Python Guidelines for Begineers **DAY - 1**

Day-1 Activities

- Python Intro
- Python Environment Setup
- Basic Syntax & Variables
- Data Types
- User Input
- Arithmetic Operator
- String & F-String

1. Python Introduction

```
*   It was created by Guido van Rossum, and released in 1991.
*   We can easily build Desktop GUI applications.
*   Generel purpose as well as Object Oriented Language.
*   Readable and Maintainable Code
*   It provides libraries for Game Development.
*   Compatible with Major Platforms and Systems.
*   It provides Framework for building websites. i.e, Django, Fla
sk etc.
*   It makes data science tasks easier.
```

2. Python Environment Setup

- Python Interpreter
- Code-Editor
- Jupyter Notebook
- Google Colaborator

Download Link For Windows 8/10: `[Version 3.9 will not work in Windows 7]`

3. Python basic Syntax & variables

First Python Program

In [1]:
```python
print("Hello World. ")
```
```
Hello World.
```

Variable is declared automatically during Assingment

In [2]:
```python
x = 10
print(type(x))
y = x + 44
print(y)
```
```
<class 'int'>
54
```

Double qoutation & Single qoutation

In [3]:
```python
first_name  = "Hello"
last_name = 'World'

print(first_name)
print(last_name)
```
```
Hello
World
```

Case Sensitivity

In [4]:
```python
age = 50
Age = 25
print(age, Age)
```
```
50 25
```

Assigning values to multiple variable

In [5]:
```python
num1 = 10
num2 = 20
num3 = 30
print(num1, num2, num3)
```
```
10 20 30
```

In [6]:
```python
num1, num2, num3 = 10, 20, 30
print(num1, num2, num3)
```
```
10 20 30
```

Adding Strings & adding Numbers

In [8]:
```python
full_name = first_name + " " + last_name
print(full_name)
```
```
Hello World
```

In [9]:
```python
number1 = 100
number2 = 200
add = number1 + number2
print(add)
```
```
300
```

Python Comments

In [10]:
```python
#This is a Comment.
"""Multi line comment.
This is a Comment.
This is a Comment.
"""

print("Comment Example. ")
```
```
Comment Example.
```

4. Data Types

Basic Types of Data

In [11]:
```python
#String Type
name = "Test User"
print(name, type(name))
```
```
Test User <class 'str'>
```

In [12]:
```python
#Integer Type
number = 250
print(number, type(number))
```
```
250 <class 'int'>
```

In [13]:
```python
#Float Type
number = 5.000
print(number, type(number))
number2 = 1e4
print(number2, type(number2))
number3 = 12E3
print(number3, type(number3))
```
```
5.0 <class 'float'>
10000.0 <class 'float'>
12000.0 <class 'float'>
```

In [14]:
```python
#Boolean Type
variable1 = True
print(variable1, type(variable1))
variable2 = False
print(variable2, type(variable2))
```
```
True <class 'bool'>
False <class 'bool'>
```

In [15]:
```python
#Range Type
OneToTen = range(1, 11)
print(OneToTen, type(OneToTen))
```
```
range(1, 11) <class 'range'>
```

Other Types

- List Type
- Tuple Type
- Dictionary Type

```
# [We will learn this on the next session]
```

Type Casting

In [16]:
```python
#String To Integer
number = "12345"
print(number, type(number))

number2 = int(number)
print(number2, type(number2))
```
```
12345 <class 'str'>
12345 <class 'int'>
```

In [17]:
```python
#Integer To String
value = 12345
print(value, type(value))

value2 = str(value)
print(value2, type(value2))
```
```
12345 <class 'int'>
12345 <class 'str'>
```

In [18]:
```python
#String To Float
number = "12345"
print(number, type(number))

number2 = float(number)
print(number2, type(number2))

#Float To String (Try Yourself)
```
```
12345 <class 'str'>
12345.0 <class 'float'>
```

In [19]:
```python
#Integer To Float
value = 10
print(value, type(value))

value2 = float(value)
print(value2, type(value2))

#Float to Integer (Try Yourself)
```
```
10 <class 'int'>
10.0 <class 'float'>
```

5. User Input

Input Function

In [20]:
```python
# Taking String Input
name = input()
print(name)
```
```
python
python
```

In [ ]:
```python

```

In [21]:
```python
# Integer Input
value = input()
print(type(value))
value_int = int(value)
print(type(value_int))
```
```
19
<class 'str'>
<class 'int'>
```

In [22]:
```python
#shortcut way
value = int(input("Enter a number : "))
print(value)
```
```
Enter a number : 10
10
```

In [23]:
```python
#Printing multiple value

name = input("Enter your name : ")
age = int(input("Enter your age : "))
print("Hello", name, ". You are", age, " years old. ")
```
```
Enter your name : karim
Enter your age : 20
Hello karim . You are 20  years old.
```

In [24]:
```python
#Using f-string

print(f"Hello, {name}. You are {age} years old. ")
```
```
Hello, karim. You are 20 years old.
```

6. Python Operators

Arithmetic Operator

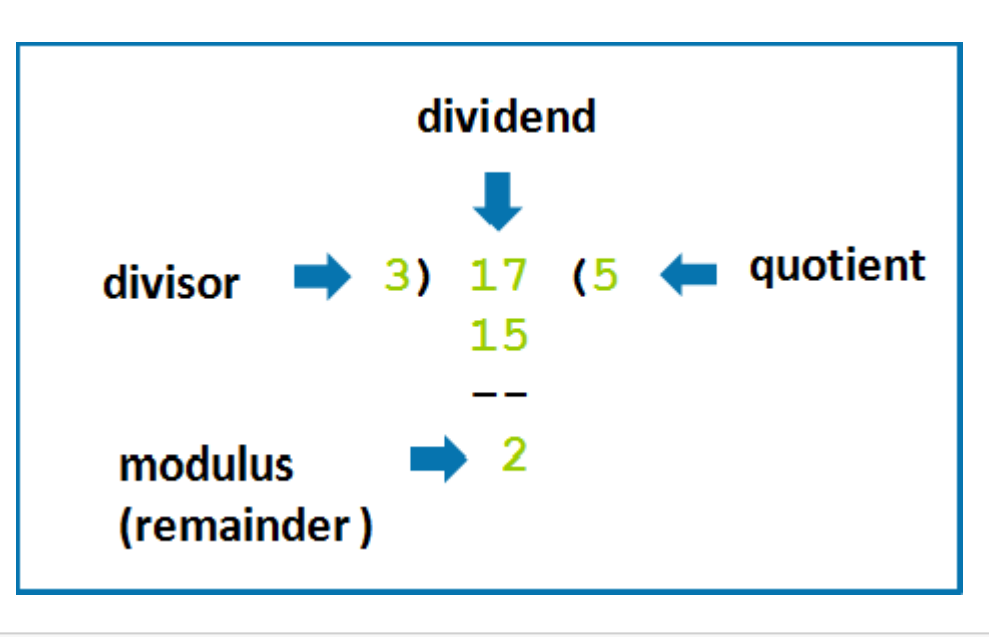
In [25]:
```python
#addition
x = 50
y = 12
z = x + y
print(z)
```
```
62
```

In [27]:
```python
#Subtraction with user input value
x=int(input())
y=int(input())

z= x-y
print(z)
```
```
10
5
5
```

In [28]:
```python
#Multiplication with user input
x=int(input())
y=int(input())

z= x*y
print(z)
```
```
10
20
200
```



In [29]:
```python
#Division with user input
x=int(input())
y=int(input())

z= x/y
print(z)
```
```
10
2
5.0
```

In [30]:
```python
#Modulus with user input
x=int(input())
y=int(input())

z= x%y
print(z)
```
```
10
3
1
```

In [31]:
```python
#Exponential operator
x=int(input())
y=int(input())

z= x**y
print(z)
```
```
2
2
4
```

In [32]:
```python
#Floor Division
x=int(input())
y=int(input())

z= x/y
print(z)
```
```
10
3
3.3333333333333335
```

Assignment Operator

In [33]:
```python
#swaping values of two variable (the common way)
a = 10
b = 5
print(f"a={a}, b={b}")
temp = a
a = b
b = temp
print(f"a={a}, b={b}")
```
```
a=10, b=5
a=5, b=10
```

In [34]:
```python
#swaping values of two variable (the python way)
a = 10
b = 20
print(f"a={a}, b={b}")
a, b = b, a
print(f"a={a}, b={b}")
```
```
a=10, b=20
a=20, b=10
```

In [35]:
```python
#Addition & Assignment
x = 20
x = x + 5
print(x)

#shortcut assignment operator
y = 10
y += 5
print(y)
```
```
25
15
```

Operator	Example	Same As
=	x = 5	x = 5
+=	x += 3	x = x + 3
-=	x -= 3	x = x - 3
*=	x *= 3	x = x * 3
/=	x /= 3	x = x / 3
%=	x %= 3	x = x % 3
//=	x //= 3	x = x // 3
**=	x **= 3	x = x ** 3

Comparison Operator

In [36]:
```python
# == operator (double equal)
print(5==5)
```
```
True
```

In [37]:
```python
# != operator (not equal)
x = 10
y = 6
print(x != y)
```
```
True
```

In [38]:
```python
# > operator  (greater than)
x = 5
y = 10
print(x > y)
```
```
False
```

In [39]:
```python
# < operator (less than)
x = 5
y = 10
print(x < y)
```
```
True
```

In [40]:
```python
# <= operator (less than or equal)
x = 5
y = 10
print(x <= y)
```
```
True
```

In [41]:
```python
# >= operator (greater than or equal)
x = 5
y = 10
print(x >= y)
```
```
False
```

Logical Operator

In [42]:
```python
# and operator
a = 6; b = 7; x = 10; y = 5
print(a < b and x > y)
```
```
True
```

In [43]:
```python
# or operator
a = 6; b = 7; x = 10; y = 5
print(a>b or x > y)
```
```
True
```

In [44]:
```python
# not operator
x = 10
print(not(x<5))
```
```
True
```

Membership Operator

In [45]:
```python
# in operator
movie_title = "Aynabaji"
print('Ayna' in movie_title)
print('bazi' in movie_title)
```
```
True
False
```

In [46]:
```python
# not in operator
quote = "Time stops, when lies become the truth."
print('Time' not in quote)
print('truth' in quote)
```
```
False
True
```

Identity Operators, Bitwise Operator

```
[We can skip it for now]
```

7. Some Common String Methods

Multiline String

In [47]:
```python
a = """Lorem ipsum dolor sit amet,
consectetur adipiscing elit,
sed do eiusmod tempor incididunt
ut labore et dolore magna aliqua."""

print(a)
```
```
Lorem ipsum dolor sit amet,
consectetur adipiscing elit,
sed do eiusmod tempor incididunt
ut labore et dolore magna aliqua.
```

Length of a String

In [48]:
```python
length = len(a)
print(length)

name = input()
length = len(name)
print(length)
```
```
123
python
6
```

Making string UpperCase

In [49]:
```python
name = "Test Name"
upper_name = name.upper()
print(name)
```
```
Test Name
```

Making String LoweCase

In [50]:
```python
name = "Test Name"
lower_name = name.lower()
print(lower_name)
```
```
test name
```

Strip Method

In [51]:
```python
string = "   Hello, World!    "
string2 = string.strip()
print(string2)

print(len(string), len(string2))
```
```
Hello, World!
20 13
```

Split Method

In [52]:
```python
string = "person1,person2,person3"
names = string.split(',')
print(names)
```
```
['person1', 'person2', 'person3']
```

Replace Method

In [53]:
```python
message = "Hello World."
new_message = message.replace('Hello', 'GoodBye')
print(new_message)
```
```
GoodBye World.
```

In [ ]:
```python

```