# Introduction to Python

**Vishal Patel** 

Spring 2022

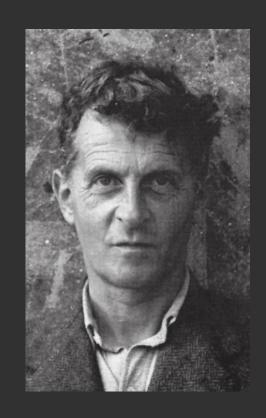




Linguistic Relativity / Sapir–Whorf hypothesis

# The limits of my language mean the limits of my world.

Ludwig Wittgenstein



# **Pythonic**

**Uncluttered** 

Simple



Readable

**Explicit** 

Concise

# The Zen of Python

Beautiful is better than ugly.

Explicit is better than implicit.

Simple is better than complex.

Complex is better than complicated.

Sparse is better than dense.

Readability counts.

Special cases aren't special enough to break the rules.

Although practicality beats purity.

Errors should never pass silently.

Unless explicitly silenced. (contd.)

# **Pearl Programming Motto**

# There is more than one way to do it.

TIMTOWTDI, pronounced Tim Toady

# The Zen of Python

- "There should be one
- and preferably only oneobvious way to do it."

Python is powerful... and fast; plays well with others; runs everywhere; is friendly & easy to learn; is Open.



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

IN DECEMBER 1989
Guido van Rossum,
a Dutch computer scientist,
set himself a Christmas project.

Irked by shortcomings in other programming languages, he wanted to build his own.



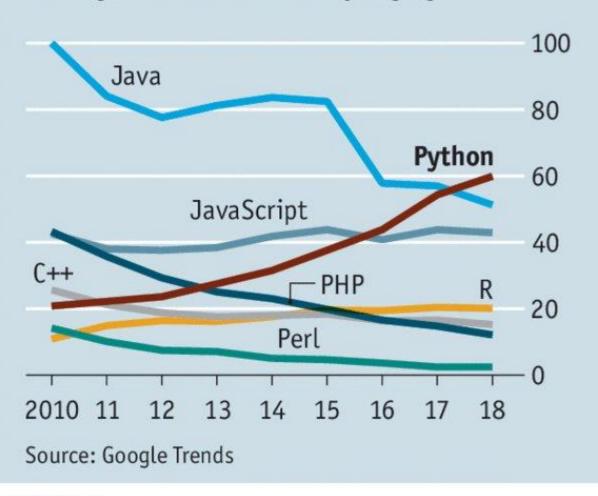
#### **Guido van Rossum**

Python's creator and emeritus BDFL (Benevolent Dictator For Life)
<a href="mailto:ogvanrossum">ogvanrossum</a>



#### **Biggus uptickus**

US, Google searches for coding languages 100=highest annual traffic for any language





# Who Uses Python?





















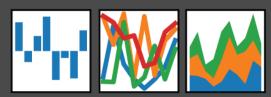
# **Trying to Learn Any Programming Language 100%**





- Anaconda is a free and open-source distribution of the Python and R programming languages for scientific computing, that aims to simplify package management and deployment.
- The Anaconda distribution is used by over 20 million users in settings ranging from small research institutions to academia to Fortune 100 companies. Commercial usage of our packages has increased more than 700% since 2016. [source]





Provides high-performance, easy-to-use data structures and data analysis tools



Simple and efficient tools for data mining and data analysis

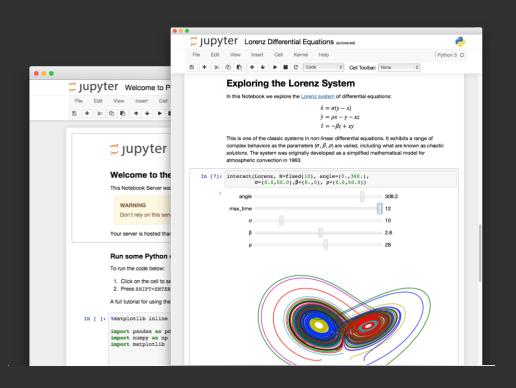
# seaborn

Provides a high-level interface for drawing attractive and informative statistical graphics.



Project Jupyter exists to develop open-source software, open-standards, and services for interactive computing across dozens of programming languages.

Julia, Python and R



The **Jupyter Notebook** is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations and narrative text.



Language of choice

The Notebook has support for over 40 programming languages, including Python, R, Julia, and Scala.



Share notebooks

Notebooks can be shared with others using email, Dropbox, GitHub and the Jupyter Notebook Viewer.



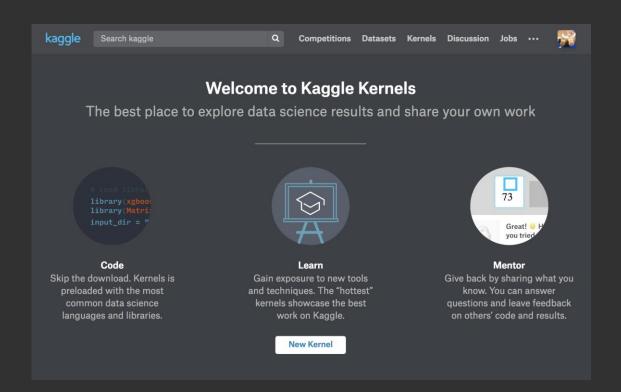
Interactive output

Your code can produce rich, interactive output: HTML, images, videos, LaTeX, and custom MIME types.



Big data integration

Leverage big data tools, such as Apache Spark, from Python, R and Scala. Explore that same data with pandas, scikit-learn, ggplot2, TensorFlow.





#### Amazon SageMaker

Machine learning for every data scientist and developer

Get Started with SageMaker

#### A gallery of interesting Jupyter Notebooks

Erwan Pannier edited this page 6 days ago · 124 revisions

# Jupyter Notebook

- 1. The **Jupyter Notebook App** (formerly IPython Notebook) is a server-client application that allows editing and running notebook documents via a web browser.
- 2. A notebook **kernel** is a "computational engine" that executes the code contained in a Notebook document.
- 3. The Jupyter Notebook App gets installed automatically when you install Anaconda.
- 4. The Jupyter Notebook App can be launched by clicking on the Jupyter Notebook icon installed by Anaconda in the start menu.

# Closing the Jupyter Notebook App

- 1. Closing the browser (or the tab) **will not close** the Jupyter Notebook App. To completely shut it down you need to **close the associated terminal**.
- 2. To shut it down, go to the associated notebook and click on menu **File**  $\rightarrow$  **Close** and **Halt**.
  - 1. Alternatively, the Notebook Dashboard has a tab named *Running* that shows all the running notebooks (i.e. kernels) and allows shutting them down (by clicking on a *Shutdown* button).

# **A Few Useful Commands**

Task	Command/Short-cut
Run selected cell	Ctrl + Enter
Run selected cell and insert a new cell below	Alt + Enter
Run all cells	Menu: <i>Cell -&gt; Run All</i> .
Restart the kernel	Menu: <i>Kernel -&gt; Restart</i>
Delete a cell	



★ Visual Studio Code

Docs Updates

Blog

Extensions

Learn

Version 1.52 is now available! Read about the new features and fixes from November.

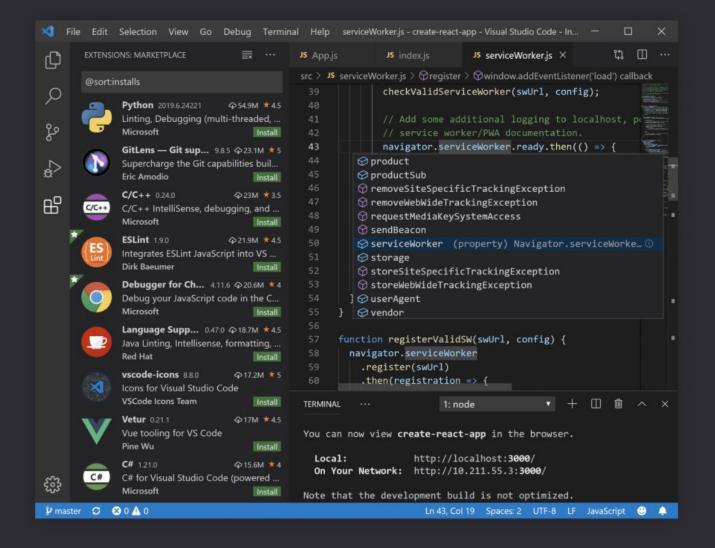
#### Code editing. Redefined.

Free. Built on open source. Runs everywhere.



Other platforms and Insiders Edition

By using VS Code, you agree to its



#### Poll 2

Do you have a preferred Python IDE (Integrated Development Environment)?

- Jupyter Notebook
- Other (e.g., VSCode)
- No preference

### Poll 3

How much of coding experience do you have?

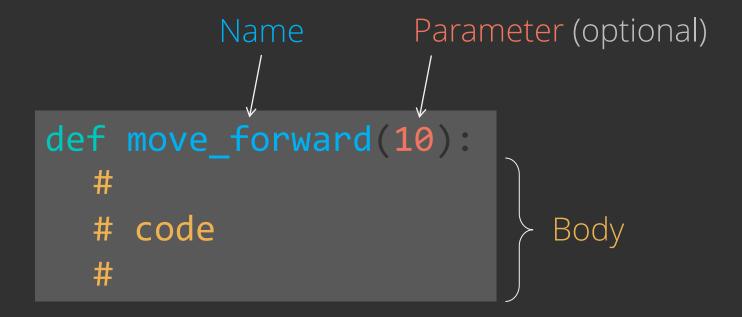
- None
- Less than 1 year
- 1-5 years
- 5+ years

# Python Intro (Caesar Cipher) Tutorial

01\_caesar\_cipher.ipynb

# **Python Functions**

- igcup Functions pprox Actions, verbs, commands
- Parameters ≈ adverbs



## **Case Sensitive**

```
temp = 1
print(temp)
 1
print(Temp)
 NameError
                                          Traceback (most recent call last)
 <ipython-input-3-086e40c91607> in <module>
 ----> 1 print(Temp)
 NameError: name 'Temp' is not defined
```

# Whitespace / Indentation

```
for i in [1, 2, 3, 4, 5]:
    print(i)
for i in [1, 2, 3, 4, 5]:
print(i)
   File "<ipython-input-2-500429ebfc34>", line 2
    print(i)
 IndentationError: expected an indented block
```

```
In [1]: (1+3) == (1 + 3)

True
```

Whitespaces within lines do not matter.

# **Dynamic Typing**

```
a = 3
print(a, type(a))
 3 <class 'int'>
a = 'Python'
print(a, type(a))
 Python <class 'str'>
a = 3.14
print(a, type(a))
 3.14 <class 'float'>
```

# **Arithmetic Operations**

Operator	Name	Description
a + b	Addition	Sum of a and b
a - b	Subtraction	Difference of a and b
a * b	Multiplication	Product of a and b
a / b	True division	Quotient of a and b
a // b	Floor division	Quotient of a and b, removing fractional parts
a % b	Modulus	Integer remainder after division of a by b
a ** b	Exponentiation	a raised to the power of b
-a	Negation	The negative of a
+a	Unary plus	a unchanged (rarely used)

```
counter = 1
counter = counter + 1
print(counter)
counter = 1
counter += 1
print(counter)
```

# Scalar Types

Туре	Example	Description
int	x = 1	integers (i.e., whole numbers)
float	x = 1.0	floating-point numbers (i.e., real numbers)
complex	x = 1 + 2j	Complex numbers (i.e., numbers with real and imaginary part)
bool	x = True	Boolean: True/False values
str	x = 'abc'	String: characters or text
NoneType	x = None	Special object indicating nulls

## **Data Structures**

Type Name	Example	Description
list	[1, 2, 3]	Ordered collection
tuple	(1, 2, 3)	Immutable ordered collection
dict	{'a':1, 'b':2, 'c':3}	Unordered (key,value) mapping
set	{1, 2, 3}	Unordered collection of unique values

# **Indexing and Slicing**

```
mylist = [1, 50, 100]
print(mylist)
 [1, 50, 100]
print(type(mylist))
 <class 'list'>
len(mylist)
mylist.append(99)
print(mylist)
 [1, 50, 100, 99]
```

```
mylist[0]
                           Indexing
mylist[3]
mylist[-1]
mylist[0:2]
 [1, 50]
mylist[::-1]
                               Slicing
 [99, 100, 50, 1]
```

# **Python Packages**

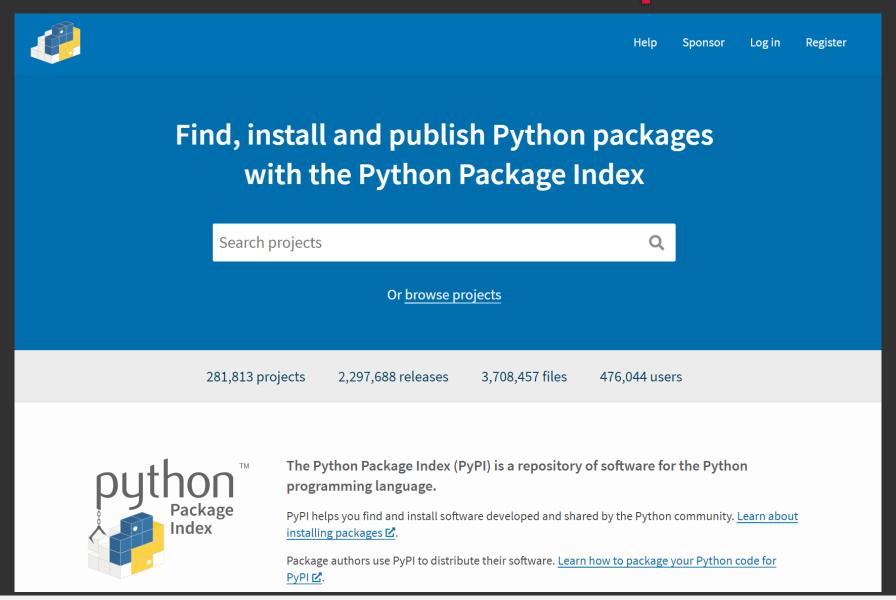
#### 1. Installing a package on your computer:

- 1. Go to Anaconda command prompt (search for it on your computer using the search bar).
- Type 'pip install <package-name>' or 'conda install <package-name>' to install a specific package.

#### **2. Importing** a package into your program:

- 1. Explicit module import: e.g., import math
- 2. Explicit module import by alias: e.g., import numpy as np
- 3. Explicit import of module contents: e.g., from math import cos, pi

# "The Cheese Shop"





Do I have to like "Monty Python's Flying Circus"?

No, but it helps. :)