

# 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 ~~Python3~~ basic programming concepts

# Objectives: Second Half

- Learn ~~Python3~~ basic programming concepts
  - Programming logic applies to all languages

# Objectives: Second Half

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

# Objectives: Second Half

- Learn ~~Python3~~ 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 ~~Python3~~ 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 ~~Python3~~ 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

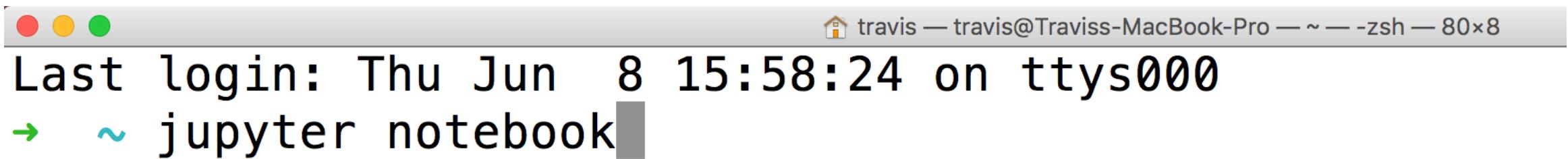
- Learn ~~Python3~~ basic programming concepts
  - Programming logic applies to all languages
  - Each programming language has pros and cons
- Why Python3?
  - Free
  - Easy to learn
  - Runs on almost anything
  - Well documented
  - Increased use among scientists

# 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



A screenshot of a terminal window on a Mac OS X desktop. The window title bar shows the system icons for red, yellow, and green buttons, followed by the user name "travis" and the host "travis@Traviss-MacBook-Pro". Below the title bar, the window frame is grey. The terminal text area shows the following output:

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

# Jupyter Server



The screenshot shows a Jupyter Server interface running on a local host. The top navigation bar includes a 'Home' button, a 'Travis' status indicator, and a 'Logout' button. The main content area features a 'jupyter' logo and a file tree. The file tree lists several directories: Applications, bin, boost1\_50, Creative Cloud Files, Desktop, Documents, Downloads, Dropbox, git, include, lib, libexec, man, and Movies. Each entry includes a checkbox for selection, a folder icon, and a timestamp indicating the last modified date. The interface also includes tabs for 'Files', 'Running', and 'Clusters', as well as buttons for 'Upload', 'New', and 'Logout'.

File/Folder	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



The screenshot shows the Jupyter Server interface running in a web browser. The title bar indicates the URL is `localhost:8888/tree`. The main content area displays a file tree with various directories and files. The toolbar at the top right includes buttons for `Upload`, `New` (highlighted with a green arrow), and a refresh icon. The file list is sorted by `Name` and `Last Modified`.

File/Folder	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



The screenshot shows the Jupyter Server interface running in a web browser. The title bar includes the Jupyter logo, the URL `localhost:8888/tree`, and a `Logout` button. The main area has tabs for `Files`, `Running` (which is selected), and `Clusters`. A message says `Select items to perform actions on them.`. On the left is a file tree with items like `Applications`, `bin`, `boost1_50`, `Creative Cloud Files`, `Desktop`, and `Documents`. On the right is a `New` menu with options for `Notebook:` (Python 2, Python 3, R), `Other:` (Text File, Folder, Terminal), and a timestamp `23 days ago`. A green arrow points to the `Python 3` option in the `Notebook:` list.

Home

localhost:8888/tree

jupyter

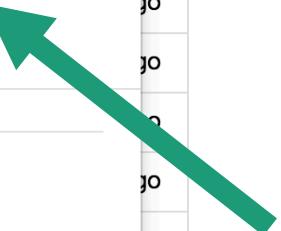
Logout

Files Running Clusters

Select items to perform actions on them.

Upload New 

Notebook:

- Python 2
- Python 3** 
- R

Other:

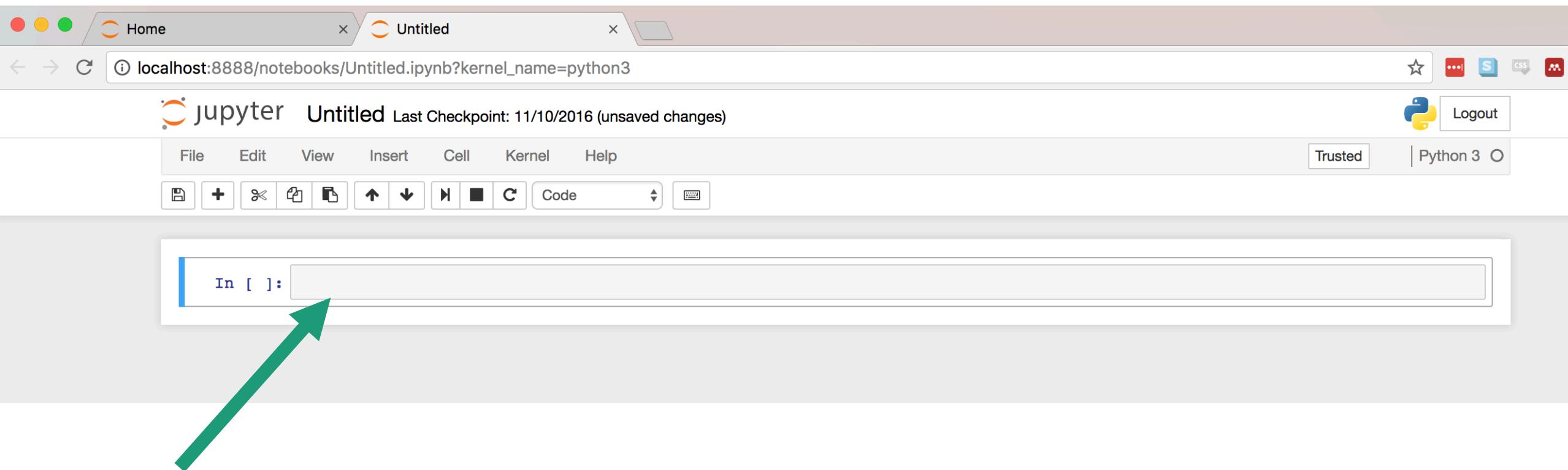
- Text File
- Folder
- Terminal

23 days ago

# Jupyter Notebook

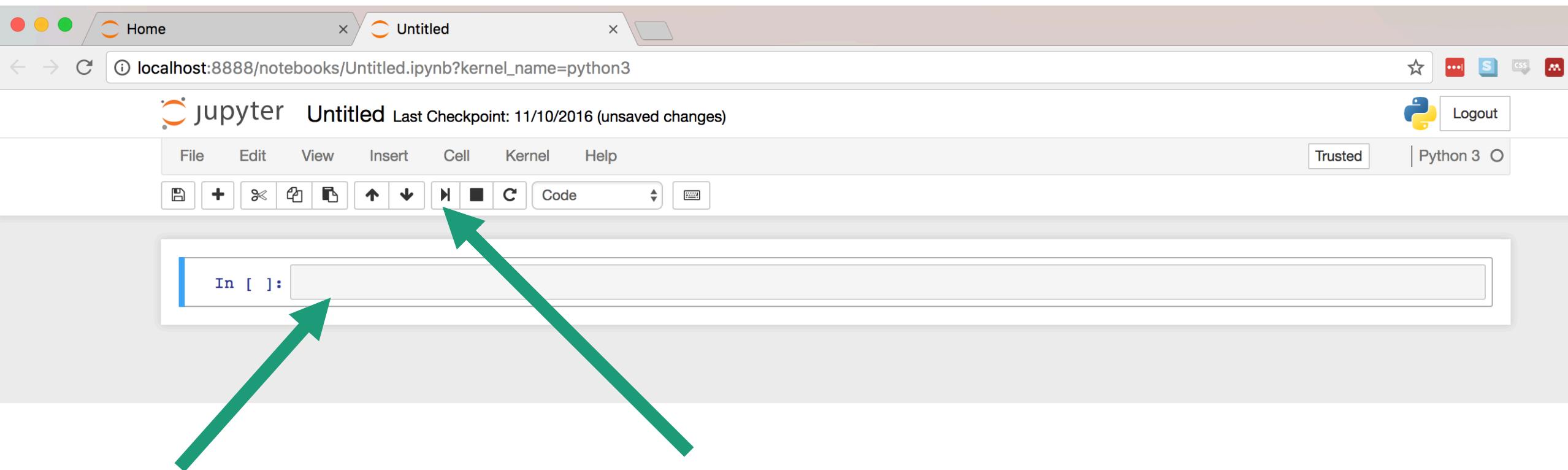
A screenshot of a Jupyter Notebook interface running in a web browser. The browser window has a tab bar with 'Home' and 'Untitled' tabs. The address bar shows the URL: `localhost:8888/notebooks/Untitled.ipynb?kernel_name=python3`. The main content area is titled 'jupyter Untitled' and shows 'Last Checkpoint: 11/10/2016 (unsaved changes)'. The top navigation bar includes 'File', 'Edit', 'View', 'Insert', 'Cell', 'Kernel', and 'Help' menus. On the right, there are buttons for 'Trusted' and 'Python 3'. Below the menu bar is a toolbar with icons for file operations (Save, New, Open, etc.) and cell controls (Run, Stop, Cell, etc.). A code cell is visible, labeled 'In [ ]:', with a blue cursor bar on the left. The background of the browser window is a light grey gradient.

# Jupyter Notebook



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

# Jupyter Notebook



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: Assignment

```
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

```
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"
```

number\_of\_samples

46

- Reassigning values

```
avg_sequence_length = number_of_samples
```

avg\_sequence\_length

?

sample\_name

Penstemon azureus

# Variables

- Think about variables as labeled boxes

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

number\_of\_samples

46

- Reassigning values

```
avg_sequence_length = 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"
```

number\_of\_samples

?

- Reassigning values

```
avg_sequence_length = number_of_samples
```

avg\_sequence\_length

?

```
number_of_samples = 57
```

sample\_name

Penstemon azureus

# Variables

- Think about variables as labeled boxes

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

number\_of\_samples

57

- Reassigning values

```
avg_sequence_length = number_of_samples
```

avg\_sequence\_length

46

```
number_of_samples = 57
```

sample\_name

Penstemon azureus

# Variables

- Think about variables as labeled boxes

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

number\_of\_samples

?

- Reassigning values

```
avg_sequence_length = number_of_samples
```

avg\_sequence\_length

46

```
number_of_samples = 57
```

```
number_of_samples = number_of_samples + 1
```

sample\_name

Penstemon azureus

# Variables

- Think about variables as labeled boxes

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

number\_of\_samples

58

- Reassigning values

```
avg_sequence_length = number_of_samples
```

avg\_sequence\_length

46

```
number_of_samples = 57
```

```
number_of_samples = number_of_samples + 1
```

sample\_name

Penstemon azureus

# Variable Types

- Integer: This type of variable is used for whole numbers

```
x = 5  
y = 4  
z = 9
```

- Float: This type of variable is used for numbers with decimals

```
x = 5.1  
y = 4.0  
z = 9.5
```

- String: The type of variable is used for text data

```
x = "5.1"  
y = "4.0"  
z = "9"
```

# 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
```