WESTCLIFF UNIVERSITY
Educate. Inspire. Empower.

Week 3: JavaScript Foundations

Agenda

- What are loops, its construct and use
- What are arrays, its construct and use
- The purpose of popup boxes, its construct and use
- Test codes using the browser console
- In-Class Demo
- Homework

Loops

What are Loops

- Loops are used in JavaScript to perform repeated tasks based on a condition that is still *true*.
- Conditions, as we already know, returns *true* or *false* when evaluated.
- A loop will continue to run until the condition returns *false*.
- There are 3 common type of loops:
  1. for
  2. while
  3. do while

How Loops are used

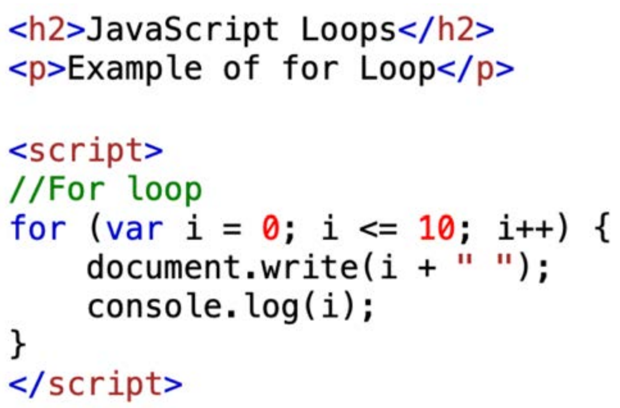
- Let’s look at the basic syntax for first type of loop: for loop

```
for (initialization; condition; counter) {
    // code block
}
```

- Breakdown:
  - *initialization*: declare the counter variable and its starting value
  - *condition*: the expression that evaluates to true or false
  - *counter*: increment or decrement of the count
- Common use:
  - count through a finite number of steps or tasks

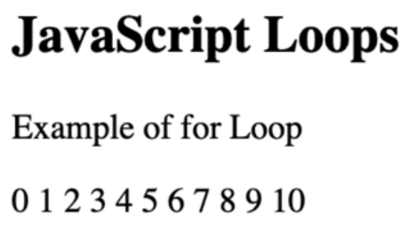
How Loops are used

- Example of for loop: say we want to iterate through integers from 0-10.



```
<h2>JavaScript Loops</h2>
<p>Example of for Loop</p>

<script>
//For loop
for (var i = 0; i <= 10; i++) {
    document.write(i + " ");
    console.log(i);
}
</script>
```

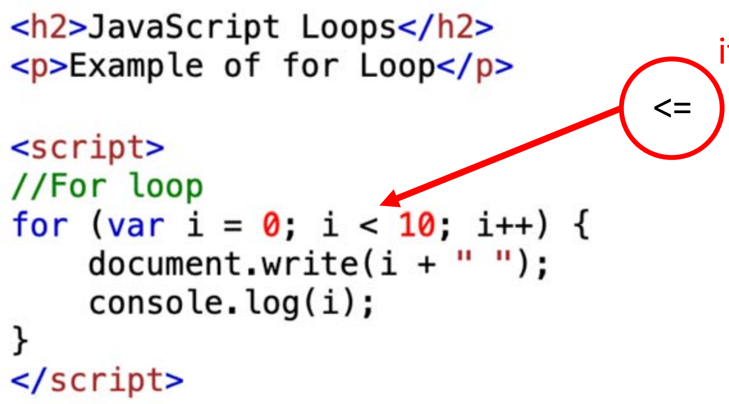
What it looks like when view on the browser:



```
JavaScript Loops

Example of for Loop

0 1 2 3 4 5 6 7 8 9 10
```

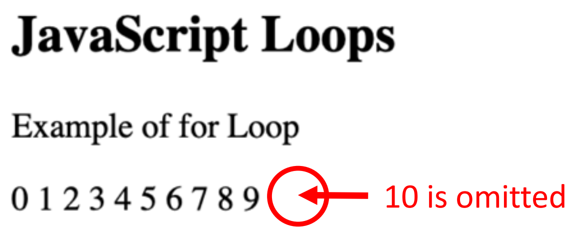
How Loops are used

- Be careful of using the incorrect operators in the condition:



```
<h2>JavaScript Loops</h2>
<p>Example of for Loop</p>

<script>
//For loop
for (var i = 0; i < 10; i++) {
    document.write(i + " ");
    console.log(i);
}
</script>
```

What it looks like when view on the browser:



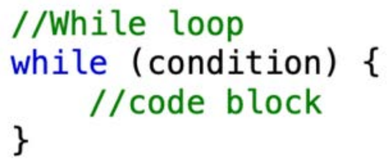
JavaScript Loops

Example of for Loop

0 1 2 3 4 5 6 7 8 9

How Loops are used

- Let’s look at the basic syntax of while loop

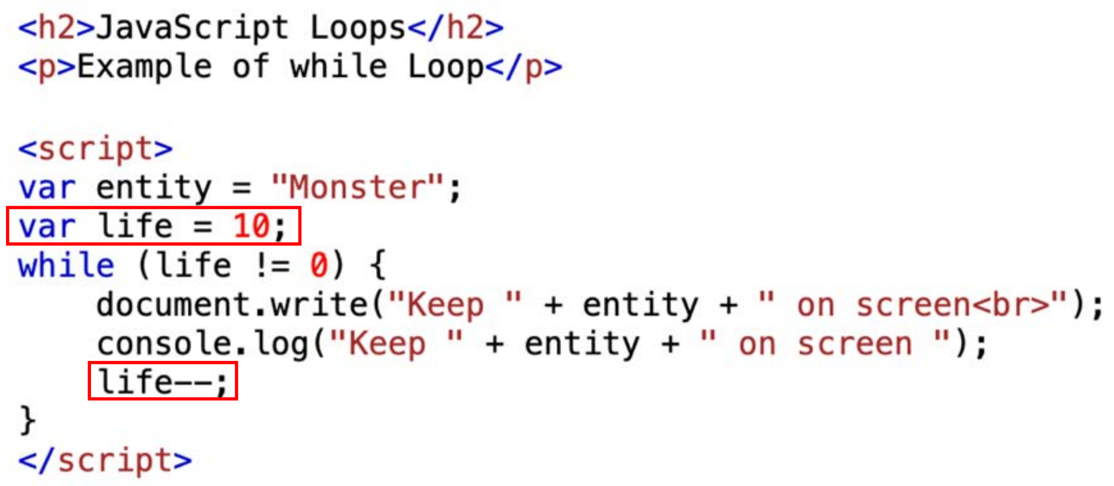


```
//While loop
while (condition) {
    //code block
}
```

- Explain:
    - while loop continues to run as long as the condition is *true*
    - loop ends if condition evaluates to *false*
- Common use:
    - to determine when a task(s) should continue or ends depending on the condition.

How Loops are used

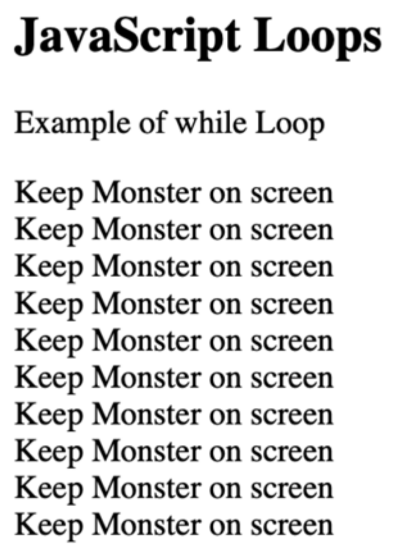
- Example of while loop: say you want to write a program for a game – keep monster on screen if its still alive.



```
<h2>JavaScript Loops</h2>
<p>Example of while Loop</p>

<script>
var entity = "Monster";
var life = 10;
while (life != 0) {
    document.write("Keep " + entity + " on screen<br>");
    console.log("Keep " + entity + " on screen ");
    life--;
}
</script>
```

You still see a counter being used. If not, the while loop will run infinitely!

What it looks like when view on the browser:



JavaScript Loops

Example of while Loop

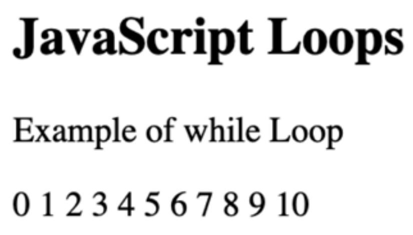
```
Keep Monster on screen
Keep Monster on screen
Keep Monster on screen
Keep Monster on screen
Keep Monster on screen
Keep Monster on screen
Keep Monster on screen
Keep Monster on screen
Keep Monster on screen
Keep Monster on screen
```

How Loops are used

- While while loops are not designed to iterate through a range of numbers like for loop, it can be used to do the same.

```
<h2>JavaScript Loops</h2>
<p>Example of while Loop</p>

<script>
var i = 0;
while (i <= 10) {
    document.write(i + " ");
    console.log(i + " ");
    i++;
}
</script>
```

What it looks like when view on the browser:



JavaScript Loops

Example of while Loop

```
0 1 2 3 4 5 6 7 8 9 10
```

How Loops are used

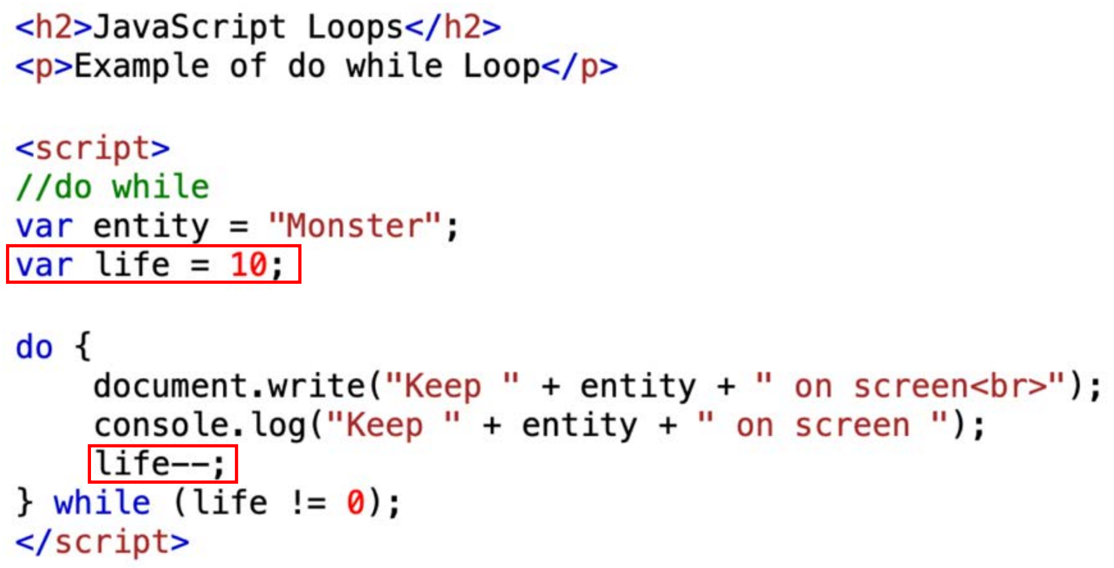
- Let’s look at the basic syntax of do while loop

```
//do while
do {
    //code block
} while (condition)
```

- Explain:
    - do while loop executes the code block at least once and continues to run as long as the condition is *true*
    - loop ends if condition evaluates to *false*
- Common use:
    - When a task is required to be executed once before the conditions are evaluated.

How Loops are used

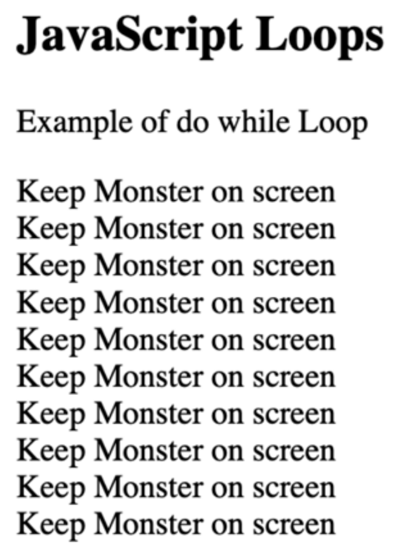
- Example of do while loop: using the same game program – keep monster on screen if its still alive.



```
<h2>JavaScript Loops</h2>
<p>Example of do while Loop</p>

<script>
//do while
var entity = "Monster";
var life = 10;

do {
    document.write("Keep " + entity + " on screen<br>");
    console.log("Keep " + entity + " on screen ");
    life--;
} while (life != 0);
</script>
```

As in the while loop, you also see a counter being used here. If not, the do while loop will run infinitely!

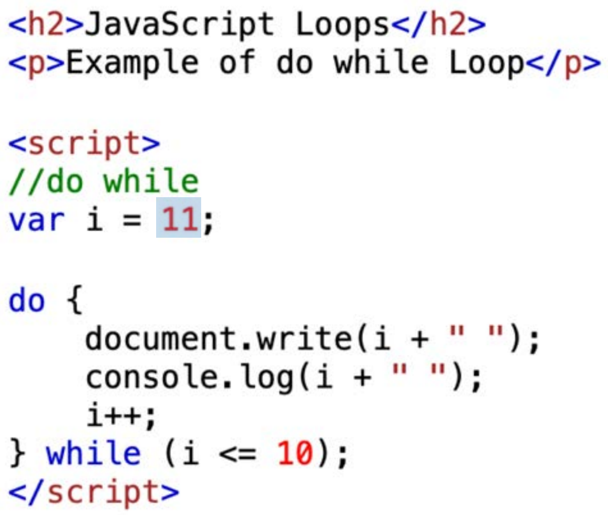
What it looks like when view on the browser:



```
JavaScript Loops

Example of do while Loop

Keep Monster on screen
Keep Monster on screen
Keep Monster on screen
Keep Monster on screen
Keep Monster on screen
Keep Monster on screen
Keep Monster on screen
Keep Monster on screen
Keep Monster on screen
Keep Monster on screen
```

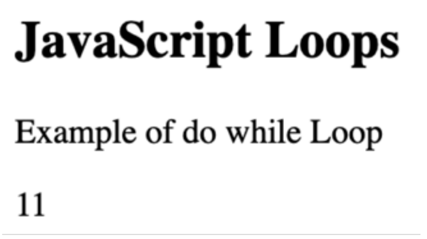
How Loops are used

- In this do while loop example, it shows how the loop will execute the code block at least once even the condition is false at the start of the program.



```
<h2>JavaScript Loops</h2>
<p>Example of do while Loop</p>

<script>
//do while
var i = 11;

do {
    document.write(i + " ");
    console.log(i + " ");
    i++;
} while (i <= 10);
</script>
```

What it looks like when view on the browser:



```
JavaScript Loops

Example of do while Loop

11
```

Arrays

What are Arrays

- In JavaScript, array is a special kind of variable.
- Unlike a regular variable which can only store one single data, array can store multiple data, like a list of values.
- To access the data in the array, call by referencing the single variable name and also the index number of the data’s position within the array.
- There are many uses of arrays. Being able to store multiple data, merge with other arrays and manipulation of data are some of key features.

How Arrays are used

- The basic syntax of array:

This declare/create a new empty array named arrayName

```
//Array
var arrayName = [];
```

This declare/create a new array named arrayName that contains a list of data

```
//Array
var arrayName = [value1, value2, value3, ....];
```

or

```
//Array
var arrayName = new Array(value1, value2, value3, ....);
```

Examples:

```
//Array
var fruits = ["apples", "oranges", "pears"];
```

```
//Array
var scores = [78, 85, 91];
```

How Arrays are used

- Accessing Array Data:
  - Data stored in an array are often accessed to be used in the program.
  - By calling the name of the array followed by a pair of square brackets:

    ```
    arrayName[];
    ```

  - Within the brackets, specify the index number of the data's position.
  - Each data in an array has a position number, starting at 0. For example, the first data's position is 0, second data's position is 1, and so on.
  - Therefore to access a data, write for example: `arrayName[2];`
  - Hence, in our fruits array example, to access "oranges", write `fruits[1];`
  - Or to access the first score in the scores array, write `scores[0];`
  - Or to access all the array data, write `fruits;`

How Arrays are used

- Array properties
  - The length property: use this property to find out the number of data in the array.
  - Example: `fruits.length;` - results in 3 because there 3 fruit items in the array
  - So therefore, to access the last fruit in the array, you may also write this:

```
fruits.length-1;
```

How Arrays are used

- Array Methods:
  - These are common methods in JavaScript array to manipulate its data:

• push()	- adds a new item at the end of the array list
• pop()	- removes the last item of the array list, returns the item removed
• shift()	- same as pop but the first item of the array list
• unshift()	- same as push but at the beginning of the array list
• splice()	- inserts new item to and also removes item on the array list (specify position to insert, how many to remove, new items)
• slice()	- slice part of array list (specify index position to start slicing)
• sort()	- sorts an array list alphabetically
• reverse()	- reverse the order of array list
• concat()	- merging arrays

How Arrays are used

- Examples of array methods
  - We will use the fruits array:

```
//Array
var fruits = ["apples", "oranges", "pears"];
```

Code: / **View on the browser:**

- push()

```
fruits.push("grapes");
document.write(fruits);
```

```
apples,oranges,pears,grapes
```

- pop()

```
fruits.pop();
document.write(fruits);
```

```
apples,oranges,pears
```

- shift()

```
fruits.shift();
document.write(fruits);
```

```
oranges,pears
```

- unshift()

```
fruits.unshift("apples");
document.write(fruits);
```

```
apples,oranges,pears
```

How Arrays are used

- Examples of array methods (con’t)
  - We will use the fruits array:

```
//Array
var fruits = ["apples", "oranges", "pears"];
```

	Code:	View on the browser:

• splice()	`fruits.splice(1,0,"grapes","kiwi");`
	`document.write(fruits);`
		apples,grapes,kiwi,oranges,pears

• slice()	`var favFruits = fruits.slice(2);`
	`document.write(favFruits);`
		kiwi,oranges,pears

• sort()	`fruits.sort();`
	`document.write(fruits);`
		apples,grapes,kiwi,oranges,pears

• reverse()	`fruits.reverse();`
	`document.write(fruits);`
		pears,oranges,kiwi,grapes,apples

How Arrays are used

- Examples of array methods (con’t)
  - We will use 2 new arrays:

```
//Arrays
var beenThereList = ["New York City","London","Rome"];
var bucketList = ["Shanghai","Santiago"];
```

  - concat() **Code:**

```
var myList = beenThereList.concat(bucketList);
document.write(myList);
```

View on the browser:

```
New York City,London,Rome,Shanghai,Santiago
```

Popup Boxes

JavaScript Popup Boxes

- **What is….**
  - Popups are common features a web user would have encounter one way or another when visiting a website or apps. It’s basically a floating small window that appears usually at the top center of the browser or center on the computer screen. Users sometimes are requested to enter information or/and click a button within the window to dismiss it.
- **Purpose**:
  - To display a simple message, request user's confirmation or take a user's input value.

JavaScript Popup Boxes

- **Popup Types**

  - JavaScript provides different built-in functions to display popup messages for different purposes, and they are:

  - Alert Box
  - Confirm Box
  - Prompt Box

JavaScript Popup Boxes

- Alert Box
  - Use alert() function to display a popup message to the user. This popup will have **OK** button to close the popup.
  - Syntax:

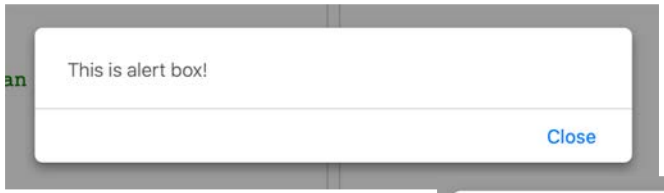
```
alert();
```

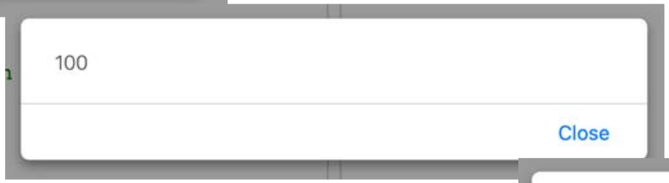
  - Examples:

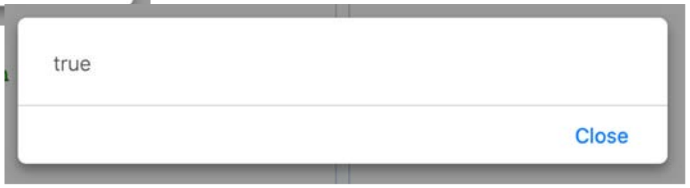
```
<h1>Demo: alert()</h1>
<script>
    alert("This is alert box!");  // display string message

    alert(100); // display number

    alert(true); // display boolean
</script>
```

JavaScript Popup Boxes

- Alert Box
  - When the page loads on the browser, a alert box will appear each time the button is click to dismiss it before the next alert box appears. The appearance of the alert boxes goes by the order they were coded starting from the top.


This is alert box!


100


true

JavaScript Popup Boxes

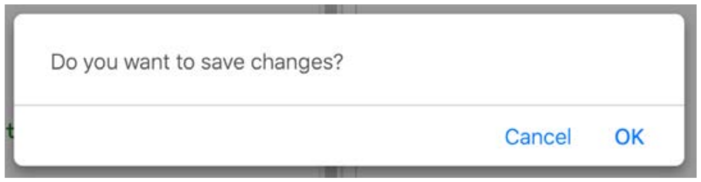
- Confirm Box
  - Use confirm() function to take the user's confirmation before allowing them to proceed. For example, saving updated data or deleting existing data. This function comes with two buttons, **OK** and **Cancel.** You can check which button the user has clicked and proceed accordingly.
  - Syntax:

```
confirm();
```

  - Example:

```
<script>
    confirm("Do you want to save changes?");
</script>
```

View on Browser:


Do you want to save changes?

JavaScript Popup Boxes

- Prompt Box
  - Use prompt() function to take the user's input to do further actions on a web page. For example, calculating interest based on users' preferred loan period. Prompt function takes two string parameters. First parameter is the message to be displayed and second parameter is the default value which will be in input text when the message is displayed. The prompt function also comes with 2 buttons, **OK** and **Cancel**.
  - Syntax:

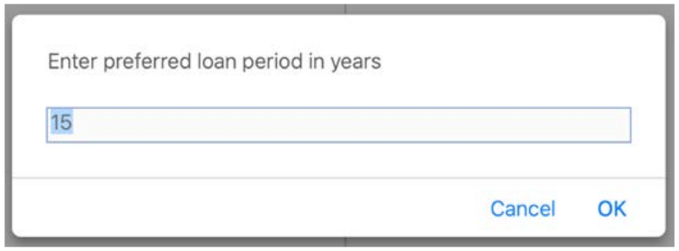
```
prompt([string message], [string defaultValue]);
```

JavaScript Popup Boxes

- Example:

```
<script>
    prompt("Enter preferred loan period in years", "15");
</script>
```

- View on Browser:



Enter preferred loan period in years

15

You can also leave the default message out:

```
<script>
    prompt("Enter preferred loan period in years");
</script>
```

Resources

https://www.geeksforgeeks.org/loops-in-javascript/

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

https://www.geeksforgeeks.org/what-are-the-types-of-popup-box-available-in-javascript/

Questions?