Section 7

Dictionaries

Mohammed Asir Shahid

2021-08-03

Contents

1 The Dictionary Data type

Like a list, a dictionary is a collection of many values, however indices can be many different types of data types, not just integers. These indices are called keys.

```
myCat={"size":"fat", "color":"gray", "disposition":"loud"}

print(myCat)

print(myCat["size"])

print("My cat has {} fur.".format(myCat["color"]))


spam={12345:"Luggage combinationn", 42:"The Answer"}
print(spam)
```

```
{'size': 'fat', 'color': 'gray', 'disposition': 'loud'}
fat
My cat has gray fur.
{12345: 'Luggage combinationn', 42: 'The Answer'}
```

Dictionaries also do not have orders. While lists with the same contents but different orders are not equivalent, dictionaries with the same content are equivalent regardless of any way in which the content is ordered.

```
print([1,2,3]==[3,2,1])

eggs={"name":"Zophie", "species":"cat", "age":8}
ham={"species":"cat", "name":"Zophie", "age":8}

print(eggs==ham)
```

```
False
True
```

We can use the in and not in operators in order to see whether or not a key exists in the dictionary.

```
eggs={"name":"Zophie", "species":"cat", "age":8}

print("name" in eggs)
print("name" not in eggs)
```

```
True
False
```

There are three common dictionary methods that will return list like values of the dictionaries keys, values, or both. These are the keys, values, and items methods.

```
eggs={"name":"Zophie", "species":"cat", "age":8}

print(list(eggs.keys()))
print(list(eggs.values()))
print(list(eggs.items()))
```

```
['name', 'species', 'age']
['Zophie', 'cat', 8]
[('name', 'Zophie'), ('species', 'cat'), ('age', 8)]
```

These can be useful in for loops.

```
eggs={"name":"Zophie", "species":"cat", "age":8}

for k in eggs.keys():
    print(k)

for v in eggs.values():
    print(v)

for k,v in eggs.items():
    print(k,v)
```

```
name
species
age
Zophie
cat
8
name Zophie
species cat
age 8
```

We can use the in and not in operators in order to see whether a certain key or value exists in the dictionary.

```
eggs={"name":"Zophie", "species":"cat", "age":8}
print("cat" in eggs.values())
```

```
True
```

The dictionary has a get() method which can be helpful for checking whether a key exists in a dictionary. The get() method takes in two different arguments, the first is what key we are looking for and the second is the value that should be returned in case the key is not in the dictionary.

```
eggs={"name":"Zophie", "species":"cat", "age":8}
print(eggs.get("age",0))
print(eggs.get("color",""))

picnicItems={"apples":5, "cups":2}
print("I am bringing {} {} to the picnic.".format(picnicItems.get("napkins",0),"napkins
```

```
8

I am bringing 0 napkins to the picnic.
```

If we didn't use get above, we would have ran into error messages when the key did not exist in the dictionary.

The opposite of the get() method is the setdefualt() method which sets the default value for a key in a dictionary.

```
eggs={"name":"Zophie", "species":"cat", "age":8}

print(eggs)

if "color" not in eggs:
    eggs["color"]="black"
print(eggs)

eggs={"name":"Zophie", "species":"cat", "age":8}

print(eggs)

eggs.setdefault("color","black")

print(eggs)
```

```
{'name': 'Zophie', 'species': 'cat', 'age': 8}
{'name': 'Zophie', 'species': 'cat', 'age': 8, 'color': 'black'}
{'name': 'Zophie', 'species': 'cat', 'age': 8}
{'name': 'Zophie', 'species': 'cat', 'age': 8, 'color': 'black'}
```

The setdefault() method is a nice shortcut to ensure that certain keys exist.

We can write a short character counting program using what we have learned.

```
message="It was a bright cold day in April, and the clocks were striking thirteen"

count={}

for char in message.upper():
    count.setdefault(char,0)
    count[char]+=1
print(count)
```

```
{'I': 7, 'T': 6, ' ': 13, 'W': 2, 'A': 5, 'S': 3, 'B': 1, 'R': 5, 'G': 2, 'H': 3, 'C':
```

The above program will work regardless of the length of the message. We could have a complete book and our program would properly figure out the amount of each character in the text.

We can use the pprint module in order to display the items in the dictionary in a cleaner way.

```
import pprint

message="It was a bright cold day in April, and the clocks were striking thirteen"

count={}

for char in message.upper():
    count.setdefault(char,0)
    count[char]+=1
pprint.pprint(count)

formattedtext=pprint.pformat(count)

print(formattedtext)
```

```
{' ': 13,
```

```
 ',': 1,
 'A': 5,
 'B': 1,
 'C': 3,
 'D': 3,
 'E': 5,
 'G': 2,
 'H': 3,
 'I': 7,
 'K': 2,
 'L': 3,
 'N': 4,
 'O': 2,
 'P': 1,
 'R': 5,
 'S': 3,
 'T': 6,
 'W': 2,
 'Y': 1}
{' ': 13,
 ',': 1,
 'A': 5,
 'B': 1,
 'C': 3,
 'D': 3,
 'E': 5,
 'G': 2,
 'H': 3,
 'I': 7,
 'K': 2,
 'L': 3,
 'N': 4,
 'O': 2,
 'P': 1,
 'R': 5,
 'S': 3,
 'T': 6,
 'W': 2,
 'Y': 1}
```