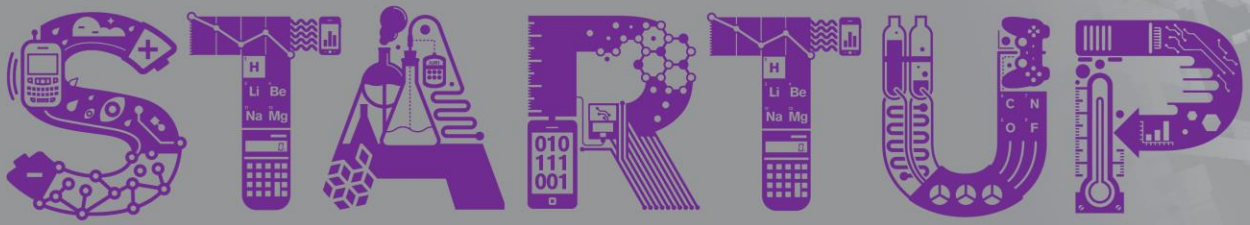
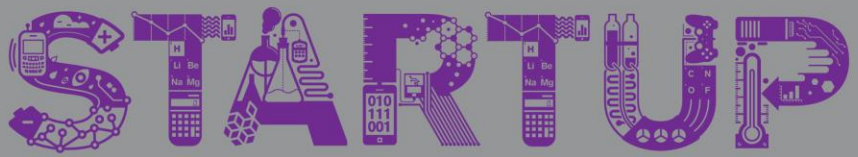
Bloomberg



STARTUP

Bloomberg



Python Lesson 9

Dictionaries

Review: Warm up

1) Create a list that contains at least three names.
2) Write a for loop that iterates over your list and prints out “Hello <Name>” for each name in your list.

How could we store the following data?

Phone Numbers:
Toby: 888-777-6666
Nico: 555-444-3333
Becca: 222-111-9999

- Each phone number has two pieces: The person it belongs to and the number itself.
- Python has a data type to store data like this: **Dictionaries**

Dictionaries: Key-Value Pairs

Dictionaries store data in the form of **key-value pairs**.

- A **key** is like the name of the data.
  - In our phone book example, the key is the person's name.
- A **value** is the piece of data you want to associate to the name.
  - In the phone book, the value is the actual phone number.
- In order to look up the **value,** you need to use the **key.**

Anatomy of a Dictionary

Dictionaries have names, just like lists.

```
phone_number_dict = {
    "toby" : "888-777-6666",
    "nico" : "555-444-3333",
    "becca": "222-111-9999"
}
```

Inside the curly brackets, a comma separated list of key-value pairs.

Key-Value pairs are written as Key: Value

Start and end with curly brackets

Keys must be unique!

Example

Create a dictionary that stores the following price data:

Oreos cost 2.75, Doritos cost 1.25, and Donuts cost 0.80.

```
snack_prices = {
    “oreos”: 2.75,
    “doritos”: 1.25,
    “donuts” : 0.80
}
```

Using and Editing Data in Dictionaries

- You can get a value from a dictionary using the key and square brackets:

```
print(snack_prices["oreos"])
```

- Using the key, you can edit values in the dictionary:

```
snack_prices["oreos"] = 2.50
```

- You can add a new key-value pair at any time:

```
snack_prices["pringles"] = 2.15
```

Dictionary Rules

- Keys can be strings or numbers.
- Values can be anything!
- Keys must be unique.
- Unlike lists, dictionaries do NOT care about the order of their keys.
  - `{"a" : "hello", "b" : "world"}` is the same as `{"b" : "world", "a" : "hello" }`
  - `[1, 2]` is NOT the same thing as `[2, 1]`

Additional Dictionary Properties

- You can loop through the keys of a dictionaries:

```
for tasty_snack in snack_prices:
    print(tasty_snack,"costs",snack_prices[tasty_snack])
```

- You can test if a key is in a dictionary:

```
if "oreos" in snack_prices:
    print("I love oreos!")
```

Recap

- Dictionaries store pairs of data: keys and values.
- To define a dictionary:

```
my_dict = {“name”: “Toby”,
           “age”: 30}
```

- To access or edit information in a dictionary, use the key and square brackets:

```
my_dict[“age”]= 31
```