



SE Bootcamp

Hyperiondev

Beginner Data Structures – Dictionaries

Objectives

- Learn about dictionaries and how to manipulate dictionaries.

Dictionaries

- ★ Dictionaries in Python are similar to those Oxford dictionaries we all used in English class.
- ★ Python **dictionaries** are similar to a **list**, however each item has **two parts**, a **key** and a **value**.
- ★ For example, an English dictionary has the word (**key**) and then its definition (**value**)

Dictionary Example

```
my_dictionary = {  
    "name" : "Terry",  
    "age" : 23,  
    "is_funny" : False  
}
```

- ★ Dictionaries are **enclosed** in **curly brackets**; key value pairs are separated by **colon** and each pair is separated by a **comma**.
- ★ On the **left is the key**, on the **right is the value**.

Dict Function

- ★ Used to create **dictionaries**.
- ★ Assign **values** to **keys** by passing in keys and values separated by an **=** sign.

```
new_dictionary = dict(name="kitty", age=0.5, kitten=True)
print(new_dictionary)
```

```
# Result >> {'name': 'kitty', 'age': 0.5, 'kitten': True}
```

Accessing a value in dictionaries

- ★ To access a value in a dictionary, we simply call the key and Python will return the value paired with said key.
- ★ Similar to indexing, however we provide a key name instead of an index number.

Example

```
new_dictionary = dict(name="kitty", age=0.5, kitten=True)
print(new_dictionary["name"])
# Result >> kitty
print(new_dictionary["age"])
# Result >> 0.5
```

Accessing all values

We are able to use the `.values()` method on a dictionary in a for loop to access every value in a dictionary.

```
new_dictionary = dict(name="kitty", age=0.5, kitten=True)

for value in new_dictionary.values():
    print(value)

# Result >> kitty
#           0.5
#           True
```


Accessing all keys

We are able to use the `.keys()` method on a dictionary in a for loop to access every key in a dictionary.

```
new_dictionary = dict(name="kitty", age=0.5, kitten=True)

for keys in new_dictionary.keys():
    print(keys)

# Result >> name
#           age
#           kitten
```

Accessing both keys & values

We are able to use the `.items()` method on a dictionary in a for loop to access all keys and values in a dictionary.

```
new_dictionary = dict(name="kitty", age=0.5, kitten=True)

for keys, values in new_dictionary.items():
    print(keys, " : ", values)

# Result >> name : kitty
#           age : 0.5
#           kitten : True
```

Popping out of a dictionary

- ★ Similar to the list, we can use `.pop()` to remove a pair out of a dictionary and return the value of the pair in a variable.
- ★ Please note that using `.pop()` must have the key as an argument for pop to work.

```
new_dictionary = dict(name="kitty", age=0.5, kitten=True)

value = new_dictionary.pop("name")

print(value)

# Result >> kitty
```

Hyperiondev

Q & A Section

Please use this time to ask any questions relating to the topic explained, should you have any



Hyperiondev

**Thank you
for joining us**