

JavaScript Loops

COEN 161

JavaScript Loops

- Loops are useful when you want to run a block of code multiple times
- Loops are very useful when working with arrays
- JavaScript has a few different kinds of loops
 - **for** - loops through a block of code a certain number of times
 - **for/in** - loops through the properties of an object
 - **while** - loops through a block while a condition is true
 - **do/while** - loops through a block of code while a condition is true, but runs at least once

The for Loop

- Syntax:

```
for (statement 1; statement 2; statement 3) {  
    code block to be executed  
}
```

- Statement 1: executed before the loop starts
- Statement 2: defines the condition for running the loop
- Statement 3: executed after each loop

Statement 1

- In JavaScript, statement 1 is optional
- Normally used to initialize the variable used in the loop (i = 0)
- You can initiate many values in statement 1

```
for (i = 0, len = cars.length, text = ""; i < len; i++) {  
    text += cars[i] + "<br>";  
}
```

```
var i = 2;  
var len = cars.length;  
var text = "";  
for (; i < len; i++) {  
    text += cars[i] + "<br>";  
}
```

Statement 2

- Statement 2 is also optional
- Usually, statement 2 is a condition that when its true, runs the block, and when its false, ends the loop.
- *If you don't have a condition in statement 2, you must use a break statement in your loop, otherwise you will have an infinite loop*

Statement 3

- This statement is meant to increment your loop variable (i++)
- However, statement 3 is also optional
- Statement 3 can do anything to change the loop variable (i += 2, i = i * 2, etc.)

```
var i = 0;
var len = cars.length;
for ( ; i < len; ) {
    text += cars[i] + "<br>";
    i++;
}
```

The for/in Loop

- The for/in loop is used to loop through the properties of an object
- Similar to looping through an array except instead of indices, we use keys

```
var person = {fname:"John", lname:"Doe", age:25};
```

```
var text = "";
```

```
var x;
```

```
for (x in person) {  
    text += person[x];  
}
```

The while Loop

- Syntax:

```
while (condition) {  
    code block to be executed  
}
```

- As long as the condition is true, the loop will continue to run

```
while (i < 10) {  
    text += "The number is " + i;  
    i++;  
}
```


The do/while Loop

- This is a slight variant where the condition is checked after the code block
- This means the code runs at least *once*
- Syntax:

```
do {  
    code block to be executed  
}  
while (condition);
```

The break Statement

- We saw the break statement used to exit a switch
- But break can also be used to end a loop early

```
for (i = 0; i < 10; i++) {  
    if (i === 3) {  
        break;  
    }  
    text += "The number is " + i + "<br>";  
}
```

The continue Statement

- The continue statement is like break, except it only skips the rest of the current block and moves on to the next iteration
- In this example, the loop skips 3

```
for (i = 0; i < 10; i++) {  
    if (i === 3) {  
        continue;  
    }  
    text += "The number is " + i + "<br>";  
}
```

Resources

https://www.w3schools.com/js/js_loop_for.asp

https://www.w3schools.com/js/js_loop_while.asp

https://www.w3schools.com/js/js_break.asp