PYTHON DICTIONARY

By
Hiba Al-Senawi

PYTHON DICTIONARIES

- A dictionary is a collection which is unordered, changeable and indexed.
- In Python dictionaries are written with curly brackets, and they have keys and values.

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

ACCESSING ITEMS

- You can access the items of a dictionary by referring to its key name, inside square brackets:

```
x = thisdict["model"]
```

There is also a method called get() that will give you the same result:

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

CHANGE VALUES

- You can change the value of a specific item by referring to its key name:

- thisdict = {
- "brand": "Ford",
- "model": "Mustang",
- "year": 1964
- }
- thisdict["year"] = 2018

DICTIONARY LENGTH

- To determine how many items (key-value pairs) a dictionary has, use the len() function.

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

ADDING 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)
```

REMOVING ITEMS

- There are several methods to remove items from a dictionary
- The pop() method removes the item with the specified key name
- The popitem() method removes the last inserted item (in versions before 3.7, a random item is removed instead)
- The del keyword removes the item with the specified key name
- The clear() method empties the dictionary:

REMOVING ITEMS

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

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

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

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

```
thisdict = {
  "brand": "Ford",
  "model": "Mustang",
  "year": 1964
}
del thisdict
print(thisdict) #this will
cause an error because
"thisdict" no longer exists.
```

COPY A DICTIONARY

- You cannot copy a dictionary simply by typing dict2 = dict1, because: dict2 will only be a reference to dict1, and changes made in dict1 will automatically also be made in dict2.

- There are ways to make a copy, one way is to use the built-in Dictionary method copy().

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

NESTED DICTIONARIES

- A dictionary can also contain many dictionaries, this is called nested 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
}
```