In [1]:

```python
#Hellow World
print("Hellow World")
```

```
Hellow World
```

In [2]:

```python
#Python uses indentation to indicate a block of code.
if 5 > 2:
  print("Five is greater than two!")
```

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

Python Variables

In [3]:

```python
"""
This is a comment
written in
more than just one line
"""
x = 5
y = "Hello, World!"
print (x)
print(y)
```

```
5
Hello, World!
```

In [4]:

```python
a = 4
A = "Sally"
#A will not overwrite a
print(a,A)
```

```
4 Sally
```

In [5]:

```python
# assign the same value to multiple variables in one line
x = y = z = "Orange"
print(x)
print(y)
print(z)
```

```
Orange
Orange
Orange
```

In [6]:

```
#use the + character to add a variable to another variable:
x = "Python is "
y = "awesome"
z =  x + y
print(z)
type(z)
```

```
Python is awesome
```

Out[6]:

```
str
```

Python List

In [7]:

```
#Lists are used to store multiple items in a single variable.
thislist = ["apple", "banana", "cherry"]
print(thislist)
```

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

In [8]:

```
#List items are indexed, the first item has index [0], the second item has index
thislist = ["apple", "banana", "cherry"]
print(thislist[1])
```

```
banana
```

In [ ]:

In [9]:

```
#Return the third, fourth, and fifth item:
thislist = ["apple", "banana", "cherry", "orange", "kiwi", "melon", "mango"]
print(thislist[2:5])
```

```
['cherry', 'orange', 'kiwi']
```

In [10]:

```
#Check if "apple" is present in the list:
thislist = ["apple", "banana", "cherry"]
if "apple" in thislist:
  print("Yes, 'apple' is in the fruits list")
```

```
Yes, 'apple' is in the fruits list
```

In [11]:

```
# change the scond item in the list
thislist = ["apple", "banana", "cherry"]
thislist[1] = "blackcurrant"
print(thislist)
```

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

In [12]:

```
# using the append() method to append an item
thislist = ["apple", "banana", "cherry"]
thislist.append("orange")
print(thislist)
```

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

In [13]:

```
#insert an item to second position
thislist = ["apple", "banana", "cherry"]
thislist.insert(1, "orange")
print(thislist)
```

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

In [14]:

```
#add the elements of one list to other
thislist = ["apple", "banana", "cherry"]
tropical = ["mango", "pineapple", "papaya"]
thislist.extend(tropical)
print(thislist)
```

```
['apple', 'banana', 'cherry', 'mango', 'pineapple', 'papaya']
```

Python Tuple

In [15]:

```
#Tuples are used to store multiple items in a single variable.
#Tuple items are ordered, unchangeable, and allow duplicate values.
thistuple = ("apple", "banana", "cherry")
print(thistuple)
```

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

Python Dictionary

Dictionaries are used to store data values in key:value pairs.A dictionary is a collection which is ordered*, changeable and does not allow duplicates.

In [16]:

```
thisdict = {
  "brand": "Ford",
  "model": "Mustang",
  "year": 1964,
  "year": 2020
}
print(thisdict)
```

```
{'brand': 'Ford', 'model': 'Mustang', 'year': 2020}
```

Python Conditions and Loops

In [17]:

```
#An "if statement" is written by using the if keyword.
a = 33
b = 200
if b > a:
  print("b is greater than a")
#print("b is greater than a")
```

```
b is greater than a
```

In [18]:

```
#elseif
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 [19]:

```
#one line if statement
if a > b: print("a is greater than b")
```

```
a is greater than b
```

In [20]:

```
#With the while loop we can execute a set of statements as long as a condition is
i = 1
while i < 6:
  print(i)
  i += 1
```

```
1
2
3
4
5
```

In [21]:

```
#A for loop is used for iterating over a sequence
fruits = ["apple", "banana", "cherry"]
for x in fruits:
  print(x)
```

```
apple
banana
cherry
```

Python Functions

In [22]:

```
#In Python a function is defined using the def keyword:
def my_function():
  print("Hello from a function")
my_function()
```

```
Hello from a function
```

Numpy

NumPy is a Python library. NumPy is used for working with arrays. It also has functions for working in domain of linear algebra, fourier transform, and matrices. NumPy is short for "Numerical Python".

In [23]:

```
import numpy as np
print(np.__version__)
```

```
1.18.5
```

In [24]:

```
import numpy as np
arr = np.array([1, 2, 3, 4, 5])
print(arr)
print(type(arr))
```

```
[1 2 3 4 5]
<class 'numpy.ndarray'>
```

Pandas

Pandas is a Python library used for working with data sets. It has functions for analyzing, cleaning, exploring, and manipulating data. The name "Pandas" has a reference to both "Panel Data", and "Python Data Analysis" and was created by Wes McKinney in 2008.

In [25]:

```
import pandas as pd

mydataset = {
  'cars': ["BMW", "Volvo", "Ford"],
  'passings': [3, 7, 2]
}

myvar = pd.DataFrame(mydataset)

print(myvar)
print(type(myvar))
```

```
    cars  passings
0    BMW         3
1  Volvo         7
2   Ford         2
<class 'pandas.core.frame.DataFrame'>
```

In [26]:

```
import pandas as pd

print(pd.__version__)
```

```
1.0.5
```

In [27]:

```
#Pandas DataFrames
#A Pandas DataFrame is a 2 dimensional data structure, like a 2 dimensional array
import pandas as pd

data = {
  "calories": [420, 380, 390],
  "duration": [50, 40, 45]
}

#load data into a DataFrame object:
df = pd.DataFrame(data)

print(df)
```

```
   calories  duration
0       420        50
1       380        40
2       390        45
```

In [28]:

```
import pandas as pd

df = pd.read_csv('data.csv')

df
```

Out[28]:

	age	gender	genre
0	20	1	HipHop
1	23	1	HipHop
2	25	1	HipHop
3	26	1	Jazz
4	29	1	Jazz
5	30	1	Jazz
6	31	1	Classical
7	33	1	Classical
8	37	1	Classical
9	20	0	Dance
10	21	0	Dance
11	25	0	Dance
12	26	0	Acoustic
13	27	0	Acoustic
14	30	0	Acoustic
15	31	0	Classical
16	34	0	Classical
17	35	0	Classical
18	27	0	Acoustic
19	30	0	Acoustic
20	31	0	Classical

In [29]:
```
print(df.duplicated())
```

```
0     False
1     False
2     False
3     False
4     False
5     False
6     False
7     False
8     False
9     False
10    False
11    False
12    False
13    False
14    False
15    False
16    False
17    False
18     True
19     True
20     True
dtype: bool
```

In [30]:

```
df.drop_duplicates(inplace = True)
df
```

Out[30]:

	age	gender	genre
0	20	1	HipHop
1	23	1	HipHop
2	25	1	HipHop
3	26	1	Jazz
4	29	1	Jazz
5	30	1	Jazz
6	31	1	Classical
7	33	1	Classical
8	37	1	Classical
9	20	0	Dance
10	21	0	Dance
11	25	0	Dance
12	26	0	Acoustic
13	27	0	Acoustic
14	30	0	Acoustic
15	31	0	Classical
16	34	0	Classical
17	35	0	Classical

In [31]:

```
print(df.duplicated())
```

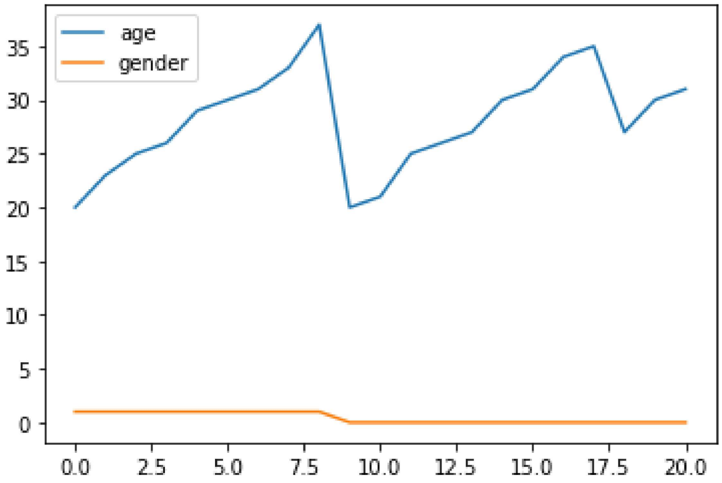
```
0     False
1     False
2     False
3     False
4     False
5     False
6     False
7     False
8     False
9     False
10    False
11    False
12    False
13    False
14    False
15    False
16    False
17    False
dtype: bool
```

```
In [32]: import pandas as pd
         import matplotlib.pyplot as plt

         df = pd.read_csv('data.csv')

         df.plot()

         plt.show()
```



```
In [ ]:
```