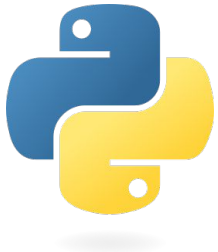
Python

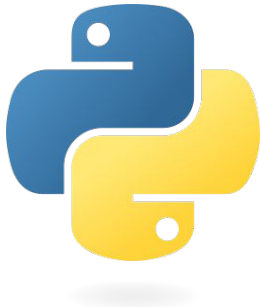


What is python and why learn it?

Python is one of the world’s most used and most popular programming languages.

Applications - Fields :

- Web development
- Data Science/Analysis
- Machine learning
- Artificial Intelligence (AI)



What we will learn

1. Variables
2. Input / Output
3. Mathematical/ Operations
4. Comparison Operations
5. Logic Operations
6. Functions
7. Loops
8. Classes
9. Data-Types such as Lists, Dictionaries, Tuples.

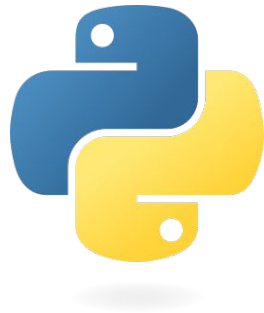
Installing python.

Either download from:

https://www.python.org/downloads/

Else use a package like **anaconda**:

https://www.anaconda.com/download



Code editor

Either use IDE like Pycharm:

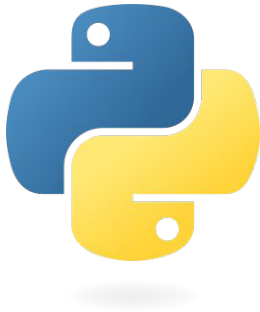
https://www.jetbrains.com/pycharm/download

Or a code Editor like Visual Studio Code:

https://code.visualstudio.com/download

Or online tools like **google collab**:

https://colab.research.google.com



History of Python

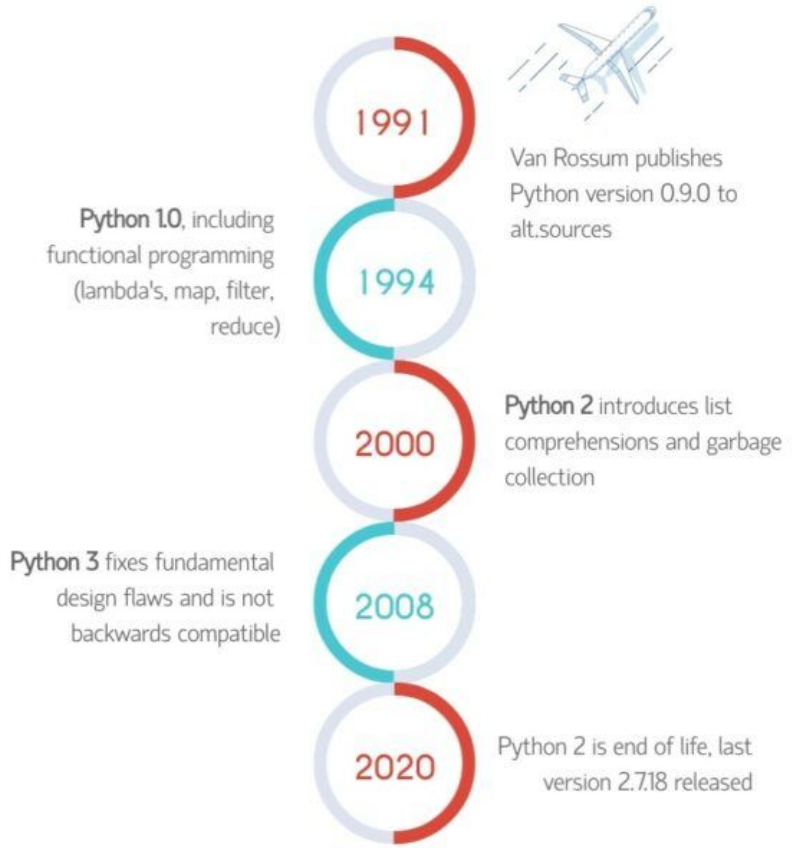


Figure from python.land

Variables : Stores information.

Can be numbers, letters or logical values, functions, Objects, dictionaries etc.

Simple variable types:

- Int : 6 , no decimals
- Float: 6.5555, decimal points
- Strings : ‘yes’, “7”
- Booleans: True, False
- None type : None

Initialize a variable:

X = 1

Mathematical/Numerical Operators

- Addition:
- Substract:
- Divide:
- Multiply:
- Power:
- Modulo:
- Floor division

→ PEMDAS :

Parenthesis - Exponent, Multiplication, Division, Addition, Substraction

Comparison Operations

- >
- <
- >=
- <=
- == : equal
- != unequal

Logical Operations

- and
- or
- not

AND Truth Table

A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

OR Truth Table

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

NOT Truth Table

A	B
0	1
1	0

If/else statement

- If
- Elif
- Else
- Nested If

Operations on Strings/Sequences

- + : concatenate
- * : multiply
- [1,...] : indexing
  - Inverse indexing using [- 1,..]
- [:] : slice
  - Inverse slice using [-1:]
- For i in string : iterate

Other Data types

1. Tuple : (x,y,z,...) : x,y,z can be different datatypes!
    a. Access using indexing : e.g a[1]
    b. Can add tuples e.g: () + ()
    c. Can check equality using ==, e.g (,) == (,)
2. Lists : [1,5,None] contains objects → mutable. (Change inplace)
    a. .append : adds an element to the last position of a list (adds one element)
    b. .extend : adds a object (iteratable) at the last position (adds multiple elements)
    c. for item in mylist: print(item)
    d. .pop() – returns last element then removes it from the list
    e. .remove('x') – finds and removes first 'x' from list
    f. .reverse() – reverses the elements in the list
    g. .sort() – sorts the list alphabetically in ascending order, or numerical in ascending order
3. Dictionaries : LOVE IT
    a. Key- value pair
    b. Init with {}, dict()
    c. A['loveit'] = True, A['woah'] = [1,2,3] etc. (add new key-value pairs)
    d. .keys() to get the keys
    e. A[key] to get the value of a key.
    f. A.get(key,default value)
4. Sets : contains unique elements **– no duplicates!**
    a. A = {1,2,3} , a = set()
    b. .add to add elements e.g a.add(1)

Loops

- For : loop over an iteratable.
- While : keep looping till condition is False.

Functions - Classes

More on this on the advanced course… (TBA)

Classes and functions help with re-usability.

1. Functions
    a. Def name(arguments):
    b. Return to bring back a value
    c. Call by using name(args)
2. Class
    a. Able to store information in an instance of the class.
    b. Has functions inside (called methods)
    c. Class Name: define a class
    d. __init__(self, arg): to initialize a object of the class
    e. A = Name() : create object of class Name
    f. Self argument used to access the objects methods - values (variables)

Again: More in the advanced course!

Advanced course (Stay tuned on YT + Udemy)

1. Classes (indepth)
2. Functions (indepth)
3. Lambda functions
4. Reading/ writing files
5. Exceptions
6. Assertions- defensive programming
7. Using external libraries
8. Intro to ML/ Data science

Useful built-in functions

1. max()
2. min()
3. len()
4. str()
5. sum()