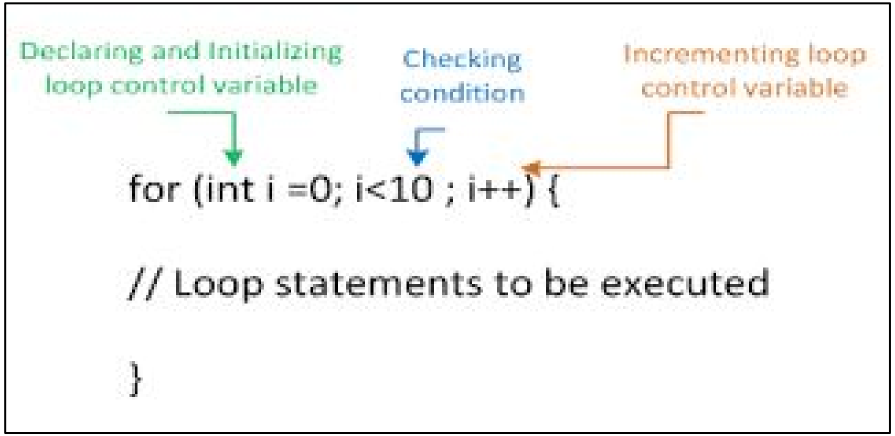
# Method to Madness

Finishing up loops and looking at methods

# For loops

Like a while loop...but different



Declaring and Initializing loop control variable

Checking condition

Incrementing loop control variable

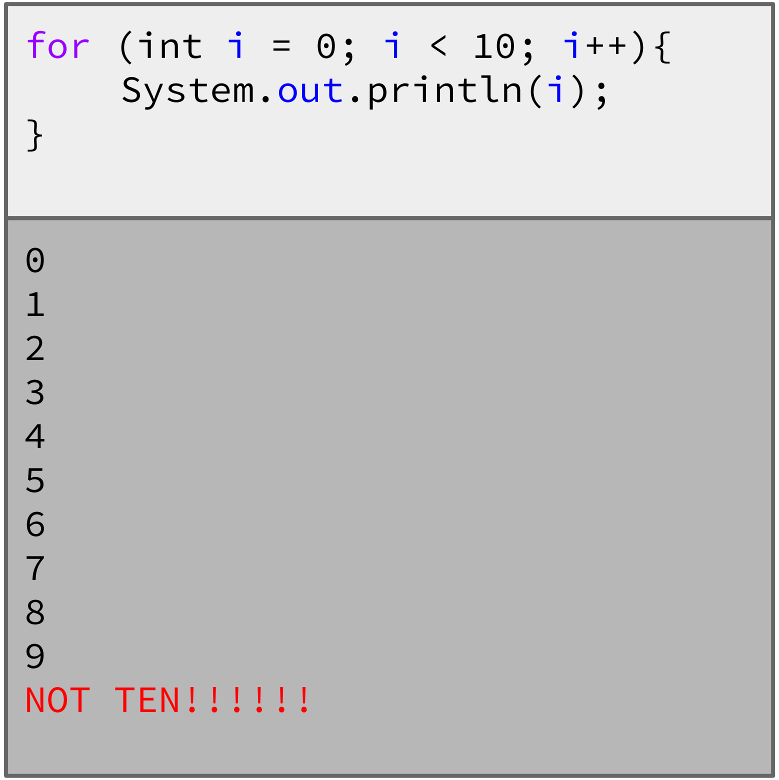
```
for (int i =0; i<10 ; i++) {

// Loop statements to be executed

}
```

```
for (int i = 0; i < 10; i++){
    System.out.println(i);
}
```

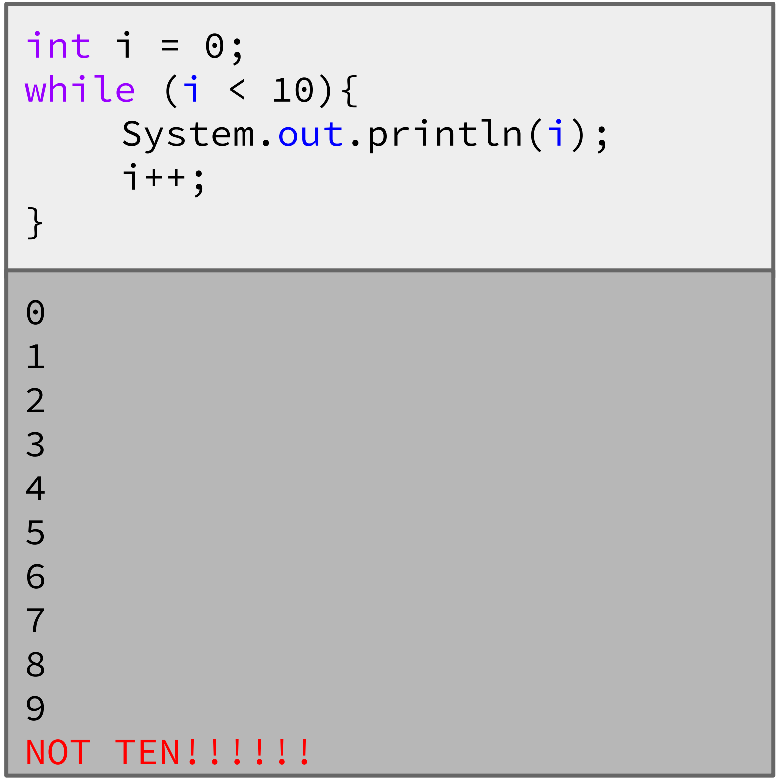
```
0
1
2
3
4
5
6
7
8
9
NOT TEN!!!!!!
```



```
for (int i = 0; i < 10; i++){
    System.out.println(i);
}
```

```
0
1
2
3
4
5
6
7
8
9
NOT TEN!!!!!!
```

=



```
int i = 0;
while (i < 10){
    System.out.println(i);
    i++;
}
```

```
0
1
2
3
4
5
6
7
8
9
NOT TEN!!!!!!
```

# For loop Challenge

Write a for loop that prints out every odd number between 1 and 99

```
Remember:

for (<declare>; <check>; <increment>){

        //Code here gets repeated

}

<increment> - an expression like i += 5;
```

# Picture This...

I have 18 integer variables (a,b,c,d,e…)that have a bunch of different values (that don’t matter).

```
int a = 12;
int b = 19;
int c = 39;
int d = 8;
int e = 2;
int f = 99;
int g = 42;
int h = 72;
int i = 78;
...
```

For every one of them, I want to

1) Print them out
2) Multiply them by 10
3) Print that new number
4) Add 5
5) Print that new number

```java
int a = 12;
int b = 19;
int c = 39;
…
System.out.println(a);
a = a * 10;
System.out.println(a);
a = a + 5;
System.out.println(a);

System.out.println(b);
b = b * 10;
```

```
12
120
125
19
...
```

If only there were a way to make a ‘routine’ of sorts that could do something like this in more ‘general’ circumstances...

Ex: After teaching someone how to make a PB and J sandwich step by step, when you want them to make one, instead of saying those steps again and again you just ask for them to make one and they execute those steps.

# Methods! Example!

```
public static void doThis(int num) {
     System.out.println(num);
     num = num * 10;
     System.out.println(num);
     num = num + 5;
     System.out.println(num);
}

public static void main(String[] args){
     int a = 12;
     int b = 19;
     int c = 39;
     …
     doThis(a);
     doThis(b);
     doThis(c);
}
```

```
12
120
125
```

# Methods Explanation

```
public (static) <return-type> <method-name>(<parameters>){
    //code goes here
    <return-statement>
}
```

# Returning

The thing that the method “returns” is what replaces the “call” to the method in the main code.

Ex: addFive(2) → 7

String, int, char....

void ---> no return value

```
public static void main(String[] args) {
    //Get a number as input
    Scanner scannerName = new Scanner(System.in);
    //Tell the user to enter a number
    System.out.println(“Enter a number”);
    //Store input in a variable
    int inputName;
--> inputName = 9;scannerName.nextInt();

--> System.out.println(inpu9tName);

}
```

```
Enter a number
> 9
9
```

# Video!

# Method Challenge

Create a method – hello – that prints out “Hello World!” and use it in your main method

```
public static void hello(){
     //write code here
}
public static void main(String[] args){
     hello();
}
```