Python Programming for Data Analytics

Python Programming

Python is a general-purpose programming language, so it can be used for many things. Python is used for web development, AI, machine learning, operating systems, mobile application development, and video games

Libraries

- 1 Pandas
  - Data Analysis
  - Data Manipulation
- 2 Numpy
  - Numerical Python
  - Array Operations
- 3 Scipy
  - Scientific Python
  - Scientific Operations
- 4 Matplotlib
  - Data Visualization
- 5 Seaborn
  - Also used for Data visualization with additional plot types
- 5 Scikit-Learn
- 6 Tensorflow
- 7 Keras

Python Program

1 Print a Statement

```
print("Hello Champs")

Hello Champs
```

Syntax

2 Variable and Types

```
a = 5
b = "Champ"
c = [1,2,3]
d = 5.2
%whos

Variable   Type     Data/Info
-----------------------------
a          int      5
b          str      Champ
c          list     n=3
d          float    5.2
e          list     n=2
```

Syntax

3 Delete a variable

```
a = 5
b = "Champ"
c = [1,2,3]
d = 5.2
del d
%whos

Variable   Type    Data/Info
----------------------------
a          int     5
b          str     Champ
c          list    n=3
```

Syntax

4 Arithmetic Operator

```
a=10
b=5
print("a+b: ",a+b)
print("a-b: ",a-b)
print("a/b: ",a/b)
print("a%b: ",a%b)
print("a*b: ",a*b)
print("a//b: ",a//b)
print("a**b: ",a**b) #pow(a,b) #power

a+b:  15
a-b:  5
a/b:  2.0
a%b:  0
a*b:  50
a//b:  2
a**b:  100000
```

Syntax

5 Boolean & Comparison

```
a=False
b=True
print(type(a))
print(type(b))
print((a and b) != (a or b))

<class 'bool'>
<class 'bool'>
True
```

Syntax - Boolean

```
x=10
y=5
print("x less than y: ",x<y)
print("x greater than y: ",x>y)
print("x equivalent to y: ",x==y)
print("x not equal to y: ",x<y)
print("x less than or equal to y: ",x<=y)
print("x greater than equal to y: ",x>=y)

x less than y:  False
x greater than y:  True
x equivalent to y:  False
x not equal to y:  False
x less than or equal to y:  False
x greater than equal to y:  True
```

Comparison

6 Getting user Input

```
name = str(input("Enter your name: "))
yearofbirth = int(input("Enter your year of Birth: "))
height = float(input("Enter your Height: "))
print("Name: {0}, YearOfBirth: {1}, Height: {2}".format(name,yearofbirth,height))

Enter your name: Sanjay
Enter your year of Birth: 1996
Enter your Height: 5.5
Name: Sanjay, YearOfBirth: 1996, Height: 5.5
```

Syntax

7 If Condition

```
a = 5
b = 5
if a<b:
  print("a is less than b")
elif a==b:
  print("a is equal to b")
  print("a is not less than b")
  print("a is not greater than b")
else:
  print("a is greater than or equal to b")

a is equal to b
a is not less than b
a is not greater than b
```

Syntax

8 While

```
i = 0
while i<10:
  print(i)
  i+=1

0
1
2
3
4
5
6
7
8
9
```

Syntax

```
i = 1
while True:
  if i%10 !=0:
    print("Checking...")
    i+=1
    continue
  print("If-Completed")
  break
print("Completed")

Checking...
Checking...
Checking...
Checking...
Checking...
Checking...
Checking...
Checking...
Checking...
If-Completed
Completed
```

If & While with Continue & Break

9 For

```
for i in range(5):
  print(i)

0
1
2
3
4
```

Syntax

```
for i in range(1,5):
  print(i)

1
2
3
4
```

Syntax

10 Data Structure

List

- Example: data = ["Champ", 67, "Red"]
- Locate: 1 — data[2]
- Add Values: 2 — data = data + "India"
- Delete: 3 — del data
- Copy: 4 — data2 = data.copy()

Tuple

- Example: data = ("Champ", 67, "Red")
- Locate: 1 — data[2]
- Add Value: 2 — Immutable
- delete: 3 — del data — not possible for individual data
- copy: 4 — data2 = data

Set

- Example: data = {"Champ", 67, "Red"}
- Locate: 1 — "Red" in data
- Add values: 2 — data.add("India") — data.update({24,"Reading Books"})
- delete: 3 — del data — data.Remove("Red")
- Copy: 4 — data2 = data.copy()

Dictionary

- Example: data = {"Name":"Champ", "Weight":67, "Color":"Red"}
- Locate: 1 — data["Color"]
- Adding Values: 2 — data["Hobby"] = "Reading Books"
- Delete: 3 — del data — del data["Hobby"]
- Copy: 4 — data2 = data.copy()

11 List

```
numList = []
for i in range (1,10):
  print(i)
  numList.append(i*16)
print(numList)

1
2
3
4
5
6
7
8
9
[16, 32, 48, 64, 80, 96, 112, 128, 144]
```

Syntax

12 Functions

```
def display():
  print("Hello Champs")
display()

Hello Champs
```

Without Arguement

13 String

```
[95] data = "Hello Champs"

print(len(data)) #length of data
print(data[2]) #letter by its index
print(data[:6]) #Until Indexx
print(data[3:]) #From Index
print(data[1:11]) #From & To Index
print(data[::-1]) #reverse by index

12
l
Hello
lo Champs
ello Champ
spmahC olleH
```

String and Index