Python

- Medium to communicate with computers.
- Every programming language has Keywords / Reserved words.
- Syntax ⇝ Set of rules
- High level language.
- Easy to simple, learn, troubleshoot.

① Literals ⇝ (Data) (Building block of Python)

- Numeric Literal → 1000 (integers) → −500 → 0 } Integer Literal
- Float Literal → 3.14 → −5.89

→ 10.0 ⇝ Classified as float.
→ 10 ⇝ Integer.

- String literal ⇝ Group/Sequence of characters Enclosed within quotes.
  - 'Hello'
  - "Hello"
  - '''Hello'''
  - """Hello"""

→ """Hello, How are you ?""" → Multi-line

```
Hello,
How are you ?
```

"Hello, \n How are you ?"

→ escape sequence ⇝ next line

- Boolean Literal ⇝ True/False
- None literal ⇝ No value

② Variables → Reserved memory location used to store values.

- a = 1
  print(a) ⇝ 1
- x = 4
  x = 3 ⇝ Replace value

- del a ⇝ Deleting 'a' variable
- a = "Python"

- Begin with alphabet or —.
- Can't begin with number.
- Variable names are case-sensitive

③ Keywords → Special reserved words.

- For, while, def, break, if, else ....

import keyword
keyword.kwlist } → Shows you all the Keywords.

④ Primitive Datatypes.

a) Integers ⇝ age = 25
   type(age) ⇝ <class 'int'>

b) Floats ⇝ percent = 87.5
   type(percent) ⇝ <class 'float'>

→ 4 + 3.5
→ 7.5
} We can do this

c) Strings ⇝ name = "Mark"
   type(name) ⇝ <class 'str'>

d) Boolean ⇝ val1 = True
   type(val1) ⇝ <class 'bool'>

e) None ⇝ val2 = None, type(val2) ⇝ <class 'NoneType'>

⑤ Strings

→ name = 'Python'
len(name) → 6

Python
0 1 2 3 4 5 ⇝ index (Positive)
-6 -5 -4 -3 -2 -1 ⇝ Negative Index.

- name[0] ⇝ P
- name[6] ⇝ Error
- name[-1] ⇝ last character ⇝ n
- name[-2] ⇝ o

- Concatenating 2 strings ⇝ S1 + S2 (Print as it is)
  ⇝ S1 + ' ' + S2 (with space)

⑥ Type Conversion / Type Casting

- Convert data-type from one another.
- num1 = 100
  num2 = float(num1) → 100.0 } → num1 doesn't change.
- num1 = float(num1) → num1 changes.

- num3 = 56.45
  int(num3) → 56

num4 = 76.78
int(num4) → 76 ⇝ Doesn't round off.

- str(num1) → '100'
- S1 = 'Hello'
- int(S1) ⇝ Error

S1 = '123'
int(S1) → Works

S1 = 'Py3.14'
int(S1) ⇝ Error

- language = "Python"
- version = 3.14
- language + version → Error

version_str = str(version)
language + version_str → ⟶

(7) Type Conversion (Boolean)

- val1 = True
- Str(val1) ~> 'True'
- bool('Python') ~> True
- bool(100) ~> True
- bool(1.5) ~> True
- bool(0) ~> False

bool(0.0) ~> False
bool(104) ~> True
:
bool(-100) ~> True

Only 0 gives us false.

bool(0.5) → True

- bool('hi') ~> True
- bool('a') ~> True
- bool(' ') ~> True (Space is valid character)

→ Empty String ~> '', "", ''' ''', """ """

This gives us false.

- bool(None) ~> False
- int(True) → 1, int(False) → 0
- float(True) → 1.0, float(False) → 0.0

(8) Arithmetic, Assignment Stuff.

a) Arithmetic Operator

- num1 = 10
- num2 = 5
- num1 + num2 ~> (15)
- num1 - num2 ~> (5)
- num1 * num2 → 10*5 = (50)
- num1 / num2 ~> (2) (Data type is float)
- num1 // num2 ~> (2) (Data type is int) → W/o decimal value.

- num1 % num2 → 10 % 5 → 0

Remainder

- 3**4 ~> 3^4 → Exponentiation.

b) Assignment operator

- Value = 100
- Value += 100 ⇝ Compound Assignment Operator → Value = value + 100
- Value -= 150 ⇝ Value = value - 150
- Value *= 2 ⇝ Value = value * 2
- Value /= 4 ⇝ value = value / 4

(9) Comparison & Logical Operators

- ==, !=, >, <, >=, <=
- Equality operator → True / False
- 'Python' == 'python' → False

- Logical → and, or, not.

and	or	not
F and F → F	F or F → F	not True → False
F and T → F	F or T → T	not False → True
T & F → F	T or F → T	
T & T → T	T or T → T	

(10) Precedence

- 5 + 10 * 6 ⇝ (65). name = "Mark", age = 25
- name == "Mark" or name = "John" and age < 18.

T ; F ; F ⇝ False → X ; T

True or False → True → ✓

→ And operators first

- 2**1**3

↓

→ 2^1 **3 → 2^3 = (8) → X

→ 1^3 = 1, 2^1 = (2) → ✓

→ Associativity → Right to left exponentiation.

* Classification of operators based on no. of operators.

i) Unary → ++x, --x etc.
↳ not True, not False

ii) Binary → 10-5 => 5
↳ True or False → True

iii) Ternary → ?:

(11) Printf()

- name = "John"
print(name)

- print(name, age) ~> John 20

↓

Space Separator.

- print(10, 20, 30, 40, sep="_") → 10_20_30_40
- print(10, 20, 30, 40, sep=",") → 10,20,30,40
- print("Addition of", num1, "and", num2, "is", num1+num2)

(12) input()

- first_name = input()
- ☆ John.
- print(first_name) ⇝ John
- last_name = input("Enter your last name")
- ☆ Enter your last name Radha.
- num = input("Enter number: ")
- ☆ Enter number: 10.

→ Input is recieved as String.
- ~~print~~ - Convert numbers to integers for any operations.

(13) Pycharm Video.

(14) Numeric Functions

- print(max(1,2,3,4,5)) → (5)
- print(min(1,2,3,4,5)) → (1)
- print(abs(-2020)) → (2020)
- print(pow(2,3)) → 2^3 = (8)