Python Basics

0

Introduction

What is Python?
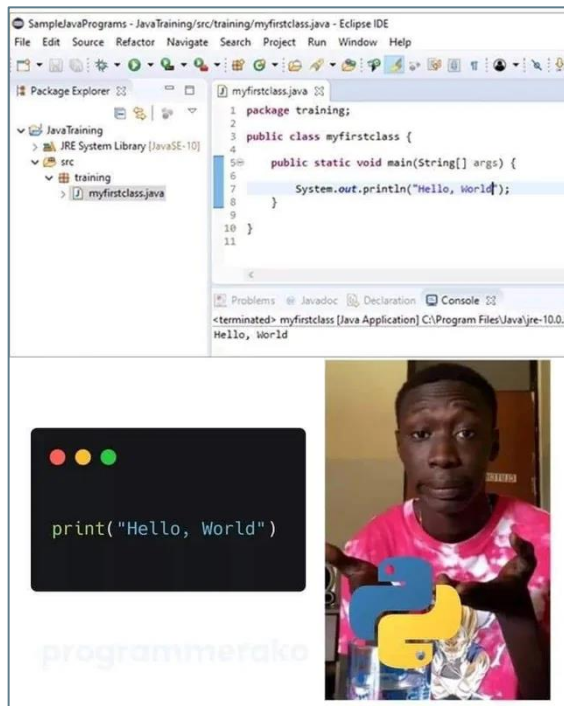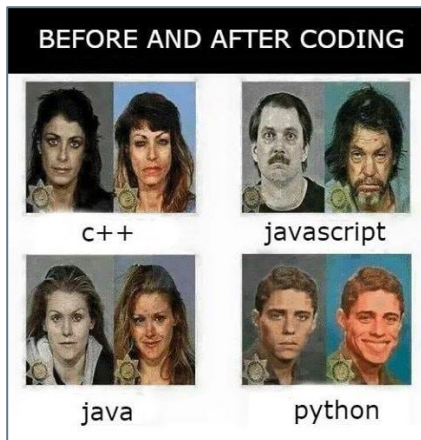
Python is a popular programming language.

It was created by Guido van Rossum, and released in 1991.

It is used for web development, software development, mathematics, system scripting, etc.

Why Python?

Syntax

Variables

```
x = 5
y = 'Hello World!'
```

Comments

```
# This is a comment.
```

Indentation

```
if 5 > 2:
    print('Five is greater than two!')
```

Output and Input

Display output on the screen

```python
print('Hello World!')
print('Hello' + 'World!')
print('Hello', 'World!')
print('Hello', 'World', sep = '-', end = '!')
```

Take input from user

```python
name = input('Enter your name: ')
print('Hello', name)
```

1
Variables

Variables

Containers for storing data values

```
x = 4
x = 'Sally'
print(x)
```

Assign multiple values

```
x, y, z = 'Orange', 'Banana', 'Cherry'
```

One value to multiple variables
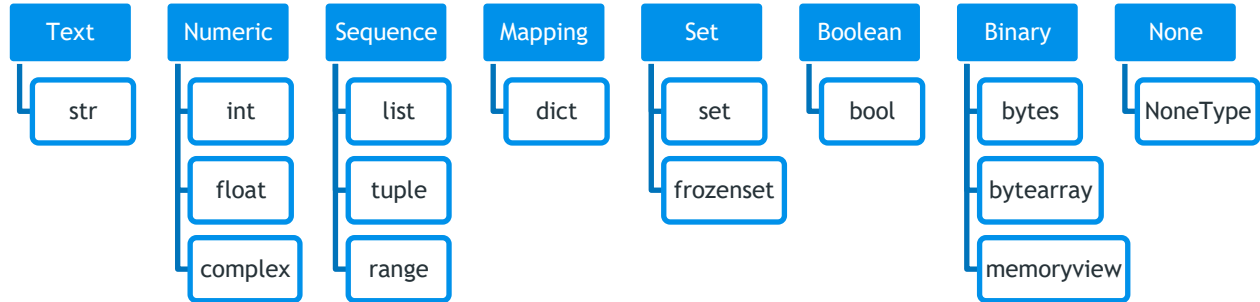
```
x = y = z = 'Orange'
```

Data Types

Get the type

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

Casting

```
a = str(3)
b = int(3)
c = float(3)
d = bool(3)
```

Text	Numeric	Sequence	Mapping	Set	Boolean	Binary	None
str	int	list	dict	set	bool	bytes	NoneType
	float	tuple		frozenset		bytearray	
	complex	range				memoryview	

Numbers

Three numeric types

```
a = 1    # int
b = 2.8  # float
c = 1j   # complex
```

Functions

```
min()  max()  abs()
```

Operators

```
+  -  *  /  %  **  //
```

Strings

Single line and multiline string

```
a = 'Hello!'
b = '''Hello!
How are you?'''
```

F-strings

```
name, age = 'Alireza' , 24
message = f'Hello {name}, you are {age} years old!'
```

Operators

```
+   *
```

Booleans

True or False

```
a = True
b = False
c = 10 >= 9
d = 10 == 9
e = 10 > 9 and 2 + 2 == 5
e = 10 > 9 or 2 + 2 == 5
f = not 2 + 2 == 5
```

Operators

```
<   <=   >   >=   ==   !=   and   or   not
```

Collections

Lists

```
fruits = ['apple', 'banana', 'cherry']
```

Tuples

```
fruits = ('apple', 'banana', 'cherry')
```

Sets

```
fruits = {'apple', 'banana', 'cherry'}
```

Dictionaries

```
fruits = {'apple': 'green', 'banana': 'yellow', 'cherry': 'red'}
```

Lists

Ordered, changeable, and allow duplicate values

```python
fruits = ['apple', 'banana', 'cherry', 'apple', 'cherry']
print(fruits[0], fruits[-3], fruits[1:3], fruits[:-1], fruits[0:-1:2], fruits[::-1])
print('orange' in fruits)
print(len(fruits))
```

Update Lists

```python
fruits[3] = 'orange'
fruits.append('kiwi')
fruits.remove('banana')
fruits.pop(3)
```

Tuples

Ordered, unchangeable, and allow duplicate values

```python
fruits = ('apple', 'banana', 'cherry', 'apple', 'cherry')
print(fruits[0], fruits[-3], fruits[1:3], fruits[:-1], fruits[0:-1:2], fruits[::-1])
print('orange' in fruits)
print(len(fruits))
```

Update tuples

```python
fruitsList = list(fruits)
fruitsList[3] = 'orange'
fruitsList.remove('banana')
fruits = tuple(fruitsList)
```

Sets

Unordered, unchangeable, and no duplicate values

```python
fruits = {'apple', 'banana', 'cherry'}
print('orange' in fruits)
print(len(fruits))
```

Update sets

```python
fruits.add('orange')
fruits.remove('banana')
fruits.pop()
```

Dictionaries

Ordered, changeable, and no duplicate values

```
fruits = {'apple': 'green', 'banana': 'yellow', 'cherry': 'red'}
print(fruits['apple'])
print('orange' in fruits)
print(len(fruits))
print(fruits.keys())
print(fruits.values())
```

Update Dictionaries

```
fruits['apple'] = 'red'
fruits['kiwi'] = 'green'
fruits.pop('banana')
```

2
Conditions

If Statements

If statement

```python
a, b = 200, 100
if a > b:
    print('a')
elif a < b:
    print('b')
else:
    print('=')
```
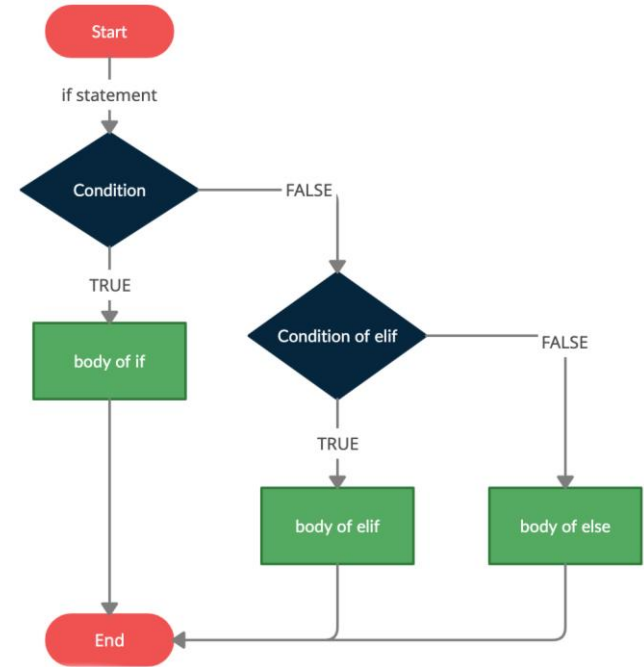
Shorthand if statement

```python
a, b = 200, 100
print('a') if a > b else print('b') if a < b else print('=')
```

3

Loops

While Loops

Execute a set of statements as long as a condition is true

```python
i = 1
while i < 10:
    print(i)
    i += 1
else:
    print(f'Finished after {i} loops')
```

For Loops

Iterating over a collection

```python
for fruit in ['apple', 'banana', 'cherry']:
    print(fruit)
else:
    print('Finished!')
```

Iterating over a string

```python
for letter in 'apple':
    print(letter)
```

Looping Through a Range

```python
for number in range(20):
    print(number)


for number in range(5, 20):
    print(number)


for number in range(5, 20, 2):
    print(number)
```

Break and Continue

Break and continue

```python
for i in range(10):
    if i == 2:
        continue
    if i == 5:
        break
    print(i)
else:
    print(f'Finished after {i} loops')
```

4

Functions

Functions

Blocks of code which only run when they are called

```python
def greeter():
    print('Hello!')
greeter()
```

Arguments and return

```python
def greeter(firstName, lastName):
    greeting = f'Hello {name}!'
    return greeting
message = greeter('Alireza', 'Nezhadshamsi')
print(message)
```

Functions

Keyword arguments

```python
def greeter(firstName, lastName):
    return f'Hello {firstName} {lastName}!'
print(greeter(lastName = 'Nezhadshamsi', firstName = 'Alireza'))
```

Default parameter value

```python
def greeter(firstName = 'Dear', lastName = 'user'):
    return f'Hello {firstName} {lastName}!'
print(greeter('Alireza', 'Nezhadshamsi'))
```

Lambda

Small anonymous functions

```
greeter = lambda: print('Hello')
print(greeter())
```

Arguments

```
greeter = (lambda name = 'Dear user': f'Hello {name}!')
print(greeter(name = 'Alireza'))
```

5

Errors

Try Except

Test a block of code for errors and handle the error

```python
try:
    print(x)
except NameError:
    print('Variable x is not defined')
except:
    print('Something else went wrong')
else:
    print('Nothing went wrong')
```

Exceptions

Handle the error

```
try:
    print(x)
except Exception as e:
    print(e)
```

Raise an error

```
x = -1
if x < 0:
    raise Exception("Please enter a posetive number")
```

Thanks!

You can find me at:

github.com/AleeRezaa

t.me/Alee_Rezaa

alee_rezaa@outlook.com