Python

Python Functions

We've seen many built-in functions:

- `print(x)`
- `max(x)`, `min(x)`, `sum(x)`, `len(x)`
- `int(x)`, `float(x)`, `str(x)` (change the type)
- `type(x)` (check the type)
- `range(start, stop, step)` (list # in range)
- `round(float, # digits)` (round a #)

Python Functions

We also saw some functions imported from packages.

`import math`	
`math.abs(x)`	Absolute value
`math.factorial(x)`	Factorial
`import random`	
`random.randint(from, to)`	Choose random integer in range
`random.random()`	Choose random float from 0 - 1
`random.choice(x)`	Choose random item from list/set/string
`import time`	
`time.sleep(seconds)`	Pause Python's execution of a program
`import string`	
`string.ascii_letters`	Returns all letters, 'abc...xyzABC...XYZ'

Python Functions

We can even write our own functions, using `def`:

```
def f_name(f_arguments):
    ... some code ...
    return f_result
```

Recall that:

- Arguments are variables or values that you pass into the function
  - These values are used within the function's code
  - They are optional; without them, the function takes no input
- Return statements are used to pass values out of the function
  - They return a result to the code that called your function
  - They are optional; without them, the function returns nothing

Python Functions

Some simple examples:

A function that takes no argument, and returns nothing:

i.e. just prints "hello"

```
def print_hello():
    print ('hello!')
```

A function that takes an argument, and returns a string:

i.e. `make_hello("world")` returns "hello! world"

```
def make_hello(name):
    return 'hello! ' + name
```