# JavaScript Loops

SOFTWARE DEVELOPMENT WORKSHOP I

*Some of the following codes and examples are taken w3schools.com*

# Different Kinds of Loops

- **for** - loops through a block of code a number of times
- **for/in** - loops through the properties of an object
- **while** - loops through a block of code while a specified condition is true
- **do/while** - also loops through a block of code while a specified condition is true

Loops are handy, if you want to run the same code over and over again, each time with a different value.

# The For Loop

- The for loop is often the tool you will use when you want to create a loop.

- The for loop has the following syntax:

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

# The For Loop

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

- **Statement 1** is executed before the loop (the code block) starts.
- **Statement 2** defines the condition for running the loop (the code block).
- **Statement 3** is executed each time after the loop (the code block) has been executed.

# The For Loop

```
for (i = 0; i < 5; i++) {
    text += "The number is " + i + "<br>";
}
```

- Statement 1 sets a variable before the loop starts

  **(var i = 0).**
- Statement 2 defines the condition for the loop to run

  **(i must be less than 5).**
- Statement 3 increases a value each time the code block in the loop has been executed.

  **(i++)**

# The For Loop

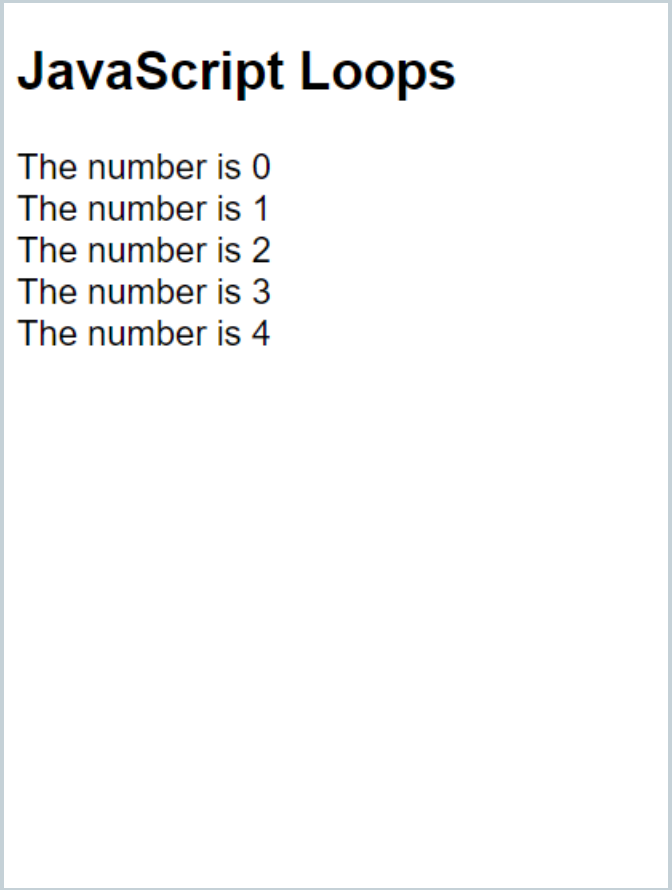
```
<!DOCTYPE html>
<html>
<body>

<h2>JavaScript Loops</h2>

<p id="demo"></p>

<script>
var text = "";
var i;
for (i = 0; i < 5; i++) {
    text += "The number is " + i + "<br>";
}
document.getElementById("demo").innerHTML = text;
</script>

</body>
</html>
```




forLoop1.html

# Statement 1

- Normally you will use statement 1 to initialize the variable used in the loop (i = 0).
- This is not always the case, JavaScript doesn't care. Statement 1 is optional.
- You can initiate many values in statement 1 (separated by comma):

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

# Statement 2

- Often statement 2 is used to evaluate the condition of the initial variable.
- This is not always the case, JavaScript doesn't care. Statement 2 is also optional.
- If statement 2 returns true, the loop will start over again, if it returns false, the loop will end.
- If you omit statement 2, you must provide a **break** inside the loop. Otherwise the loop will never end. This will crash your browser.

# Statement 3

- Often statement 3 increments the value of the initial variable.
- This is not always the case, JavaScript doesn't care, and statement 3 is optional.
- Statement 3 can do anything like negative increment (i--), positive increment (i = i + 15), or anything else.
- Statement 3 can also be omitted (you can increment your values inside the loop):

# Class Exercise

- Open **forLoopExercise.html**
- In the for loop, change *num1* to **0** and *num2* to **10** and run the code.
- Make the loop start counting from 5 instead of 0:
- Make the loop start counting from 5. Count up to (including) 50, and count only every fifth number.
- Make the loop start counting downwards from 10 and stop at 1
- output the numbers **1 3 5 7 9** with line breaks between each number. .

# Example

```
<!DOCTYPE html>
<html>
<body>

<p id="demo"></p>

<script>
var cars = ["BMW", "Volvo", "Saab", "Ford"];

var i = 0;
var len = cars.length;
var text = "";

for (; i < len; ) {
    text += cars[i] + "<br>";
    i++;
}
document.getElementById("demo").innerHTML = text;
</script>

</body>
</html>
```

BMW
Volvo
Saab
Ford

forLoop2.html

# For loops through arrays

- JavaScript arrays are used to store multiple values in a single variable. (next class topic)
- Instead of the code below, which is 3 variables

```
var car1 = "Saab";
var car2 = "Volvo";
var car3 = "BMW";
```

- We can use an array literal

```
var cars = ["Saab", "Volvo", "BMW"];
```

# For loops through arrays

- Same thing

```
text += cars[0] + "<br>";
text += cars[1] + "<br>";
text += cars[2] + "<br>";
text += cars[3] + "<br>";
text += cars[4] + "<br>";
text += cars[5] + "<br>";
```

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

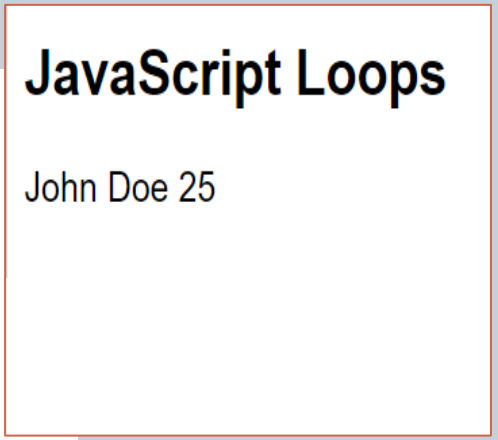
# The For/In Loop

- The JavaScript for/in statement loops through the properties of an object:

```
<h2>JavaScript Loops</h2>

<p id="demo"></p>

<script>
var txt = "";
var person = {fname:"John", lname:"Doe", age:25};
var x;
for (x in person) {
    txt += person[x] + " ";
}
document.getElementById("demo").innerHTML = txt;
</script>
```




forInLoop.html

# While Loop

- The while loop loops through a block of code as long as a specified condition is true.
- Syntax

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

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

If you forget to increase the variable used in the condition, the loop will never end. This will crash your browser.

# While Loop

```
<!DOCTYPE html>
<html>
<body>

<h2>JavaScript while</h2>

<p id="demo"></p>

<script>
var text = "";
var i = 0;
while (i < 10) {
    text += "<br>The number is " + i;
    i++;
}
document.getElementById("demo").innerHTML = text;
</script>

</body>
</html>
```

**JavaScript while**

The number is 0
The number is 1
The number is 2
The number is 3
The number is 4
The number is 5
The number is 6
The number is 7
The number is 8
The number is 9

whileLoop.html

# Do/While Loop

- The do/while loop is a variant of the while loop.
- This loop will execute the code block once,
  - before checking if the condition is true,
  - then it will repeat the loop as long as the condition is true.

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

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

# Do/While Loop

```
<h2>JavaScript do ... while</h2>

<p id="demo"></p>

<script>
var text = ""
var i = 0;

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

document.getElementById("demo").innerHTML = text;
</script>
```

**JavaScript do ... while**

The number is 0
The number is 1
The number is 2
The number is 3
The number is 4
The number is 5
The number is 6
The number is 7
The number is 8
The number is 9

The loop will always be executed at least once, even if the condition is false, because the code block is executed before the condition is tested:

doWhileLoop.html

# Comparing For and While

```
var cars = ["BMW", "Volvo", "Saab", "Ford"];
var i = 0;
var text = "";

for (;cars[i];) {
    text += cars[i] + "<br>";
    i++;
}
```

```
var cars = ["BMW", "Volvo", "Saab", "Ford"];
var i = 0;
var text = "";

while (cars[i]) {
    text += cars[i] + "<br>";
    i++;
}
```

A while loop is much the same as a for loop, with statement 1 and statement 3 omitted.

in both examples the **loop** collects the car names from the cars array:

# Class Exercise 2

## Loop Exercise

| Number | Number x 10 | Number ^ 2 |
|---|---|---|
| 1 | 10 | 1 |
| 2 | 20 | 4 |
| 3 | 30 | 9 |
| 4 | 40 | 16 |
| 5 | 50 | 25 |
| 6 | 60 | 36 |
| 7 | 70 | 49 |
| 8 | 80 | 64 |
| 9 | 90 | 81 |
| 10 | 100 | 100 |

# Class Exercise 2

**Loop Exercise**

| Number | Number x 10 | Number ^ 2 |
|---|---|---|
| 1 | 10 | 1 |
| 2 | 20 | 4 |
| 3 | 30 | 9 |
| 4 | 40 | 16 |
| 5 | 50 | 25 |

| Number | Number x 10 | Number ^ 2 |
|---|---|---|
| 6 | 60 | 36 |
| 7 | 70 | 49 |
| 8 | 80 | 64 |
| 9 | 90 | 81 |
| 10 | 100 | 100 |

| Number | Number x 10 | Number ^ 2 |
|---|---|---|
| 11 | 110 | 121 |
| 12 | 120 | 144 |
| 13 | 130 | 169 |
| 14 | 140 | 196 |
| 15 | 150 | 225 |

| Number | Number x 10 | Number ^ 2 |
|---|---|---|
| 16 | 160 | 256 |
| 17 | 170 | 289 |

There are many ways to do this, but try to do it in the most simple way (minimum amount of code)

# Radio Button Array

```
<input type="radio" name="gender" value="male" checked> Male<br>
<input type="radio" name="gender" value="female"> Female<br>
<input type="radio" name="gender" value="other"> Other<br><br>
<input type="submit" onclick="radioButton()">
<script>
function radioButton(){
        var radios = document.getElementsByName('gender');

        for (var i = 0, length = radios.length; i < length; i++) {
                if (radios[i].checked) {
                // do whatever you want with the checked radio
                alert(radios[i].value);

                // only one radio can be logically checked, don't check the rest
                break;
                }
        }
}
```

radioLoop.html

# Radio Button Array

```
<input type="radio" name="gender" value="male" checked> Male<br>
<input type="radio" name="gender" value="female"> Female<br>
<input type="radio" name="gender" value="other"> Other<br><br>
<input type="submit" onclick="radioButton()">
```

radioLoop.html

# Radio Button Array

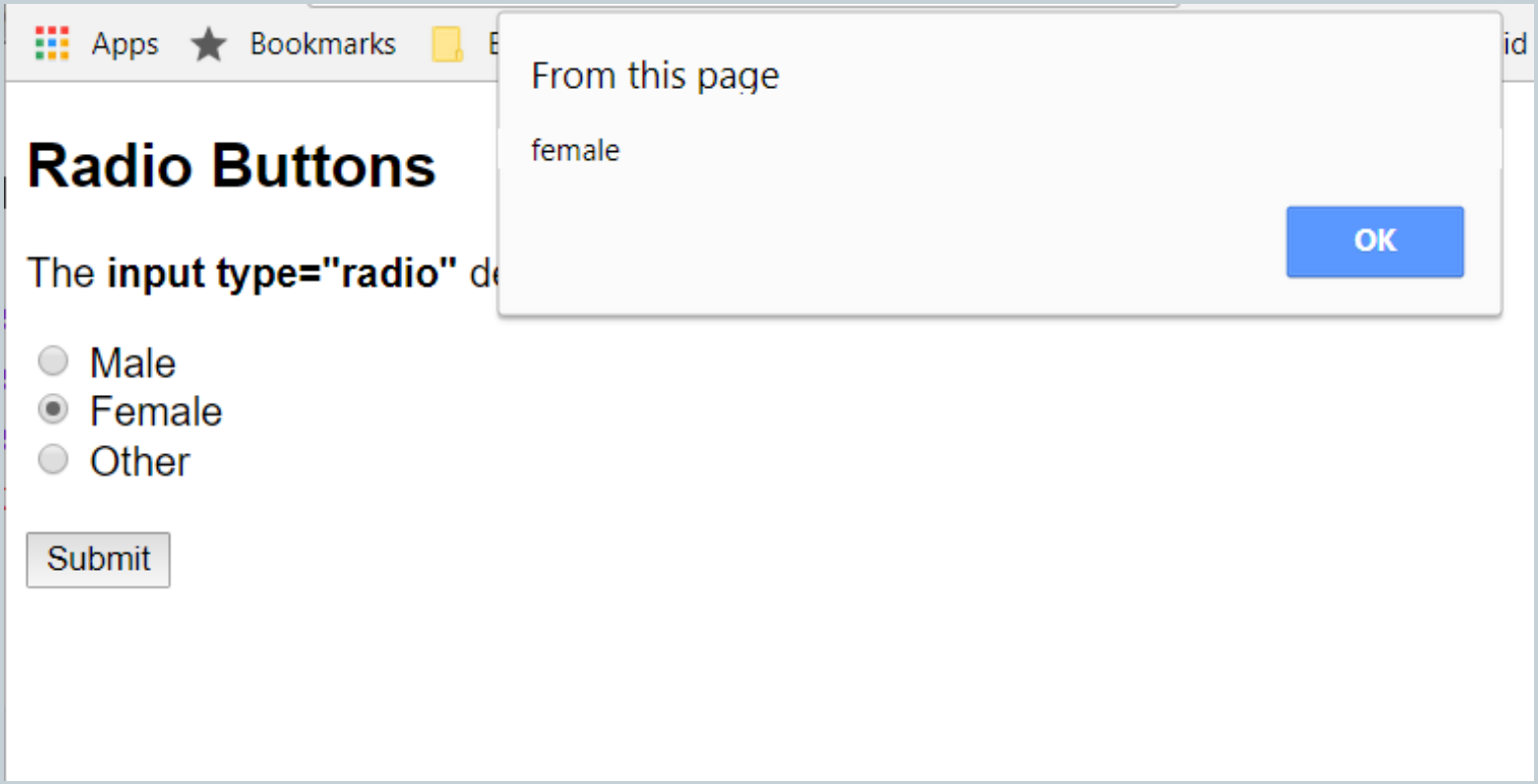
```
<script>
function radioButton(){
        var radios = document.getElementsByName('gender');

        for (var i = 0, length = radios.length; i < length; i++) {
                if (radios[i].checked) {
                // do whatever you want with the checked radio
                alert(radios[i].value);

                // only one radio can be logically checked, don't check the rest
                break;
                }
        }
}
</script>
```

radioLoop.html

# Radio Button Array



# Radio Button: Class Exercise

- Download your homework 2
- Add radio buttons to the code (none checked)
  - Mr
  - Ms
  - Dr
  - Prof
- Depending on the selection, place that title in front of the name once the submit button is clicked
- If no radio button is selected, then leave the title blank