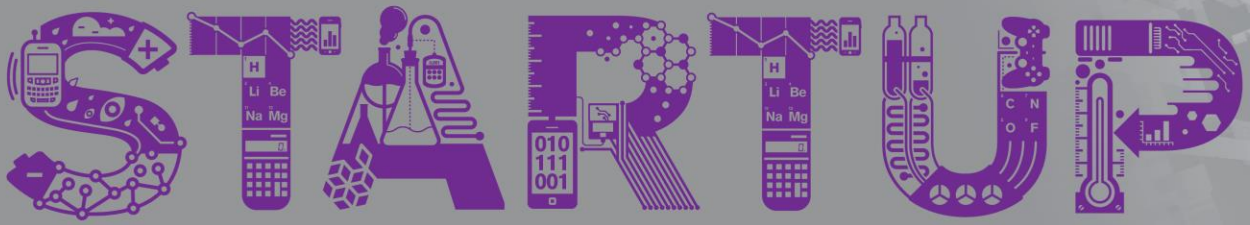
Bloomberg



Bloomberg

STARTUP

Python Lesson 6

Advanced Functions

Review: Warm up

Write a function that takes a number between 0 and 100 and prints out the letter grade (A is 90-100, B is 80-89, C is 70-79, D is 60-69, F is 0-59)

Review: What does this function do?

```
def computeScore(totalPoints, studentScore):
    ratio = studentScore/totalPoints
    print(ratio)
```

What if we want to do more than print the score?

Returning Values

Functions give us a way to take a piece of code that does a specific action, give it a name, and then call it without re-writing the code every time.

We already know how to pass inputs INTO a function.

Now we want to get outputs OUT of a function.

How? Using return values!

Anatomy of a Function Part II

Example:

```
def computeScore(totalPoints, studentScore):
    ratio = studentScore/totalPoints
    return ratio
```

To return a value, just type return your_value!
You can return any python object.

- What are the parameters of this function?
- What is the return value?
- If we define score = computeScore(40, 37), what is the value of score?
- Caution: After you return a value, the function stops executing!

Example: Input function

We've actually already seen a function with a return value:

```
name = input("Enter your name: ")
```

What are the parameters?
What is the return value?

Functions: Practice

- Work on exercises

Python Explorer: Walk

```
def walk(current_location, direction):
    if direction.upper() == "N" and current_location < 4:
        current_location += 1
    elif direction.upper() == "S" and current_location > 0:
        current_location -= 1
    return current_location
```

Recap

- Functions can return values: they can output objects as well as take inputs.
- Because of this, we can call functions within other functions.
- This helps to keep code well organized and easy to read.