

Python Lecture 3 Notes: Functions & Other Building Blocks

Miles Martinez

10/6/22

1 Introduction and learning outcomes from class

Here's a quick summary of what we went over & learned in class!

- Know the building blocks for defining 'functions'
- Learn about how to use 'functions' to transform data
- Learn the different ways to create 'functions'
- Learn the difference between 'functions' and 'methods'

2 Functions

A function is something that maps inputs to outputs! You give a function something, and it gives you something back. There are built-in functions, and there are functions you can create yourself.

Functions are good because they:

- reduce code duplication
- increase abstraction: let us use functions without needing to know their inner workings
- efficiency: people have spent a lot of time making functions work fast in Python, and there are a lot of functions that are already built in to Python! So we don't have to create eVERYTHING from scratch, and the things we don't create will work well.

2.1 Making Functions

We make functions with the following pseudocode:

```
def function_name(argument1, argument2, ..., argumentN):  
  
    code of function  
  
    return something
```

So to make a function we need the word 'def', we need a name for the function, and a set of parentheses with the 'arguments' our function takes inside. 'Arguments' are the inputs to our function. These parentheses need to be followed by a colon and then an indented block of code. The code generally does something to our arguments, then returns some value. To use, or CALL a function, we use this syntax:

```
variable = function(arguments)
```

2.2 keyword arguments

We can also have named arguments in our function, we put these things in our function definition like this:

```
def function_name(arg1, arg2, named_arg1=value2, named_arg2=value2):  
    do code stuff  
  
    return something
```

We use this with:

```
variable = function(arg1, arg2, named_arg1=some value, named_arg2 = another value)
```

Finally, a method is a function that belongs to a specific class (see lecture 4)