# Recursion in Java

- Let's write the recursive countdown function
  - Algorithm: Countdown from n to 1 then output "Blastoff!"

```
countdown(n)
    if  n == 0 output "Blastoff!"
    else  output n and call countdown( n-1 )
```

- Let's write the recursive countdown function

```
public static void countdown(int n) {
        if (n == 0) {
            System.out.println("Blastoff!");
        } else {
            System.out.println(n);
            countdown(n - 1);
        }
}
```

# Recursion in Java

- Let's write the factorial function
  - f(1) = 1 // I know, I know f(0) =1, but I have my reasons!
  - f(n) = n (n – 1) !

- Let's write the factorial function // looks like magic, doesn't it?!

```
public static int factorial(int n) {
  if (n == 1)   return 1 ;
  else return  n * factorial(n-1) ;
}
```

# Recursion in Java

- Let's write the factorial function again with extra variables, so that I can show you how things work in a simulation

```
public static int factorial(int n) {
  if (n == 0) {
        return 1;
  } else {
        int minus1 = factorial(n-1);
        int result = n * minus1;
        return result;
     }
}
```

# Fibonaci two ways: Print the nth Fibonacci number

Using a loop:

```
static int fib1(int n){
    int curr = 1;
    int prev = 0;
    int next = -1;          //meaningless but needed to compile
    for (int i = 0; i < n-1; i++ ){
      next = curr + prev;
      prev = curr;
      curr = next;
    }
  return curr;
}
```

# Fibonacci two ways: Print the nth Fibonacci number

Using recursion:

```
static int fib2(int n) {
    if (n==0 || n==1) {
      return n;
    }
    else {
      return fib2(n-1) + fib2(n-2);
    }
 }
```