

Python for Accounting Applications

Foundations of Python Programming

1121PAA03

ACC2, NTPU (M5265) (Fall 2023)

Wed 6, 7, 8, (14:10-17:00) (9:10-12:00) (B3F10)



Min-Yuh Day, Ph.D,
Associate Professor

Institute of Information Management, National Taipei University

<https://web.ntpu.edu.tw/~myday>

2023-09-27



Syllabus

Week	Date	Subject/Topics
1	2023/09/13	Introduction to Python for Accounting Applications
2	2023/09/20	Python Programming and Data Science
3	2023/09/27	Foundations of Python Programming
4	2023/10/04	Data Structures
5	2023/10/11	Control Logic and Loops
6	2023/10/18	Functions and Modules
7	2023/10/25	Files and Exception Handling
8	2023/11/01	Midterm Project Report

Syllabus

Week Date Subject/Topics

9 2023/11/08 Data Analytics and Visualization with Python

10 2023/11/15 Obtaining Data From the Web with Python

11 2023/11/22 Statistical Analysis with Python

12 2023/11/29 Machine Learning with Python

**13 2023/12/06 Text Analytics with Python and
Large Language Models (LLMs)**

14 2023/12/13 Applications of Accounting Data Analytics with Python

15 2023/12/20 Applications of ESG Data Analytics with Python

16 2023/12/27 Final Project Report

Foundations of Python Programming

Outline

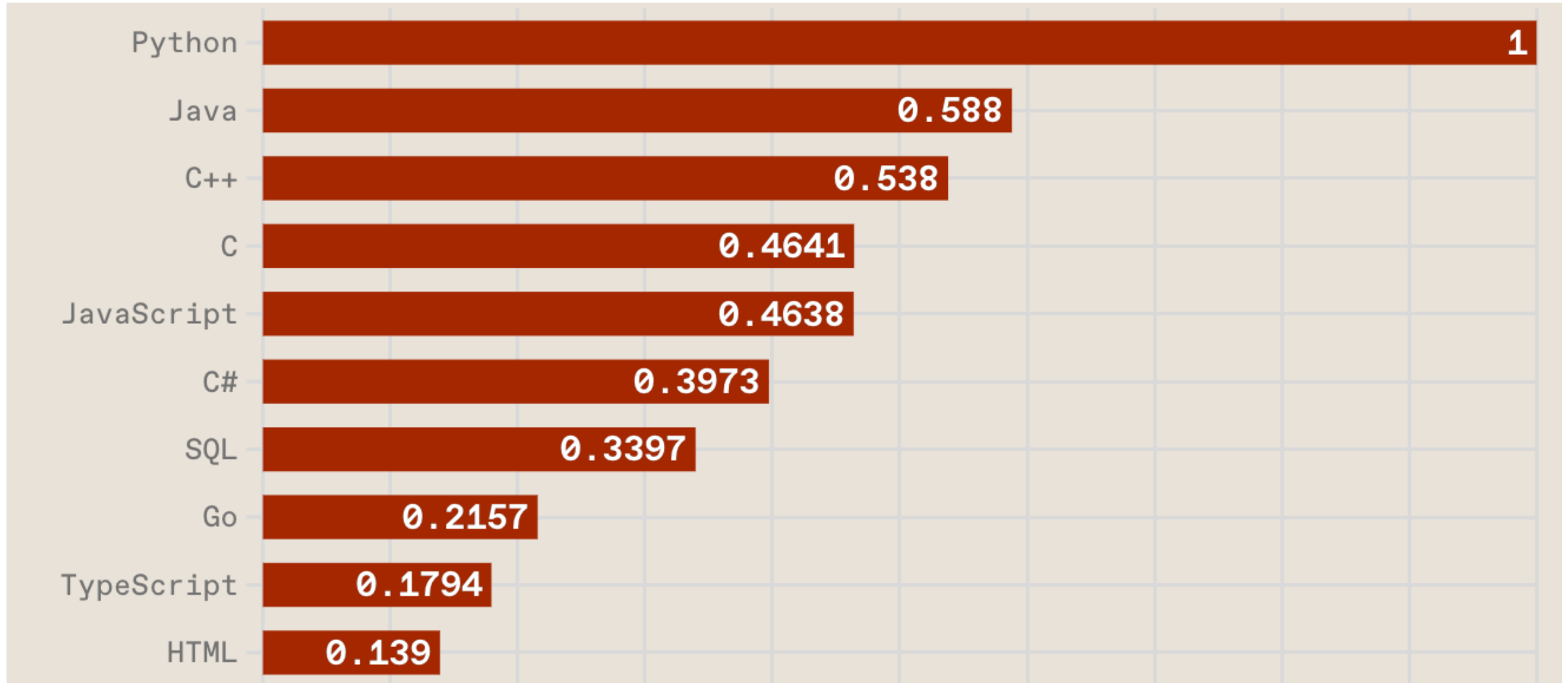
- **Python Syntax**
 - **Python Comments**
- **Python Variables**
- **Python Data Types**
 - **Python Numbers**
 - **Python Casting**
 - **Python Strings**
- **Python Operators**
- **Python Booleans**



Python

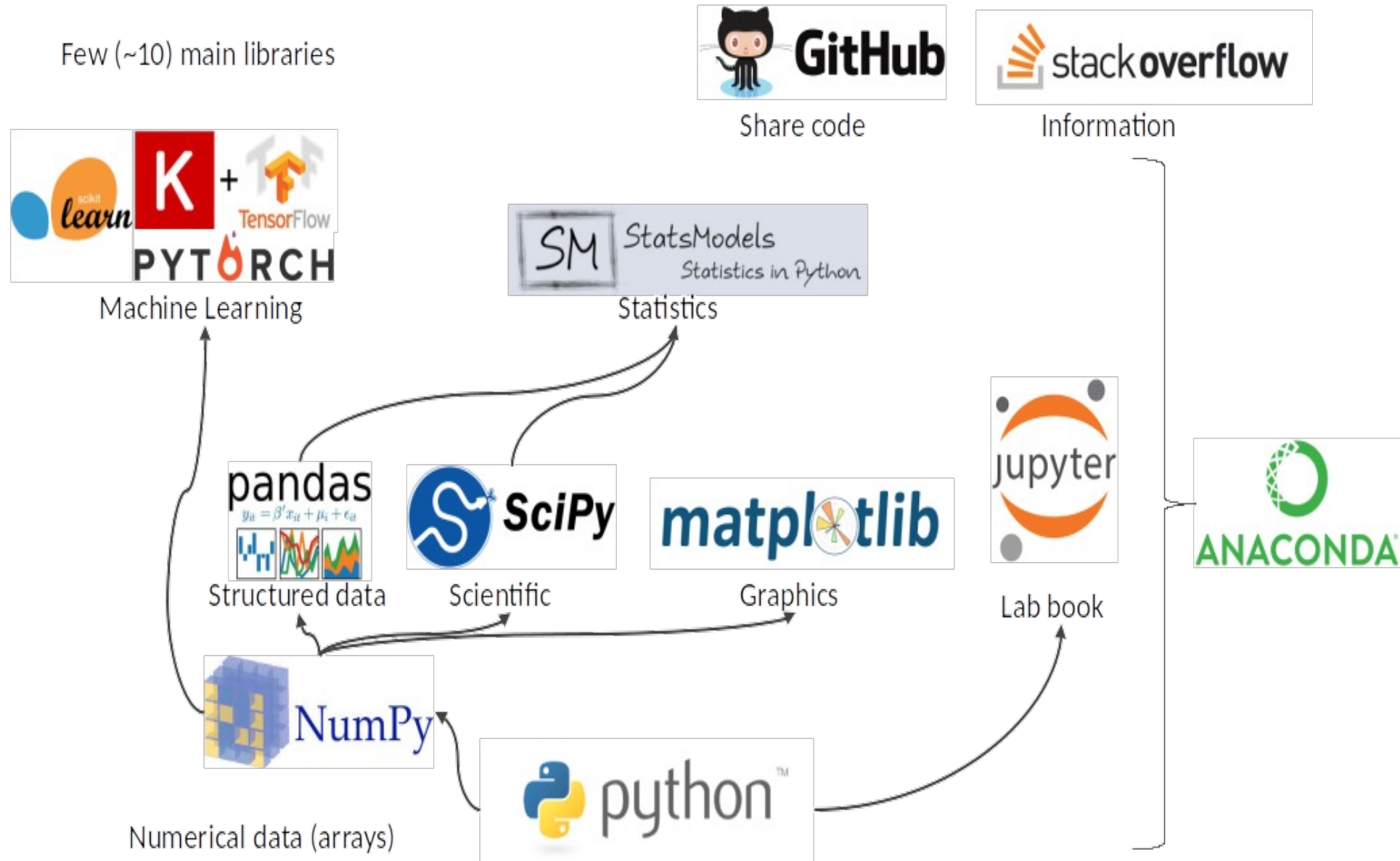
Programming

Top Programming Languages

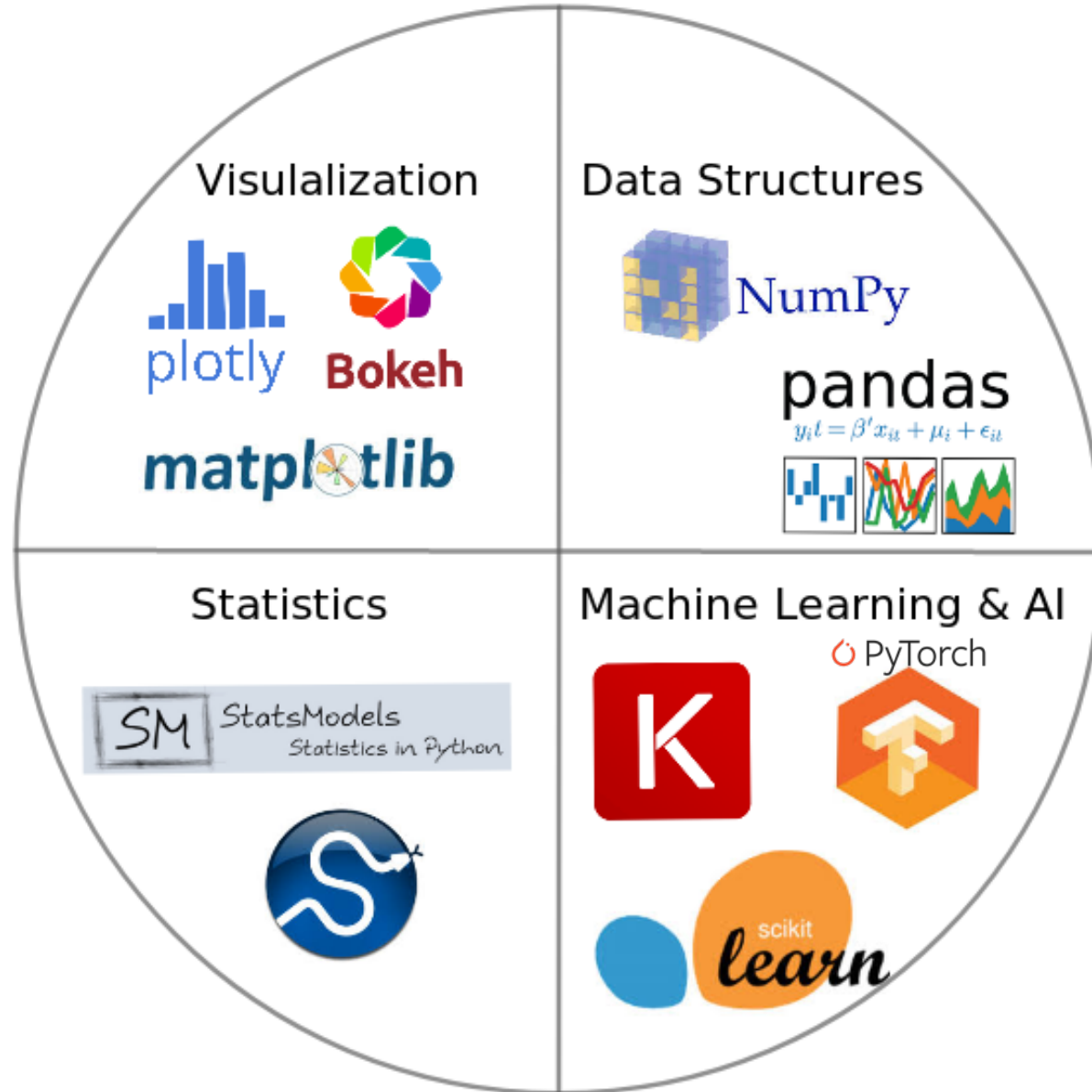


Python is an
interpreted,
object-oriented,
high-level
programming language
with
dynamic semantics.

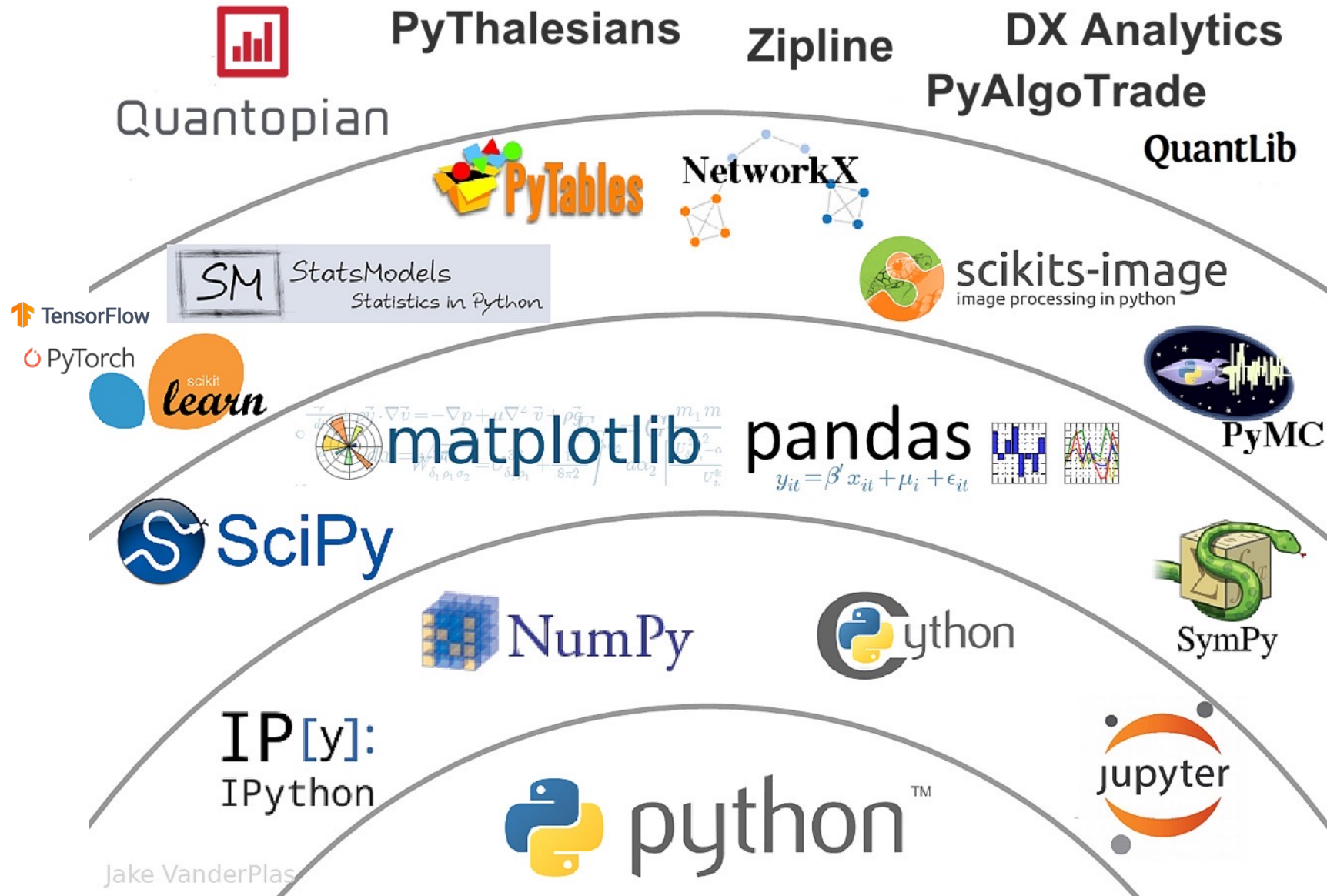
Python Ecosystem for Data Science



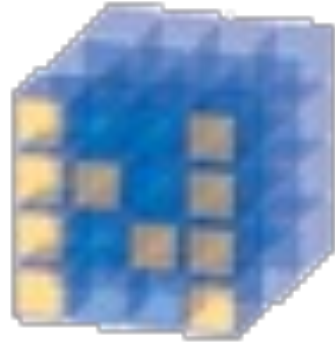
Python Ecosystem for Data Science



The Quant Finance PyData Stack



NumPy



NumPy

Base

N-dimensional array
package

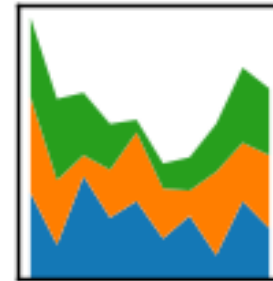
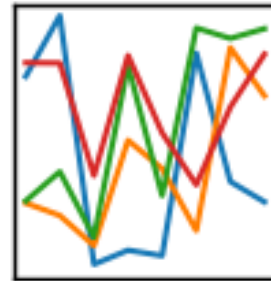
Python
matplotlib
matplotlib

Python

Pandas

pandas

$$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$$



- Python Tutorial
- Python HOME**
- Python Intro
- Python Get Started
- Python Syntax
- Python Comments
- Python Variables
- Python Data Types
- Python Numbers
- Python Casting
- Python Strings
- Python Booleans
- Python Operators
- Python Lists
- Python Tuples
- Python Sets
- Python Dictionaries
- Python If...Else
- Python While Loops
- Python For Loops
- Python Functions

Python Tutorial

[← Home](#)

[Next >](#)

Learn Python

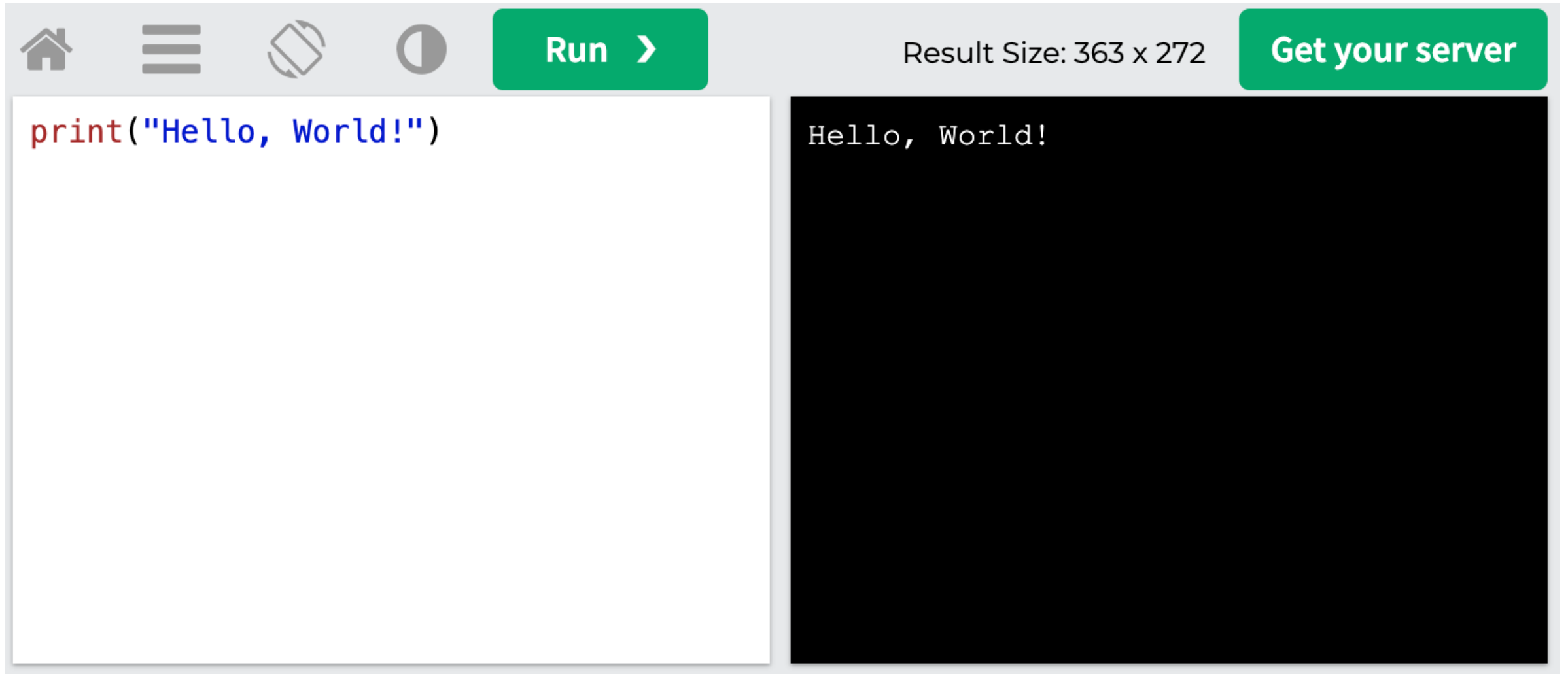
Python is a popular programming language.
Python can be used on a server to create web applications.

[Start learning Python now »](#)

Learning by Examples

With our "Try it Yourself" editor, you can edit Python code and view the result.

W3Schools Python: Try Python

A screenshot of the W3Schools Python Try Python interface. The interface has a light gray header with navigation icons (home, menu, refresh, moon) and a green 'Run >' button. To the right of the 'Run' button, it says 'Result Size: 363 x 272' and a green button that says 'Get your server'. Below the header, there is a white text area on the left containing the Python code `print("Hello, World!")` and a black output area on the right displaying the result 'Hello, World!'.

LearnPython.org



learnpython.org

[Home](#)

[About](#)

[Certify](#)

[More Languages](#) ▾

[Python](#)

[Java](#)

[HTML](#)

[Go](#)

[C](#)

[C++](#)

[JavaScript](#)

[PHP](#)

[Shell](#)

[C#](#)

[Perl](#)

[Ruby](#)

[Scala](#)

[SQL](#)

Get started learning Python with [DataCamp's](#) free [Intro to Python tutorial](#). Learn Data Science by completing interactive coding challenges and watching videos by expert instructors. [Start Now!](#)

Ready to take the test? Head onto [LearnX](#) and get your Python Certification!

This site is generously supported by [DataCamp](#). DataCamp offers online interactive [Python Tutorials](#) for Data Science. Join **11 millions** other learners and get started learning Python for data science today!

Good news! You can save 25% off your Datacamp annual subscription with the code [LEARNPYTHON23ALE25](#) - [Click here to redeem your discount!](#)

Welcome

Welcome to the LearnPython.org interactive Python tutorial.

Whether you are an experienced programmer or not, this website is intended for everyone who wishes to learn the Python programming language.

You are welcome to join our group on [Facebook](#) for questions, discussions and updates.

After you complete the tutorials, you can get certified at [LearnX](#) and add your certification to your LinkedIn profile.

Just click on the chapter you wish to begin from, and follow the instructions. Good luck!

<https://www.learnpython.org/>

Google's Python Class

Google for Education > Python

Search

English



Filter

Overview

Python Set Up

Python Intro

Strings

Lists

Sorting

Dicts and Files

Regular Expressions

Utilities

Lecture Videos

1.1 Introduction, strings [↗](#)

1.2 Lists and sorting [↗](#)

1.3 Dicts and files [↗](#)

2.1 Regular expr [↗](#)

2.2 Utilities [↗](#)

2.3 Utilities urllib [↗](#)

2.4 Conclusions [↗](#)

Python Exercises



Home > Products > Google for Education > Python

Was this helpful? [👍](#) [🗨️](#)

Google's Python Class [📄](#)

Welcome to Google's Python Class -- this is a free class for people with a little bit of programming experience who want to learn Python. The class includes written materials, lecture videos, and lots of code exercises to practice Python coding. These materials are used within Google to introduce Python to people who have just a little programming experience. The first exercises work on basic Python concepts like strings and lists, building up to the later exercises which are full programs dealing with text files, processes, and http connections. The class is geared for people who have a little bit of programming experience in some language, enough to know what a "variable" or "if statement" is. Beyond that, you do not need to be an expert programmer to use this material.

To get started, the Python sections are linked at the left -- [Python Set Up](#) to get Python installed on your machine, [Python Introduction](#) for an introduction to the language, and then [Python Strings](#) starts the coding material, leading to the first exercise. The end of each written section includes a link to the code exercise for that section's material. The lecture videos parallel the written materials, introducing Python, then strings, then first exercises, and so on. At Google, all this material makes up an intensive 2-day class, so the videos are organized as the day-1 and day-2 sections.

This material was created by [Nick Parlante](#) working in the engEDU group at Google. Special thanks for the help from my Google colleagues John Cox, Steve Glassman, Piotr Kaminski, and Antoine Picard. And finally thanks to Google and my director Maggie Johnson for the enlightened generosity to put these materials out on the internet for free under the [Creative Commons Attribution 2.5](#) license -- share and enjoy!

<https://developers.google.com/edu/python>

Google Colab

Table of contents

- Getting Started
- Highlighted Features
 - TensorFlow execution
- GitHub
- Visualization
- Forms
- Examples
- Local runtime support

SECTION

Welcome to Colaboratory!

Colaboratory is a free Jupyter notebook environment that requires no setup and runs entirely in the cloud. See our [FAQ](#) for more info.

Getting Started

- [Overview of Colaboratory](#)
- [Loading and saving data: Local files, Drive, Sheets, Google Cloud Storage](#)
- [Importing libraries and installing dependencies](#)
- [Using Google Cloud BigQuery](#)
- [Forms, Charts, Markdown, & Widgets](#)
- [TensorFlow with GPU](#)
- [Machine Learning Crash Course: Intro to Pandas & First Steps with TensorFlow](#)

Highlighted Features

Seedbank

Looking for Colab notebooks to learn from? Check out [Seedbank](#), a place to discover interactive machine learning examples.

TensorFlow execution

Colaboratory allows you to execute TensorFlow code in your browser with a single click. The example below adds two matrices.

$$\begin{bmatrix} 1. & 1. & 1. \end{bmatrix} + \begin{bmatrix} 1. & 2. & 3. \end{bmatrix} = \begin{bmatrix} 2. & 3. & 4. \end{bmatrix}$$

Connect Google Colab in Google Drive

The image shows a browser window with the Google Drive interface. The address bar displays 'https://drive.google.com/drive/u/2/my-drive'. The main content area shows the 'My Drive' section with a 'Quick Access' sidebar on the left. The sidebar includes options like 'Computers', 'Shared with me', 'Recent', 'Starred', 'Trash', 'Backups', and 'Storage'. The 'Storage' section indicates '0 bytes of 15 GB used' and provides a link to 'UPGRADE STORAGE'. A 'New' button is located at the top left of the main content area. A red dashed box highlights the 'New' button and the 'My Drive' option in the sidebar. A dropdown menu is open from the 'New' button, showing options: 'New folder...', 'Upload files...', 'Upload folder...', 'Google Docs', 'Google Sheets', 'Google Slides', and 'More'. The 'More' option is highlighted with a red dashed box. A second dropdown menu is open from the 'More' option, showing options: 'Google Forms', 'Google Drawings', 'Google My Maps', 'Google Sites', and 'Connect more apps'. The 'Connect more apps' option is highlighted with a red dashed box. The 'Files' section is visible at the bottom, showing a 'Name' column header and a vertical scrollbar.

Google Colab

The screenshot shows the Google Drive interface with a 'Connect apps to Drive' dialog box open. The dialog box has a search bar at the top with 'colab' entered and highlighted by a red dashed border. Below the search bar, there are six app cards arranged in a 2x3 grid:

- ZIP Extractor**: Extract ZIP files to Google Drive. Extraction complete. 307,585 users.
- Lumin PDF**: Beautiful PDF Editor. 289,310 users.
- CloudConvert**: 373,161 users.
- Sejda**: Merge PDF - Split PDF - Sejda.com. 1106 reviews.
- DocHub**: Edit and Sign PDF Documents. 2,131,600 users.
- Google Forms**: 4,803,614 users.

The background shows the Google Drive sidebar with options like 'My Drive', 'Computers', 'Shared with me', 'Recent', 'Starred', 'Trash', 'Backups', and 'Storage'. The top navigation bar includes a search bar and various utility icons.

Google Colab

The image shows a browser window with the Google Drive interface. A modal dialog titled "Connect apps to Drive" is open, displaying search results for the app "Colaboratory". The search bar contains the text "colab". The Colaboratory app entry is highlighted with a red dashed border and includes a blue "+ CONNECT" button, also highlighted with a red dashed border. The app's details include its logo, name, URL, description, category, and rating.

My Drive - Google Drive x +

https://drive.google.com/drive/u/2/my-drive

Drive Search Drive

New

My Drive

Computers

Shared with me

Recent

Starred

Trash

Backups

Storage

0 bytes of 15 GB used
[UPGRADE STORAGE](#)

Get Backup and Sync for Mac

Access anywhere Share easily

Connect apps to Drive

All colab

Colaboratory
offered by <https://colab.research.google.com>
A data analysis tool that combines code, output, and descriptive text into one collaborative document.
Productivity
★★★★★ (195)

+ CONNECT

Name ↑

Connect Colaboratory to Google Drive

The screenshot shows the Google Drive web interface. A dialog box titled "Connect apps to Drive" is open, displaying a search for "colab". A confirmation message from Colaboratory is centered in the dialog, stating "Colaboratory was connected to Google Drive." and "Make Colaboratory the default app for files it can open" with a checked checkbox. An "OK" button is visible at the bottom right of the message. The background shows the Drive sidebar with categories like "My Drive", "Computers", "Shared with me", "Recent", "Starred", "Trash", "Backups", and "Storage". The top navigation bar includes the Drive logo, search bar, and various utility icons.

Google Colab

The image shows a browser window with the Google Drive interface. The address bar shows the URL `https://drive.google.com/drive/u/2/my-drive`. The main header includes the Drive logo, a search bar, and navigation icons. On the left, a sidebar contains navigation options like 'My Drive', 'Computers', 'Shared with me', 'Recent', 'Starred', 'Trash', 'Backups', and 'Storage'. The 'New' button is highlighted with a red dashed box, and its dropdown menu is open. The 'More' option in the dropdown is also highlighted with a red dashed box. A secondary dropdown menu is visible, listing various Google apps, with 'Colaboratory' highlighted by a red dashed box. Other visible elements include a 'Get Backup and Sync for Mac' notification and a 'Files' section with 'Store safely' and 'Sync seamlessly' options.

My Drive - Google Drive

https://drive.google.com/drive/u/2/my-drive

Drive

Search Drive

My Drive

Quick Access

New

My Drive

Computers

Shared with me

Recent

Starred

Trash

Backups

Storage

0 bytes of 15 GB used

UPGRADE STORAGE

Get Backup and Sync for Mac

New folder...

Upload files...

Upload folder...

Google Docs

Google Sheets

Google Slides

More

Google Forms

Google Drawings

Google My Maps

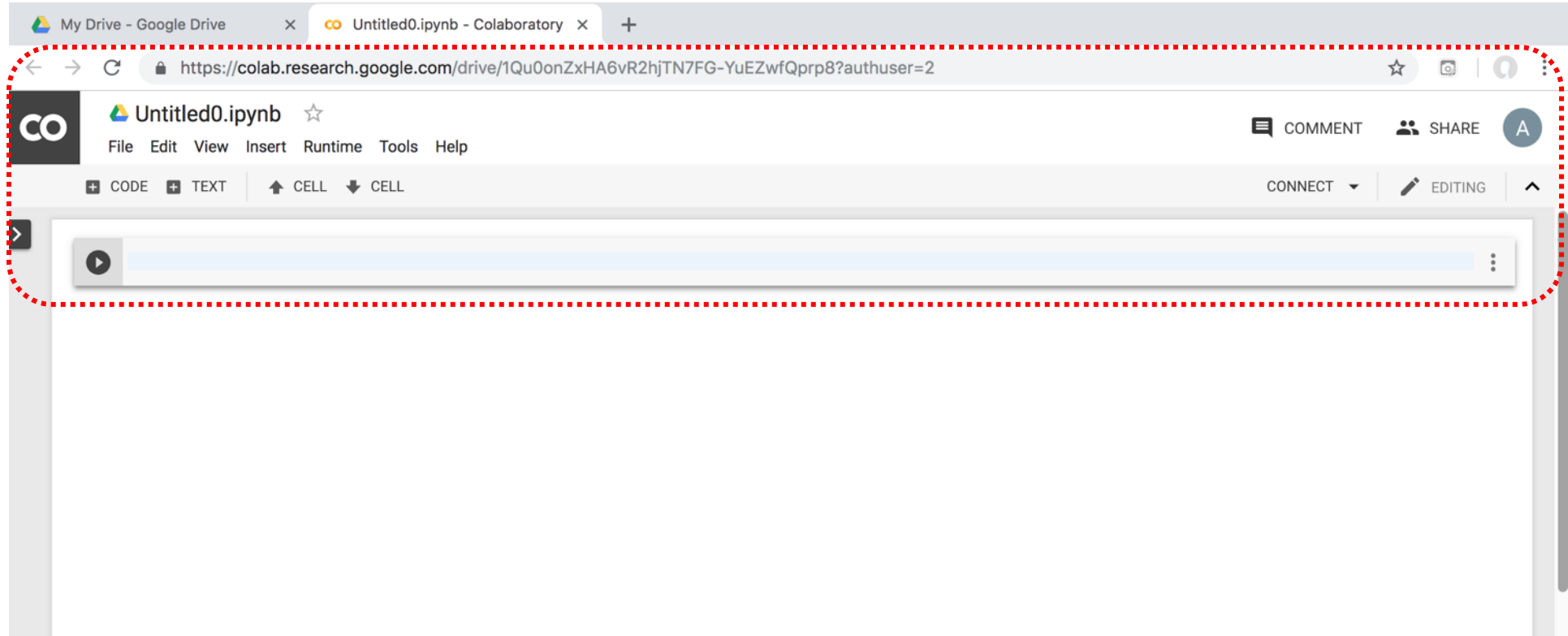
Google Sites

Colaboratory

Connect more apps

Name ↑

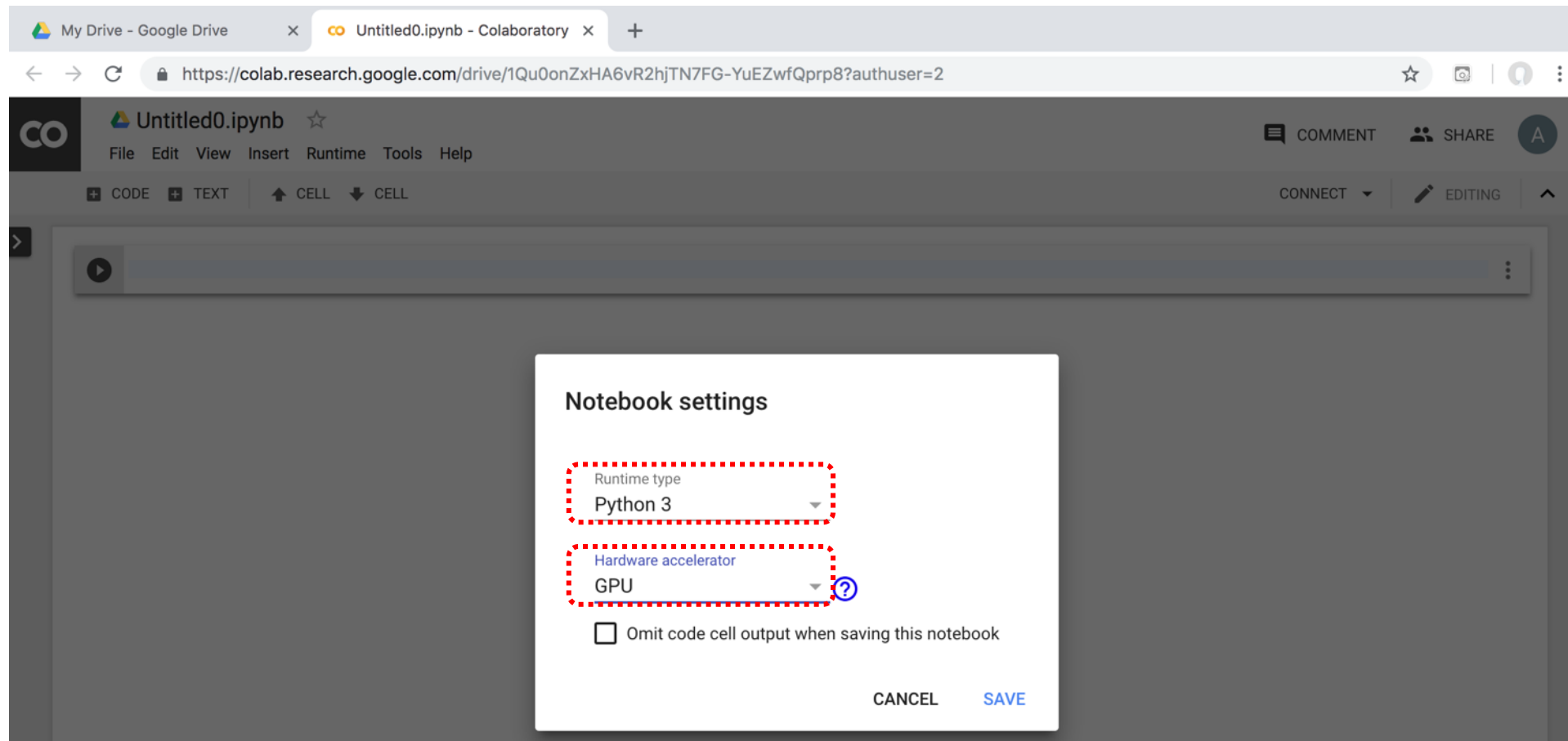
Google Colab



Google Colab

The image shows a screenshot of the Google Colab web interface. The browser's address bar displays the URL: <https://colab.research.google.com/drive/1Qu0onZxHA6vR2hjTN7FG-YuEZwfQprp8?authuser=2>. The main interface features a menu bar with 'File', 'Edit', 'View', 'Insert', 'Runtime', 'Tools', and 'Help'. The 'Runtime' menu is open, showing various options: 'Run all' (⌘/Ctrl+F9), 'Run before' (⌘/Ctrl+F8), 'Run the focused cell' (⌘/Ctrl+Enter), 'Run selection' (⌘/Ctrl+Shift+Enter), 'Run after' (⌘/Ctrl+F10), 'Interrupt execution' (⌘/Ctrl+M I), 'Restart runtime...' (⌘/Ctrl+M .), 'Restart and run all...', 'Reset all runtimes...', 'Change runtime type', and 'Manage sessions'. The 'Change runtime type' option is highlighted with a red dashed border. The interface also includes a 'CONNECT' dropdown, an 'EDITING' mode indicator, and a user profile icon.

Run Jupyter Notebook Python3 GPU Google Colab



The screenshot shows the Google Colab web interface. The browser tabs include 'My Drive - Google Drive' and 'Untitled0.ipynb - Colaboratory'. The address bar shows the URL: <https://colab.research.google.com/drive/1Qu0onZxHA6vR2hjTN7FG-YuEZwfQprp8?authuser=2>. The interface includes a menu bar with 'File', 'Edit', 'View', 'Insert', 'Runtime', 'Tools', and 'Help'. Below the menu bar are buttons for '+ CODE', '+ TEXT', '↑ CELL', and '↓ CELL'. On the right side, there are buttons for 'COMMENT', 'SHARE', 'CONNECT', and 'EDITING'. A 'Notebook settings' dialog box is open in the center, featuring two dropdown menus: 'Runtime type' set to 'Python 3' and 'Hardware accelerator' set to 'GPU'. A checkbox labeled 'Omit code cell output when saving this notebook' is unchecked. The dialog box has 'CANCEL' and 'SAVE' buttons at the bottom right.

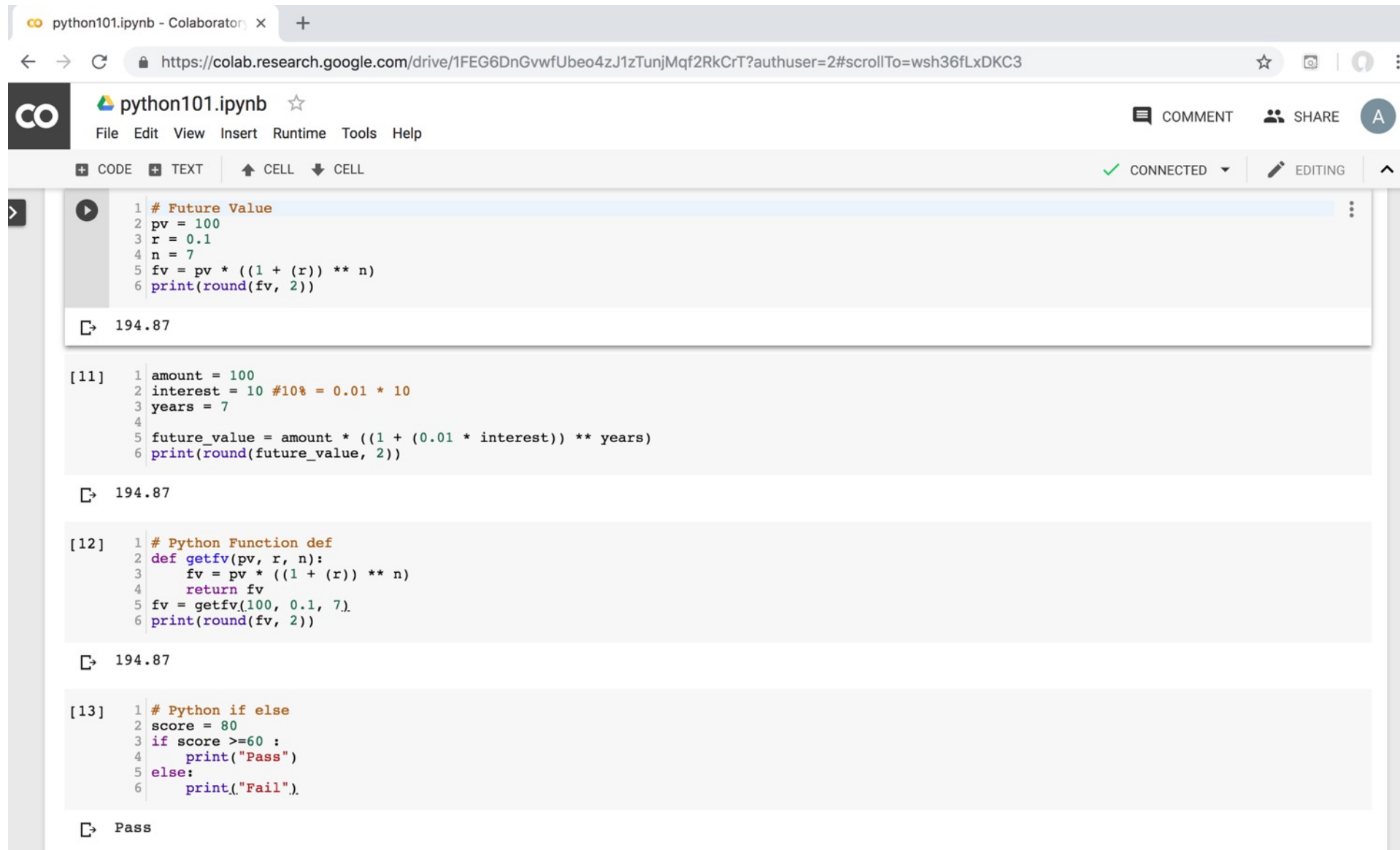
Google Colab Python Hello World

```
print('Hello World')
```



Python in Google Colab (Python101)

<https://colab.research.google.com/drive/1FEG6DnGvwfUbeo4zJ1zTunjMqf2RkCrT>



The screenshot shows a Google Colab notebook interface. The browser address bar displays the URL: <https://colab.research.google.com/drive/1FEG6DnGvwfUbeo4zJ1zTunjMqf2RkCrT?authuser=2#scrollTo=wsh36fLxDKC3>. The notebook title is "python101.ipynb". The interface includes a menu bar (File, Edit, View, Insert, Runtime, Tools, Help) and a toolbar with options for CODE, TEXT, CELL, and a status indicator showing "CONNECTED" and "EDITING".

The notebook contains four code cells:

- Cell 1:** A code cell with the following Python code:

```
1 # Future Value
2 pv = 100
3 r = 0.1
4 n = 7
5 fv = pv * ((1 + (r)) ** n)
6 print(round(fv, 2))
```

The output of this cell is "194.87".
- Cell 2:** A code cell with the following Python code:

```
[11] 1 amount = 100
2 interest = 10 #10% = 0.01 * 10
3 years = 7
4
5 future_value = amount * ((1 + (0.01 * interest)) ** years)
6 print(round(future_value, 2))
```

The output of this cell is "194.87".
- Cell 3:** A code cell with the following Python code:

```
[12] 1 # Python Function def
2 def getfv(pv, r, n):
3     fv = pv * ((1 + (r)) ** n)
4     return fv
5 fv = getfv(100, 0.1, 7).
6 print(round(fv, 2))
```

The output of this cell is "194.87".
- Cell 4:** A code cell with the following Python code:

```
[13] 1 # Python if else
2 score = 80
3 if score >=60 :
4     print("Pass")
5 else:
6     print("Fail").
```

The output of this cell is "Pass".

<https://tinyurl.com/aintpupython101>



Anaconda
The Most Popular
Python
Data Science Platform

Download Anaconda



Products ▾

Pricing

Solutions ▾

Resources ▾

Partners ▾

Blog

Company ▾

Contact Sales

Data science technology for a better world.

Anaconda offers the easiest way to perform Python/R data science and machine learning on a single machine. Start working with thousands of open-source packages and libraries today.

Download 

Get Additional Installers



<https://www.anaconda.com/download>

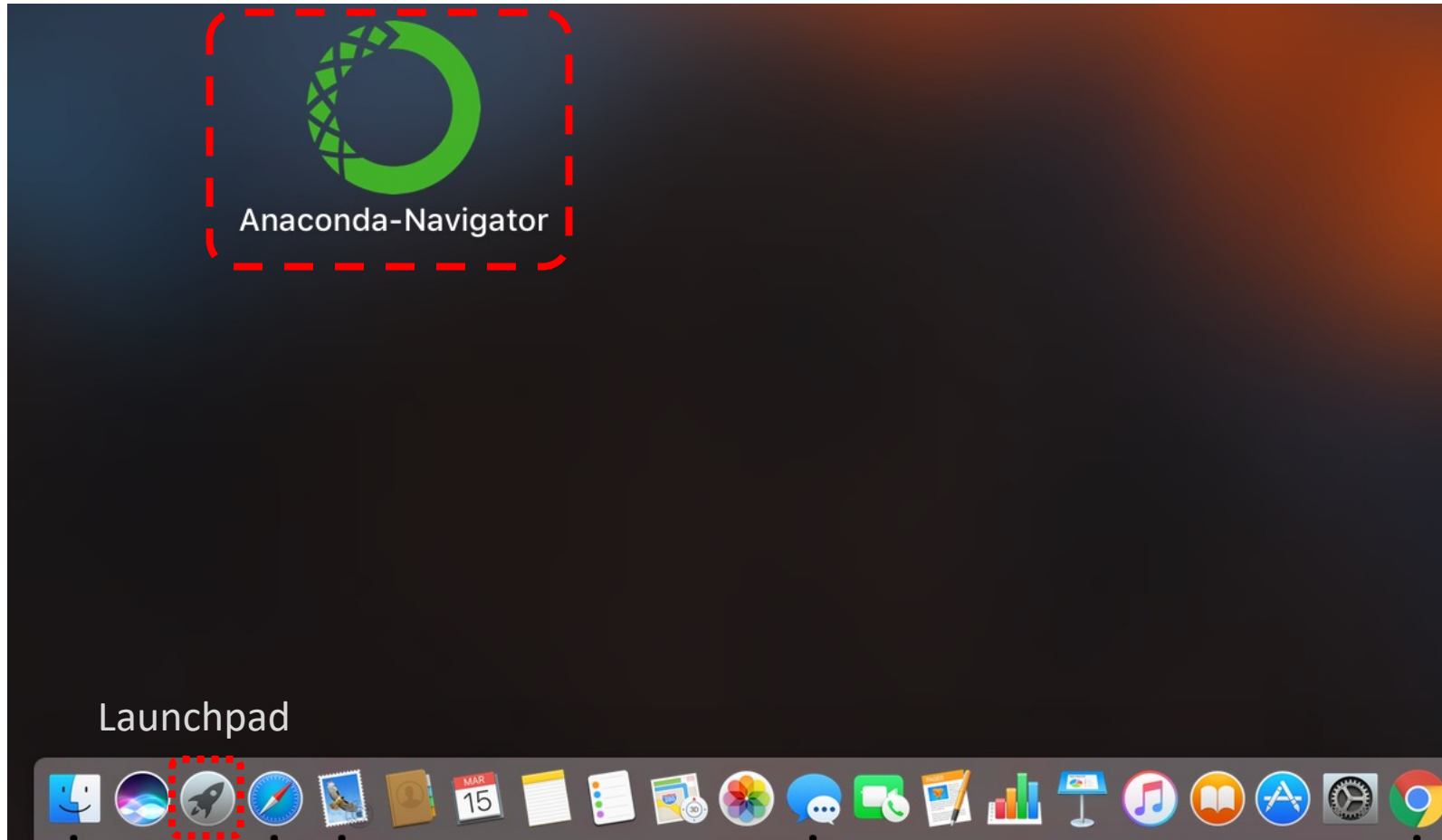




Python

HelloWorld

Anaconda-Navigator



Anaconda Navigator

The screenshot displays the Anaconda Navigator desktop application. At the top, the title bar reads "Anaconda Navigator". Below it, the application header features the "ANACONDA NAVIGATOR" logo on the left and a "Sign in to Anaconda Cloud" button on the right. A left-hand sidebar contains navigation options: "Home", "Environments", "Learning", and "Community". At the bottom of the sidebar are links for "Documentation", "Developer Blog", and "Feedback", along with social media icons for Twitter, YouTube, and GitHub.

The main content area is titled "Applications on" and shows a dropdown menu set to "base (root)" and a "Channels" button. A "Refresh" button is located in the top right of this section. The applications are arranged in a grid:

- jupyterlab** (0.31.5): An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture. [Launch]
- jupyter notebook** (5.4.0): Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis. [Launch]
- qtconsole** (4.3.1): PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more. [Launch]
- spyder** (3.2.6): Scientific PYTHON Development Environment. Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features. [Launch]
- vscode** (1.22.2): Streamlined code editor with support for development operations like debugging, task running and version control. [Launch]
- glueviz** (0.12.4): Multidimensional data visualization across files. Explore relationships within and among related datasets. [Install]

The "jupyter notebook" application card is highlighted with a red dashed border, and its "Launch" button is enclosed in a solid red box.

Jupyter Notebook

The screenshot shows a web browser window with the Jupyter Notebook interface. The browser's address bar displays the URL `localhost:8888/tree/Documents/Data/BDA`. The Jupyter logo and a "Logout" button are visible at the top. Below the navigation tabs ("Files", "Running", "Clusters"), there is a prompt "Select items to perform actions on them." followed by "Upload", "New", and "Refresh" buttons. A file browser table is shown with a red dashed border around its content area. The table header includes a selection column with a count of "0", a breadcrumb path `/ Documents / Data / BDA`, and columns for "Name" and "Last Modified". The table body contains a single entry for a parent directory `..` with a "seconds ago" timestamp. A message at the bottom of the table states "The notebook list is empty."

<input type="checkbox"/> 0	<code>/ Documents / Data / BDA</code>	Name ↓	Last Modified
<input type="checkbox"/>	<code>..</code>		seconds ago

The notebook list is empty.

Jupyter Notebook

New Python 3

The screenshot shows a web browser window with the Jupyter Notebook interface. The browser's address bar displays `localhost:8888/tree/Documents/Data/BDA`. The Jupyter logo and a 'Logout' button are visible at the top. Below the logo, there are tabs for 'Files', 'Running', and 'Clusters'. A message reads 'Select items to perform actions on them.' To the right of this message are 'Upload', 'New', and a refresh icon. The 'New' dropdown menu is open, showing options: 'Notebook: Python 3', 'Other: Text File', 'Folder', and 'Terminal'. The 'Python 3' option is highlighted with a red box. The file browser below shows the path `/ Documents / Data / BDA` and a message 'The notebook list is empty.'

```
print("hello, world")
```

The screenshot shows a Jupyter Notebook interface in a web browser. The browser tabs include 'Home' and 'HelloWorld'. The address bar shows the URL 'localhost:8888/notebooks/Documents/Data/BDA/HelloWorld.ipynb'. The Jupyter logo and 'HelloWorld (autosaved)' are visible in the top left, along with a 'Logout' button and a Python logo. A menu bar contains 'File', 'Edit', 'View', 'Insert', 'Cell', 'Kernel', 'Widgets', and 'Help'. Below the menu is a toolbar with icons for saving, adding, deleting, copying, pasting, and running. The 'Run' button is highlighted with a red box. The main area contains a code cell with the text 'In [1]: print("hello, world")' and its output 'hello, world'. The code line is also highlighted with a red box. Below the code cell is an empty input field labeled 'In []:'.





Python

Programming

Python Hello World

```
print("Hello World")
```

```
print("Hello World")
```


Python Syntax

comment

```
# comment
```

Python Syntax

Indentation

the spaces at the beginning of a code line
4 spaces

```
score = 80
if score >= 60 :
    print("Pass")
```

Python Variables

```
# Python Variables  
x = 2  
price = 2.5  
word = 'Hello'  
  
word = 'Hello'  
word = "Hello"  
word = '''Hello'''
```

Python Variables

```
x = 2
```

```
y = x + 1
```

python_version()

```
# comment  
from platform import python_version  
print("Python Version:", python_version())
```

Python Version: 3.10.12

Python Data Types

```
x = "Hello World"    #str
x = 2                #int
x = 2.5              #float
x = 7j               #complex
```

Python Data Types

```
x = ["apple", "banana", "cherry"] #list
x = ("apple", "banana", "cherry") #tuple
x = range(6) #range
x = {"name" : "Tom", "age" : 20} #dict
x = {"apple", "banana", "cherry"} #set
x = frozenset({"apple", "banana", "cherry"})
#frozenset
```

Python Data Types

```
x = True #bool
x = b"Hello" #bytes
x = bytearray(5) #bytearray
x = memoryview(bytes(5)) #memoryview
x = None #NoneType
```


Python Casting

```
x = str(3) # x will be '3'  
y = int(3) # y will be 3  
z = float(3) # z will be 3.0  
print(x, type(x))  
print(y, type(y))  
print(z, type(z))
```

```
3 <class 'str'>  
3 <class 'int'>  
3.0 <class 'float'>
```

Python Numbers

```
x = 2 # int
y = 3.4 # float
z = 7j #complex
print(x, type(x))
print(y, type(y))
print(z, type(z))
```

```
2 <class 'int'>
3.4 <class 'float'>
7j <class 'complex'>
```

Python Arithmetic Operators

Operator	Name	Example
+	Addition	$7 + 2 = 9$
-	Subtraction	$7 - 2 = 5$
*	Multiplication	$7 * 2 = 14$
/	Division	$7 / 2 = 3.5$
//	Floor division	$7 // 2 = 3$ (Quotient)
%	Modulus	$7 \% 2 = 1$ (Remainder)
**	Exponentiation	$7 ** 2 = 49$

Python Basic Operators

```
print('7 + 2 =', 7 + 2)
print('7 - 2 =', 7 - 2)
print('7 * 2 =', 7 * 2)
print('7 / 2 =', 7 / 2)
print('7 // 2 =', 7 // 2)
print('7 % 2 =', 7 % 2)
print('7 ** 2 =', 7 ** 2)
```

7 + 2 = 9
7 - 2 = 5
7 * 2 = 14
7 / 2 = 3.5
7 // 2 = 3
7 % 2 = 1
7 ** 2 = 49

Python Booleans: True or False

```
# Python Booleans: True or False  
print(3 > 2)  
print(3 == 2)  
print(3 < 2)
```

Python BMI Calculator

```
# BMI Calculator in Python
height_cm = 170
weight_kg = 60
height_m = height_cm/100
BMI = (weight_kg/(height_m**2))

print("Your BMI is: " + str(round(BMI,1)))
```

Your BMI is: 20.8

Future value
of a specified
principal amount,
rate of interest, and
a number of years

How much is your \$100 worth after 7 years?

```
# How much is your $100 worth after 7 years?  
fv = 100 * 1.1 ** 7  
print('fv = ', round(fv, 2))  
# output = 194.87
```

```
fv = 194.87
```


Future Value

```
# Future Value
pv = 100
r = 0.1
n = 7

fv = pv * ((1 + (r)) ** n)
print(round(fv, 2))
```

194.87

Future Value

```
# Future Value
amount = 100
interest = 10 #10% = 0.01 * 10
years = 7

future_value = amount * ((1 + (0.01 * interest)) ** years)
print(round(future_value, 2))
```

194.87

Summary

- **Python Syntax**
 - **Python Comments**
- **Python Variables**
- **Python Data Types**
 - **Python Numbers**
 - **Python Casting**
 - **Python Strings**
- **Python Operators**
- **Python Booleans**

References

- Wes McKinney (2022), "Python for Data Analysis: Data Wrangling with pandas, NumPy, and Jupyter", 3rd Edition, O'Reilly Media.
- Aurélien Géron (2023), Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems, 3rd Edition, O'Reilly Media.
- Steven D'Ascoli (2022), Artificial Intelligence and Deep Learning with Python: Every Line of Code Explained For Readers New to AI and New to Python, Independently published.
- Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson.
- Varun Grover, Roger HL Chiang, Ting-Peng Liang, and Dongsong Zhang (2018), "Creating Strategic Business Value from Big Data Analytics: A Research Framework", Journal of Management Information Systems, 35, no. 2, pp. 388-423.
- Junliang Wang, Chuqiao Xu, Jie Zhang, and Ray Zhong (2022). "Big data analytics for intelligent manufacturing systems: A review." Journal of Manufacturing Systems 62 (2022): 738-752.
- Ramesh Sharda, Dursun Delen, and Efraim Turban (2017), Business Intelligence, Analytics, and Data Science: A Managerial Perspective, 4th Edition, Pearson
- Python Programming, <https://pythonprogramming.net/>
- Python, <https://www.python.org/>
- Python Programming Language, <http://pythonprogramminglanguage.com/>
- Numpy, <http://www.numpy.org/>
- Pandas, <http://pandas.pydata.org/>
- Skikit-learn, <http://scikit-learn.org/>
- W3Schools Python, <https://www.w3schools.com/python/>
- Learn Python, <https://www.learnpython.org/>
- Google's Python Class, <https://developers.google.com/edu/python>
- Min-Yuh Day (2023), Python 101, <https://tinyurl.com/aintpupython101>