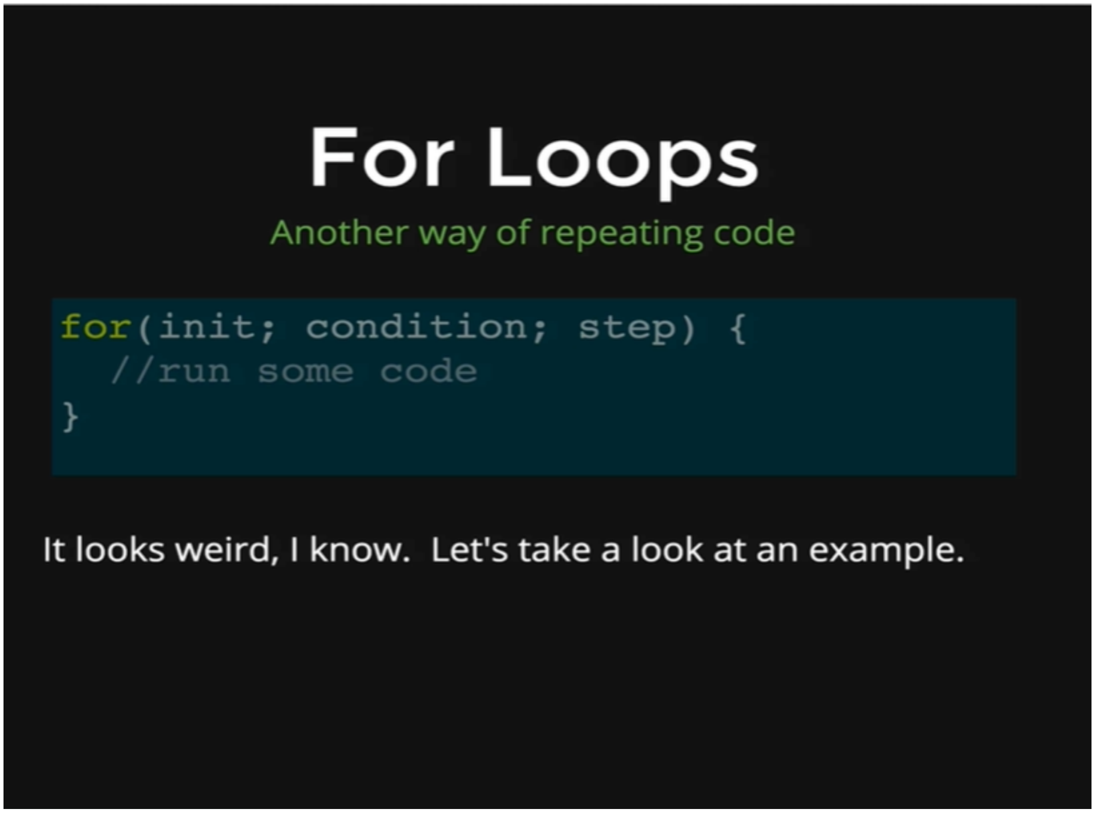
Objectives:

- *Understanding the purpose of for loops*
- *Write valid for loops*
- *Compare and contrast while loops and for loops*

For loop is like the *while loop* that they both let us repeat code. They are little bit different in their syntax.



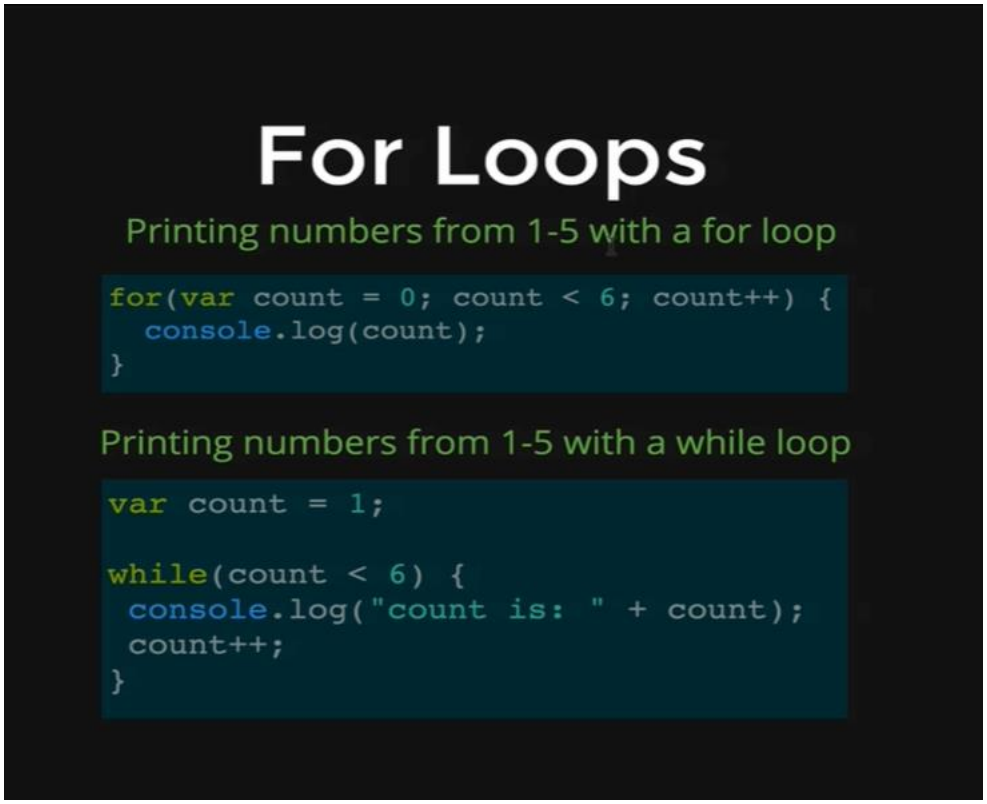
For Loops

Another way of repeating code

```
for(init; condition; step) {
  //run some code
}
```

It looks weird, I know. Let's take a look at an example.

So, to start here is a recipe for *for loop*. We write the word *for*, it's a keyword, then in parentheses there are three parts. They are called *initialize*, *condition* and *step*. Then we have our curly braces that contains the code which executes once the *condition* in the parentheses is met.



For Loops

Printing numbers from 1-5 with a for loop

```
for(var count = 0; count < 6; count++) {
  console.log(count);
}
```

Printing numbers from 1-5 with a while loop

```
var count = 1;

while(count < 6) {
 console.log("count is: " + count);
 count++;
}
```

Here’s an example, this is how we print the numbers from 1 to 5 with a for loop. We start with for and then we create a variable, here we are calling it count, and we set it equal to 0. Then we have our condition, while count is less than 6, and then we have our increment which is going to add 1 to count.

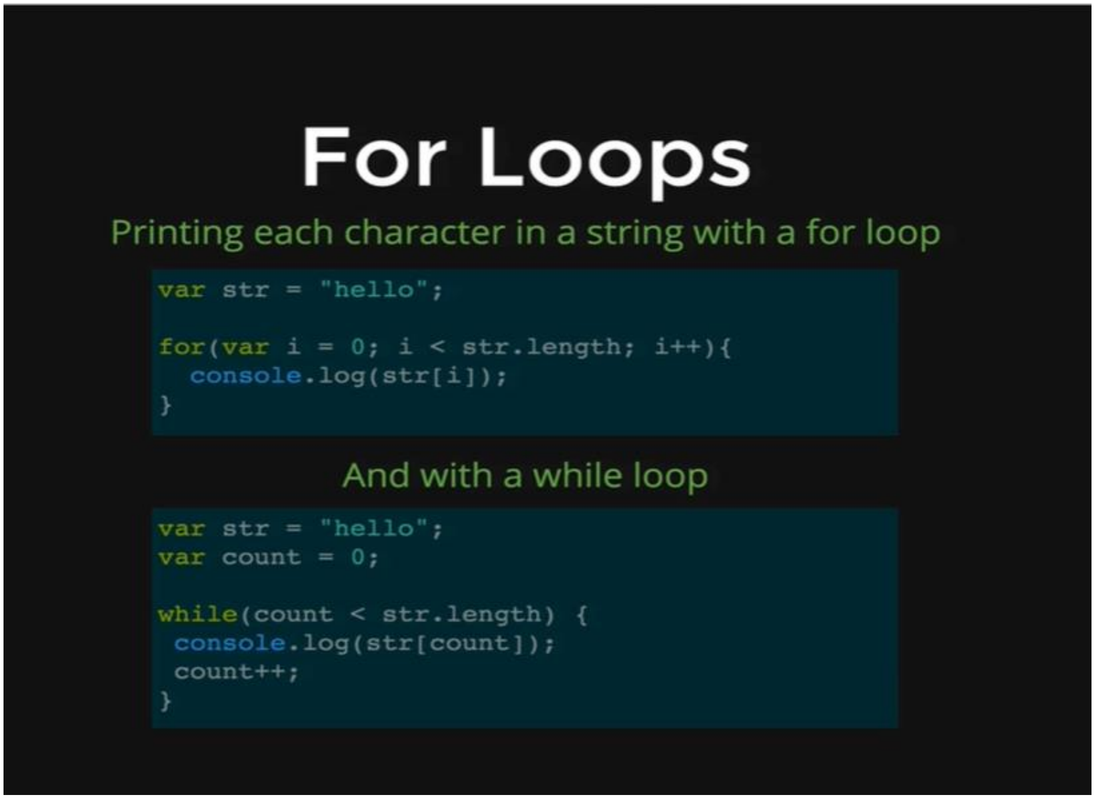
This is conceptually very similar to the while loops we have been writing. You can see the equivalent while loop below our for loop, the big difference is that, the for loop is shorter and the other difference is that the variable count only exists inside the loop of for loop, in the case of while loop the variable must be initialized outside so that we can use it inside the parentheses of while loop. In the for loop a variable can be created which exists only for the moment the loop runs.

In the for loop we are initializing our variable count to be 0, every time through the loop we add 1 to count, and we keep going while count is less than 6. First the count is zero and we add 1 to it and 1 gets printed out, then we add 1 again to count and count is now 2, and 2 gets printed out, and it continues like this till count is 5 and then 5 gets printed out, then count is set to 6 and the condition *count* < *6* is not true anymore, which is why the for loop stops executing its code.

```
for(var count = 0; count < 6; count++) {
        console.log(count);
}
```

Output

```
0
1
2
3
4
5
```



For Loops

Printing each character in a string with a for loop

```
var str = "hello";

for(var i = 0; i < str.length; i++){
  console.log(str[i]);
}
```

And with a while loop

```
var str = "hello";
var count = 0;

while(count < str.length) {
 console.log(str[count]);
 count++;
}
```

So, here is another example taking what we did with the *while* loop where we printed every character in the string but doing it with a for loop, so once again you can see we do not have to create the count variable outside of a loop that we do for a while loop, instead we can create a temporary variable inside the for loop, as we did in this example with a single character ‘i’, and it is pretty conventional to have single character variable used as for loop variables like i, j, x etc. That is because they do not exist for any other purpose but live inside the loop.

In the example above the logic between the for loop and the while loop is same where we are setting our variable to be 0 and keep on increasing the value of the variable, till we reach the length of the string.

```
var str = “hello”;


for (var i = 0; i < str.length; i++) {

        console.log(str[i]);

}
```

Output:

```
h

e

l

l

o
```