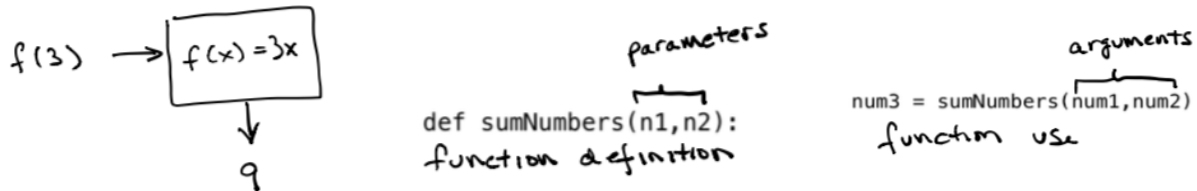- Readings: *Liang* section 6.1 – 6.5 (skip 6.3.1 *Call Stacks*)

A function is like a "Black Box". You can put things in and get things back out, but you don't necessarily have to know anything about the internals to use it.

$f(3) \longrightarrow \boxed{f(x) = 3x}$

9

parameters

```
def sumNumbers(n1,n2):
```
function definition

arguments

```
num3 = sumNumbers(num1,num2)
```
function use

A few notes about functions:

- ***Scope*** can get tricky. Scope means determining which variables apply to which section of code. To make things simple (for now) treat scope this way: the box is closed. Only use variables in the function in which they were declared or passed as arguments. *The only way __in__ is through argument passing, the only way __out__ is with* `return`.

- For functions with default parameters, the default parameters must come *after* all non-default parameters.

- Like everything else in python, indentation is critical inside of functions.

- Functions reduce duplication of code.

- Functions allow us to decompose complex problems into simpler pieces.

- Functions improve the clarity of code.

- Functions facilitate code reuse.