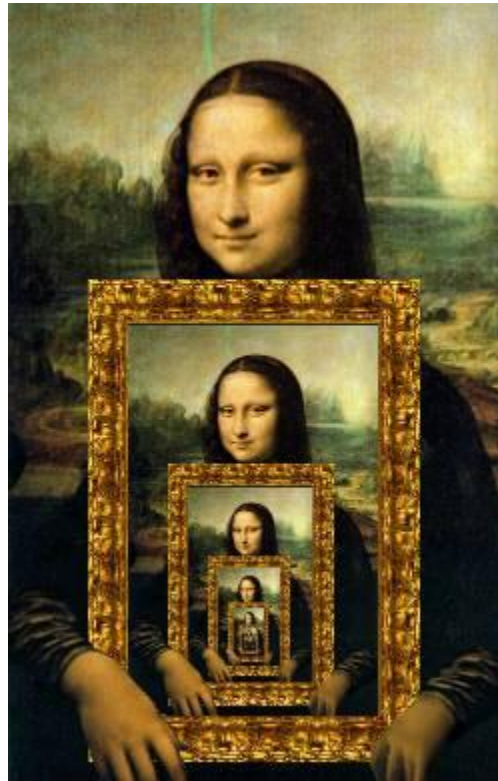


In order to Understand Recursion...



Functions can call other functions

```
def f(a):  
    return a + 1
```

```
def g(b):  
    return f(b) * 2
```

```
g(6)
```

Functions can also call themselves

```
def forever(a):  
    print a  
    forever(a+1)
```

```
forever(0)
```

Oops.

Functions can *sometimes* call themselves.

```
def saner(a):  
    if a < 10:  
        print a  
        saner(a + 1)
```

```
saner(0)
```

Where is this useful?

- For the simplest case, can you just tell me the answer?
- For a more complicated case, can you break it into slightly smaller cases?
- Then recursion might be a good strategy!

Example: Factorial

Factorial of n: All the natural numbers less than or equal to n, multiplied together. For example, 5 factorial (sometimes written 5!) is

$$5 * 4 * 3 * 2 * 1 = 120$$
$$= 5 * 4! \dots$$

```
def factorial(n):  
    if n <= 1:  
        return 1  
    else:  
        return n*factorial(n-1)
```



Base Case



Recursive

What's what

- The **base case** just flat out returns the answer for simple cases.
- The **recursive case** builds the answer for more complicated cases out of simple cases.

Example: Fibonacci

The fibonacci sequence is a sequence where the first two numbers are 1 and 1, and every subsequent number is the sum of the previous two (sound familiar?)

- What's the base case?
- What's the recursive case?

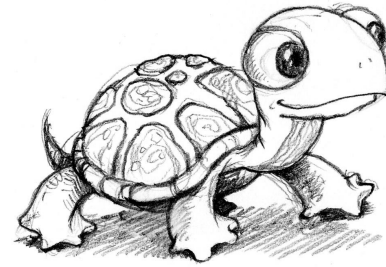
```
def fibonacci(x):  
    if ____:  
        return ____  
    else:  
        return _____
```


Example: Making change

What different ways are there to make change for a given amount x ?

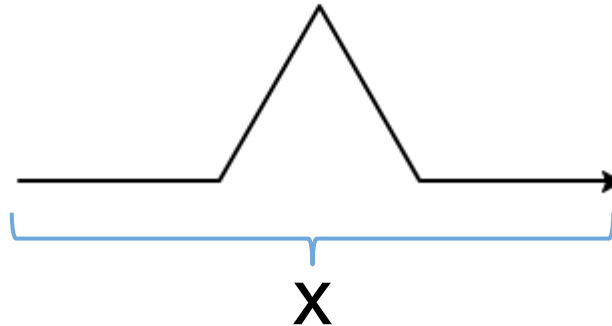
- What's the simplest case here?
1 way for 1 cent: 1 penny
- If a quarter fits, what ways are there to make change for $x - 25$ using quarters or smaller
- If a dime fits, what ways are there to make change for $x - 10$ using dimes or smaller...

In-class Exercise



Using turtle:

- Write a function that draws an equilateral triangle
- Write a function that takes in a length measurement x , and draws this shape:



- Modify your function to replace every straight line with that shape, and so on smaller and smaller.
- Don't forget your base case!