List

The list is a most versatile data type available in Python which can be written as a list of comma-separated values (items) between square brackets. Important thing about a list is that items in a list need not be of the same type.

Creating a list is as simple as putting different comma-separated values between square brackets.

**colors = ['red', 'blue', 'green']**

**print (colors[0]) ## red**

**print (colors[2]) ## green**

**print (len(colors)) ## 3**

**print ("Creating List:")**

**colors = ['red', 'blue', 'green']**

**print (colors[0]) ## red**

**print (colors[1]) ## blue**

**print (colors[2]) ## green**

**print (len(colors)) ## 3**

**print ("Append to the List")**

**colors.append("orange")**

**print (colors[3]) ##orange**

**print ("Insert to the List")**

**colors.insert(3, "yellow")**

**print (colors[3]) ##yellow**

**print (colors[4]) ##orange**

**print ("Remove from the List")**

**print (colors[1]) ## blue**

**colors.remove("blue") ## deletes blue and shifts elements to the left**

**print (colors[1]) ## green**

**print ("Sorting Ascending order using sorted")**

**nums = [98,22,45,30]**

**numsAsc = sorted(nums)**

**print (numsAsc[0]) ## 22**

**print (numsAsc[1]) ## 30**

**print (numsAsc[2]) ## 45**

**print ("Sorting Descending order using sorted")**

**numsDesc = sorted(nums,reverse=True)**

**print (numsDesc[0]) ## 98**

**print (numsDesc[1]) ## 45**

**print (numsDesc[2]) ## 30**

Using for loops with lists

**nos = [1, 4, 9, 16, 25]**

**sum = 0**

**for num in nos:**

**sum += num**

**print (sum) ## 55**

#### Using if loop with lists

**colors = ['red', 'blue', 'green']**

**if 'blue' in colors:**

**print ('cool') # blue found in the list**

#### Using while loops with lists

**i = 0**

**a = [1,2,3,4,5,6,7,8,9]**

**while i < len(a):**

**print (a[i]) #prints 1 4 7**

**i = i + 3**

# List Slices

list = ['a', 'b', 'c', 'd']  
  print list[1:-1]   ## ['b', 'c']  
  list[0:2] = 'z'    ## replace ['a', 'b'] with ['z']  
  print list         ## ['z', 'c', 'd']

### List Methods

* list.append(elem) -- adds a single element to the end of the list. Common error: does not return the new list, just modifies the original.
* list.insert(index, elem) -- inserts the element at the given index, shifting elements to the right.
* list.extend(list2) adds the elements in list2 to the end of the list. Using + or += on a list is similar to using extend().
* list.index(elem) -- searches for the given element from the start of the list and returns its index. Throws a ValueError if the element does not appear (use "in" to check without a ValueError).
* list.remove(elem) -- searches for the first instance of the given element and removes it (throws ValueError if not present)
* list.sort() -- sorts the list in place (does not return it). (The sorted() function shown below is preferred.)
* list.reverse() -- reverses the list in place (does not return it)
* list.pop(index) -- removes and returns the element at the given index. Returns the rightmost element if index is omitted (roughly the opposite of append()).

List Comprehensions

 S = [x\*\*2 for x in range(10)]

Print s

[0, 1, 4, 9, 16, 25, 36, 49, 64, 81]  
  
V = [2\*\*i for i in range(13)]

Print v

[1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048, 4096]

M = [x for x in S if x % 2 == 0]  
print M

|  |
| --- |
| [0, 4, 16, 36, 64] |

noprimes = [j for i in range(2, 8) for j in range(i\*2, 50, i)]  
primes = [x for x in range(2, 50) if x not in noprimes]  
print primes

|  |
| --- |
| [2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47] |

words = 'The quick brown fox jumps over the lazy dog'.split()  
print words

|  |
| --- |
| ['The', 'quick', 'brown', 'fox', 'jumps', 'over', 'the', 'lazy', 'dog'] |

stuff = [[w.upper(), w.lower(), len(w)] for w in words]  
for i in stuff:  
     print i

output

|  |
| --- |
| ['THE', 'the', 3] ['QUICK', 'quick', 5] ['BROWN', 'brown', 5] ['FOX', 'fox', 3] ['JUMPS', 'jumps', 5] ['OVER', 'over', 4] ['THE', 'the', 3] ['LAZY', 'lazy', 4] ['DOG', 'dog', 3] |