

# PYTHON TUTORIAL'S

EASY WAY TO UNDERSTAND

# INTRO

## What is Python?

Python is a popular programming language. It was created by Guido van Rossum, and released in 1991.

It is used for:

1. web development (server-side),
2. software development,
3. mathematics,
4. system scripting.

## What can Python do?

- Python can be used on a server to create web applications.
- Python can be used alongside software to create workflows.
- Python can connect to database systems. It can also read and modify files.
- Python can be used to handle big data and perform complex mathematics.
- Python can be used for rapid prototyping, or for production-ready software development.

# WHY PYTHON

- Python works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc).
- Python has a simple syntax similar to the English language.
- Python has syntax that allows developers to write programs with fewer lines than some other programming languages.
- Python runs on an interpreter system, meaning that code can be executed as soon as it is written. This means that prototyping can be very quick.
- Python can be treated in a procedural way, an object-oriented way or a functional way.



# THE VISION

## Python Syntax compared to other programming languages

- Python was designed for readability, and has some similarities to the English language with influence from mathematics.
- Python uses new lines to complete a command, as opposed to other programming languages which often use semicolons or parentheses.
- Python relies on indentation, using whitespace, to define scope; such as the scope of loops, functions and classes. Other programming languages often use curly-brackets for this purpose.



# PYTHON - VARIABLE NAMES

A variable can have a short name (like x and y) or a more descriptive name (age, carname, total\_volume).

## RULES FOR PYTHON VARIABLES:

- A variable name must start with a letter or the underscore character
- A variable name cannot start with a number
- A variable name can only contain alpha-numeric characters and underscores (A-z, 0-9, and \_ )
- Variable names are case-sensitive (age, Age and AGE are three different variables)
- A variable name cannot be any of the Python keywords.

## EXAMPLE:

### LEGAL VARIABLE NAMES:

```
myvar = "shaikh"  
my_var = "shaikh"  
_my_var = "shaikh"  
myVar = "shaikh"  
MYVAR = "shaikh"  
myvar2 = "shaikh"
```

### ILLEGAL VARIABLE NAMES:

```
2myvar = "shaikh"  
my-var = "shaikh"  
my var = "shaikh"
```



# PYTHON DATA TYPES

## BUILT-IN DATA TYPES:

In programming, data type is an important concept.

Variables can store data of different types, and different types can do different things.

Python has the following data types built-in by default, in these categories:

Text Type: `str`

Numeric Types: `int`, `float`, `complex`

Sequence Types: `list`, `tuple`, `range`

Mapping Type: `dict`

Set Types: `set`, `frozenset`

Boolean Type: `bool`

Binary Types: `bytes`, `bytearray`, `memoryview`

None Type: `NoneType`

## GETTING THE DATA TYPE:

You can get the data type of any object by using the `type()` function:

## EXAMPLE:

Print the data type of the variable `x`:

```
x = 5  
print(type(x))
```

## SETTING THE DATA TYPE

In Python, the data type is set when you assign a value to a variable:

### Example

1. x = "Hello World"

### Data Type

str

2. x = 20

int

3. x = 20.5

float

4. x = 1j

complex

5. x = ["apple", "banana", "cherry"]

list

6. x = ("apple", "banana", "cherry")

tuple

7. x = range(6)

range

8. x = {"name" : "John", "age" : 36}

dict

9. x = {"apple", "banana", "cherry"}

set

10. x = frozenset({"apple", "banana", "cherry"})

frozenset

11. x = True

bool

12. x = b"Hello"

bytes

13. x = bytearray(5)

bytearray

14. x = memoryview(bytes(5))

memoryview

15. x = None

NoneType

**when you assign a value to a variable the data type is set in python**

# SETTING THE SPECIFIC DATA TYPE

If you want to specify the data type, you can use the following constructor functions:

Example

x = str("Hello World")

x = int(20)

x = float(20.5)

x = complex(1j)

x = list(("apple", "banana", "cherry"))

x = tuple(("apple", "banana", "cherry"))

x = range(6)

x = dict(name="John", age=36)

x = set(("apple", "banana", "cherry"))

x = frozenset(("apple", "banana", "cherry"))

x = bool(5)

x = bytes(5)

x = bytearray(5)

x = memoryview(bytes(5))

Data Type

str

int

float

complex

list

tuple

range

dict

set

frozenset

bool

bytes

bytearray

memoryview

Variables can store data of different types, and different types can do different things.

# PYTHON SYNTAX

## EXECUTE PYTHON SYNTAX

```
>>> print("Hello, world!")
```

```
Hello, world
```



Creating a python file on the server, using the .py file extension, and running it in the Command Line: C:\Users\Your Name>python myfile.py

## PYTHON INDENTATION

Indentation refers to the spaces at the beginning of a code line.

Where in other programming languages the indentation in code is for readability only, the indentation in Python is very important.

Python uses indentation to indicate a block of code

**EXAMPLE:** if 5 > 2:

```
    print("Five is greater than two!")
```

The number of spaces is up to you as a programmer, the most common use is four, but it has to be at least one.

**EXAMPLE:** if 5 > 2:

```
        print("Five is greater than two!")
```

if 5 > 2:

```
    print("Five is greater than two!")
```



## CONTACT US

+91-6206423920 

[hassanshaikh77866@gmail.com](mailto:hassanshaikh77866@gmail.com) 

[www.hassanshaikh.com](http://www.hassanshaikh.com) 

INDIA 

# THANK YOU!

FOR YOUR ATTENTION