

# Global Variables & Methods

Java  
Mr. Poole

What would happen here?

```
int x = 5;
int y = 6;


if(x < y){
    int sum = x + y;
    System.out.print("x is smaller than y");
}
System.out.println(sum);
```

# What would happen here?

**It won't compile!**

Variables constructed within brackets (squigglys)  
are only usable within the same brackets

Since sum is constructed within the **if** statement, sum can't be accessed outside.




```
int x = 5;
int y = 6;

if(x < y){
    int sum = x + y;
    System.out.print("x is smaller than y");
}
System.out.println(sum);
```

# What would happen here?

Construct sum outside of the if statement and access it within.  
This is an example of variables being used “globally”



```
int x = 5;
int y = 6;
int sum = 0;

if(x < y){
    sum = x + y;
    System.out.print("x is smaller than y");
}
System.out.println(sum);
```

# Let's apply this to Classes/Methods!

Would this compile?

```
class BaseClass{
    public BaseClass(){
        int x = 0;
    }
    public void toString(){
        System.out.println(x);
    }
}

class starter {
    public static void main(String args[]) {
        // Your code goes below here
        BaseClass test = new BaseClass();
        test.toString();
    }
}
```

# Let's apply this to Classes/Methods!

It wouldn't!

`int x` is constructed in the **constructor**, the `toString` method can't access `x`.

To fix this,  
`int x` will be a global variable!

```
class BaseClass{
    public BaseClass(){
        int x = 0;
    }
    public void toString(){
        System.out.println(x);
    }
}

class starter {
    public static void main(String args[]) {
        // Your code goes below here
        BaseClass test = new BaseClass();
        test.toString();
    }
}
```

# Let's apply this to Classes/Methods!

`int x` is now a global variable!

`int x` is constructed in the class  
But declared in the constructor.

This means `toString`  
can now use `x`.

```
class BaseClass{
    int x;
    public BaseClass(){
        x = 0;
    }
    public void toString(){
        System.out.println(x);
    }
}

class starter {
    public static void main(String args[]) {
        // Your code goes below here
        BaseClass test = new BaseClass();
        test.toString();
    }
}
```

# Lab - Global Variables & Methods

We've sort of already created and used Global variables, if you haven't create the following global variables.

- **String role, int strength, int dexterity, int intelligence, int charisma.**
- Now create a new Method called **“myToString”** in your **“myCharacter”** class.
  - This returns nothing
  - This should **print out the role and all stats** easily for us! Example below.
  - Call this instead of printing in your starter.java

```
Your role is rogue  
Your strength trait is 0  
Your dexterity trait is 0  
Your intelligence trait is 0  
Your constitution trait is 0  
Your charisma trait is 0
```