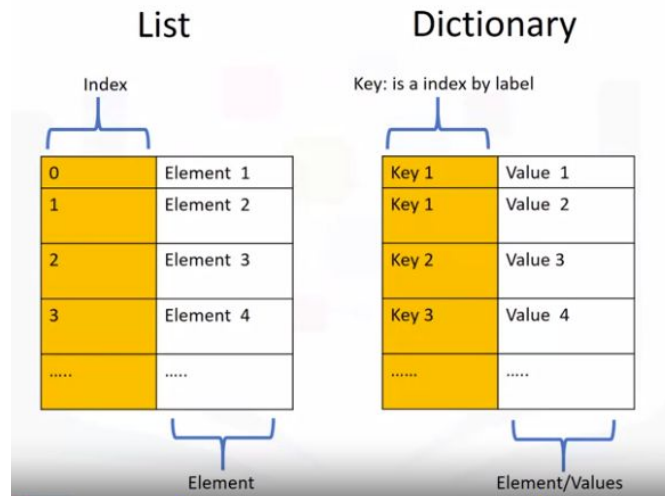


Dictionaries are a type of collection in Python. If you recall, a list has integer indexes. These are like addresses. A list also has elements.

A dictionary has keys and values. The key is analogous to the index, they are like addresses but they don't have to be integers. They are usually characters.



The values are similar to the element in a list and contain information.

To create a dictionary, we use curly brackets.

The keys are the first elements. They must be immutable and unique.

Each key is followed by a value separated by a colon.

The values can be immutable, mutable, and duplicates.

Each key and value pair is separated by a comma.

Dictionaries

- Dictionaries are denoted with curly Brackets `{}`
- The keys have to be immutable and unique
- The values can be can immutable, mutable and duplicates
- Each key and value pair is separated by a comma

```
{ "key1":1, "key2 " : "2", "key3" : [3,3,3], "key4":(4,4,4), ('key5'):5 }
```

Consider the following example of a dictionary.

```
Dict={"Thriller": 1982, "Back in Black": 1980, "The Dark Side of the Moon": 1973, "The Bodyguard":1992}
```

The album title is the key and the value is the released data.

We can use yellow to highlight the keys and leave the values in white.

It is helpful to use the table to visualize a dictionary where the first column represents the keys, and the second column represents the values.

"Thriller"	"1982"
"Back in Black"	"1980"
"The Dark Side of the Moon"	"1973"
"The Bodyguard"	"1992"

We can add a few more examples to the dictionary.

We can also assign the dictionary to a variable. The key is used to look at the value.

We use square brackets. The argument is the key. This outputs the value. Using the key of "Back in Black", this returns the value of 1980.

```
Dict["The Dark Side of the Moon"]  
= 1980
```

The key, "The Dark Side Of The Moon", gives us the value of 1973.

Using the key, "The bodyguard", gives us the value 1992, and so on.

What is the result of the following operation: Dict["D"]

```
Dict={"A":1,"B":2,"C":[3,3,3],"D":(4,4,4),'E':5,'F':6}  
=(444)
```

We can add a new entry to the dictionary as follows. This will add the value 2007 with a new key called, "Graduation."

"The Dark Side of the Moon"	"1973"
"The Bodyguard"	"1992"
"Bat Out of Hell"	"1977"
"Their Greatest..."	"1976"
Saturday Night Fever	"1977"
"Rumors"	"1977"
'Graduation'	"2007"

DICT['Graduation']='2007'

We can delete an entry as follows. This gets rid of the key, "Thriller" and it's value.

```
del(DICT['Thriller'])
```

We can verify if an element is in the dictionary using the "in" command as follows.

'The Bodyguard' in DICT

The command checks the keys. If they are in the dictionary, they return a true.

If we try the same command with a key that is not in the dictionary, we get a false. If we try another key that is not in the dictionary, we get a false. In order to see all the keys in the dictionary, we can use the method keys to get the keys.

"Thriller"	"1982"
"Back in Black"	"1980"
"The Dark Side of the Moon"	"1973"
"The Bodyguard"	"1992"
"Bat Out of Hell"	"1977"
"Their Greatest..."	"1976"
"Saturday Night Fever"	"1977"
"Rumors"	"1977"

```
DICT.keys()=[ "Thriller", "Back in Black", "The Dark Side of the Moon", "The Bodyguard",  
              "Bat Out of Hell", "Their Greatest...", "Saturday Night Fever", "Rumors" ]
```

The output is a list like object with all the keys. In the same way, we can obtain the values using the method values.

"Thriller"	"1982"
"Back in Black"	"1980"
"The Dark Side of the Moon"	"1973"
"The Bodyguard"	"1992"
"Bat Out of Hell"	"1977"
"Their Greatest..."	"1976"
Saturday Night Fever	"1977"
"Rumors"	"1977"

```
DICT.values()=[ "1982","1980","1973","1992", "1977","1976" "1977", "1977" ]
```

Check out the labs for more examples and info on dictionaries.