Introduction to Programming

Unit 4: Functions

What are functions?

Functions are units of code that we can re-use by executing them whenever we need them.

Code that we can re-use by executing functions

This means that instead of writing the same code multiple times, we can write it once inside of a function and re-use by calling the function anytime we need the run that code.

We've already been using functions.

We've already been using functions.

`print` and `input` are both functions that come with Python.

Code that we can re-use by executing functions

Anytime we want to get the user's input, we can use the `input` function.

Code that we can re-use by executing functions

Anytime we want show something to the user, we can use the `print` function.

We can also create and use our own functions.

Creating our own functions

Creating our own functions let us re-use our own code.

Let's write some code

Executing functions

Functions are *executed* (made to run) by putting parenthesis next to their name.

Executing functions, an example

```
show_message()
```

Executing functions

We execute the `show_message` function by putting parenthesis next to it.

Executing functions

Like variables, functions have names.

Executing functions

Unlike variables, functions must be executed to get their value.

Passing arguments to a function

Sometimes our functions will depend on values/data that live outside of the function.

Passing arguments to a function

We can pass these values as arguments to functions.

Passing arguments to a function

```
def show_greeting(name):
    print("Hello " + name + ", welcome to class")

show_greeting("Ryan")
show_greeting("Olivia")
```

Returning values from a function

We can also *return* values from a function. This is helpful when a function is used to calculate a result that we want to use after.

Returning values from a function

```
def convert_to_inches(feet, inches):
    return feet * 12 + inches
```

Returning values from a function

Values are returned from functions with the `return` keyword.

Caveat

Not everything that looks like a function is a function.

Caveat

- `str(3)`
- `int("23")`

Caveat

`str` and `int` are *classes*, not *functions*. They look similar but are different things. We'll talk more about classes in the future.