Python Crash Course

Taking Input

This is how you can take inputs from the user

In [4]:
```
name = input("Enter your name: ")
print('hello ' + name)
```

```
Enter your name: Taha
hello Taha
```

In [5]:
```
Age = int(input("Enter your age: "))
print("Your actual age is : " , Age + 2)
```

```
Enter your age: 23
Your actual age is :  25
```

Evaluating

`eval:` lets you evaluate expressions or variables

In [6]:
```
math = "(25+5) * 10"
eval(math)
```

Out[6]:
```
300
```

You can also Evaluate Variables like below

In [7]:
```
e_value = 45
print('e_value')
print(eval('e_value'))
```

```
e_value
45
```

Now lets make a Calculator in 2 lines of Python

In [8]:
```
exp = input("Enter any Math Expression: ")
print('Answer: ' + str(eval(exp))  )
```

```
Enter any Math Expression:

Traceback (most recent call last):

  File "C:\Users\hp-pc\Anaconda3\lib\site-packages\IPython\core\interactiveshel
l.py", line 3325, in run_code
    exec(code_obj, self.user_global_ns, self.user_ns)

  File "<ipython-input-8-362853a5869d>", line 2, in <module>
    print('Answer: ' + str(eval(exp))  )

  File "<string>", line unknown

    ^
SyntaxError: unexpected EOF while parsing
```

Assignment 2: Dynamic Story Maker

Create a Dynamic e 2-3 lines story in which you first ask the user his name and a few other things and make a story out of it

In [9]:
```
##ADD CODE BELOW
# ~ 8 lines
```

Lists

In Python programming, a list is created by placing all the items (elements) inside a square bracket [ ], separated by commas. It can have any number of items and they may be of different types (integer, float, string etc.).

A list can be of Numbers

In [10]:
```
number_list = [1,2,3]
print(number_list)
```

```
[1, 2, 3]
```

It can be of Letters

In [11]:
```
letter_list = ['s','d','f','d']
print(letter_list)
```

```
['s', 'd', 'f', 'd']
```

It Can be of Words

In [12]:
```
word_list = ['hello', 'Who','Mister','what']
print(word_list)
```

```
['hello', 'Who', 'Mister', 'what']
```

A list can be a list of Lists

In [13]:
```
list_of_lists =[[1,2,3], [12,2,213,212,3],['sdsa','df','fd']]
print(list_of_lists)
```

```
[[1, 2, 3], [12, 2, 213, 212, 3], ['sdsa', 'df', 'fd']]
```

It can be a combination of All of them

In [14]:
```
mix_list = ['h',1,'world',[2,'ad']]
print(mix_list)
```

```
['h', 1, 'world', [2, 'ad']]
```

Some List methods

In [15]:
```
my_list = ['a','b','c','d','e']
```

Pop out the last Element

In [16]:
```
my_list.pop()
my_list
```

Out[16]:
```
['a', 'b', 'c', 'd']
```

Add an Element to the List

In [17]:
```
my_list.append('z')
```

In [18]:
```
my_list
```

Out[18]:
```
['a', 'b', 'c', 'd', 'z']
```

Indexing

You can index into the list and pull out a specific Element

In [19]:
```
my_list[0]
```

Out[19]:
```
'a'
```

In [20]:
```
my_list[1]
```

Out[20]:
```
'b'
```

You can also Reverse Index in a list
For e.g `-1` will give you the the last element of the list

In [21]:
```
my_list[-1]
```

Out[21]:
```
'z'
```

In [22]:
```
my_list[-2] # 2nd last
```

Out[22]:
```
'd'
```

Slicing

You can Slice the list, Upper Limit is not Included
So `0:3` Means Give me Elements ranging from 0 to 2

In [23]:
```
my_list[:1]  # the upper limit is not included
```

Out[23]:
```
['a']
```

In [24]:
```
my_list[1:3]
```

Out[24]:
```
['b', 'c']
```

Below Command is Equal to the Above command
So `:3` Means give me Elements from Start to 2

In [25]:
```
my_list[:3]
```

Out[25]:
```
['a', 'b', 'c']
```

Similarly you can slice from some index to End of list

In [26]:
```
my_list[3:]
```

Out[26]:
```
['d', 'z']
```

You Can Modify any existing value of the list

In [27]:
```
my_list[0] = 'NEW'
```

In [28]:
```
my_list
```

Out[28]:
```
['NEW', 'b', 'c', 'd', 'z']
```

Lets take Look at a Nested list

Can you print out the string target through Indexing

In [29]:
```
nest = [1,2,3,[4,5,['target']]]  # the third element of thrid element is also a
```

In [30]:
```
nest[3]
```

Out[30]:
```
[4, 5, ['target']]
```

In [31]:
```
nest[3][2]
```

Out[31]:
```
['target']
```

In [32]:
```
nest[3][2][0]
```

Out[32]:
```
'target'
```

You can also check if a particular element exists in a list

In [33]:
```
'x' in ['x','y','z']
```

Out[33]:
```
True
```

Dictionaries

A dictionary is a collection which is unordered, changeable and indexed. In Python dictionaries are written with curly brackets, and they have keys and values.

In [34]:
```
d = {'key1': 'Value1' ,  'key2':'Value'}
```

Consider Below Example of a Dictionary in which Key is the name of a person and the value is his/her phone number

In [35]:
```
d = {'Taha': 923132283550,  'Sara' : 923332234450 , 'Mustafa': 923212062374}
```

Now if Needed Mustafa's Phone Number I would do this:

In [36]:
```
d['Mustafa']
```

Out[36]:
```
923212062374
```

You can use anything as a key, integers, strings etc.

Note: You can't Append to a dictionary

In [37]:
```
d.append['Mouse'] = "Cheese"          #we can't use append keyword in dictionary
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
<ipython-input-37-69de06ce8208> in <module>
----> 1 d.append['Mouse'] = "Cheese"          #we can't use append keyword in di
ctionary

AttributeError: 'dict' object has no attribute 'append'
```

Now this is how you add a new Key Value Pair

In [38]:
```
d['Mouse'] = 'Cheese'    # Set a new entry in a dictionary
d
```

Out[38]:
```
{'Taha': 923132283550,
 'Sara': 923332234450,
 'Mustafa': 923212062374,
 'Mouse': 'Cheese'}
```

This is How you Delete a Key Value Pair

In [39]:
```
del d['Sara']
```

In [40]:
```
d
```

Out[40]:
```
{'Taha': 923132283550, 'Mustafa': 923212062374, 'Mouse': 'Cheese'}
```

You Can Update Existing Values

In [41]:
```
d['Taha'] = '92323358330'      #update key1 value with chicken
d
```

Out[41]:
```
{'Taha': '92323358330', 'Mustafa': 923212062374, 'Mouse': 'Cheese'}
```

Here are all your dict keys

In [42]:
```
d.keys()
```

Out[42]:
```
dict_keys(['Taha', 'Mustafa', 'Mouse'])
```

Here are all you dict values

In [43]:
```
d.values()
```

Out[43]:
```
dict_values(['92323358330', 923212062374, 'Cheese'])
```

Here are all your dict items (key/value pairs)

In [44]:
```
d.items()
```

Out[44]:
```
dict_items([('Taha', '92323358330'), ('Mustafa', 923212062374), ('Mouse', 'Chee
se')])
```