



ShiPy

—

Learn to Py while Shelter-in-Place
**L0: Logistics + Introduction +
Colab/Jupyter setup**



A volunteering educational initiative during COVID-19





ShiPy

Learn to Py while Shelter-in-Place

L0-A: Course Logistics



A volunteering educational initiative during COVID-19



HOWDY and Welcome Aboard



A volunteering educational initiative during COVID-19



ShiP Crew



Find us on LinkedIn
(see hyperlinks)



★ [JD](#)



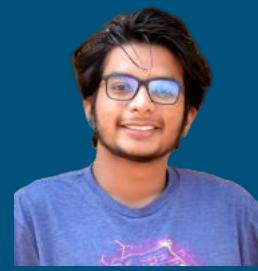
★ [Teddy](#)



★ [Chinmay](#)



★ [Pratik](#)



★ [Siddharth](#)



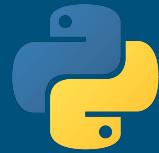
★ [Umang](#)



★ [Waseem](#)



A volunteering educational initiative during COVID-19



Logistics

AGENDA

- Itinerary
- Topics
- Pedagogy
- Tools
- Class Demographic
- Tips for Success



—

Itinerary

PHASE I: Foundations 3 weeks
(Sat, April 18 - Sat, May 9)

PHASE II: Libraries # In pipeline



Topics

PHASE I: Foundations

All times are in CDT (GMT-5)

1. Variables, Expressions, Simple I/O
2. Boolean Decisions (branching)
3. Repetitions (loops)
4. Collective Data Structures
5. Functions
6. File I/O
7. X

Sat, April 18 (11 am-12 noon)



Wed, April 22 (9 pm-10 pm)



Sat, April 25 (11 am-12 noon)



Wed, April 29 (9 pm-10 pm)



Sat, May 02 (11 am-12 noon)



Wed, May 06 (9 pm-10 pm)



Sat, May 09 (11 am-12 noon)



Pedagogy

1. Lectures

- Each is a 60 minute *LIVE Zoom* session
- **Slides:** 25-30 minutes
- **LIVE Coding:** 30-35 minutes

2. Programming Assignments

- Weekly take-home assignment with multiple questions ([Basic](#), [Intermediate](#), [Advanced](#))
- Code on *Google Colab*
- Multiple 30-minutes *LIVE Zoom* Help Sessions hosted by Crew (~daily)
- Submit on *Hackerrank* to get auto-graded and join the leaderboard

3. Quizzes

- Take-home quiz to reinforce lecture topics
- Designed to help with programming assignments



Tools



Campuswire- Q&A forum, chatrooms, DMs, materials, announcements
Refer to *Campuswire How Tos* post (#6) in class feed



Zoom- Lectures and Help sessions



Google Colab- Coding assignments in Python

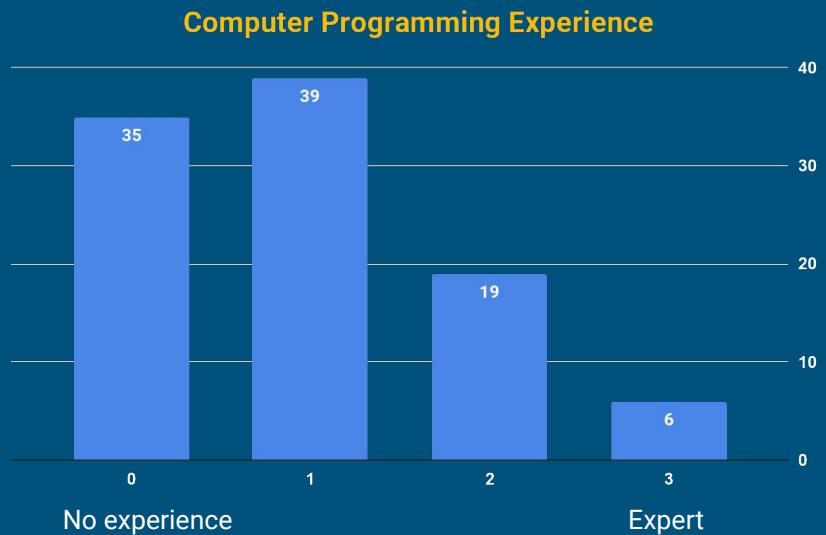
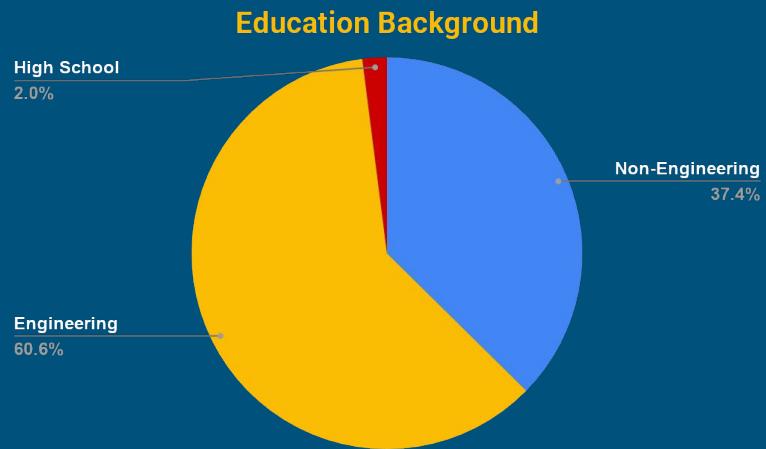


Hackerrank- Assignment submission and leaderboard

ShiPy Assignment opens at 12:30 pm CDT Today
<https://www.hackerrank.com/ship-py-assignments>



Class Demographic



Tips for Success

- Make most the LIVE sessions
- Attend lectures regularly
- Code with us in the Lectures regularly
- Practice by coding take-home assignments regularly
- Create elegant codes
- Form your own study groups/ chatrooms to build focussed communities
- Attend Live Help sessions for assignments whenever you are stuck
- Submit only your own assignments and compete on the Leaderboard
- Stay on Track with class
- No Grades, so Stay self-motivated
- Make most of this time and the opportunity
- DO NOT PLAGIARIZE , Respect everyone's time and efforts
- Share your positivity, encouragement, and feedback with our volunteering Crew



CODE . EAT . SLEEP . REPEAT

Next,

L0-B: Introduction





ShiPy

—
Learn to Py while Shelter-in-Place

L0-B: Introduction



A volunteering educational initiative during COVID-19



ShiP Crew



★ JD



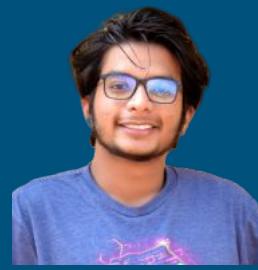
★ Teddy



★ Chinmay



★ Pratik



★ Siddharth



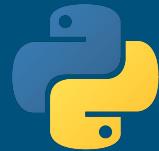
★ Umang



★ Waseem



A volunteering educational initiative during COVID-19



Presail

AGENDA

- What is a Code ?
- Programming Process
- Intro to Python
- My 1st Python Code
- Code Style



What is a Code Recipe ?



Ingredients

- 2 cups sifted all-purpose flour
- 2 teaspoons baking soda
- 1 teaspoon salt (heaping)
- 1 tablespoon sugar
- 2 eggs
- 2 cups buttermilk
- oil (for cooking) or butter (for cooking)
- butter, and
- maple syrup, for serving

Directions

Mix all ingredients together until incorporated, but do not over mix.

Heat a griddle or non stick pan under med. - med. high heat, and grease with a little butter or oil (more if you like a crispy edge).

Drop batter using a 1/4 cup measuring cup onto hot pan. Once the bottom side is golden, flip and brown remaining side.

Serve with butter and real maple syrup.





What is a Code ? Code = Computer Program

Animate U
Spread Wisdom in Our World

Chapter 1:

Introduction to Python Programming



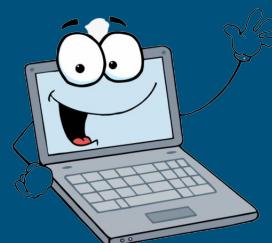
Programming Process: Problem to Solution

Problem

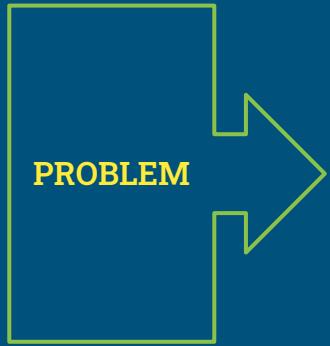
WAP to calculate Gross Pay (\$) from user-input values for #hours worked and hourly pay rate

Solution

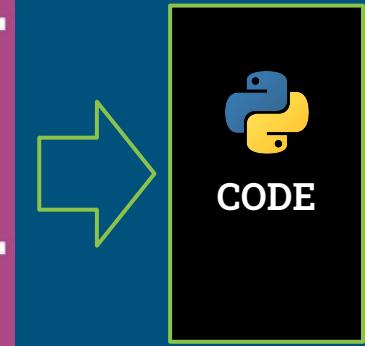
```
Enter the number of hours: 3  
Enter the hourly pay rate: 15.5  
The gross pay is = $ 46.5
```



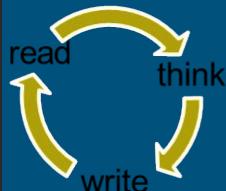
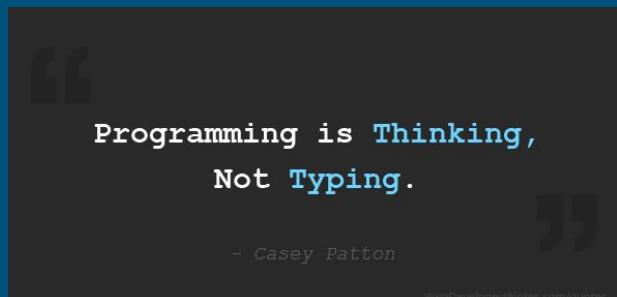
Algorithm vs PseudoCode



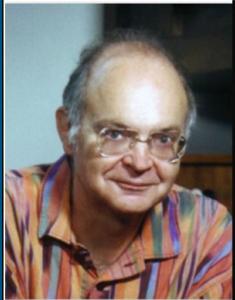
ALGORITHM	PSEUDOCODE
An unambiguous specification of how to solve a problem	An informal high-level description of the operating principle of a computer program or other algorithm
Helps to simplify and understand the problem	A method of developing an algorithm



Solution



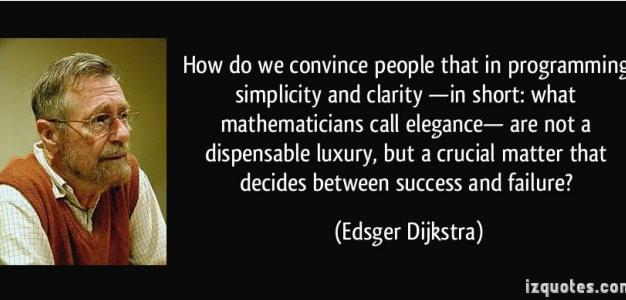
➤ Art of Programming



Programming is the art of telling another human being what one wants the computer to do.

— Donald Knuth —

AZ QUOTES



How do we convince people that in programming simplicity and clarity—in short: what mathematicians call elegance—are not a dispensable luxury, but a crucial matter that decides between success and failure?

(Edsger Dijkstra)

izquotes.com



You might not think that programmers are artists, but programming is an extremely creative profession. Its logic-based creativity.

— John Romero —

AZ QUOTES



Programs must be written for people to read, and only incidentally for machines to execute.

— Hal Abelson —

AZ QUOTES

Everyday life is like programming, I guess. If you love something you can put beauty into it.

- Donald Knuth



Programming Process: Example

Problem

WAP to calculate Gross Pay (\$) from user-input values for #hours worked and hourly pay rate



Algorithm

1. Get the number of hours worked
2. Get the hourly pay rate
3. Multiple the number of hours worked by the hourly pay rate
4. Display the result of the calculation that was performed in Step 3

[source](#)



Pseudocode

- 1-1: Display "Enter the number of hours"
- 1-2: Input hours
- 2-1: Display "Enter the hourly pay rate"
- 2-2: Input payRate
- 3: Set grossPay = hours * payRate
- 4: Display "The gross pay is \$", grossPay



Source Code

```
# 1. get the number of hours worked
hours = int(input("Enter the number of hours"))

# 2. get the hourly payrate
payRate = float(input("Enter the hourly pay rate"))

# 3. calculate grossPay
grossPay = hours * payRate

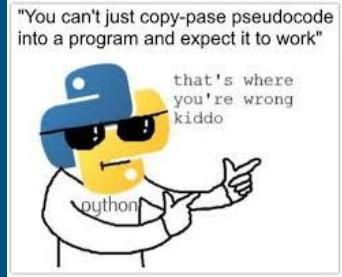
# 4. display the result
print("The gross pay is $", grossPay)
```



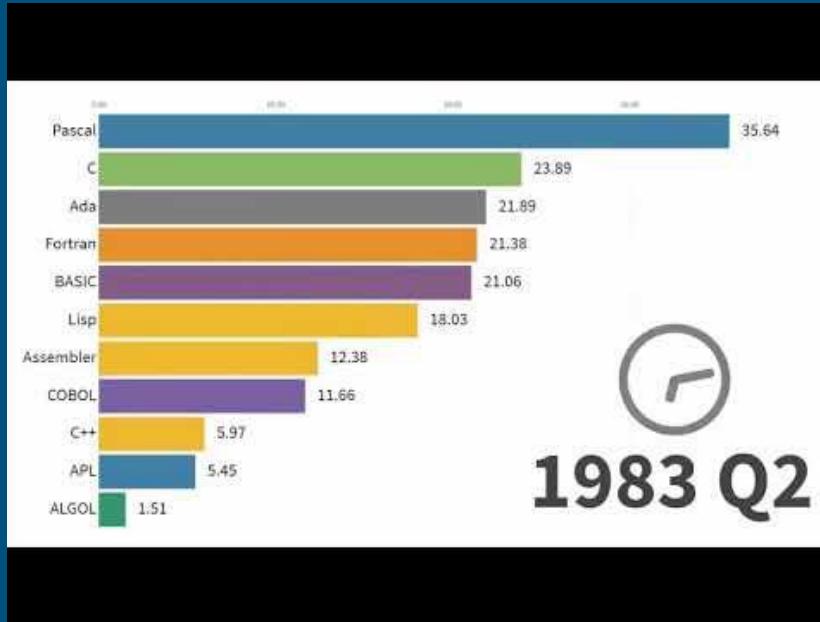
RUN
THE CODE

```
Enter the number of hours: 3
Enter the hourly pay rate: 15.5
The gross pay is = $ 46.5
```

Output solution



Popular Programming Languages



PYPL PopularitY of Programming Language

Worldwide, Apr 2020 compared to a year ago:

Rank	Change	Language	Share	Trend
1		Python	30.61 %	+3.9 %
2		Java	18.45 %	-1.9 %
3		Javascript	7.91 %	-0.4 %
4		C#	7.27 %	-0.0 %
5		PHP	6.07 %	-1.1 %
6		C/C++	5.76 %	-0.2 %
7		R	3.8 %	-0.2 %
8		Objective-C	2.4 %	-0.4 %
9		Swift	2.23 %	-0.2 %
10	↑	TypeScript	1.85 %	+0.2 %
11	↓	Matlab	1.77 %	-0.2 %

The PYPL PopularitY of Programming Language Index is created by analyzing how often language tutorials are searched on Google.

The more a language tutorial is searched, the more popular the language is assumed to be. It is a leading indicator. The raw data comes from Google Trends.

Intro to Python

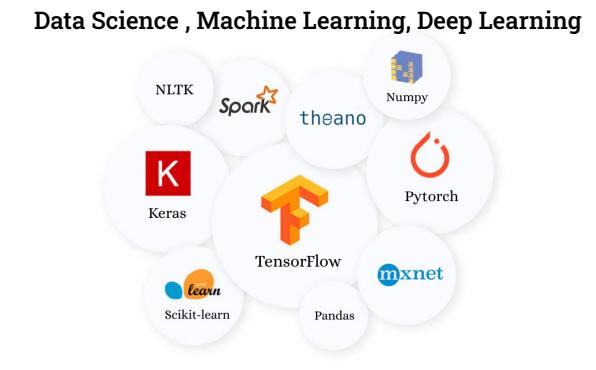
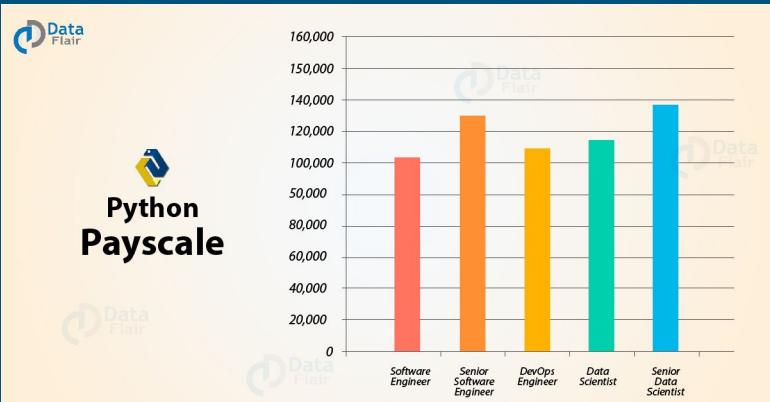
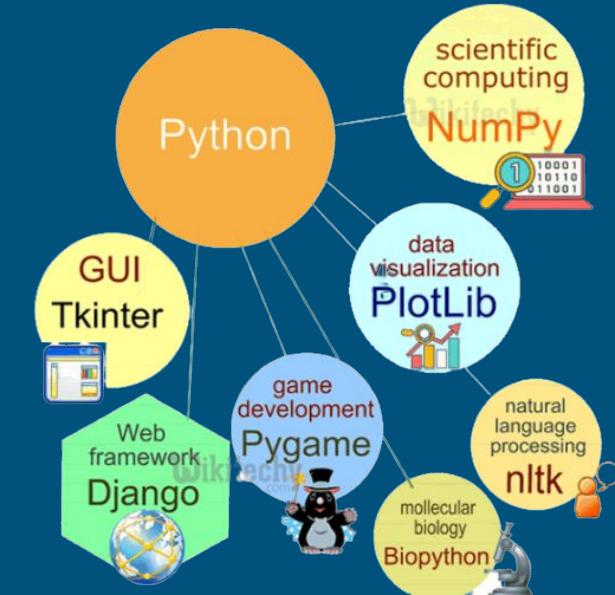


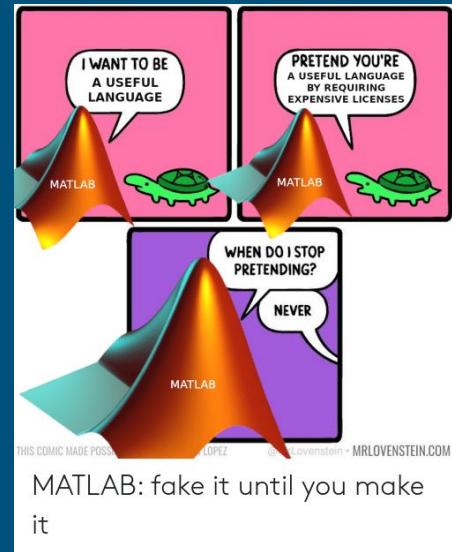
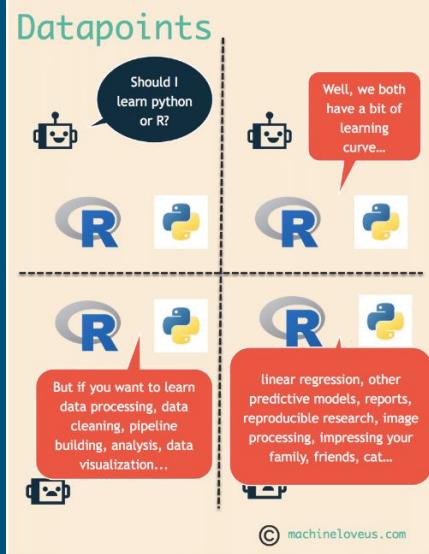
The slide features the Animate U logo with a graduation cap icon and the tagline "Spread Wisdom in Our World". It is titled "Chapter 1: Introduction to Python Programming".

Chapter 1:
Introduction to
Python Programming

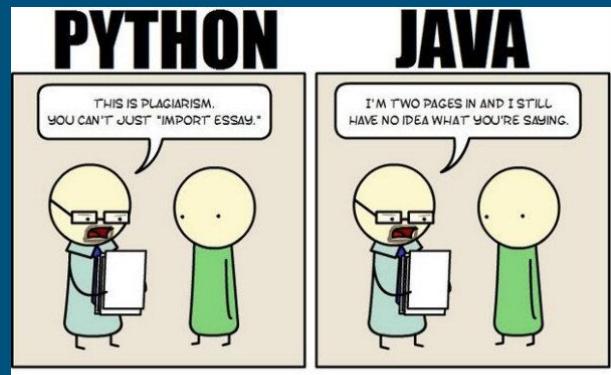
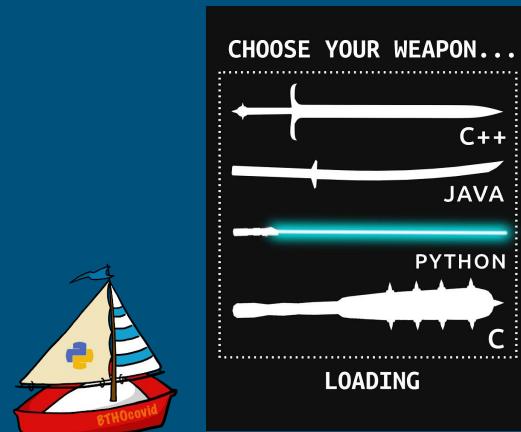


Why Python ?





MATLAB: fake it until you make it



My 1st Python code (use version 3.x)



IDLE Shell
(locally)

```
Python 3.5.2 (v3.5.2:4def2a2901a5, Jun 26 2016, 10:47:25)
[GCC 4.2.1 (Apple Inc. build 5666) (dot 3)] on darwin
Type "copyright", "credits" or "license()" for more information.

>>> WARNING: The version of Tk/Tk (8.5.9) in use may be unstable.
Visit http://www.python.org/download/mac/tcltk/ for current information.
print("Howdy World!")
Howdy World!
>>> |
```

Ln: 8 Col: 4

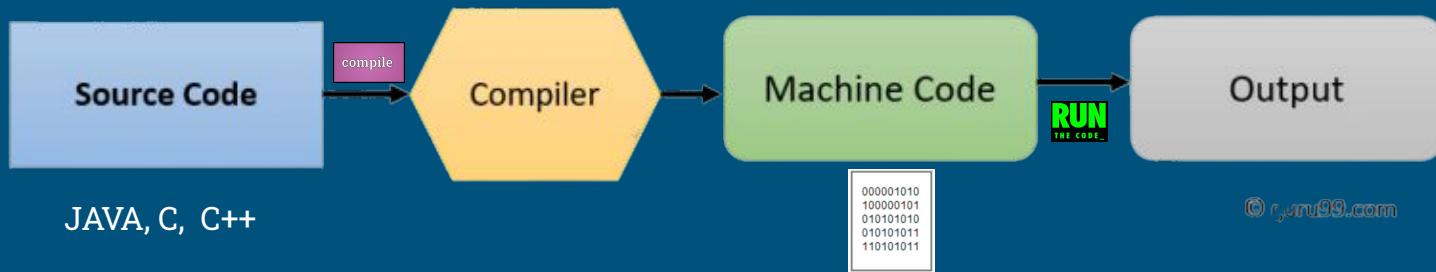
Google Colab
(online)

▶ print("Howdy World!")

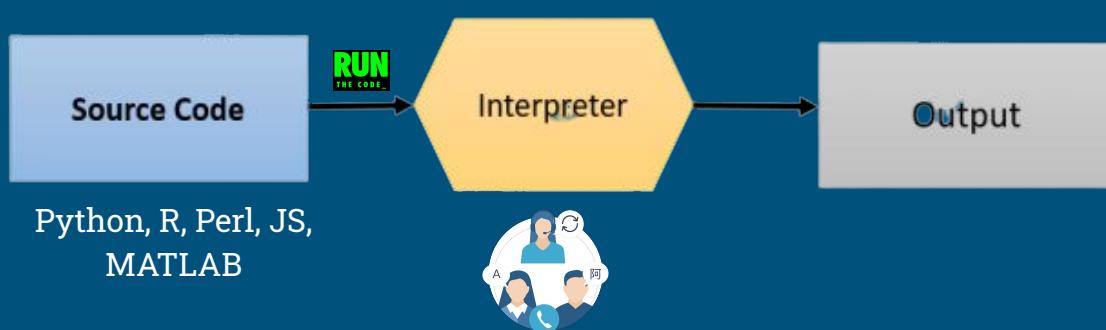
⇨ Howdy World! CO



➤ Journey of a Source code



© guru99.com



Code Styles

Comments



```
# This is a single line comment  
print("I am the first statement")
```



I am the first statement



```
# Multi-line comment  
# Comments are not considered code statements  
# Comments are ignored (not executed) when code runs  
# Comments help programmers to read, understand, and review the code  
print("I am still the first statement")
```



I am still the first statement



MY PYTHON CODE

```
#include<stdio.h>  
#include<conio.h>  
#include<ctype.h>  
#include<string.h>  
#include<iostream.h>  
#include<dos.h>  
#include<limits.h>  
print("Python")
```



C PROGRAMMERS

Why did this idiot include all those header files to print a string?



ME

is the syntax for comments in Python



Code Styles

Indentation



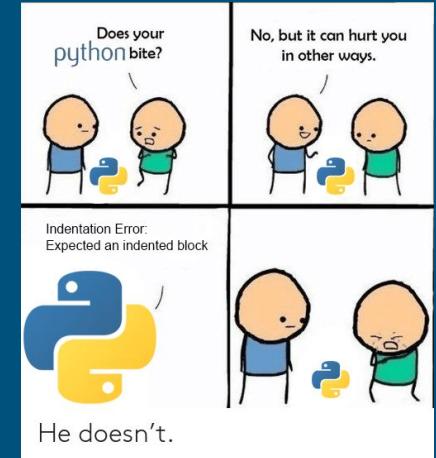
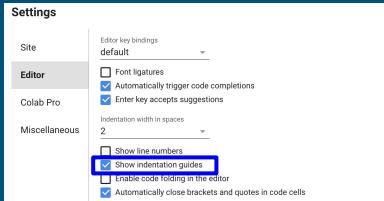
```
# define a calculator function
def calculator(a, b, key):
    print("This is a calculator")

    if(key == '+'): # if value of key is +, perform addition
        print("Sum = ",a+b)
    elif(key == '-'): # else if value of key is -, perform subtraction
        print("Difference = ",a-b)

# call calculator function with values 10, 12 and -
calculator(10, 12, '-')


```

→ This is a calculator
Difference = -2



- `def` is **Level 0** indentation
- `print("This is a calculator")`, `if`, and `elif` are **Level 1** (single tab) indentation within `def` block
- `print("Sum = ",a+b)` is **Level 2** indentation (2 tabs from `def` block or single tab from `if` block)
- `print("Difference = ",a-b)` is **Level 2** indentation (2 tabs from `def` block or single tab from `elif` block)
- `calculator(10, 12, '-')` is **Level 0** i.e. same level as `def` block

Next,

Getting Google Colab Fired Up and Sailing





ShiPy

—

Learn to Py while Shelter-in-Place

Getting Google Colab Fired Up and Sailing

co



A volunteering educational initiative during COVID-19

ShiP Crew



JD



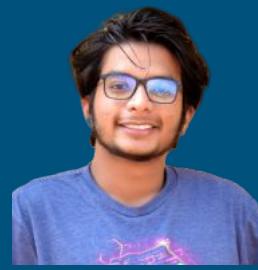
Teddy



Chinmay



Pratik



Siddharth



Umang



Waseem



A volunteering educational initiative during COVID-19



Setting up to Sail

- Computer
- Good Internet Connection
- Google Account The Google 'G' logo, which is a multi-colored letter 'G' composed of several smaller colored letters.
- Creative Mind



Step 1: Switch on your Computer



Step 2: Open your Browser



Mozilla Firefox



Google Chrome



Microsoft Edge



Safari

Step 3: Open Google Colab



<https://colab.research.google.com/>





Step 4: Login in using Google Account

Welcome To Colaboratory

File Edit View Insert Runtime Tools Help

Table of contents

- Getting started
- Data science
- Machine learning
- More Resources
- Machine Learning Examples
- Section

+ Code + Text Copy to Drive

Connect Editing

Login Button

Sign In

What is Colaboratory?

Colaboratory, or "Colab" for short, allows you to write and execute Python in your browser, with

- Zero configuration required
- Free access to GPUs
- Easy sharing

Whether you're a student, a data scientist or an AI researcher, Colab can make your work easier. Watch [Introduction to Colab](#) to learn more, or just get started below!

Getting started

The document you are reading is not a static web page, but an interactive environment called a **Colab notebook** that lets you write and execute code.

For example, here is a **code cell** with a short Python script that computes a value, stores it in a variable, and prints the result:

```
[ ] seconds_in_a_day = 24 * 60 * 60
seconds_in_a_day
```

86400

To execute the code in the above cell, select it with a click and then either press the play button to the left of the code, or use the keyboard shortcut "Command/Ctrl+Enter". To edit the code, just click the cell and start editing.

Variables that you define in one cell can later be used in other cells:

```
[ ] seconds_in_a_week = 7 * seconds_in_a_day
seconds_in_a_week
```

604800

Colab notebooks allow you to combine **executable code** and **rich text** in a single document, along with **images**, **HTML**, **LaTeX** and more. When you create your own Colab notebooks, they are stored in your Google Drive account. You can easily share your Colab notebooks with co-workers or friends, allowing them to comment on your notebooks or even edit them. To learn more, see [Overview of Colab](#). To create a new Colab notebook you can use the File menu above, or use the following link: [create a new Colab notebook](#).



Create New Colab Notebook

Welcome To Colaboratory

File Edit View Insert Runtime Tools Help

GD Share ⚙️

Table of contents

- Getting started
- Data science
- Machine learning
- More Resources
- Machine Learning Examples
- Section

+ Code + Text Copy to Drive

Connect Editing

What is Colaboratory?

Colaboratory, or "Colab", is a free Jupyter notebook environment that requires nothing more than browser access.

- Zero configuration
- Free access to GPU
- Easy sharing

Whether you're a student just getting started below!

Getting started

The document you are currently viewing is a Colaboratory notebook.

For example, here is a cell containing executable Python code:

```
[ ] seconds_in_a_day = 24 * 60 * 60
```

To execute the code in this cell, press the shortcut "Command/Control+Enter".

Variables that you define in this notebook can be referenced later:

```
[ ] seconds_in_a_week = 7 * seconds_in_a_day
```

Colab notebooks allow you to combine **executable code** and rich text in a single document, along with **images**, **HTML**, **LaTeX** and more. When you create your own Colab notebooks, they are stored in your Google Drive account. You can easily share your Colab notebooks with co-workers or friends, allowing them to comment on your notebooks or even edit them. To learn more, see [Overview of Colab](#). To create a new Colab notebook, click the button below.

Examples Recent Google Drive GitHub Upload

Filter notebooks

Title	First opened	Last opened	⋮
Welcome To Colaboratory	6 days ago	0 minutes ago	⋮
Untitled1.ipynb	20 hours ago	20 hours ago	⋮
HW3_STUDENT	4 days ago	4 days ago	⋮
Untitled0.ipynb	6 days ago	6 days ago	⋮
BlazingSQL_vs_PySpark_Netflow.ipynb	Aug 6, 2019	6 days ago	⋮

Click This → NEW NOTEBOOK CANCEL





New Notebook Opened

A screenshot of a Jupyter Notebook interface. The title bar shows 'Untitled2.ipynb'. The menu bar includes File, Edit, View, Insert, Runtime, Tools, Help, and a status message 'All changes saved'. The toolbar on the right includes Comment, Share, and settings. The main area shows a single code cell with a play button. The sidebar on the left has '+ Code' and '+ Text' buttons. The bottom left corner features a small sailboat icon with the text '5THOovid'.





Add Text cell on Notebook

A screenshot of a Jupyter Notebook interface titled "Hello World". The top navigation bar includes File, Edit, View, Insert, Runtime, Tools, Help, and a message indicating "All changes saved". On the right side, there are buttons for Comment, Share, and Settings. Below the menu, there are two buttons: "+ Code" and "+ Text", with "+ Text" being highlighted by a blue rectangular box. A blue arrow points from this box down to a larger blue box containing the text "Add Text cell".



A screenshot of the same Jupyter Notebook interface after adding a Text cell. The "+ Text" button is no longer highlighted. The notebook now contains two cells: a code cell and a text cell. The code cell contains the Python code "# This is my First Program". The text cell contains the text "This is my First Program". The top navigation bar and right-side controls remain the same.



Run the Code cell

co

A screenshot of a code editor window titled "Hello World". The menu bar includes File, Edit, View, Insert, Runtime, Tools, and Help. The toolbar on the right includes Comment, Share, Connect, and Editing. The main area shows a single text cell containing the text "This is my Program".

A screenshot of a code editor window titled "Hello World". The menu bar includes File, Edit, View, Insert, Runtime, Tools, Help, and All changes saved. The toolbar on the right includes Comment, Share, Connect, and Editing. The main area shows a code cell containing the Python code "print('Hello World')". A blue box highlights the "Run the code cell" button, which is located below the code cell and has a blue arrow pointing to it from the bottom left.

A screenshot of a code editor window titled "Hello World". The menu bar includes File, Edit, View, Insert, Runtime, Tools, Help, and All changes saved. The toolbar on the right includes Comment, Share, Connect, RAM, Disk, and Editing. The main area shows the code cell "print('Hello World')". Below it, the output cell displays the result "Hello World". A blue checkmark icon is visible on the toolbar.

Assignment 0

[CLICK HERE](#)



Next

Offline Google Colab aka Jupyter Notebook





OPTIONAL

ShiPy

Learn to Py while Shelter-in-Place

Getting Jupyter Fired Up and Sailing



A volunteering educational initiative during COVID-19



ShiP Crew



JD



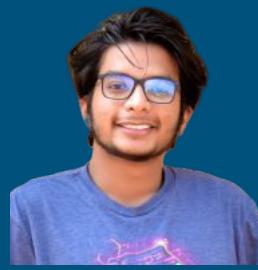
Teddy



Chinmay



Pratik



Siddharth



Umang



Waseem



A volunteering educational initiative during COVID-19



Setting up to Sail

- Computer
 - Good Internet Connection
 - Google Account 
 - Creative Mind
-



Step 1: Switch on your Computer



Step 2: Open your Browser



Mozilla Firefox



Google Chrome



Microsoft Edge



Safari

Step 2: Download Anaconda

The screenshot shows the Anaconda download page for macOS. At the top, there are links for Windows, macOS, and Linux. A blue box labeled "Choose Correct Operating System" points to the Linux link. Another blue box labeled "Always Install Python 3.xx version. For this course we will follow this version." points to the "Python 3.7 version" section. This section contains a "Download" button and links for "64-Bit Graphical Installer (442 MB)" and "64-Bit Command Line Installer (430 MB)". To the right, there is a "Python 2.7 version" section with its own "Download" button and links for "64-Bit Graphical Installer (637 MB)" and "64-Bit Command Line Installer (409 MB)".

Always Install Python 3.xx version. For this course we will follow this version.

Choose Correct Operating System

Anaconda 2020.02 for macOS Installer

Python 3.7 version

Download

64-Bit Graphical Installer (442 MB)
64-Bit Command Line Installer (430 MB)

Python 2.7 version

Download

64-Bit Graphical Installer (637 MB)
64-Bit Command Line Installer (409 MB)

<https://www.anaconda.com/distribution/#download-section>



Step 3: Follow Setup for Respective OS

- For Windows

<https://youtu.be/YQWAiUhZJX8>



- For Mac

<https://youtu.be/jhFyTv9vLi4?t=76>

Watch until 3:21



Mac™OS

- For Linux

https://youtu.be/DY0DB_NwEu0?t=51

Watch until 7:16



No Internet No Problem Keep Coding





Step 4: Open Anaconda Navigator

ANACONDA NAVIGATOR

Applications on base (root) Channels Refresh

JupyterLab 1.0.2 An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture. [Launch](#)

jupyter Notebook 6.0.0 Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis. [Launch](#)

IPy[ly] 4.5.1 PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more. [Launch](#)

Spyder 3.3.6 Scientific PYthon Development Environment. Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features. [Launch](#)

Glueviz 0.15.2 Multidimensional data visualization across files. Explore relationships within and among related datasets. [Install](#)

Orange 3 3.23.1 Component based data mining framework. Data visualization and data analysis for novice and expert. Interactive workflows with a large toolbox. [Install](#)

RStudio 1.1.456 A set of integrated tools designed to help you be more productive with R. Includes R essentials and notebooks. [Install](#)

VS Code 1.44.0 Streamlined code editor with support for development operations like debugging, task running and version control. [Install](#)

[Documentation](#) [Developer Blog](#)

[Twitter](#) [YouTube](#) [GitHub](#)





Step 5: Launch Jupyter Notebook

ANACONDA NAVIGATOR

Applications on base (root) Channels Refresh

Jupyter Notebook

An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.

Launch

Click on it

Glueviz

Multidimensional data visualization across files. Explore relationships within and among related datasets.

Install

Orange 3

Component based data mining framework. Data visualization and data analysis for novice and expert. Interactive workflows with a large toolbox.

Install

RStudio

A set of integrated tools designed to help you be more productive with R. Includes R essentials and notebooks.

Install

VS Code

Streamlined code editor with support for development operations like debugging, task running and version control.

Install

Documentation Developer Blog

Twitter YouTube GitHub



Upgrade Now Sign in to Anaconda Cloud



Jupyter Notebook Landing Page

jupyter

Files Running Clusters

Select items to perform actions on them.

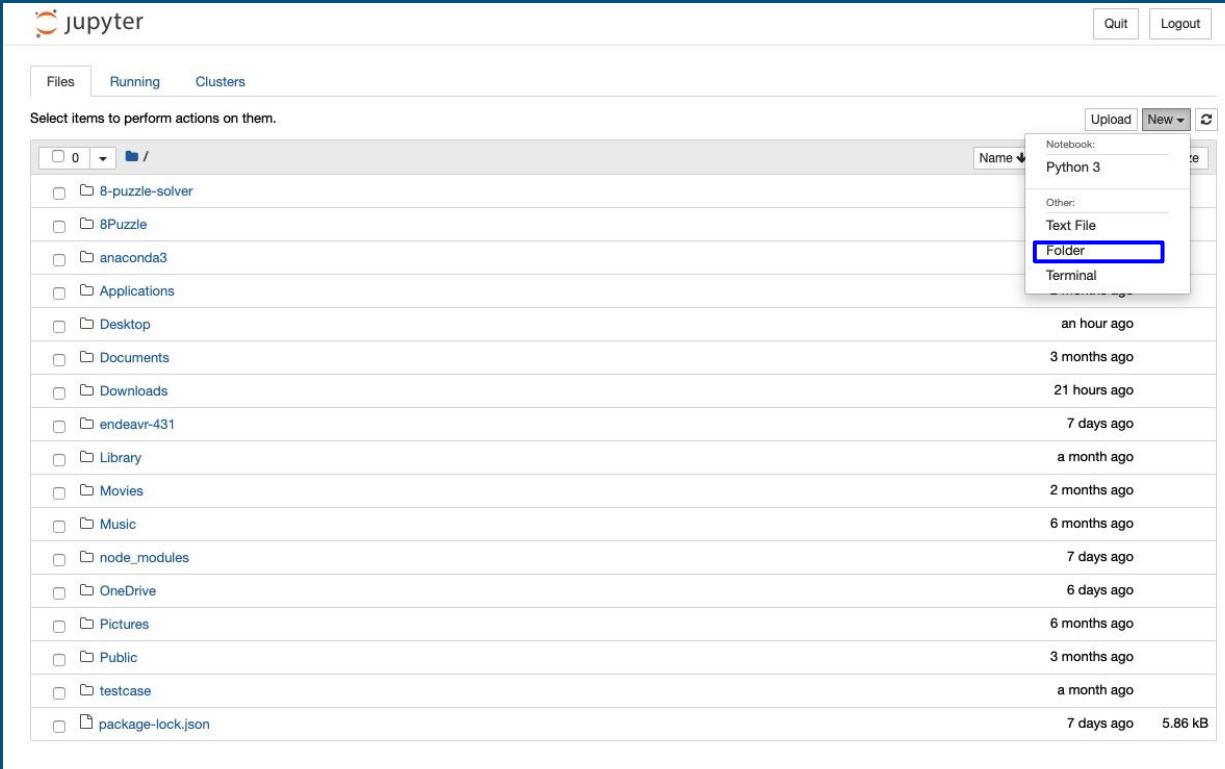
Upload New

	Name	Last Modified	File size
<input type="checkbox"/>	0		
<input type="checkbox"/>	8-puzzle-solver	2 months ago	
<input type="checkbox"/>	8Puzzle	2 months ago	
<input type="checkbox"/>	anaconda3	seconds ago	
<input type="checkbox"/>	Applications	2 months ago	
<input type="checkbox"/>	Desktop	an hour ago	
<input type="checkbox"/>	Documents	3 months ago	
<input type="checkbox"/>	Downloads	21 hours ago	
<input type="checkbox"/>	endeavr-431	7 days ago	
<input type="checkbox"/>	Library	a month ago	
<input type="checkbox"/>	Movies	2 months ago	
<input type="checkbox"/>	Music	6 months ago	
<input type="checkbox"/>	node_modules	7 days ago	
<input type="checkbox"/>	OneDrive	6 days ago	
<input type="checkbox"/>	Pictures	6 months ago	
<input type="checkbox"/>	Public	3 months ago	
<input type="checkbox"/>	testcase	a month ago	
<input type="checkbox"/>	package-lock.json	7 days ago	5.86 kB



A small illustration of a sailboat with a single mast and a single sail. The sail has a blue cross on it. The boat is red with a white hull. The name "STHOovid" is written on the side of the boat.

Step 6: Create New Folder



The screenshot shows the Jupyter Notebook interface. On the left, there's a sidebar with a file tree containing various folders like '8-puzzle-solver', 'anaconda3', and 'Downloads'. The main area shows a list of files and notebooks, each with a timestamp. A context menu is open over a notebook named 'Python 3', with 'Folder' selected. The menu also includes options for 'Text File' and 'Terminal'. The top navigation bar has tabs for 'Files', 'Running', and 'Clusters', along with 'Upload' and 'New' buttons. The status bar at the bottom shows the path '/home/steve' and the file 'package-lock.json' with a size of 5.86 kB.

Select items to perform actions on them.

Files Running Clusters

Upload New

Notebook: Python 3

Other:

- Text File
- Folder**
- Terminal

Name	Last Modified	Size
0	an hour ago	
8-puzzle-solver	3 months ago	
8Puzzle	21 hours ago	
anaconda3	7 days ago	
Applications	a month ago	
Desktop	2 months ago	
Documents	6 months ago	
Downloads	7 days ago	
endeavor-431	6 days ago	
Library	6 months ago	
Movies	3 months ago	
Music	a month ago	
node_modules	7 days ago	
OneDrive	6 days ago	
Pictures	6 months ago	
Public	3 months ago	
testcase	7 days ago	
package-lock.json	5.86 kB	



Step 7: Rename Folder to Ship.py or anything

jupyter

Files Running Clusters

Rename Move

Upload New

	Name	Last Modified	File size
<input type="checkbox"/>	8-puzzle-solver	2 months ago	
<input type="checkbox"/>	8Puzzle	2 months ago	
<input type="checkbox"/>	anaconda3	seconds ago	
<input type="checkbox"/>	Applications	2 months ago	
<input type="checkbox"/>	Desktop	an hour ago	
<input type="checkbox"/>	Documents	3 months ago	
<input type="checkbox"/>	Downloads	21 hours ago	
<input type="checkbox"/>	endeavr-431	7 days ago	
<input type="checkbox"/>	Library	a month ago	
<input type="checkbox"/>	Movies	2 months ago	
<input type="checkbox"/>	Music	6 months ago	
<input type="checkbox"/>	node_modules	7 days ago	
<input type="checkbox"/>	OneDrive	6 days ago	
<input type="checkbox"/>	Pictures	6 months ago	
<input type="checkbox"/>	Public	3 months ago	
<input type="checkbox"/>	testcase	a month ago	
<input checked="" type="checkbox"/>	Untitled Folder	seconds ago	
<input type="checkbox"/>	package-lock.json	7 days ago	5.86 kB



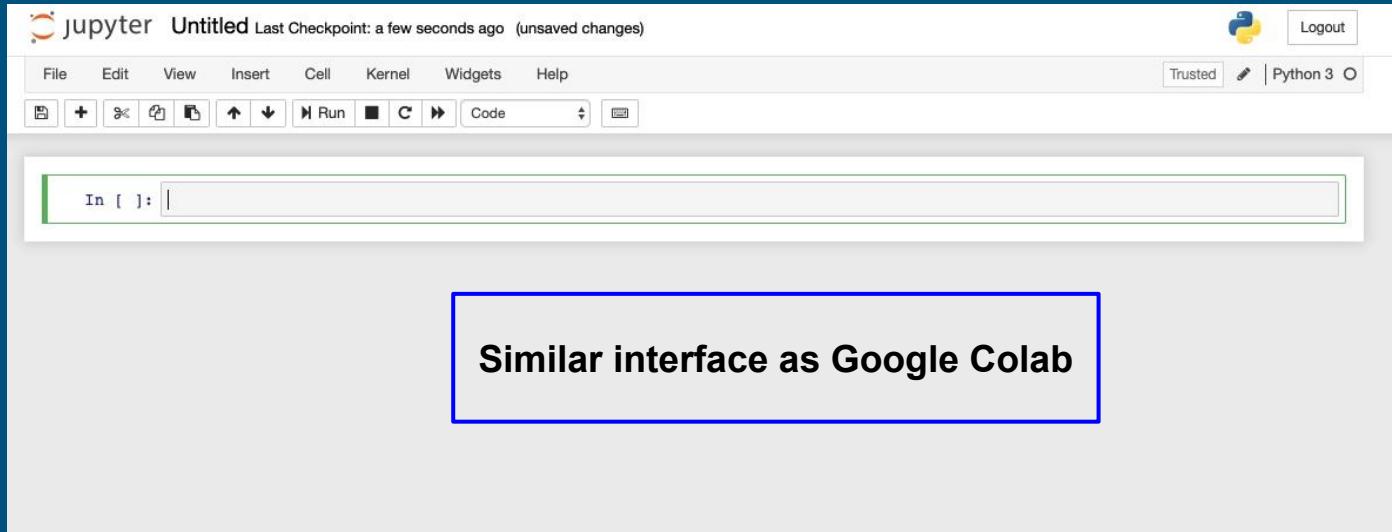
Step 8: Visit the folder and create a new notebook



The screenshot shows the Jupyter Notebook interface. At the top, there is a navigation bar with the Jupyter logo, 'Files' (selected), 'Running', 'Clusters', 'Quit', and 'Logout'. Below the navigation bar, a message says 'Select items to perform actions on them.' A file list shows '0' items and a folder named 'Ship.py'. To the right, a modal window titled 'New' is open, showing a dropdown menu for 'Notebook:' with 'Python 3' selected (highlighted with a blue border). Other options in the 'Notebook:' dropdown are 'Text File', 'Folder', and 'Terminal'. Below the dropdown, there are sections for 'Other:' with options 'Text File', 'Folder', and 'Terminal'.



Landing Page of new notebook



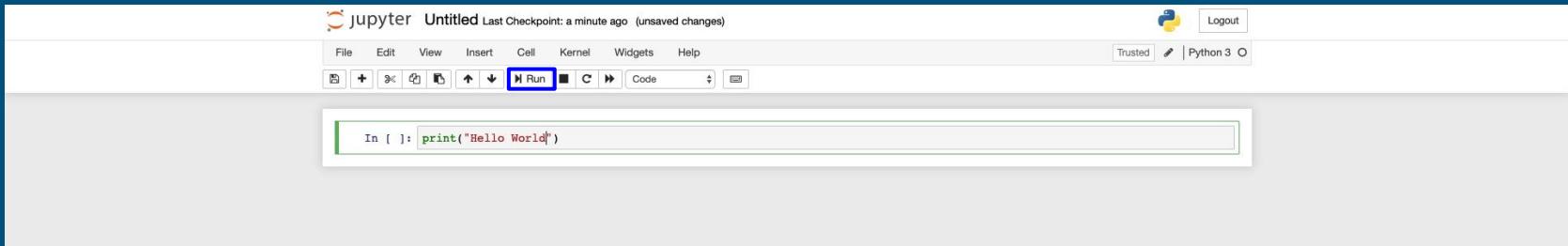
A screenshot of the Jupyter Notebook interface. At the top, there's a header bar with the "jupyter" logo, the title "Untitled" (Last Checkpoint: a few seconds ago (unsaved changes)), and user information (Trusted, Logout). Below the header is a toolbar with various icons for file operations like opening, saving, and running cells. The main area shows a code cell starting with "In []:" followed by a large empty text input field. A blue rectangular box highlights the text input field. Inside this box, the text "Similar interface as Google Colab" is displayed in bold black font.

In []: |

Similar interface as Google Colab

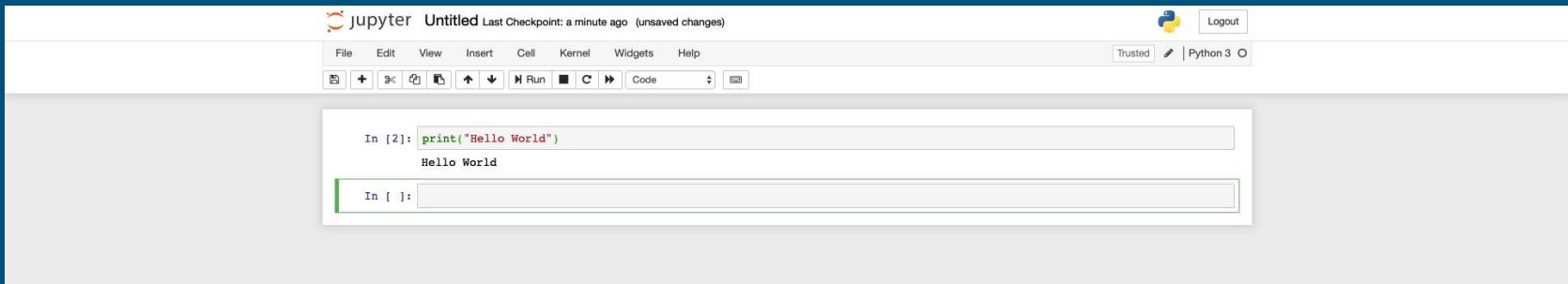


Step 9: Print Hello World and Run



A screenshot of a Jupyter Notebook interface. The title bar says "jupyter Untitled Last Checkpoint: a minute ago (unsaved changes)". The toolbar includes File, Edit, View, Insert, Cell, Kernel, Widgets, Help, and a Python 3 kernel icon. Below the toolbar is a toolbar with icons for cell operations like Run, Cell, Kernel, and Help. A large green arrow points downwards from this interface to the next one.

```
In [ ]: print("Hello World")
```



A screenshot of a Jupyter Notebook interface, identical to the one above but showing the result of the execution. The code cell "In []: print('Hello World')" has been run, and the output "Hello World" is displayed below it. A large green arrow points downwards from the previous interface to this one.

```
In [2]: print("Hello World")
Hello World
```

```
In [ ]:
```



END

