Python 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 (server-side),
- software development,
- mathematics,
- 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.

Python Syntax Vs Other Programing 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.

Installing Python

Refer this video to install Python Programming Language

How to Install Python on Windows (https://www.youtube.com/watch?v=VWgs_iTojoA)

Printing Hello TechEdu !!

Now we will start writing our first line of code in Python

In [1]:
```
1 print('Hello TechEdu !!')
```

```
Hello TechEdu !!
```

In [1]:
```
1 print(True)
2 print(False)
3 print(45)
4 print(45.2)
5 print('TechEdu')
```

```
True
False
45
45.2
TechEdu
```

Yeah !!! We have Written our First Line of Core

Python Identation

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('5 is greater than 2')
```

In [4]:
```
if 5>2:
    print('Yes')
```

```
Yes
```

Python will give you an error if you skip the indentation:

code below will produce Identation Error

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

Python Comments

- Comments can be used to explain Python code.

- Comments can be used to make the code more readable.
- Comments can be used to prevent execution when testing code.

In [11]:
```
print('Hello Learners!!!') #This is my first line

#I am Student
#currently in Final Year

```

```
Hello Learners!!!
```

Creating Comment

Comments starts with a #, and Python will ignore them:

In [3]:
```
#this is comment
print('Hello Team !')
```

```
Hello Team !
```

Comments can be placed at the end of a line, and Python will ignore the rest of the line:

In [4]:
```
print("Hello, TechEdu!") #This is a comment
```

```
Hello, TechEdu!
```

A comment does not necessarily have to be text that explains the code, it can also be used to prevent Python from executing code:

In [5]:
```
#print("Hello, Team")
print("Cheers, TechEdu !")
```

```
Cheers, TechEdu !
```

Python Variable

What is Python Variables ?

Variables are containers for storing data values.

Creating Variables

- Python has no command for declaring a variable.
- A variable is created the moment you first assign a value to it.

In [16]:
```
a = 11
print(10) #line 1
print(a) #line 2
print('a')
```

```
10
11
a
```

In [9]:
```
x = 2022
y = "TechEdu"
print(x)
print(y)
```

```
2022
TechEdu
```

In [18]:
```
techedu = 10
print(techedu)
techedu = 15
print(techedu)
techedu = 20
print(techedu)
```

```
10
15
20
```

Variables do not need to be declared with any particular type, and can even change type after they have been set.**

Casting Variables

If you want to specify the data type of a variable, this cxan be done using casting

In [24]:
```
a = int(14)
b = float(14)
c = str(14)
d = 14.0
```

In [22]:
```
print(a)
print(b)
print(c)
print(d)
```

```
14
14.0
14
14
```

Getting the type of Variable

You can get the data type of variable with **type()** function

In [28]:
```
print(type(a))
print(type(b))
print(type(c))
print(type(d))
```

```
<class 'int'>
<class 'float'>
<class 'str'>
<class 'float'>
```

In [29]:
```
x = 10
y = 'TechEdu'
z = 'true'
a = '45'

print(type(x))
print(type(y))
print(type(z))
```

```
<class 'int'>
<class 'str'>
<class 'bool'>
```

Single and Double Quote

String variable can be declared either by single or double inverted comma

In [4]:
```
a = 'Techedu'

#this is same as

a = "Techedu"
```

Case Sensitive

Variable name are case Sensitive

In [6]:
```
a = 14
 #is not same as

A = 'Upskilling'
```

In [ ]:
```
techedu = 10
Techedu = 15
TechEdu = 20
```

Here capital A will not overwrite a.

Variable name

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

Rules for Python variable:

- A variable name must start with a letter or the underscore characters
- A variable name can only contain alpha-numaric characters and underscore(A-z, 0-9, and_)
- Variable names are case_sensitive (age, Age and AGE are three different variable)

EXAMPLE

Legal variable name

In [10]:
```
myvar  =  'John'
my_var =  'John'
_my_var = 'John'
MYVAR   = 'John'
myvar2  =  'John'
```

EXAMPLE

illegal variable name.

Following code will be showing error as the names of Variables are illigal.

In [9]:
```
2myvar  = 'John'
my-var  = 'John'
my var  = 'John'
```

```
  Input In [9]
    illegal variable name
                ^
SyntaxError: invalid syntax
```

Multi Words Variable Names

Variable names with more than one word can be difficult to read.

There are several tecniques you can use to make them more readable;

- **Camle Case**

In [1]:
```
prime_minister_of_India = 2019
primeMinisterOfIndia =
```

#camle Case

myTeamTechedu

#pascle

MyTeamTeched

#snake

my_team_techedu

Every Word, except the first starts with capital letter.

In [ ]:
```
my_Team_Techedu = 14
```

In [11]:
```
myTeamTechedu = 14
```

- **Pascal Case**

Each word starts with capital letter

In [12]:
```
MyTeamTechedu = 14
```

- **Snake Case**

Each word is separated by an underscore character.

In [15]:
```
my_team_techedu = 14
```

Many Values to multiple variables

In [4]:
```
x = 10
y = 20
z = 30

x,y,z = 10,20,30
print(x)
print(y)
print(z)
print(x,y,z)
```

```
10
20
30
10 20 30
```

In [16]:
```
a,b,c = 1,2,3

print(a)
print(b)
print(c)
```

```
1
2
3
```

In [5]:
```
ram , shaym , Ravi = 1000,2000,0
print(Ravi)
```

```
0
```

One value to multiple Variables

In [11]:
```
a = b = c = 'Techedu'
print(a,b,c)
```

```
Techedu Techedu Techedu
```

In [7]:
```
print(a)
print(b)
print(c)
```

```
Techedu
Techedu
Techedu
```

Unpack a Collection

Unpack a Collection

If you have a collection of values in a list,tuple,etc. Python allows you to extract the values into variables. This is called **Unpacking**

In [13]:
```
fruits = ['apple','banana','mango']
x,y,z = fruits
```

In [20]:
```
print(x)
print(y)
print(z)
```

```
apple
banana
mango
```

Output Variables

The python **print** statement is often used to output variables.

To combine both text and a variable, python uses the **+** Character.

In [22]:
```
a,b =8,10

print('This number is 10')
print('This number is',a)
print('This number is',a, 'and we have stored it as',b)
```

```
This number is 10
This number is 8
This number is 8 and we have stored it as 10
```

In [27]:
```
x = 'Upskilling'

print('We are ' + x)
```

```
We are Upskilling
```

You can also use **+** character to add a variable to another variable.

In [30]:
```
a = 'Information'
b = 'Technology'

print(a+b)
print(a,b)
print(a+' '+b)
```

```
InformationTechnology
Information Technology
Information Technology
```

In [48]:
```
a+b
```

Out[48]:
```
'InformationTechnology'
```

In [24]:
```
a+' '+b
```

Out[24]:
```
'Information Technology'
```

Global Variables

Variable that are created outside of a function (as in all of the example above) are known as global vriables.

Global variables can be used by everyone, both inside of function and outside.

Example

Create a variable can be used by outside of a function, and use it inside the function

In [26]:
```
x  =  "awesome"
def myfune () :
    print("python is" +  X)

myfune()
```

In [27]:
```
if you create  a variable with the same name inside a function. The global v

```

```
  Input In [27]
    if you create  a variable with the same name inside a function. The global
 variable with the same name will remain as it was, global and with the orignal
value.
           ^
SyntaxError: invalid syntax
```

In [28]:
```
Example

Create a variable inside a function, with the same name as the global variab

x =  'awesome'


 def myfunc () :
        x = "fantastic"
        print("Python is "  + x)
```

```
  Input In [28]
    Create a variable inside a function, with the same name as the global varia
ble
            ^
SyntaxError: invalid syntax
```

In [ ]:
```
The global keyword

Normally, when you create a variabvle inside a function, the variable is loc

To create a global variable inside a function,you can use the global keyword


Example

 if you use the global keyword,the variable belongs to the global scope.87
```

In [ ]:
```

```

In [ ]:
```

```

Data Types

- Numbers : Int, Float, Complex
- String
- Boolean (True and False)
- List
- Sets
- Dictionaries
- Tuples

Numbers

1. Integer

All whole Numbers and their negations comes under Int

In [31]:
```
a = 10
type(a)
b = -10
print(type(b))
```

```
<class 'int'>
```

2.Float

All Natural numbers and their negations comes under **FLOAT**

In [51]:
```
x = 10.3256897
type(x)
```

Out[51]:
```
float
```

Inverting Int to float

In [34]:
```
a = 10
print(type(10))

b = float(a)
print(type(b))

c = complex(b)
print(c)

```

```
<class 'int'>
<class 'float'>
(10+0j)
```

3.Complex Numbers

In [57]:
```
a = 10 + 2j
```

type(a)

String

All the alphanumerical data which may also contain symbols and signs comes under strings

In [60]:
```
a = '#upskilling9'
```

In [61]:
```
type(a)
```

Out[61]: str

In [62]:
```
a = 10
str(10)
```

Out[62]: '10'

In [63]:
```
a = 'techedu'
```

In [64]:
```
a.upper()
```

Out[64]: 'TECHEDU'

In [65]:
```
a.lower()
```

Out[65]: 'techedu'

In [66]:
```
a.title()
```

Out[66]: 'Techedu'

In [67]:
```
a.capitalize()
```

Out[67]: 'Techedu'

In [69]:
```
b = 'The man is human'
b.lower()
```

Out[69]: 'the man is human'

In [70]:
```
b.upper()
```

Out[70]: 'THE MAN IS HUMAN'

In [72]:
```
b.capitalize()
```

Out[72]: 'The man is human'

In [73]:
```
b.title()
```

Out[73]: 'The Man Is Human'

In [83]:
```
a = 'TECHEDU'
a[-1]
a[::2]
#a[start:end:step size]
```

Out[83]: 'TCEU'

In [84]:
```
a
```

Out[84]: 'TECHEDU'

In [85]:
```
a+'4'
```

Out[85]: 'TECHEDU4'

In [86]:
```
'4'+a
```

Out[86]: '4TECHEDU'

In [87]:
```
a
```

Out[87]: 'TECHEDU'

In [92]:
```
a[0:3]+'4'+a[4:]
```

Out[92]: 'TEC4EDU'

In [ ]:
```
T1C2E3U #reversed.

```

In [35]:
```
a ='T1C2E3U'
```

In [40]:
```
a[::-2]
```

Out[40]: 'UECT'

In [37]:
```
a[0::2]
```

Out[37]: 'TCEU'

In [38]:
```
a[1::2]
```

Out[38]: '123'

In [39]:
```
a[1::2] + a[0::2]
```

Out[39]: '123TCEU'

In [44]:
```
a = 'TECHEDU'
a[6:2:-1]
```

Out[44]: 'UDEH'

List

In [53]:
```
vowels = ['a','e','i','o','u',1,2,0.25,4+2j, False]
```

In [57]:
```
vowels[::-1]
```

Out[57]: [False, (4+2j), 0.25, 2, 1, 'u', 'o', 'i', 'e', 'a']

In [58]:
```
vowels.append('2')
```

In [59]:
```
vowels
```

Out[59]: ['a', 'e', 'i', 'o', 'u', 1, 2, 0.25, (4+2j), False, '2']

In [60]:
```
vowels.clear()
```

In [61]:
```
vowels
```

Out[61]: []

In [69]:
```
a = [1,2,3,1.2,1]
```

In [70]:
```
b = a.copy()
```

In [71]:
```
b
```

Out[71]: [1, 2, 3, 1.2, 1]

In [72]:
```
len(b)
```

Out[72]: 5

In [73]:
```
b.count(1)
```

Out[73]: 2

In [78]:
```
b.insert(1,0) # list_name.insert(index,value)

#the index number must be less than the lenghth of the list
```

In [79]:
```
1 b.pop(0)
```

Out[79]: 1

In [84]:
```
1 b = ['a','b','c']
2
3 b.pop(0) #list_name.pop(index_position)
```

Out[84]: 'a'

In [85]:
```
1 b
```

Out[85]: ['b', 'c']

In [90]:
```
1 a = [[1,2,3],[4,5,6]] #array
```

In [92]:
```
1 b.append('d')
```

In [93]:
```
1 b
```

Out[93]: ['b', 'c', 'd']

In [94]:
```
1 b.remove('d')
```

In [95]:
```
1 b
```

Out[95]: ['b', 'c']

In [96]:
```
1 a
```

Out[96]: [[1, 2, 3], [4, 5, 6]]

In [97]:
```
1 b
```

Out[97]: ['b', 'c']

In [98]:
```
1 a+b
```

Out[98]: [[1, 2, 3], [4, 5, 6], 'b', 'c']

In [103]:
```
(b + a)*3
```

Out[103]:
```
['b',
 'c',
 [1, 2, 3],
 [4, 5, 6],
 'b',
 'c',
 [1, 2, 3],
 [4, 5, 6],
 'b',
 'c',
 [1, 2, 3],
 [4, 5, 6]]
```

In [107]:
```
statement = [100,12,45,78,78,4578,96]
statement[-3:]
```

Out[107]:
```
[78, 4578, 96]
```

In [108]:
```
b
```

Out[108]:
```
['b', 'c']
```

In [111]:
```
string = []
integer = []a
```

Out[111]:
```
False
```

In [116]:
```
a
```

Out[116]:
```
['g']
```

In [117]:
```
a = ['z',True,45.9,48,20+4j]
```

In [118]:
```
a.append(2)
```

In [119]:
```
a
```

Out[119]:
```
['z', True, 45.9, 48, (20+4j), 2]
```

In [120]:
```
b = a.copy()
```

In [121]:
```
b
```

Out[121]:
```
['z', True, 45.9, 48, (20+4j), 2]
```

In [122]:
```
b.clear()
```

In [124]:
```
1 a.append(2)
```

In [125]:
```
1 a
```

Out[125]: ['z', True, 45.9, 48, (20+4j), 2, 2]

In [126]:
```
1 a.count(2)
```

Out[126]: 2

In [127]:
```
1 len(a)
```

Out[127]: 7

In [128]:
```
1 a.insert(1,'Techedu')
```

In [129]:
```
1 a
```

Out[129]: ['z', 'Techedu', True, 45.9, 48, (20+4j), 2, 2]

In [130]:
```
1 a.pop(1)
```

Out[130]: 'Techedu'

In [131]:
```
1 a
```

Out[131]: ['z', True, 45.9, 48, (20+4j), 2, 2]

In [132]:
```
1 a.remove(2)
```

In [133]:
```
1 a
```

Out[133]: ['z', True, 45.9, 48, (20+4j), 2]

In [135]:
```
1 a[::-1]
```

Out[135]: [2, (20+4j), 48, 45.9, True, 'z']

In [136]:
```
1 a[:2]
```

Out[136]: ['z', True]

In [137]:
```
1 a[0:4]
```

Out[137]: ['z', True, 45.9, 48]

In [138]:
```
a[::-2]
```

Out[138]: [2, 48, True]

In [141]:
```
a
```

Out[141]: ['om,9,8']

In [145]:
```
a.extend(['a','b'])
```

In [146]:
```
a
```

Out[146]: ['om,9,8', 'U', 'p', 's', 'k', 'i', 'l', 'l', 'i', 'n', 'g', 'a', 'b']

In [147]:
```
a = [8,4,8,9,2,1,0]
```

In [148]:
```
a.sort()
```

In [150]:
```
a[::-1]
```

Out[150]: [9, 8, 8, 4, 2, 1, 0]

In [1]:
```
#separating Integers, Float,String and Boolean

l = [12,10.2,'Mohanish',True, 2+2j]
```

In [3]:
```
integer = []
integer.append(l[0])
integer
```

Out[3]: [12]

In [5]:
```
flo = []
flo.append(l[1])
flo
```

Out[5]: [10.2]

In [6]:
```
string = []
string.append(l[2])
string
```

Out[6]: ['Mohanish']