

## List:

Sr.No.	Methods & Description
1	<b>list.append(obj)</b> Appends object obj to list
2	<b>list.count(obj)</b> Returns count of how many times obj occurs in list
3	<b>list.extend(seq)</b> Appends the contents of seq to list
4	<b>list.index(obj)</b> Returns the lowest index in list that obj appears
5	<b>list.insert(index, obj)</b> Inserts object obj into list at offset index
6	<b>list.pop(obj = list[-1])</b> Removes and returns last object or obj from list
7	<b>list.remove(obj)</b> Removes object obj from list
8	<b>list.reverse()</b> Reverses objects of list in place
9	<b>list.sort([func])</b> Sorts objects of list, use compare func if given

## Dictionary:

Sr.No.	Method & Description
1	<b>dict.clear()</b> Removes all elements of dictionary <i>dict</i>
2	<b>dict.copy()</b> Returns a shallow copy of dictionary <i>dict</i>
3	<b>dict.fromkeys()</b> Create a new dictionary with keys from seq and values set to <i>value</i> .
4	<b>dict.get(key, default=None)</b> For key key, returns value or default if key not in dictionary
5	<b>dict.has_key(key)</b> Removed, use the <i>in</i> operation instead.
6	<b>dict.items()</b> Returns a list of <i>dict</i> 's (key, value) tuple pairs
7	<b>dict.keys()</b> Returns list of dictionary dict's keys
8	<b>dict.setdefault(key, default = None)</b> Similar to get(), but will set dict[key] = default if key is not already in dict
9	<b>dict.update(dict2)</b> Adds dictionary <i>dict2</i> 's key-values pairs to <i>dict</i>
10	<b>dict.values()</b> Returns list of dictionary <i>dict</i> 's values

## Strings:

# Unicode String

In Python 3, all strings are represented in Unicode. In Python 2 are stored internally as 8-bit ASCII, hence it is required to attach 'u' to make it Unicode. It is no longer necessary now.

## Built-in String Methods

Python includes the following built-in methods to manipulate strings –

Sr.No.	Methods & Description
1	<b>capitalize()</b>  Capitalizes first letter of string
2	<b>center(width, fillchar)</b> Returns a string padded with <i>fillchar</i> with the original string centered to a total of <i>width</i> columns.
3	<b>count(str, beg = 0, end = len(string))</b> Counts how many times str occurs in string or in a substring of string if starting index beg and ending index end are given.
4	<b>decode(encoding = 'UTF-8', errors = 'strict')</b> Decodes the string using the codec registered for encoding. encoding defaults to the default string encoding.
5	<b>encode(encoding = 'UTF-8', errors = 'strict')</b> Returns encoded string version of string; on error, default is to raise a ValueError unless errors is given with 'ignore' or 'replace'.
6	<b>endswith(suffix, beg = 0, end = len(string))</b>

	Determines if string or a substring of string (if starting index beg and ending index end are given) ends with suffix; returns true if so and false otherwise.
7	<b>expandtabs(tabsize = 8)</b> Expands tabs in string to multiple spaces; defaults to 8 spaces per tab if tabsize not provided.
8	<b>find(str, beg = 0 end = len(string))</b> Determine if str occurs in string or in a substring of string if starting index beg and ending index end are given returns index if found and -1 otherwise.
9	<b>index(str, beg = 0, end = len(string))</b> Same as find(), but raises an exception if str not found.
10	<b>isalnum()</b> Returns true if string has at least 1 character and all characters are alphanumeric and false otherwise.
11	<b>isalpha()</b> Returns true if string has at least 1 character and all characters are alphabetic and false otherwise.
12	<b>isdigit()</b> Returns true if string contains only digits and false otherwise.
13	<b>islower()</b> Returns true if string has at least 1 cased character and all cased characters are in lowercase and false otherwise.
14	<b>isnumeric()</b> Returns true if a unicode string contains only numeric characters and false otherwise.
15	<b>isspace()</b>

	Returns true if string contains only whitespace characters and false otherwise.
16	<b>istitle()</b> Returns true if string is properly "titlecased" and false otherwise.
17	<b>isupper()</b> Returns true if string has at least one cased character and all cased characters are in uppercase and false otherwise.
18	<b>join(seq)</b> Merges (concatenates) the string representations of elements in sequence seq into a string, with separator string.
19	<b>len(string)</b> Returns the length of the string
20	<b>ljust(width[, fillchar])</b> Returns a space-padded string with the original string left-justified to a total of width columns.
21	<b>lower()</b> Converts all uppercase letters in string to lowercase.
22	<b>lstrip()</b> Removes all leading whitespace in string.
23	<b>maketrans()</b> Returns a translation table to be used in translate function.
24	<b>max(str)</b> Returns the max alphabetical character from the string str.
25	<b>min(str)</b> Returns the min alphabetical character from the string str.
26	<b>replace(old, new [, max])</b>

	Replaces all occurrences of old in string with new or at most max occurrences if max given.
27	<b>rfind(str, beg = 0, end = len(string))</b> Same as find(), but search backwards in string.
28	<b>rindex( str, beg = 0, end = len(string))</b> Same as index(), but search backwards in string.
29	<b>rjust(width,[, fillchar])</b> Returns a space-padded string with the original string right-justified to a total of width columns.
30	<b>rstrip()</b> Removes all trailing whitespace of string.
31	<b>split(str="", num=string.count(str))</b> Splits string according to delimiter str (space if not provided) and returns list of substrings; split into at most num substrings if given.
32	<b>splitlines( num=string.count('\n'))</b> Splits string at all (or num) NEWLINES and returns a list of each line with NEWLINES removed.
33	<b>startswith(str, beg=0, end=len(string))</b> Determines if string or a substring of string (if starting index beg and ending index end are given) starts with substring str; returns true if so and false otherwise.
34	<b>strip([chars])</b> Performs both lstrip() and rstrip() on string
35	<b>swapcase()</b> Inverts case for all letters in string.
36	<b>title()</b>

	Returns "titlecased" version of string, that is, all words begin with uppercase and the rest are lowercase.
37	<b>translate(table, deletechars="")</b> Translates string according to translation table str(256 chars), removing those in the del string.
38	<b>upper()</b> Converts lowercase letters in string to uppercase.
39	<b>zfill (width)</b> Returns original string leftpadded with zeros to a total of width characters; intended for numbers, zfill() retains any sign given (less one zero).
40	<b>isdecimal()</b> Returns true if a unicode string contains only decimal characters and false otherwise.

## Multiple Statements on a Single Line

The semicolon ( ; ) allows multiple statements on a single line given that no statement starts a new code block. Here is a sample snip using the semicolon –

```
import sys; x = 'foo'; sys.stdout.write(x + '\n')
```

## Multi-Line Statements

Statements in Python typically end with a new line. Python, however, allows the use of the line continuation character ( \ ) to denote that the line should continue. For example –

```
total = item_one + \
    item_two + \
    item_three
```

## Multiple Assignment

Python allows you to assign a single value to several variables simultaneously.

For example –

```
a = b = c = 1
```

### **Project:**

#### **1. Rock paper scissor**



## **2. Encryption decrypt using public private key**

### **Random:**

```
# Python code to demonstrate the working of  
# choice() and randrange()
```

```
# importing "random" for random operations  
import random
```

```
# using choice() to generate a random number from a  
# given list of numbers.  
print ("A random number from list is : ",end="")  
print (random.choice([1, 4, 8, 10, 3]))
```

```
# using randrange() to generate in range from 20  
# to 50. The last parameter 3 is step size to skip  
# three numbers when selecting.  
print ("A random number from range is : ",end="")  
print (random.randrange(20, 50, 3))
```

```
# importing "random" for random operations  
import random
```

```
# using random() to generate a random number  
# between 0 and 1  
print ("A random number between 0 and 1 is : ", end="")  
print (random.random())
```

```
# using seed() to seed a random number  
random.seed(5)
```

```
# printing mapped random number  
print ("The mapped random number with 5 is : ", end="")  
print (random.random())
```

```
# using shuffle() to shuffle the list
```

```
random.shuffle(li)
```

```
from random import randint  
print(randint(0, 9))
```