Assignment Week2 Notebook

1. Assigning Variables

In [30]:
```
import pandas as pd
```

In [31]:
```
x = 7
y = 20 #variables can be assigned as int or strings
print(x*y)
print(x-y)
z='Hello World'
print(z)
print(z,x,y)
```

```
140
-13
Hello World
Hello World 7 20
```

assigning multiple variables with the same value

In [36]:
```
x = y =10
print(x,'/',y)
```

```
10 / 10
```

2. built in comparison and mathematical functions

In [38]:
```
17-5
```

Out[38]:
```
12
```

In [39]:
```
5/4
```

Out[39]:
```
1.25
```

In [41]:
```
5//4 #Floor division converting the result so obtained to the nearest integer.
```

Out[41]:
```
1
```

Relational Operators

Symbol	Task Performed
==	True, if it is equal
!=	True, if not equal to
<	less than
>	greater than
<=	less than or equal to
>=	greater than or equal to

```
In [47]: x=10
         y=10.2
         x==y

Out[47]: False
```

```
In [52]: x=10
         y=10.2
         x==y

Out[52]: False
```

```
In [57]: x!=y

Out[57]: True
```

```
In [54]: x>y

Out[54]: False
```

```
In [55]: x<y

Out[55]: True
```

```
In [58]: x>=y

Out[58]: False
```

```
In [59]: x<=y

Out[59]: True
```

int(x,8) one is value and the other its base **int(0xaa,16)** 0xaa is hexadecimal number (hex) to calculate this, do the following:

A x 1 = 10 multiply the last digit by 1

A x 16 = 160 multiply the second last digit by 16

multiply the third to the last digit by 16 x 16, Multiply the fourth to the last digit by 16 x 16 x 16, Multiply the fifth to the last digit by 16 x 16 x 16 x 16 and so on until all the digits are used. then add up all products from step1 to get answer to 0xaa **10 + 160 = 170**

In [68]:
```
print(int('0xaa',16))
```

```
170
```

float() creates a float (decimal, real number)

In [74]:
```
print(float(5642))
print(float(5))
```

```
5642.0
5.0
```

In [115]:
```
print(round(97.75)) #round( ) function rounds the input value to a specified number of places or to the nearest integer.
print(round(4.4557,2))
```

```
98
4.46
```

pow(x,y,z) can be used to find the power x^y also the mod of the resulting value with the third specified number can be found i.e. : (x^y % z).

In [131]:
```
print (pow(3,4))
print (pow(3,3,5)) # power of 3^3 and modulus 5
print ((3*3*3)%5) # this example is the same as the above one to demonstrate
print (10%2) # remainder is zero from division
```

```
81
2
2
0
```

```
- %s -> string
- %d -> Integer
- %f -> Float
- %o -> Octal
- %x -> Hexadecimal
- %e -> exponential
```

This can be used for conversions inside the print statement itself.

In [138]:

```python
print ("Integer = %d" %25)
print ("Float of the number = %f" %25)
print ("Octal equivalent of the number = %o" %25)
print ("Hexadecimal equivalent of the number = %x" %25)
print ("Exponential equivalent of the number = %e" %25)
```

```
Integer = 25
Float of the number = 25.000000
Octal equivalent of the number = 31
Hexadecimal equivalent of the number = 19
Exponential equivalent of the number = 2.500000e+01
```

In [145]:

```python
print ("Hello " *10)
```

```
Hello Hello Hello Hello Hello Hello Hello Hello Hello Hello
```

3. Python Functions

a function must be called with the correct number of arguments. Meaning that if your function expects 2 arguments, you have to call the function with 2 arguments, not more, and not less.

In [156]:

```python
def function():
  print("Hello from a function")
function()
```

```
Hello from a function
```

In [161]:

```
x = range(20)
for n in x:
    print(n)
```

```
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
```

Create a sequence of numbers from 0 to 19, and print each item in the sequence

range

range(start, stop, step)

In [181]:

```
x = range(1, 10)
for n in x:
  print(n) #create a sequence of numbers from 1 to 9
```

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

In [175]:

```
x = range(2,20,2)
for n in x:
    print(n) #create a sequence of numbers from 2 to 18 by increment of 2
```

```
2
4
6
8
10
12
14
16
18
```

help in python

In [180]:

```
help('print')
```

```
Help on built-in function print in module builtins:

print(...)
    print(value, ..., sep=' ', end='\n', file=sys.stdout, flush=False)

    Prints the values to a stream, or to sys.stdout by default.
    Optional keyword arguments:
    file:  a file-like object (stream); defaults to the current sys.stdout.
    sep:   string inserted between values, default a space.
    end:   string appended after the last value, default a newline.
    flush: whether to forcibly flush the stream.
```

In [212]:

```
num = [0,1,2,3,4,5]
len(num)
```

Out[212]:

```
6
```

In [216]:

```
print(num[0:4])
print(num[2:])
max(num)
```

```
[0, 1, 2, 3]
[2, 3, 4, 5]
```

Out[216]:

```
5
```

4. Loop over a list and print to the console

List is equivalent to arrays in other languages

In [182]:

```
# to iterate over a list
```

In [186]:

```
list = [1, 3, 5, 7, 10]

#using for loop
for i in list:
    print(i)
```

```
1
3
5
7
10
```

In [218]:

```
for i in range(20):
    print(i)
```

```
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
```

5. Import Pandas

In [219]:

```
import pandas as pd
import datetime
import os
```

6. Create a data frame with a few columns and rows

In [224]:

```
# intialise data of lists.
data = {'Name':['Peter', 'John', 'Dan', 'Mike','Brad'],
        'Age':[31, 35, 45, 25, 52],
        'Position':['HR','IT','Engineer','Accountant','Consultant']}

# Create DataFrame
df = pd.DataFrame(data)

# Print the output.
df
```

Out[224]:

	Name	Age	Position
0	Peter	31	HR
1	John	35	IT
2	Dan	45	Engineer
3	Mike	25	Accountant
4	Brad	52	Consultant

7. Index a pandas data frame and print different cells, rows, columns

In [241]:

```
frame = pd.DataFrame(data, columns=['Name', 'Age', 'Position'],
                      index=['one', 'two', 'three', 'four', 'five'])

frame
```

Out[241]:

	Name	Age	Position
one	Peter	31	HR
two	John	35	IT
three	Dan	45	Engineer
four	Mike	25	Accountant
five	Brad	52	Consultant

In [242]:

```
frame.columns
```

Out[242]:

```
Index(['Name', 'Age', 'Position'], dtype='object')
```

In [246]:
```
frame['Age']
```

Out[246]:
```
one      31
two      35
three    45
four     25
five     52
Name: Age, dtype: int64
```

In [312]:
```
frame['Position']
```

Out[312]:
```
one              HR
two              IT
three      Engineer
four     Accountant
five     Consultant
Name: Position, dtype: object
```

In [313]:
```
frame2 = pd.DataFrame(data, columns=['Name', 'Age', 'Position', 'Salary'],
                      index=['one', 'two', 'three', 'four', 'five'])

frame2
```

Out[313]:

	Name	Age	Position	Salary
one	Peter	31	HR	NaN
two	John	35	IT	NaN
three	Dan	45	Engineer	NaN
four	Mike	25	Accountant	NaN
five	Brad	52	Consultant	NaN

In [314]:
```
frame2['Salary']=('55000', '65000','75000','80000','85000')
frame2
```

Out[314]:

	Name	Age	Position	Salary
one	Peter	31	HR	55000
two	John	35	IT	65000
three	Dan	45	Engineer	75000
four	Mike	25	Accountant	80000
five	Brad	52	Consultant	85000

In [315]:
```
frame2.columns
```

Out[315]:
```
Index(['Name', 'Age', 'Position', 'Salary'], dtype='object')
```

In [317]:

```
frame2.Salary
```

Out[317]:

```
one      55000
two      65000
three    75000
four     80000
five     85000
Name: Salary, dtype: object
```

In [318]:

```
frame2.loc['four']
```

Out[318]:

```
Name              Mike
Age                 25
Position    Accountant
Salary           80000
Name: four, dtype: object
```

End of week2 notebook

In [ ]:

```

```