Dictionaries and JSON

—

IT Adventures: Smart IT

Reading Assignment

- How to Think Like a Computer Scientist: Interactive Edition
- **Required: 12.1-5**

Dictionaries

- A dictionary is a variable very similar to the list, in that it can store numerous values inside of a single variable.
- They are ordered.
  - When you add a value to the list it will be placed at the end of the dictionary.
- They are malleable.
  - You can change the contents of each index whenever you want.
- They are **not** indexed.
  - Instead of an index, dictionaries use a key to reference element. As a consequence, this means that there can be only unique, not repeated, elements in a dictionary.
    - Elements sharing a key will overwrite to use the value most recently assigned.
- Other languages would refer to this as a hashmap.

How to Make a Dictionary

- Assign the variable name to a group of key value pairs connected with colons, and separated from other pairs with commas, all surrounded by curly braces.
- This is commonly done on separate lines to show the connection between key pairs

```
class = {

“teacher”: “Mr. Jones”,

“subject”: “History”,

“num_of_students”: 30

}
```

Basic Tools for Dictionaries

- “dictionary.get(key)” will return the value associated with the key pair.
- “dictionary.keys()” will return a list of all keys.
- “dictionary.values()” will return a list of all values.
- “dictionary.items()” will return a list of all key pairs.
- “dictionary[key] = value” will add or overwrite a key pair.
- “dictionary.update(key, value)” will update the key pair with the new value.
- “dictionary.pop(key)” will remove the key pair associated with the given key.
- “dictionary.popitem()” will remove the most recently added key pair.
- “dictionary.clear()” will remove all key pairs from the dictionary.
- “if key in dictionary” will return true if the key is found in the dictionary.

JSON

- A JSON is a variable written in JavaScript Object Notation
- For functional purposes it will be turned into a dictionary most often.
    - This makes them a fantastic example, especially as you'll be using them with the RVR
- It is possible to convert python variables into JSON as well. Making JSONs very versatile.
- JSONs are used in every other programming language as well so being familiar with them is worthwhile.

Basic Tools for JSON

- “import json” will import the json module and allow us to use all of the tools.
  - Like all import statements this should be put at the very beginning of the program.
- “json.loads(var)” will return the json “var” as a python object, likely a dictionary.
- “json.dumps(var)” will return the variable “var” as a JSON.
- “json.dumps(var, indent=4)” will return the same as above but with an indentation of 4 making it much easier to read.
- “json.dumps(var, sort_keys=True)” will return the JSON but with keys sorted alphabetically.

JSON Example

```
import json
x = {
        name:”Frank”,
        age:25,
        job:”Teacher”,
}
#y is now a JSON that is sorted and indented.
y=json.dumps(x,is_sorted=true,indent=4)
#z is now a dictionary
z=json.loads(y)
```

Technical Challenge

- See the Sphero RVR Challenge