KEY_Lesson18_Dictionaries

February 4, 2020

1 Dictionaries

Today we're going to learn about a super useful Python data type called **dictionaries**. Dictionaries have **key-value pairs**, and act kind of like a dictionary where you can look up a value using a key. Let's jump in with an example.

Here, we're going to make a dictionary of books where:

- The *keys* are the authors
- The *values* are the books the authors have written

[1]:
```
# create dictionary
books = {'Madeline L\'Engle':'A wrinkle in time',
        'J.K. Rowling':['Harry Potter and the socerer\'s stone',
                        'Harry Potter and the chamber of secrets']}

# print type of books
print(type(books))

# print dictionary
print(books)
```

```
<class 'dict'>
{"Madeline L'Engle": 'A wrinkle in time', 'J.K. Rowling': ["Harry Potter and the
socerer's stone", 'Harry Potter and the chamber of secrets']}
```

Say we want to add an entry to our dictionary. We can do it like this:

[2]:
```
# add entry to dictionary
books['Laura Ingalls Wilder'] = 'Little House on the Prarie'

# print dictionary
print(books)
```

```
{"Madeline L'Engle": 'A wrinkle in time', 'J.K. Rowling': ["Harry Potter and the
socerer's stone", 'Harry Potter and the chamber of secrets'], 'Laura Ingalls
Wilder': 'Little House on the Prarie'}
```

What if we want to try to get the first element in our books dictionary?

[3]:
```
#books[0]
```

Whoops! This didn't work. That's because dictionaries are *unordered*. That means that there's no first, second, third etc. element. This is not like Python lists, which are *ordered*. Because of this, we always have to access dictionaries using their keys.

Let's try accessing something in our dictionary using our new knowledge. Let's look up the book(s) an author has written (the *value*) using the author's name (the *key*):

[4]:
```
# get value in dictionary using key
books['J.K. Rowling']
```

[4]:
```
["Harry Potter and the socerer's stone",
 'Harry Potter and the chamber of secrets']
```

Cool! As long as we know the author's name, we can look up the books that they've written in our dictionary.

What if we want to print out all the keys or all the values? We can do that like this:

[5]:
```
# get all the keys:
print(books.keys())

# get all the values:
print(books.values())
```

```
dict_keys(["Madeline L'Engle", 'J.K. Rowling', 'Laura Ingalls Wilder'])
dict_values(['A wrinkle in time', ["Harry Potter and the socerer's stone",
'Harry Potter and the chamber of secrets'], 'Little House on the Prarie'])
```

Now let's get a little fancier. Let's try to loop through our dictionary! We're going to use the method `.items()` to access key-value pairs in our dictionary:

[6]:
```
# print items in dictionary
books.items()
```

[6]:
```
dict_items([("Madeline L'Engle", 'A wrinkle in time'), ('J.K. Rowling', ["Harry
Potter and the socerer's stone", 'Harry Potter and the chamber of secrets']),
('Laura Ingalls Wilder', 'Little House on the Prarie')])
```

[7]:
```
for author, title in books.items():
    print(author)
    print(title)
    print('\n')
```

```
Madeline L'Engle
A wrinkle in time


J.K. Rowling
```

```
["Harry Potter and the socerer's stone", 'Harry Potter and the chamber of
secrets']


Laura Ingalls Wilder
Little House on the Prarie
```

Say we decide that we actually want to take Laura Ingals Wilder out of our dictionary for some reason. We can do that using the `del` function:

[12]:
```
# delete entry in dictionary and save it to variable
del books['Laura Ingalls Wilder']

# print dictionary
print(books)
```

```
{"Madeline L'Engle": 'A wrinkle in time'}
```

Alternatively, if you want to save the values to a variable and remove it from the dictionary at the same time, you can use the `.pop` method:

[10]:
```
# delete entry in dictionary and save it to variable
jk = books.pop('J.K. Rowling')

# print jk
print(jk)

# print dictionary
print(books)
```

[10]:
```
["Harry Potter and the socerer's stone",
 'Harry Potter and the chamber of secrets']
```

Nice job! You just learned:

- How to create a dictionary in Python
- How to access keys, values, and key-value pairs in a dictionary
- How to loop over elemets in a dictionary
- How to delete elements from a dictionary