Python Basics

Python syntax

In [3]:
```
if 5 > 2:
  print("Five is greater than two!")
```

```
Five is greater than two!
```

Python Comments

In [14]:
```
#This is a comment
print("Hello, World!")
```

```
Hello, World!
```

In [15]:
```
"""
This is a comment
written in
more than just one line
"""
print("Hello, World!")
```

```
Hello, World!
```

Python Variables

In [16]:
```
#A variable is created the moment you first assign a value to it.
x = 5
y = "John"
print(x)
print(y)
#for diplaying the data type of a var
print(type(x))
```

```
5
John
<class 'int'>
```

In [17]:
```
#Global Variables
x = "awesome"

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

myfunc()

print("Python is " + x)
```

```
Python is fantastic
Python is awesome
```

Python Data Types

In [18]:
```
x = "Hello World" #str
x = 20 #int
x = 20.5 #float
x = ["apple", "banana", "cherry"] #list
x = {"name" : "John", "age" : 36} #dic
```

Python Lists

In [19]:
```
#List items are ordered, changeable, and allow duplicate values but tuples is unchangeable
#A list can contain different data types:
thislist = ["abc", 34, True, 40, "male","kiwi", "melon", "mango"]
print(thislist[1])
print(thislist[-1])
print(thislist[2:5])
print(thislist[2:])
```

```
34
mango
[True, 40, 'male']
[True, 40, 'male', 'kiwi', 'melon', 'mango']
```

In [20]:
```
#To change the value of a specific item, refer to the index number:
thislist = ["apple", "banana", "cherry"]
thislist[1] = "blackcurrant"
print(thislist)
thislist.append("orange")
print(thislist)
thislist.insert(1, "orange")
print(thislist)
```

```
['apple', 'blackcurrant', 'cherry']
['apple', 'blackcurrant', 'cherry', 'orange']
['apple', 'orange', 'blackcurrant', 'cherry', 'orange']
```

In [21]:
```
thislist = ["apple", "banana", "cherry"]
thislist.remove("banana")
print(thislist)
```

```
['apple', 'cherry']
```

In [22]:
```
thislist = ["apple", "banana", "cherry"]
for x in thislist:
  print(x)
```

```
apple
banana
cherry
```

Python If statements

In [24]:
```
a = 33
b = 200
if b > a:
print("b is greater than a")# you will get an error
```

```
b is greater than a
```

In [25]:
```
a = 33
b = 33
if b > a:
  print("b is greater than a")
elif a == b:
  print("a and b are equal")
```

```
a and b are equal
```

In [26]:
```
a = 200
b = 33
if b > a:
  print("b is greater than a")
elif a == b:
  print("a and b are equal")
else:
  print("a is greater than b")
```

```
a is greater than b
```

In [27]:
```
#nested if
x = 41

if x > 10:
  print("Above ten,")
  if x > 20:
    print("and also above 20!")
  else:
    print("but not above 20.")
```

```
Above ten,
and also above 20!
```

Python Loops

In [28]:
```
#while loops
i=1
while i<6:
    print(i)
    i +=1
```

```
1
2
3
4
5
```

In [29]:
```
#the break statement we can stop the loop even if the while condition is true:
i = 1
while i < 6:
  print(i)
  if i == 3:
    break
  i += 1
```

```
1
2
3
```

In [30]:
```
#the continue statement we can stop the current iteration, and continue with the next:
i = 0
while i < 6:
  i += 1
  if i == 3:
    continue
  print(i)
```

```
1
2
4
5
6
```

In [31]:
```
#for loop
#A for loop is used for iterating over a sequence (that is either a list, a tuple, a dictionary, a set, or a string).
fruits = ["apple", "banana", "cherry"]
for x in fruits:
  print(x)
```

```
apple
banana
cherry
```

In [32]:
```
#loop through the letters
for x in "banana":
  print(x)
#break statement
fruits = ["apple", "banana", "cherry"]
for x in fruits:
  print(x)
  if x == "banana":
    break
```

```
b
a
n
a
n
a
apple
banana
```

In [33]:
```
#continue statement
fruits = ["apple", "banana", "cherry"]
for x in fruits:
  if x == "banana":
    continue
  print(x)
```

```
apple
cherry
```

In [34]:
```
#The range() function returns a sequence of numbers, starting from 0 by default, and increments by 1 (by default), and ends at a specified number.
for x in range(6):
  print(x)
```

```
0
1
2
3
4
5
```

In [35]:
```
for x in range(2, 6):
  print(x)
```

```
2
3
4
5
```

In [36]:
```
for x in range(2, 30, 3):
  print(x)
```

```
2
5
8
11
14
17
20
23
26
29
```

In [37]:
```
#nested loop
adj = ["red", "big", "tasty"]
fruits = ["apple", "banana", "cherry"]

for x in adj:
  for y in fruits:
    print(x, y)
```

```
red apple
red banana
red cherry
big apple
big banana
big cherry
tasty apple
tasty banana
tasty cherry
```