Beyond Blocks

Session 1 Challenge Problems

October 17th, 2013

Instructions: return n^2

```
>>> def square(n):
        #Your code here

>>> square(8)
64
```

Instructions: return the average of x and y

```
>>> def average(x, y):
        #Your code here

>>> average(7, 30)
18.5
```

Instructions: return the bigger number from x and y

```
>>> def bigger(x, y):
        #Your code here

>>> bigger(23, 67)
67
```

Instructions: return the biggest of x, y, and z

```
>>> def biggest(x, y, z):
        #Your code here

>>> biggest(11, 20, 3)
20
```

Instructions: write a factorial function USING a “while” loop (i.e. n!)

```
>>> def while_factorial(n):
        #Your code here

>>> while_factorial(5)
120
```

Instructions: write a factorial function USING a “for” loop (i.e. n!)

```
>>> def for_factorial(n):
        #Your code here

>>> for_factorial(5)
120
```

Instructions: print all the factors of **factor** from numbers between **start** and **end**. Print “done” when finished. (assume start <= end)

```
>>> def print_factors_between(start, end, factor):

        #Your code here

>>> print_factors_between(3, 10, 2)
4
6
8
10
done
```

Instructions: write a function that sums the factorials of all numbers from start to end (assume start <= end)

```
>>> def sum_factorials(start, end):
        #Your code here

>>> sum_factorials(4)
32
```

Instructions: if n is even, divide n by 2, if n is odd, multiply n by 3 and add 1. Repeat until n = 1. Print n for each iteration. **Return number of iterations**.

(assume n > 0)

```
>>> def hailstone(n):
        #Your code here


>>> hailstone(5):
5
16
8
4
2
1
5 ← returns number of iterations
```