

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

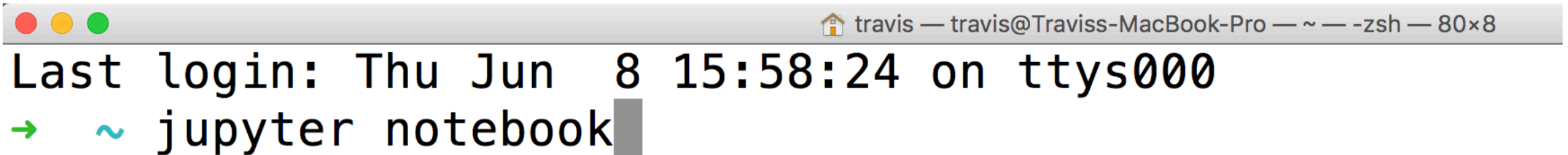
- Learn ~~Python3~~ basic programming concepts
 - Programming logic applies to all languages
 - Each programming language has pros and cons
- Why Python3?
 - Free
 - Easy to learn
 - Runs on almost anything
 - Well documented
 - Increased use among scientists

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 macOS terminal window. The title bar at the top shows three colored window control buttons (red, yellow, green) on the left and a status bar on the right that reads "travis — travis@Traviss-MacBook-Pro — ~ — -zsh — 80x8". The terminal content shows a login message: "Last login: Thu Jun 8 15:58:24 on ttys000". Below this, a prompt character (a green arrow) is followed by a tilde (~) and the text "jupyter notebook", with a grey cursor block at the end of the line.

```
travis — travis@Traviss-MacBook-Pro — ~ — -zsh — 80x8
Last login: Thu Jun 8 15:58:24 on ttys000
→ ~ jupyter notebook
```

Jupyter Server

Home

localhost:8888/tree

Travis

jupyter

Logout

Files

Running

Clusters

Select items to perform actions on them.

Upload

New

<input type="checkbox"/>		Name	Last Modified
<input type="checkbox"/>	Applications		4 months ago
<input type="checkbox"/>	bin		3 months ago
<input type="checkbox"/>	boost1_50		7 months ago
<input type="checkbox"/>	Creative Cloud Files		7 hours ago
<input type="checkbox"/>	Desktop		a day ago
<input type="checkbox"/>	Documents		23 days ago
<input type="checkbox"/>	Downloads		a day ago
<input type="checkbox"/>	Dropbox		2 months ago
<input type="checkbox"/>	git		9 days ago
<input type="checkbox"/>	include		4 months ago
<input type="checkbox"/>	lib		4 months ago
<input type="checkbox"/>	libexec		3 months ago
<input type="checkbox"/>	man		6 months ago
<input type="checkbox"/>	Movies		9 months ago

Jupyter Server

Home

localhost:8888/tree

Travis

jupyter

Logout

Files

Running

Clusters

Select items to perform actions on them.

Upload

New

Name ↑

Last Modified ↓

<input type="checkbox"/>	Applications	4 months ago
<input type="checkbox"/>	bin	3 months ago
<input type="checkbox"/>	boost1_50	7 months ago
<input type="checkbox"/>	Creative Cloud Files	7 hours ago
<input type="checkbox"/>	Desktop	a day ago
<input type="checkbox"/>	Documents	23 days ago
<input type="checkbox"/>	Downloads	a day ago
<input type="checkbox"/>	Dropbox	2 months ago
<input type="checkbox"/>	git	9 days ago
<input type="checkbox"/>	include	4 months ago
<input type="checkbox"/>	lib	4 months ago
<input type="checkbox"/>	libexec	3 months ago
<input type="checkbox"/>	man	6 months ago
<input type="checkbox"/>	Movies	9 months ago

Jupyter Server

Home

localhost:8888/tree

☆

...

S

CSS

jupyter

Logout

Files

Running

Clusters

Select items to perform actions on them.

☐

Applications

☐

bin

☐

boost1_50

☐

Creative Cloud Files

☐

Desktop

☐

Documents

Upload

New

Notebook:

Python 2

Python 3

R

Other:

Text File

Folder

Terminal

23 days ago

Jupyter Notebook

The image shows a web browser window displaying a Jupyter Notebook. The browser's address bar shows the URL `localhost:8888/notebooks/Untitled.ipynb?kernel_name=python3`. The notebook's title bar indicates the file is named "Untitled" and shows the last checkpoint as "11/10/2016 (unsaved changes)". A "Logout" button is visible in the top right corner. Below the title bar is a menu bar with options: File, Edit, View, Insert, Cell, Kernel, and Help. To the right of the menu bar are two buttons: "Trusted" and "Python 3". Below the menu bar is a toolbar containing icons for saving, creating a new file, deleting, copying, pasting, undo, redo, and running the cell. A dropdown menu next to the toolbar shows "Code" as the selected cell type. The main area of the notebook contains a single code cell with the prompt `In []:` followed by an empty text input field.

Home x Untitled x

localhost:8888/notebooks/Untitled.ipynb?kernel_name=python3

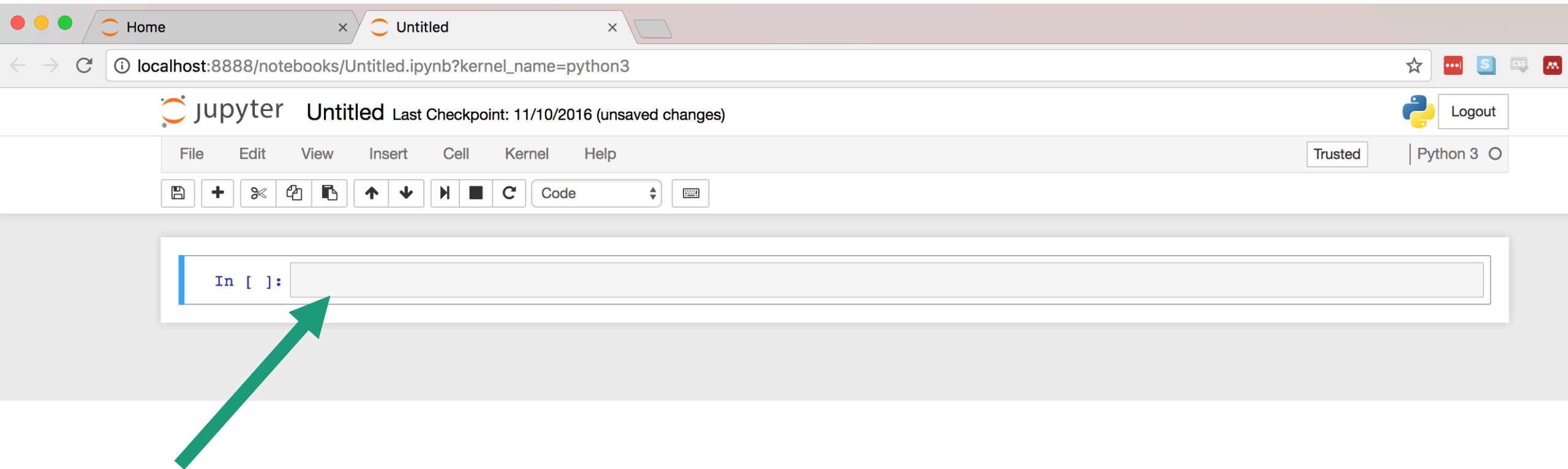
jupyter Untitled Last Checkpoint: 11/10/2016 (unsaved changes) Logout

File Edit View Insert Cell Kernel Help Trusted | Python 3

Save New Delete Copy Paste Undo Redo Run Code

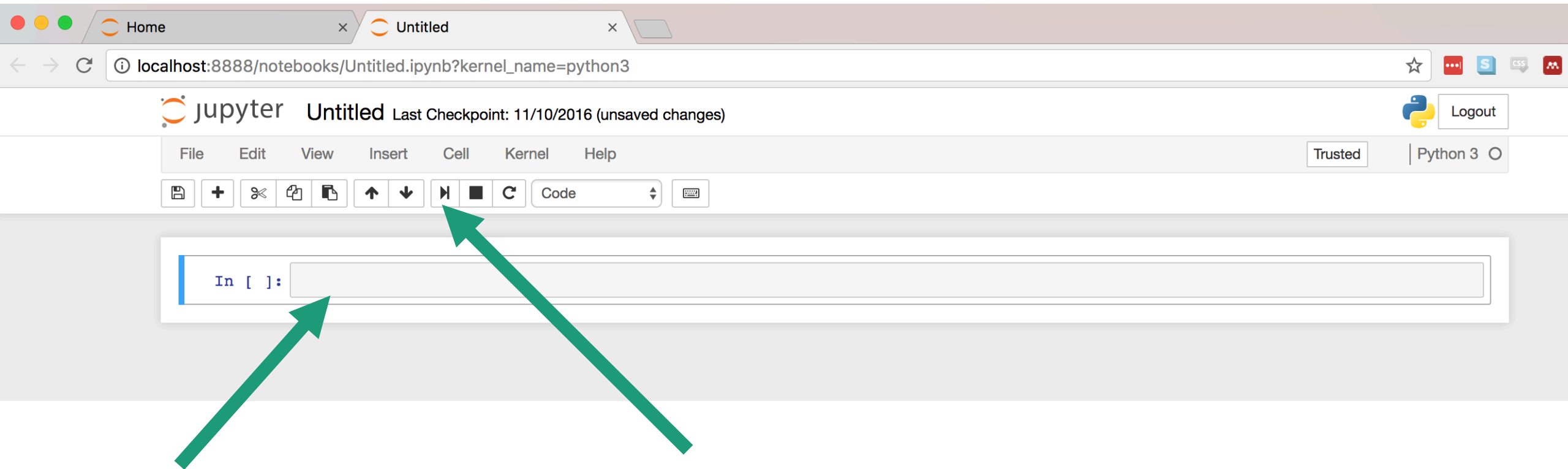
In []:

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"
```

- Reassigning values

```
avg_sequence_length = number_of_samples
```

number_of_samples

46

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"
```

- Reassigning values

```
avg_sequence_length = number_of_samples
```

number_of_samples

46

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

?

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"
```

- Reassigning values

```
avg_sequence_length = number_of_samples
```

```
number_of_samples = 57
```

number_of_samples

57

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

?

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
```