

# Python

if `__name__ == "__main__"` to ensure code only runs when the script is executed directly, and not when it is imported as module.

## **Strings**

```
message = 'Hello world'
print(message.lower())    # converts all characters to lower case
print(message.upper())    # converts all characters to upper case
print(message.count('l')) # count number of passed argument

new_message = message.replace('World', 'Universe') #str.replace returns a value
```

## **List, Tuples and Sets**

```
courses = ['History', 'Math', 'Physics', 'CompSci']
print(len(courses))      #prints length of the list
courses.append('Art')    #add 'Arts' at the end of the list
courses.insert(0, 'Chemistry') #adds the element at the specified index
courses.extend(['Bio', 'PHE']) #adds the elements of list in argument to the object list

courses.remove('PHE')    #Remove first occurrence of the argument in the list,
                          #throws error if argument not present
popped = courses.pop(0)  #removes the element at specified location, throws
                          #error if index out of range

courses.reverse()        #reverse the list
courses.sort(reverse = True) #sort the list, if reverse = True sort it in a
                             #descending order
sorted(courses)          #sorted function return sorted version of the list,
                          #without changing the list

nums = [101,45,56,87]
min(nums)                #returns min of the list
max(nums)                #returns max of the list
sum(nums)                #returns sum of the list

Min, max return even on strings, whereas sum return error when list has strings.
courses.index('Math')    #returns first occurrence of the value, return error
                          #if not present
print('Hindi' in courses) #check if value is present or not
#we want to join elements of courses by some identifier (here comma separated values)
newcourses = ', '.join(courses)
#Similarly, we want to split the string based on some identifier
newcourses.split(', ')
```

Tuples are immutable while list are mutable. Set does not have duplicates and it does not care about an order.

```
mcourses = {'Physics', 'Chemistry', 'Math'}
bcourses = {'Physics', 'Chemistry', 'Bio'}
mcourses.intersection(bcourses) #Intersection between two sets
mcourses.union(bcourses)        #Union between two sets
mcourses.difference(bcourses)   #Difference between two sets
```

### **Initialization**

```
empty_list = []
empty_list = list()
empty_tuple = ()
empty_tuple = tuple()
empty_set = set()
```

If we do `empty_set = {}`, it creates a dictionary, not a set

## **Dictionaries**

```
student = {'name': 'John', 'age': 25, 'courses': ['Math', 'CompSci']}
```

Key-Value pair, if we access the key which does not exist, returns an error, if we don't want error, we can use `get()` method

```
print(student.get('name', 'Not_Found'))    #searches for the key, return Not_Found if
                                           not present
student['phone'] = '9999999999'           #adding a new key-value pair
student.update({'name': 'John', 'age': 25}) #updating multiples values at a single time
#if we want to remove any key value-pair, we can use any of the following method
del student['Age']
age = student.pop('age')

students.keys()    #return list of all keys
student.values()   #return list of all values
student.items()    #return list of all key-value pair in tuples
```