Dictonaries

Dr. Seema Gupta Bhol

Dictionary

- Dictionaries are used to store data values in key : value pairs.
- A dictionary is a collection which is ordered, changeable and do not allow duplicates.
- Dictionaries are written with curly brackets, and have keys and values:

```
thisdict = {
  "brand": "Ford",
  "model": "Mustang",
  "year": 1964
}

print(thisdict)
```

Duplicates Not Allowed

Dictionaries cannot have two items with the same key:

Example : Duplicate values will overwrite existing values:

```
thisdict = {
  "brand": "Ford",
  "model": "Mustang",
  "year": 1964,
  "year": 2020
}
print(thisdict)
```

Dictionary Length

- To determine how many items a dictionary has, use the len() function:

Example : Print the number of items in the dictionary:

```
print(len(thisdict))
```

- The values in dictionary items can be of any data type:

Example : String, int, boolean, and list data types

```
thisdict = {
  "brand": "Ford",
  "electric": False,
  "year": 1964,
  "colors": ["red", "white", "blue"]
}
print(type(thisdict))
```

The dict() Constructor

- It is also possible to use the dict() constructor to make a dictionary.

Example : Using the dict() method to make a dictionary:

```
thisdict = dict(name = "John", age = 36, country = "Norway")
print(thisdict)
```

in keyword

To determine if a specified key is present in a dictionary use the in keyword:

Example : Check if "model" is present in the dictionary:

```
thisdict = {
  "brand": "Ford",
  "model": "Mustang",
  "year": 1964
}
if "model" in thisdict:
  print("Yes, 'model' is one of the keys in the thisdict dictionary")
```

Dictionary Items

- Dictionary items are ordered, changeable, and do not allow duplicates.
- Dictionary items are presented in key : value pairs, and can be referred to by using the key name.
- Example: Print the "brand" value of the dictionary

```
thisdict = {
  "brand": "Ford",
  "model": "Mustang",
  "year": 1964
}
print(thisdict["brand"]) or
```

```
x = thisdict.get("model")
print(x)
```

Access the items

- Get() will be used to access the items of a dictionary

Example: Get the value of the "model" key

```
x = thisdict.get("model")
```

- The keys() method will return a list of all the keys in the dictionary.

```
x = thisdict.keys()
```

- The values() method will return a list of all the values in the dictionary.

```
x = thisdict.values()
```

- The items() method will return each item in a dictionary, as tuples in a list.

```
x = thisdict.items()
```

Change Dictionary Items

- To change the value of a specific item, refer by its key name:

Example: Change the "year" to 2018

```
thisdict = { "brand": "Ford", "model": "Mustang", "year": 1964}

thisdict["year"] = 2018
```

- The update() method will update the dictionary with the items from the given argument. The argument must be a dictionary, or an iterable object with key : value pairs.

```
thisdict.update({"year": 2020})
```

Add Dictionary Items

Adding an item to the dictionary is done by using a new index key and assigning a value to it:

```
thisdict = {  "brand": "Ford", "model": "Mustang",  "year": 1964}
thisdict["color"] = "red"
print(thisdict)
```

- The update() method will update the dictionary with the items from a given argument. If the item does not exist, the item will be added.

The argument must be a dictionary, or an iterable object with key:value pairs.

Example : Add a color item to the dictionary by using the update() method:

```
thisdict = { "brand": "Ford", "model": "Mustang", "year": 1964}
thisdict.update({"color": "red"})
```

Removing Items

- The pop() method removes the item with the specified key name

```
thisdict = { "brand": "Ford",  "model": "Mustang",  "year": 1964 }
thisdict.pop("model")
print(thisdict)
```

- The popitem() method removes the last inserted item

```
thisdict.popitem()
print(thisdict)
```

- The del keyword removes the item with the specified key name:

```
del thisdict["model"]
print(thisdict)
```

- The del keyword can also delete the dictionary completely:

```
del thisdict
print(thisdict) #this will cause an error because "thisdict" no longer exists.
```

- The clear() method empties the dictionary:

```
thisdict.clear()
print(thisdict)
```

Loop Through a Dictionary

- When looping through a dictionary, the return value are the keys of the dictionary.

Example : Print all key names in the dictionary, one by one

```
for x in thisdict:
  print(x)
```

- Example :Print all values in the dictionary, one by one

```
for x in thisdict:
  print(thisdict[x])
```

- values() method can be used to return values of a dictionary:

```
for x in thisdict.values():
  print(x)
```

- The keys() method can be used to return the keys of a dictionary:

```
for x in thisdict.keys():
  print(x)
```

- Loop through both keys and values, by using the items() method:

```
for x, y in thisdict.items():
  print(x, y)
```

Copy Dictionaries

- Make a copy of a dictionary with the copy() method:

```
thisdict = {
  "brand": "Ford",
  "model": "Mustang",
  "year": 1964
}
mydict = thisdict.copy()
print(mydict)
```

- Make a copy of a dictionary with the dict() function:

```
mydict = dict(thisdict)
print(mydict)
```

Nested Dictionaries

A dictionary can contain dictionaries, this is called nested dictionaries.

Example : Create a dictionary that contain three dictionaries:

```
myfamily = {
  "child1" : {
    "name" : "Emil",
    "year" : 2004
  },
  "child2" : {
    "name" : "Tobias",
    "year" : 2007
  },
  "child3" : {
    "name" : "Linus",
    "year" : 2011
  }
}
```

Create three dictionaries, then create one dictionary that will contain the other three dictionaries

```
child1 = {
  "name" : "Emil",
  "year" : 2004
}
child2 = {
  "name" : "Tobias",
  "year" : 2007
}
child3 = {
  "name" : "Linus",
  "year" : 2011
}
myfamily = {
  "child1" : child1,
  "child2" : child2,
  "child3" : child3
}
```

Access Items in Nested Dictionaries

To access items from a nested dictionary, you use the name of the dictionaries, starting with the outer dictionary

Example: Print the name of child 2:

```
print(myfamily["child2"]["name"])
```

Exercise

- Q1. Create a dictionary of 5 students with their marks and display it.
  a) Access and print the value of a given key from a dictionary.
  b) Add a new key-value pair to an existing dictionary.
  c) Update the value of an existing key.
  d) Delete a key from a dictionary using del.
  e) Check whether a key exists in a dictionary.
  f) Print all keys and values separately.
  g) Find the length of a dictionary.
  h) Print all key-value pairs using a for loop.
  i) Find the student with highest marks.
- Q2. Count frequency of each character in a string using dictionary.
- Q3. Create a dictionary of numbers and their squares (1 to 10).
- Q4. Create nested dictionary for student details (name, marks, age).
- Q5 Convert two lists into a dictionary. Example:

```
keys = ['a','b','c']
values = [1,2,3]
```