

TLE / ICT 9 FOURTH QUARTER



LESSON
4



LESSON
4

JavaScript

LOOPS

What is a Loop?

There are times when you want a block of code repeated over and over again until a certain condition is satisfied. This can be done by creating a loop. A loop is a repetitive cycle of a block of code until the condition is met.

In JS, there are a variety of loops:

- 1. For loop – it loops through a block of code within a specified number of times.**
- 2. While loop – It loops through a block of code until the condition is satisfied.**
- 3. Do-while loop – It is like while loop; however, it will execute the set of codes at least one.**
- 4. For-in loop – It loops through the elements of an array.**

LOOPS

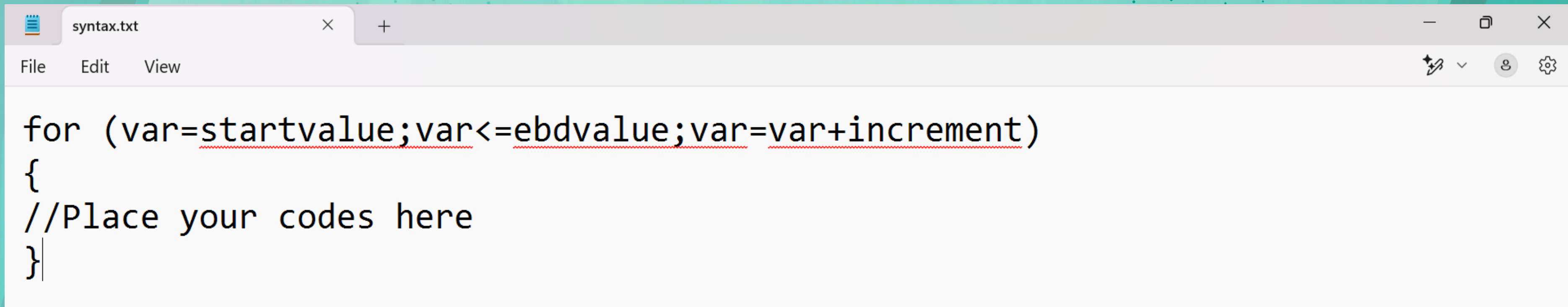
Note:

If for instance, you want to exit from the loop after a condition has been met, you can use the *break* statement. This statement will immediately break the loop and continue the execution of codes after the said loop. The *continue* statement will break the current loop and will continue to the next value.

For Loop

The for loop is used when you know how many times the block of code must be iterated.

Syntax:



```

syntax.txt
File Edit View
for (var=startvalue;var<=endvalue;var=var+increment)
{
//Place your codes here
}
  
```




forloop.txt



File

Edit

View

Sample Script

```
<!DOCTYPE html>
<html>
<head><title>ForLoop</title></head>
<body>
<script type="text/javascript">
var x;
for (x=0;x<=6;x++)
{
document.write(++x);
document.write("<br/>");
}
document.write("Kaboom!");
</script>
</body>
</html>
```

Browser Output



ForLoop



File

C:/Users/MaryGraceUminga/Documents/forloop.html

1

3

5

7

Kaboom!



```
<!DOCTYPE html>
<html>
<head><title>For Loop Input Example</title></head>
<body>
<script type="text/javascript">
  // Number of times to ask for input
  var numberOfInputs = 3;

  // Loop to get inputs
  for (var i = 0; i < numberOfInputs; i++)
  {
    // Prompt the user to input a value
    var userInput = prompt("Please enter a value (" + (i + 1) + " of " + numberOfInputs + ")");

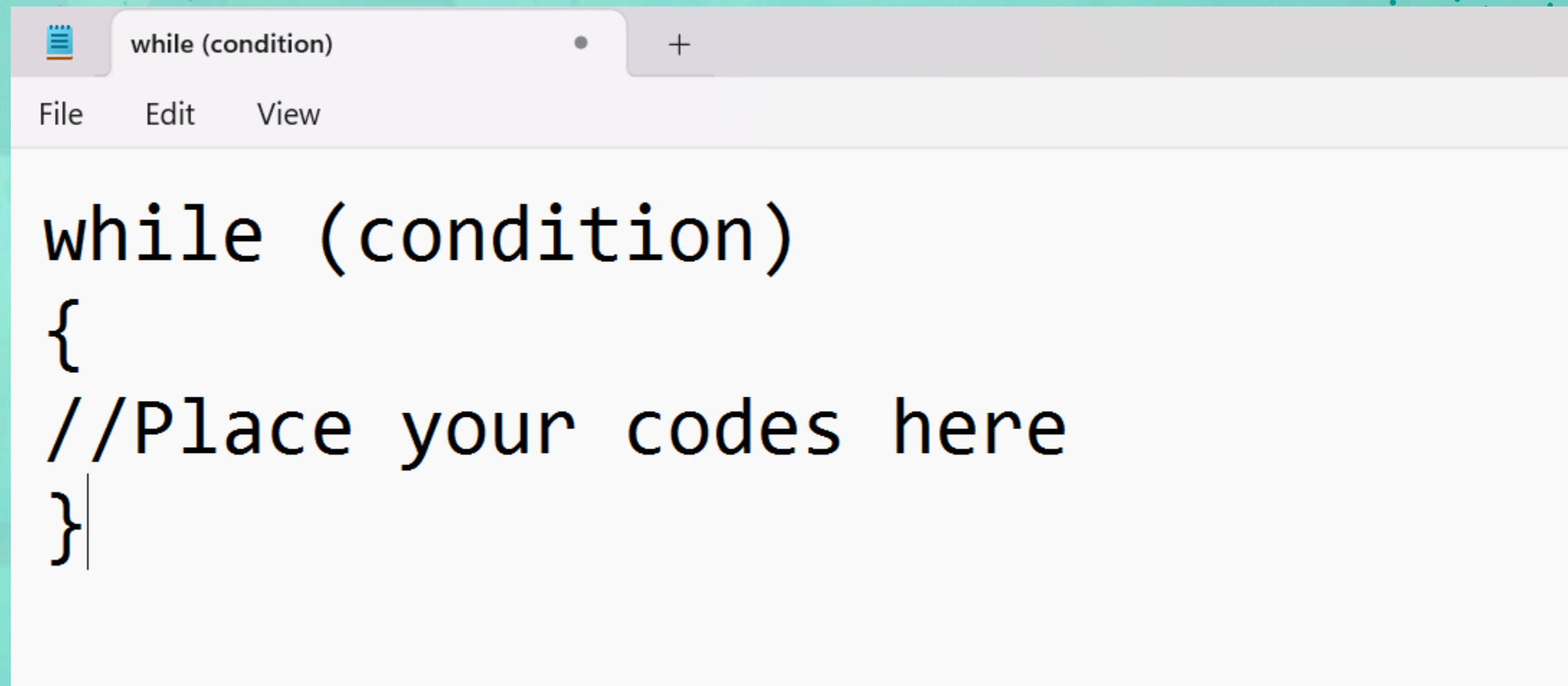
    // Display the entered value
    document.write("You entered: " + userInput + "<br/>");
  }
</script>
</body>
</html>
```

**Do this**

While Loop

The while loop is used if you know when the condition will be met. It will loop through the set of codes until the condition is met.

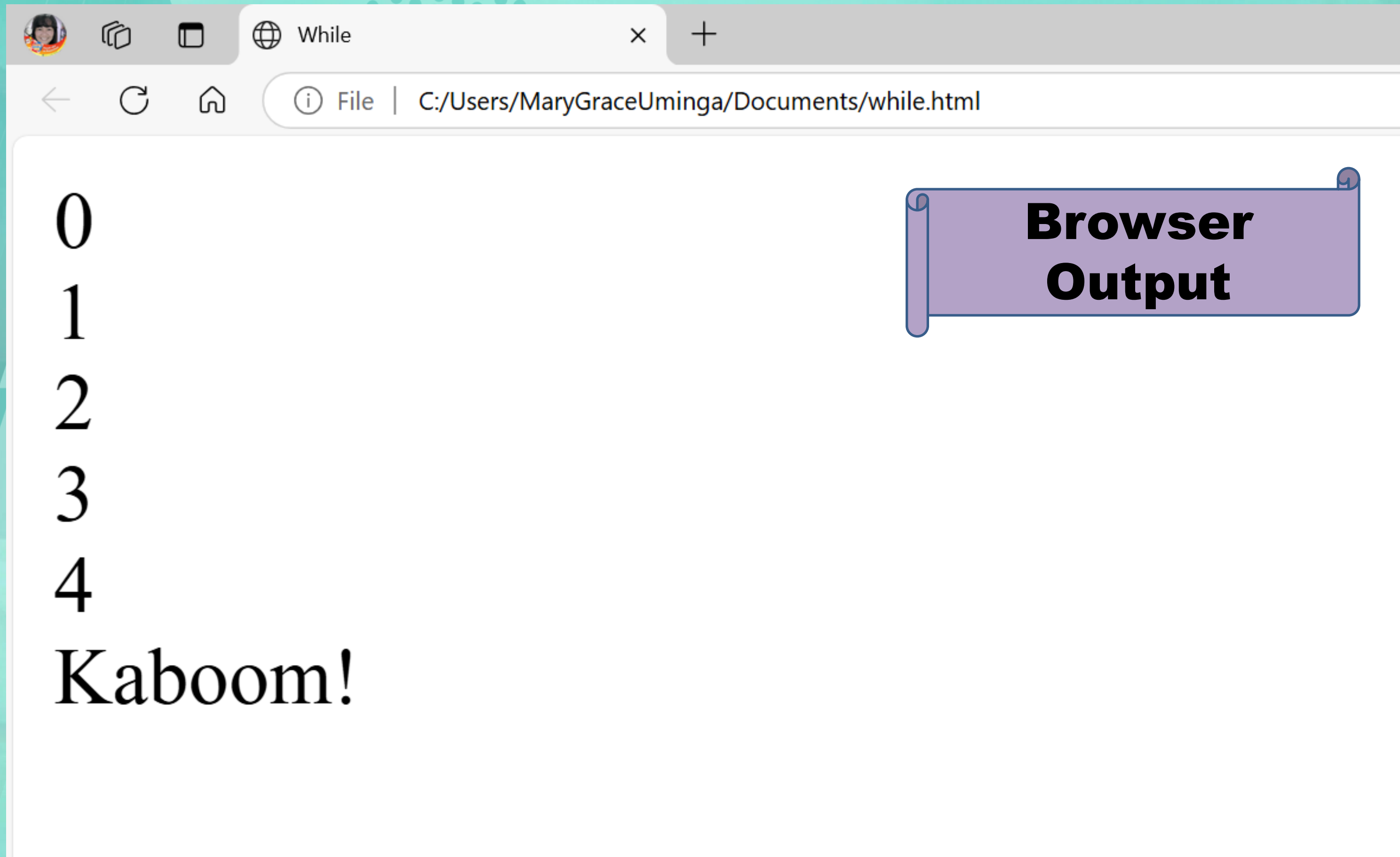
Syntax:

A screenshot of a code editor window. The title bar shows a tab labeled 'while (condition)' with a plus sign to its right. Below the title bar is a menu bar with 'File', 'Edit', and 'View'. The main area of the editor contains the following code:

```
while (condition)
{
//Place your codes here
}
```

Sample Script

```
<!DOCTYPE html>
<html>
<head><title>While</title></head>
<body>
<script type="text/javascript">
var x=0;
while (x<5)
{
document.write(x);
document.write("<br/>");
x++;
}
document.write("Kaboom!");
</script>
</body>
</html>
```

Do this

```
<!DOCTYPE html>
<html>
<head><title>While Loop Input Example</title></head>
<body>
<script type="text/javascript">
    var userInput = "";

    // While loop to continue asking for input until a valid value is entered
    while (userInput === "")
    {
        // Prompt the user for input
        userInput = prompt("Please enter a value (cannot be empty):");
        // Check if the input is empty
        if (userInput === "")
        {
            alert("Input cannot be empty. Please try again.");
        }
    }

    // Once a valid input is entered, display the value
    document.write("You entered: " + userInput);
</script>
</body>
</html>
```


Do-While Loop

Do-while loops work like the while loop except that the code will be executed once and is repeated as long as the set condition is true, but will stop once the condition is false.

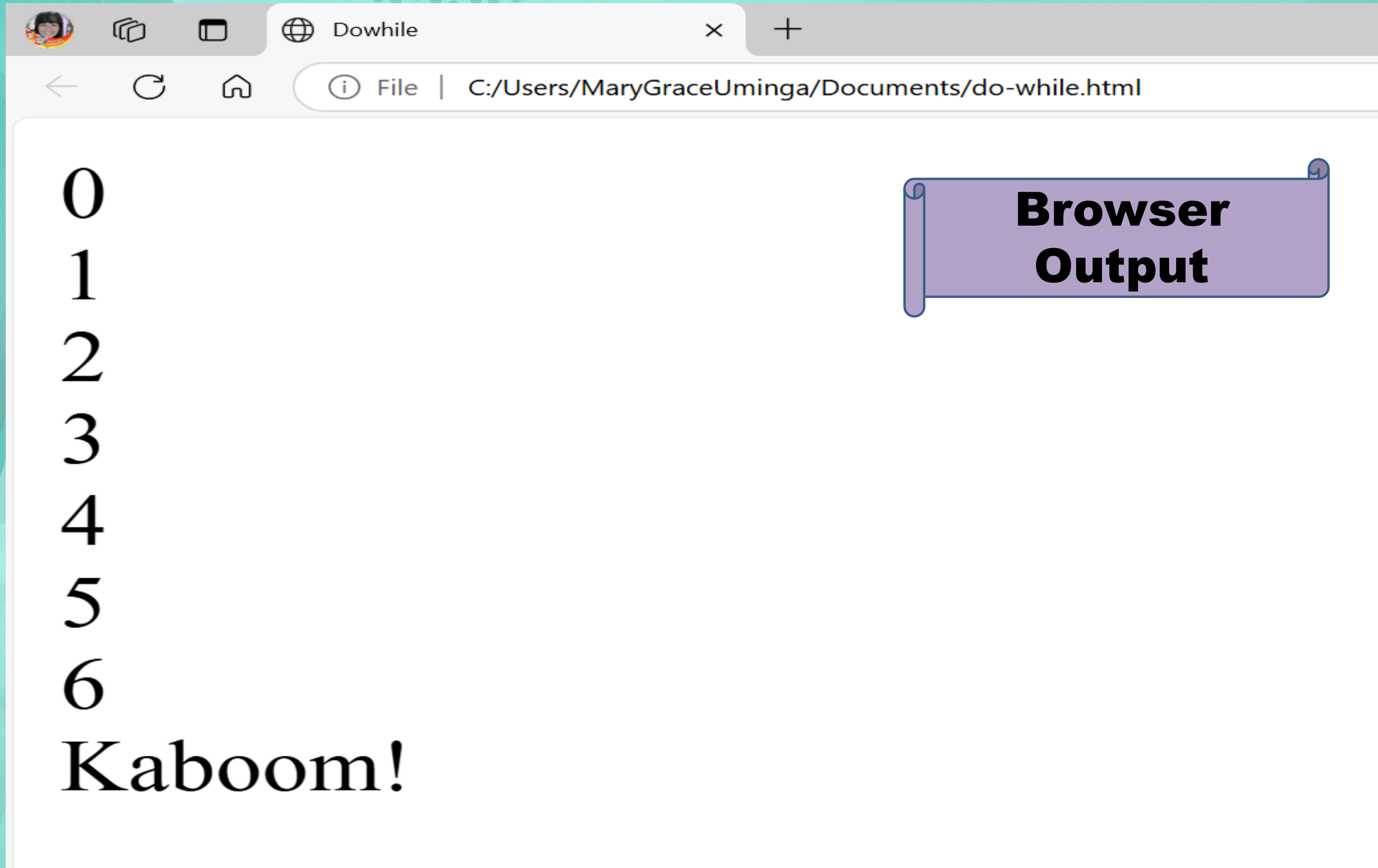
Syntax:

```
dowhile.txt
File Edit View

do
{
//Place your codes here
}
while (condition)|
```

Sample Script

```
<!DOCTYPE html>
<html>
<head><title>Dowhile</title></head>
<body>
<script type="text/javascript">
//5 sec countdown for time bomb
var x=0;
do
{
document.write(x);
document.write("<br/>");
x++;
}|
while (x<7)
document.write("Kaboom!");
</script>
</body>
</html>
```

Do this

```
<!DOCTYPE html>
<html>
<head><title>For-in Loop Input Example</title></head>
<body>
<script type="text/javascript">
    // Ask the user how many values they want to input
    var numberOfInputs = prompt("How many values do you want to input?");
    numberOfInputs = parseInt(numberOfInputs); // Convert the input to a number

    // Initialize an empty array
    var userInputs = [];

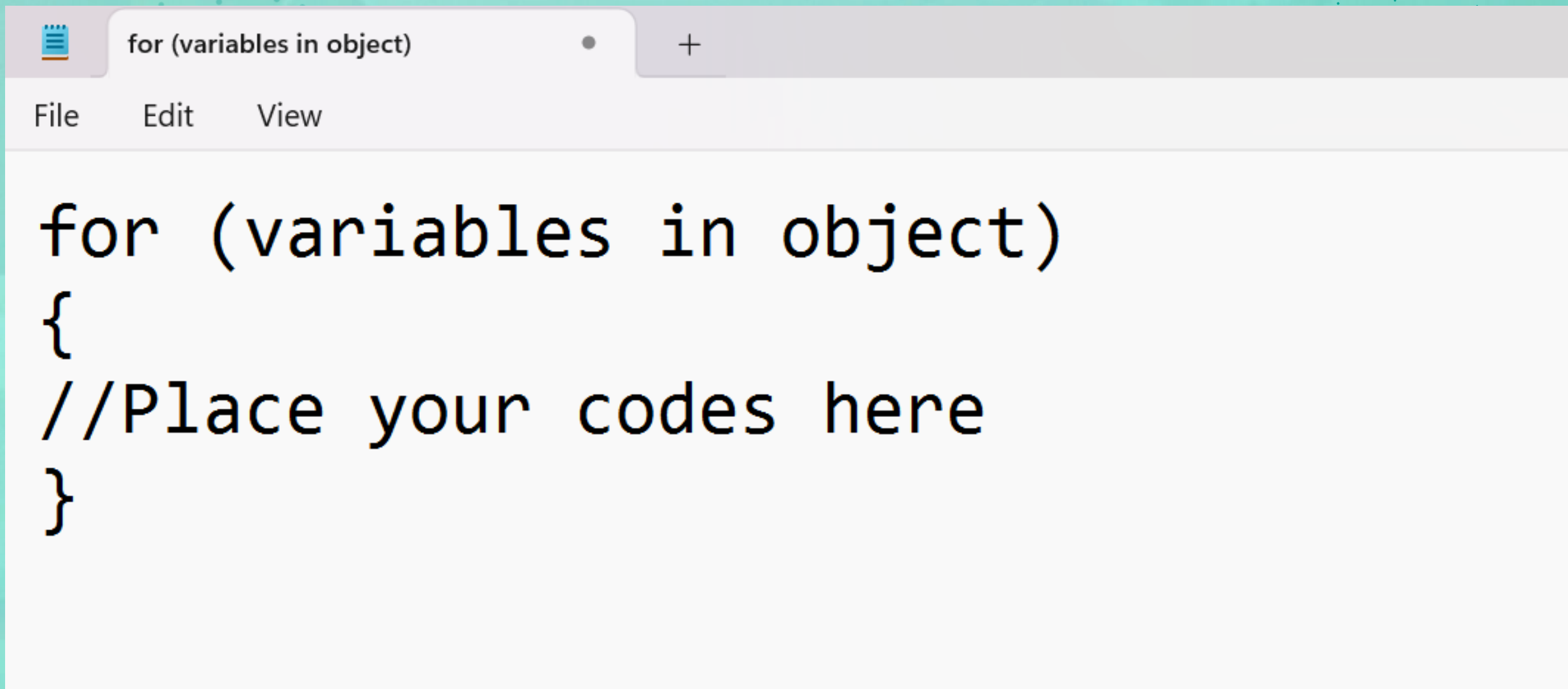
    // Use a for loop to collect the input values
    for (var i = 0; i < numberOfInputs; i++)
    {
        var input = prompt("Enter value " + (i + 1) + ":");
        userInputs[i] = input; // Store the input value in the array
    }

    // Use the for-in loop to display the collected values
    document.write("<h2>You entered the following values:</h2>");
    for (var index in userInputs)
    {
        document.write(userInputs[index] + "<br/>");
    }
</script>
</body>
</html>
```


For-in Loop

For-in loop through the elements of an array or an object.

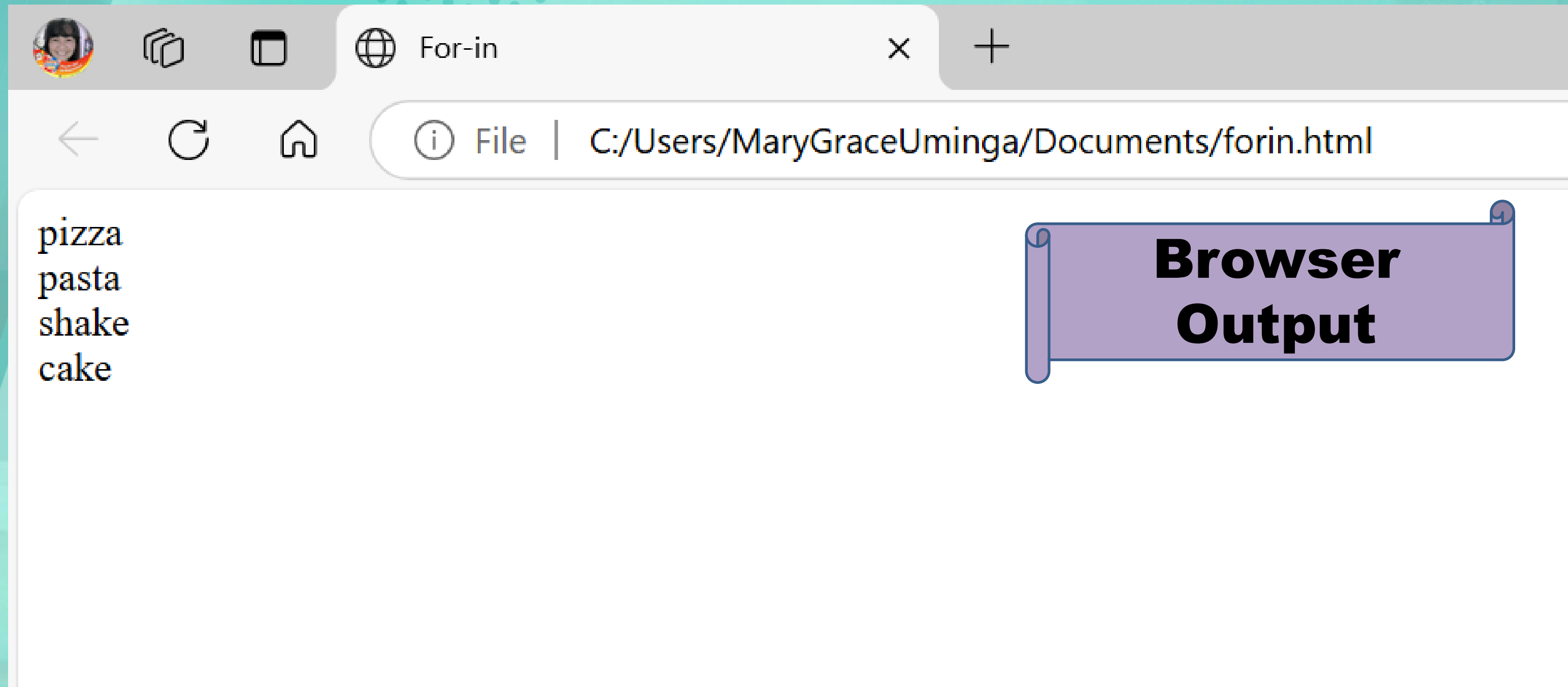
Syntax:

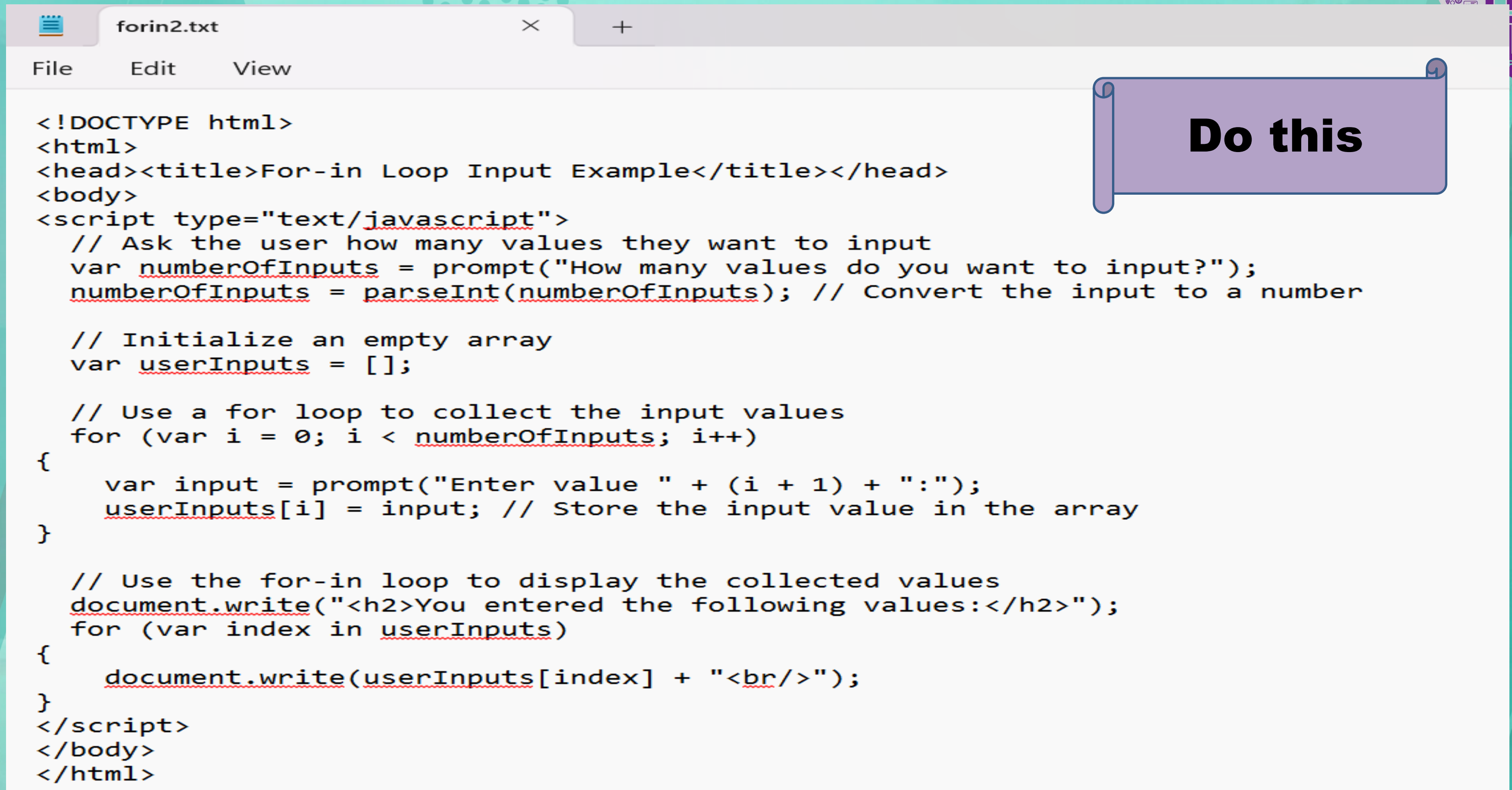


```
for (variables in object)
{
//Place your codes here
}
```

Sample Script

```
<!DOCTYPE html>
<html>
<head><title>For-in</title></head>
<body>
<script type="text/javascript">
var x;
var myFood=new Array();
myFood[0]="pizza";
myFood[1]="pasta";
myFood[2]="shake";
myFood[3]="cake";
for (x in myFood)
{
document.write(myFood[x]+"<br/>");
}
</script>
</body>
</html>
```



The image shows a web browser window with a single tab titled 'forin2.txt'. The browser's address bar is empty. The page content is a mix of HTML and JavaScript code. A purple callout box with a scroll effect on the right side of the page contains the text 'Do this'.

```
<!DOCTYPE html>
<html>
<head><title>For-in Loop Input Example</title></head>
<body>
<script type="text/javascript">
    // Ask the user how many values they want to input
    var numberOfInputs = prompt("How many values do you want to input?");
    numberOfInputs = parseInt(numberOfInputs); // Convert the input to a number

    // Initialize an empty array
    var userInputs = [];

    // Use a for loop to collect the input values
    for (var i = 0; i < numberOfInputs; i++)
    {
        var input = prompt("Enter value " + (i + 1) + ":");
        userInputs[i] = input; // Store the input value in the array
    }

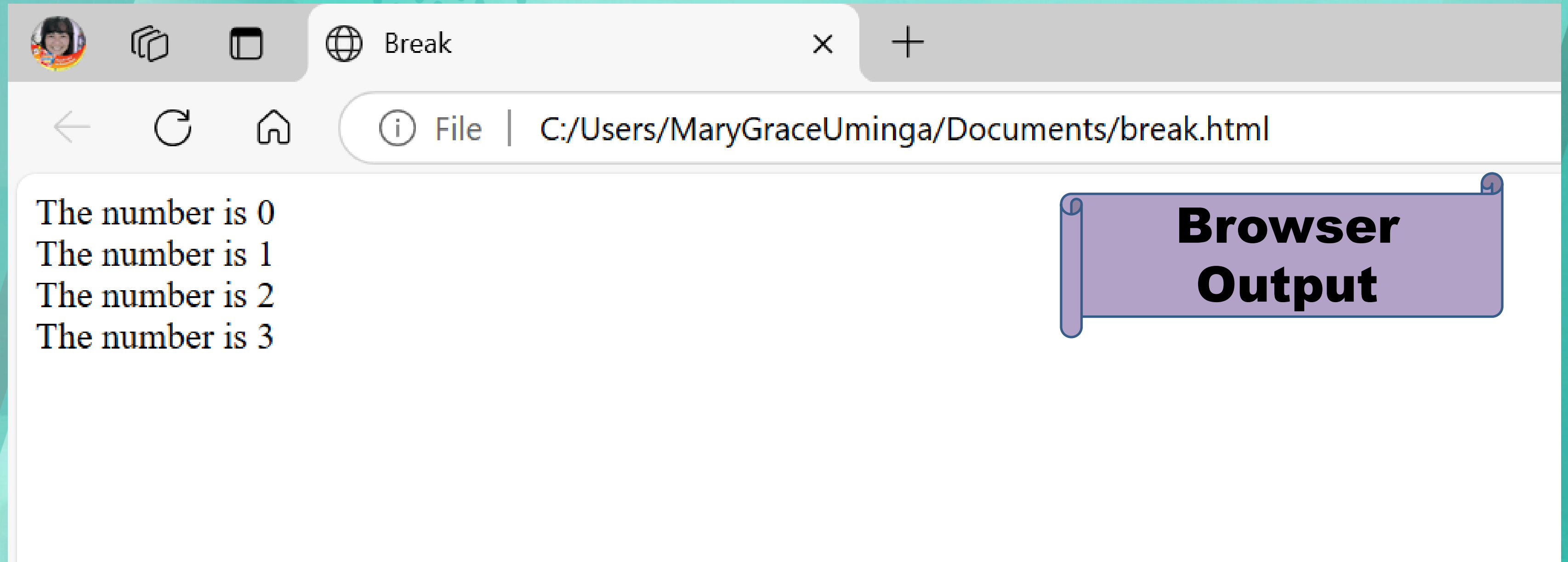
    // Use the for-in loop to display the collected values
    document.write("<h2>You entered the following values:</h2>");
    for (var index in userInputs)
    {
        document.write(userInputs[index] + "<br/>");
    }
</script>
</body>
</html>
```


Break and Continue statements

The break statement will stop or break the loop once the specified condition becomes true and then execute the next code if any after the loop.

Sample Script

```
<!DOCTYPE html>
<html>
<head><title>Break</title></head>
<body>
<script type="text/javascript">
var i=0;
for (i=0;i<=7;i++)
{
if (i==4)
{
break;
}
document.write("The number is"+ i);
document.write("<br/>");
}
</script>
</body>
</html>
```

Continue statement

The continue statement will stop or break the loop for a set value then continue to loop with the next value.

Sample Script

```
<!DOCTYPE html>
<html>
<head><title>Break</title></head>
<body>
<script type="text/javascript">
var i=0;
for (i=0;i<=12;i++)
{
if (i==5)
{
continue;
}
document.write("The number is "+ i);
document.write("<br/>");
}
</script>
</body>
</html>
```