Lists

* **List is set of values**

**e.g.: spam = ['cat','dog,'rat','elephant']**

* **Access**

1. **Using index : spam[1] 🡪 ‘dog’**
2. **Using negative index : spam[-2] 🡪 ‘rat’**

***#second last element***

1. **Using slicing : spam[1:3] 🡪 [‘dog’, ‘rat’]**
2. **Using double index :**

**(Lists inside lists)**

**e.g.: spam = [['cat’, ‘bat'], ['10','20','30']]**

**spam[1][2] --> 30**

* **List APIs**

1. **index(member)**

**Returns the index of the argument**

**e.g.:**

**If spam = ['cat', 'dog', 'rat'] spam.index ('dog')**

**returns 1**

1. **count(member)**

**returns the number of occurrences of the member**

**e.g.:**

**If spam = ['cat', 'dog', 'rat', 'cat']**

**Then spam.count ('cat') returns**

**2**

1. **append(member)**

**Add member at the end of the list**

**e.g.:**

**If spam = ['cat', 'dog', 'rat']**

**spam.append ('mongoose')**

**spam = ['cat', 'dog', 'rat','mongoose']**

1. **insert(index, member)**

**Add member at index position**

**e.g.:**

**If spam = ['cat','dog','rat']**

**Then spam.insert (1,'chicken')**

**['cat', 'chicken', 'dog', 'rat']**

1. **extend(seq)**

**Appends seq at the end of the list**

**e.g.: if spam = ['cat','dog','rat'] and seq = 'hello','hi'] then spam.extend (seq)**

**['cat’,’dog’,’rat’, ‘hello’,’hi’]**

1. **remove(member)**

**Remove the member from the list**

**e.g.:**

**If spam = ['cat','dog','rat']**

**Then spam.remove ('dog')**

**['cat', 'rat']**

1. **pop (index)**

**Removes and returns the member at index**

**e.g.:**

**If spam = ['cat','dog','rat']**

**spam.pop () --> abc**

**spam.pop (2) --> zara**

1. **clear()**

**Empty the list**

**e.g.:**

**If spam = ['1', '2','3'], spam.clear ()**

**spam = []**

1. **reverse()**

**Reverses the list**

**e.g.:**

**If spam = ['cat','dog','rat'] then spam.reverse ()**

**[‘cat’,’dog’,’rat’]**

1. **sort()**

**Sort the list**

**e.g.:**

**If spam = ['3', '1', '2'] then spam.sort()**

**[‘1’,’2’,’3’]**

**Misc.**

* **list() function to convert objects to list**

**e.g.: list('Hello') --> ['h', 'e', 'l', 'l', 'o']**

* **Operators on lists**

**e.g.: [1,2,3] + [4,5,6] --> [1,2,3,4,5,6]**

**[1,2,3] \* 3 --> [1,2,3,1,2,3,1,2,3]**

* **Augmented assignment Used to update the variable**

**e.g.: spam += spam --> spam = spam + 1**

**spam \*= 2 --> spam = spam \* 2**

**spam /= 2 --> spam = spam + 2**

**spam -=1 --> spam = spam -1**

**spam %=2 --> spam = spam % 2**

* **index(member) throws an exception if member is not in the list**
* **sort() will only work on list containing only strings or numbers. if the list contains both strings as well as numbers, it raises an exception**
* **sort (reverse=True) for sorting in reverse. sort(key=str.lower) to sort in alphabetical order.**

**\*\* do LISTDPP and STRINGDPP**