



Data types: List and Dictionary



The list is a most versatile data type available in Python which can be written as a list of comma-separated values between square brackets.

Properties of Lists:

- ❑ Lists can be accessed by offset.

```
>>>L[0]
```

- ❑ Lists are mutable.

```
>>>L[0]=5
```

The list is a most versatile data-type available in Python which can be written as a list of comma-separated values between square brackets.

Properties of Lists :

- ❑ Ordered collections of arbitrary objects i.e. lists are sequences.

```
>>>L = [1,2,3]
```

- ❑ Unlike strings, lists can grow and shrink in place, and can contain any sort of object.

```
>>>L=[1, 'python',  
1.24]
```


List Operations

Operations	Interpretation
L.append(4)	Growing list
L.extend([2,3,4])	Growing list
L.insert(i,x)	Insert x at position i
L.index(x)	Searching
L.count(x)	Count no. of occurrences of x
L.sort()	Sorting
L.reverse()	Reversing
L.pop(i)	Shrinking
L.remove(x)	Shrinking list



Hands-On



Dictionary

In a dictionary each key is separated from its value by a colon (:), the items are separated by commas, and the whole thing is enclosed in curly braces

Properties of Dictionary:

- ❑ Dictionaries are mutable

```
>>>D['name']='Brida'
```

- ❑ Accessed by keys, not offset position.

```
>>>D['name']
```

Dictionary

In a dictionary each key is separated from its value by a colon (:), the items are separated by commas, and the whole thing is enclosed in curly braces

Properties of Dictionary:

- ☐ Unordered collections of arbitrary objects.
- ☐ Variable length, heterogeneous and arbitrarily nestable.

```
>>>D={'name':'abc' , 'l_name':'xyz' , 'age':22}
```

```
>>>record={'name':{'first':'Bob','lname':'Willaim'},  
          'jobs':['manager' , 'engineer'],  
          'age':45}
```

Dictionary Operations

Operations	Interpretations
<code>d.keys()</code>	All keys
<code>d.values()</code>	All values
<code>d.items()</code>	All key+value tuples
<code>d.clear()</code>	Remove all items
<code>d.update(d2)</code>	Merge by keys
<code>d.get(key,default)</code>	Fetch by key, if absent default(or None)
<code>d.pop(key,default)</code>	Remove by key, if absent default(or error)

Hands-On

