Iterating Through Dictionaries with For Loops

Iterating Through Dictionaries with `For` Loops

When you iterate through a dictionary using a `for` loop, doing it the normal way, like

`for n in some_dict`, will only give you access to the **keys** in the dictionary - which is what you'd want in some situations. In other cases, you'd want to iterate through both the **keys** and **values** in the dictionary. Let's see how this is done in an example. Consider this dictionary that uses names of actors as keys and their characters as values.

```python
cast = {
           "Jerry Seinfeld": "Jerry Seinfeld",
           "Julia Louis-Dreyfus": "Elaine Benes",
           "Jason Alexander": "George Costanza",
           "Michael Richards": "Cosmo Kramer"
        }
```

Iterating through it in the usual way with a `for` loop would give you just the keys, as shown below:

```python
for key in cast:
    print(key)
```

This outputs:

```
Jerry Seinfeld
Julia Louis-Dreyfus
Jason Alexander
Michael Richards
```

If you wish to iterate through both keys and values, you can use the built-in method `items` like this:

```python
for key, value in cast.items():
    print("Actor: {}    Role: {}".format(key, value))
```

This outputs:

```
Actor: Jerry Seinfeld     Role: Jerry Seinfeld
Actor: Julia Louis-Dreyfus     Role: Elaine Benes
Actor: Jason Alexander     Role: George Costanza
Actor: Michael Richards     Role: Cosmo Kramer
```

`items` is an awesome method that returns tuples of key, value pairs, which you can use to iterate over dictionaries in `for` loops.

Try It Out!

You can run and experiment with the code in the workspace on the Coding Space page!
Look for the **Iterating Through Dictionary** section (see image below).

An image showing the Iterating Through Dictionaries section in the workspace

An image showing the Iterating Through Dictionaries section in the workspace