Exercise_3_solutions

January 5, 2019

In [0]:
```
import math
import matplotlib.pyplot as plt
import random
```

1.

In [0]:
```
def pow_gen(x, i):
  j = 0
  while(j < i):
    yield x**j
    j += 1

def polynomial(c_coefs, x_coors):
  return [sum([c_coefs[i]*p for i,p in enumerate(pow_gen(x, len(c_coefs)))]) for x in x

c_coefs = [1,2,3]
x_coors = [0, 2, 4, 8]
y_coors = polynomial(c_coefs, x_coors)
print(y_coors)
```

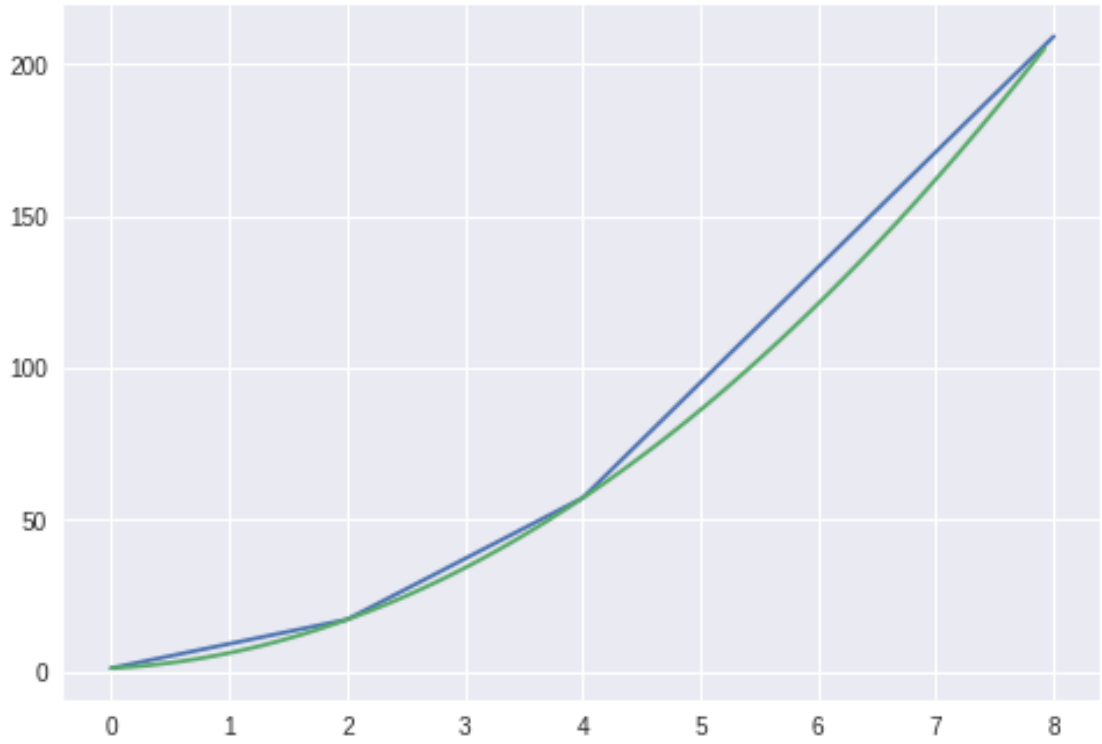
```
[1, 17, 57, 209]
```

2.

In [0]:
```
smooth_x_coors = [8*i/100 for i in range(0,100)]
smooth_y_coors = polynomial(c_coefs, smooth_x_coors)

plt.plot(x_coors, y_coors)
plt.plot(smooth_x_coors, smooth_y_coors)
```

Out[0]:
```
[<matplotlib.lines.Line2D at 0x7f30c989e2e8>]
```



3.

```
In [0]: upperstr = 'thepurposeoflife'
        upperstr = [c.upper() if c=='e' else c for c in upperstr]
        print("".join(upperstr))
```

```
thEpurposEoflifE
```

4.

```
In [0]: records = (('Sam', 19, 'CS'),
        ('Nicole', 21, 'Biochemistry'),
        ('Paul', 20, 'Fine Arts'),
        ('Ashley', 18, 'History'))

        def showrecords(records):
          """Unpack records stored in a tuple of tuples and print each one in a nice format""
          for r in records:
            r0, r1, r2 = r
            print('%s and %d and %s' % (r0, r1, r2))

        showrecords(records)
```

```
Sam and 19 and CS
Nicole and 21 and Biochemistry
Paul and 20 and Fine Arts
Ashley and 18 and History
```

5.

```
In [0]: def multiplier_of(n):
          def multiply(x):
            return x*n

          return multiply

        multiply_with_5 = multiplier_of(5)
        print(multiply_with_5(9))

        multiply_with_45 = multiplier_of(multiply_with_5(9))
        print(multiply_with_45(2))
```

```
45
90
```

6.

```
In [19]: def type_check(correct_type):
             def check(old_function):
               def wrapper(arg):
                 if(isinstance(arg, correct_type)):
                   return old_function(arg)
                 else:
                   print("Bad Type")

               return wrapper

             return check

         @type_check(int)
         def times2(num):
             return num*2

         @type_check(str)
         def first_letter(word):
             return word[0]

         print(times2(2))
         times2('Not A Number')
```

```
        print(first_letter('Hello World'))
        first_letter(['Not', 'A', 'String'])
```

```
4
Bad Type
H
Bad Type
```

7.

```
In [0]: PLUGINS = dict()

        def register(func):
            PLUGINS[func.__name__] = func

        @register
        def say_hello(name):
            return f"Hello {name}"

        @register
        def be_awesome(name):
            return f"Yo {name}, together we are the awesomest!"

        def randomly_greet(name):
            greeter, greeter_func = random.choice(list(PLUGINS.items()))
            print(f"Using {greeter!r}")
            return greeter_func(name)

        randomly_greet('John')
```

```
Using 'be_awesome'


Out[0]: 'Yo John, together we are the awesomest!'
```