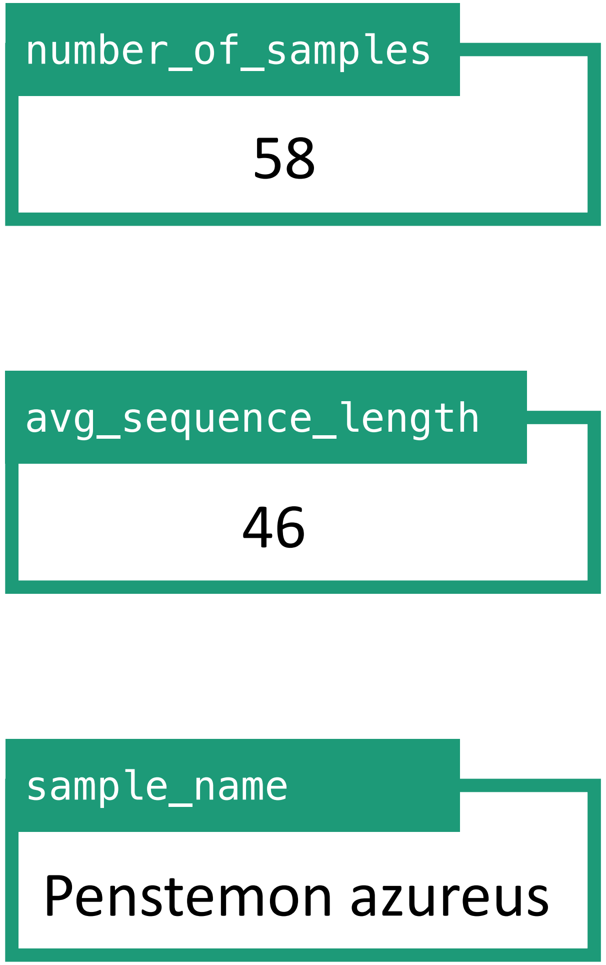
```
avg_sequence_length = number_of_samples
```



```
number_of_samples = 57
```



```
number_of_samples = number_of_samples + 1
```



number_of_samples

58

avg_sequence_length

46

sample_name

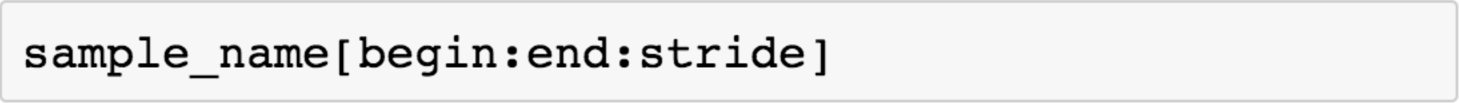
Penstemon azureus

# Variables: Operators

- + Addition
- – Subtraction
- ∗ Multiplication
- / Division
- ∗∗ Exponent
- % Remainder

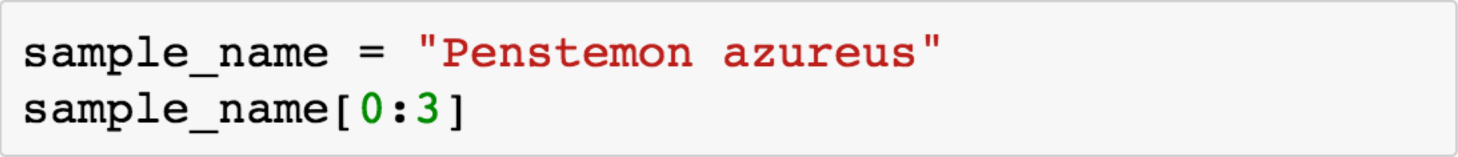
# Strings: Indexing and Slicing

- Square brackets surround slicing indices



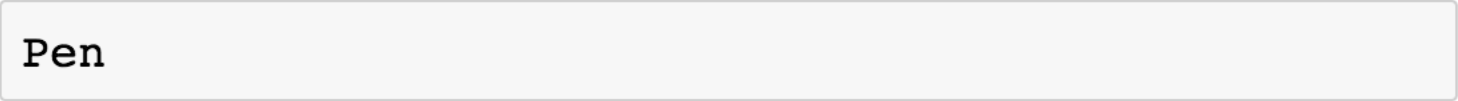
```
sample_name[begin:end:stride]
```

- Zero base indexing
  - This means the index starts at zero instead of one



```
sample_name = "Penstemon azureus"
sample_name[0:3]
```

- The ending index is not included



```
Pen
```