WESTCLIFF UNIVERSITY
Educate. Inspire. Empower.

Week 4: JavaScript Essentials

Agenda

- More Array Methods
- How to generate random numbers using the random method
- Test codes using the browser console
- In-Class Demo
- Homework

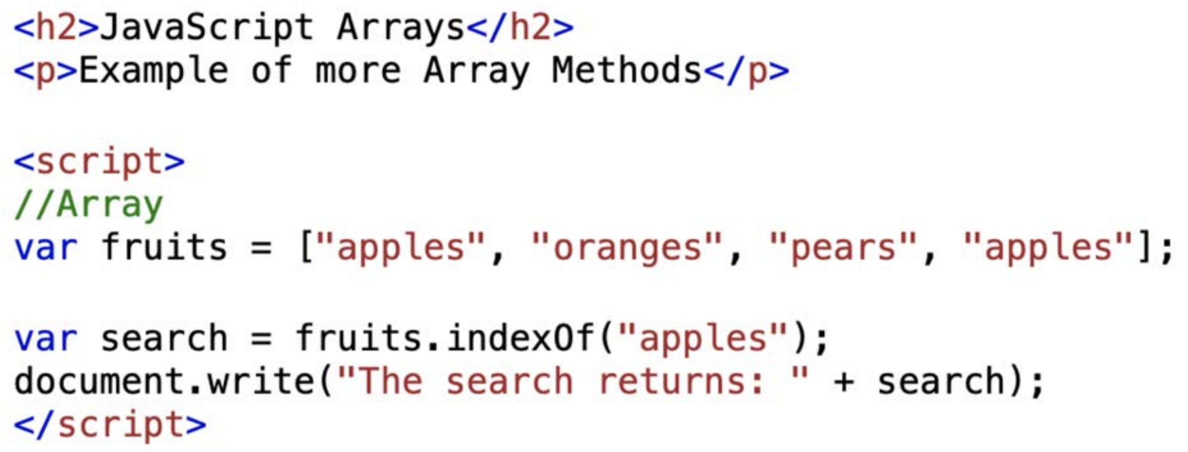
More Array Methods

Array Methods

● indexOf()	- to determine whether or not an object is in an array
● lastIndexOf()	- returns the last index at which an item is found
● forEach()	- a cleaner version of the for loop to loop through an array
● includes()	- determines if the array contains the specified item and true or false as output.
● every()	- checks every array item against a condition & returns a falsy value
● some()	- like every but the passing condition is at least one callback returns true
● map()	- loops through an array, runs a function, and create a new array built from the return values of each iteration
● filter()	- like the map but creates a new array containing only items of the original array that return a truthy value from the callback
● reduce()	- melt the items in an array down to a single value by the operations performed in its callback function

More Array Methods

- indexOf() Method
  - It returns the *first* index at which the item was found, or -1 if it was not found at all. Strict equality is used to determine that an item is present in the array.
  - Example:

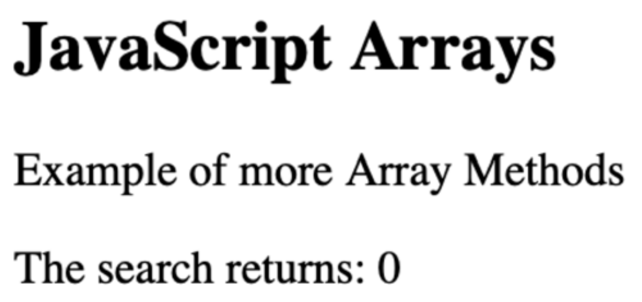


```
<h2>JavaScript Arrays</h2>
<p>Example of more Array Methods</p>

<script>
//Array
var fruits = ["apples", "oranges", "pears", "apples"];

var search = fruits.indexOf("apples");
document.write("The search returns: " + search);
</script>
```

View on Browser:



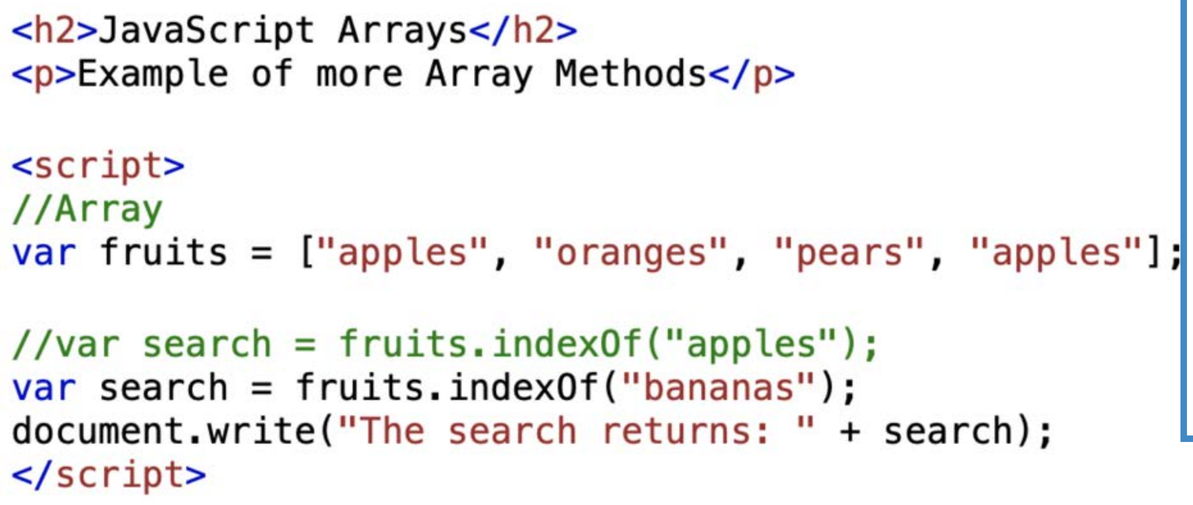
JavaScript Arrays

Example of more Array Methods

The search returns: 0

More Array Methods

- indexOf() Method
  - It returns the *first* index at which the item was found, or -1 if it was not found at all. Strict equality is used to determine that an item is present in the array.
  - Example:

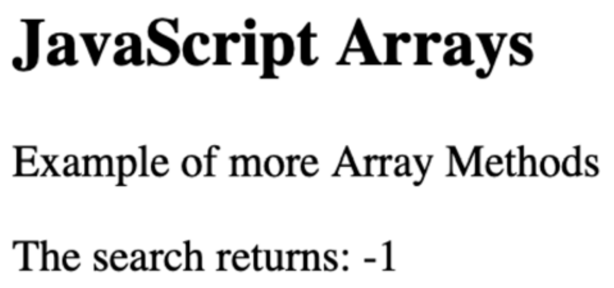


```
<h2>JavaScript Arrays</h2>
<p>Example of more Array Methods</p>

<script>
//Array
var fruits = ["apples", "oranges", "pears", "apples"];

//var search = fruits.indexOf("apples");
var search = fruits.indexOf("bananas");
document.write("The search returns: " + search);
</script>
```

View on Browser:



JavaScript Arrays

Example of more Array Methods

The search returns: -1

More Array Methods

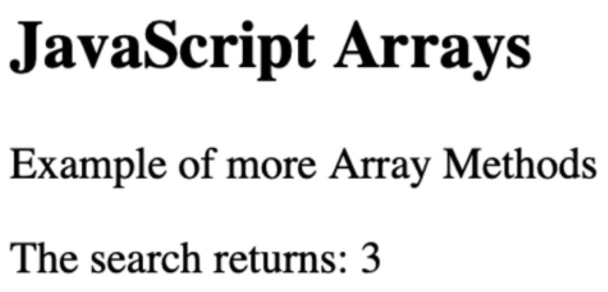
- lastIndexOf() Method
  - It returns the *last* index at which the item was found, even if an identical item was found first.
  - Example:

```
<h2>JavaScript Arrays</h2>
<p>Example of more Array Methods</p>

<script>
//Array
var fruits = ["apples", "oranges", "pears", "apples"];

var search = fruits.lastIndexOf("apples");
document.write("The search returns: " + search);
</script>
```

View on Browser:



```
JavaScript Arrays

Example of more Array Methods

The search returns: 3
```

More Array Methods

- forEach() Method
  - List each item in an array. Must call a function for each array item. Function must pass one value into it: array item. Optional second value: index of item.
  - Example:

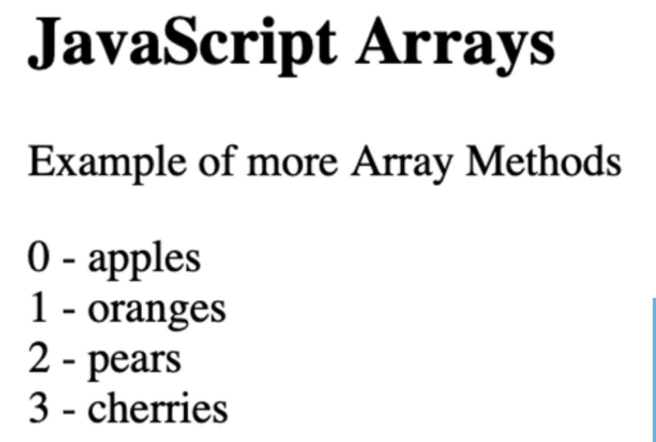
```
<h2>JavaScript Arrays</h2>
<p>Example of more Array Methods</p>

<script>
//Array
var fruits = ["apples", "oranges", "pears", "cherries"];

fruits.forEach(myFunction);

function myFunction(fruit, indexNum) {
    document.write(indexNum + " - " + fruit + "<br>");
}
</script>
```

View on Browser:



JavaScript Arrays

Example of more Array Methods

```
0 - apples
1 - oranges
2 - pears
3 - cherries
```

More Array Methods

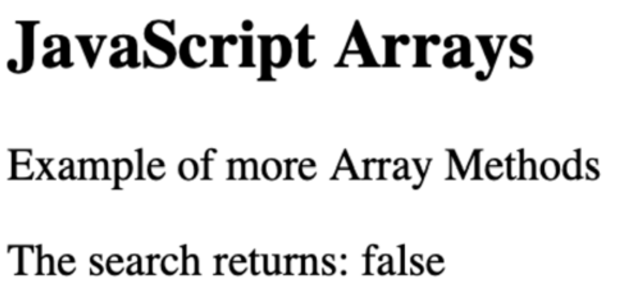
- includes() Method
  - This method determines whether the array contains the specified item. It returns **true** or **false** as output depending on the result.
  - Example:

```
<h2>JavaScript Arrays</h2>
<p>Example of more Array Methods</p>

<script>
//Array
var fruits = ["apples", "oranges", "pears", "cherries"];

var search = fruits.includes("mangoes");
document.write("The search returns: " + search);
</script>
```

View on Browser:



```
JavaScript Arrays

Example of more Array Methods

The search returns: false
```

More Array Methods

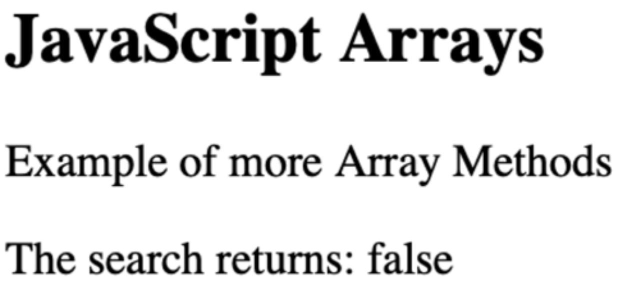
- every() Method
  - Checks if all elements in an array pass a test (provided as a function). If it finds an array element where the function returns a *false* value, every() returns *false* (and does not check the remaining values). If no false occur, every() returns *true*.
  - Example:

```
<h2>JavaScript Arrays</h2>
<p>Example of more Array Methods</p>

<script>
//Array
var fruits = ["apples", "oranges", "pears", "cherries"];

var search = fruits.every(myFunction);
function myFunction(fruit) {
    return fruit.length > 6;
}
document.write("The search returns: " + search);
</script>
```

View on Browser:



```
JavaScript Arrays

Example of more Array Methods

The search returns: false
```

More Array Methods

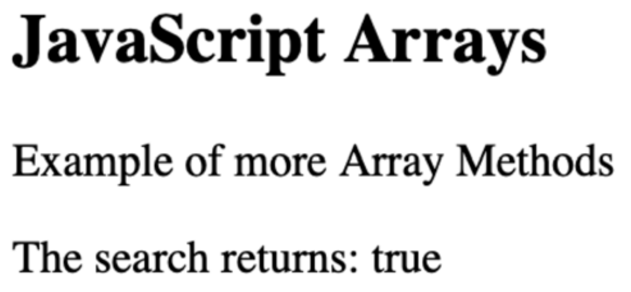
- every() Method
  - Checks if all elements in an array pass a test (provided as a function). If it finds an array element where the function returns a *false* value, every() returns *false* (and does not check the remaining values). If no false occur, every() returns *true*.
  - Example:

```
<h2>JavaScript Arrays</h2>
<p>Example of more Array Methods</p>

<script>
//Array
var fruits = ["apples", "oranges", "pears", "cherries"];

var search = fruits.every(myFunction);
function myFunction(fruit) {
    //return fruit.length > 6;
    return fruit.length > 3;
}
document.write("The search returns: " + search);
</script>
```

View on Browser:


JavaScript Arrays

Example of more Array Methods

The search returns: true

More Array Methods

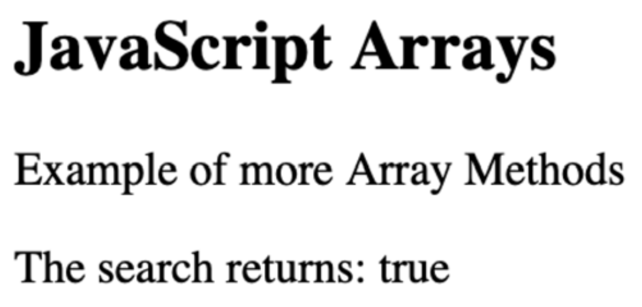
- some() Method
  - Checks if all elements in an array pass a test (provided as a function). If it finds an array element where the function returns a *true* value, some() returns *true* (and does not check the remaining values). Otherwise, it returns *false*.
  - Example:

```
<h2>JavaScript Arrays</h2>
<p>Example of more Array Methods</p>

<script>
//Array
var fruits = ["apples", "oranges", "pears", "cherries"];

var search = fruits.some(myFunction);
function myFunction(fruit) {
    return fruit.length > 6;
}
document.write("The search returns: " + search);
</script>
```

View on Browser:



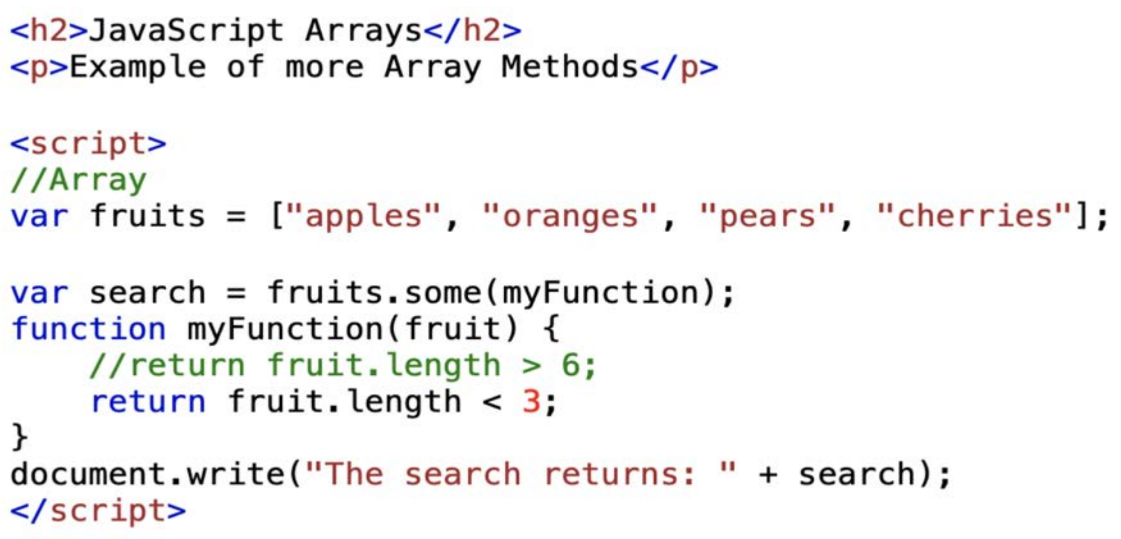
JavaScript Arrays

Example of more Array Methods

The search returns: true

More Array Methods

- some() Method
  - Checks if all elements in an array pass a test (provided as a function). If it finds an array element where the function returns a *true* value, some() returns *true* (and does not check the remaining values). Otherwise, it returns *false*.
  - Example:

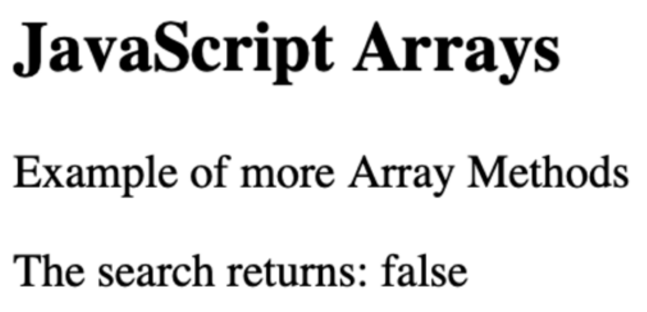


```
<h2>JavaScript Arrays</h2>
<p>Example of more Array Methods</p>

<script>
//Array
var fruits = ["apples", "oranges", "pears", "cherries"];

var search = fruits.some(myFunction);
function myFunction(fruit) {
    //return fruit.length > 6;
    return fruit.length < 3;
}
document.write("The search returns: " + search);
</script>
```

View on Browser:



JavaScript Arrays

Example of more Array Methods

The search returns: false

More Array Methods

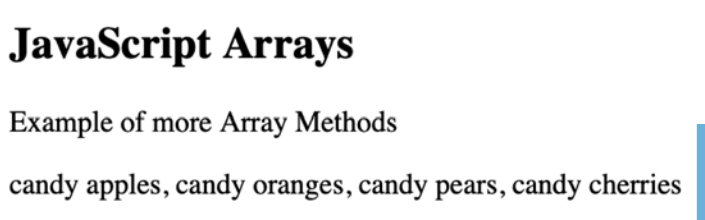
- map() Method
  - Maps each element of an existing array by calling a function for each element and assign its results in a new array. All elements in the parent array remain as it does not mutate the original array.
  - Example:

```
<h2>JavaScript Arrays</h2>
<p>Example of more Array Methods</p>

<script>
//Array
var fruits = ["apples", "oranges", "pears", "cherries"];

var candies = fruits.map(myFunction);
function myFunction(fruit) {
    return " candy " + fruit;
};
document.write(candies);
</script>
```

View on Browser:



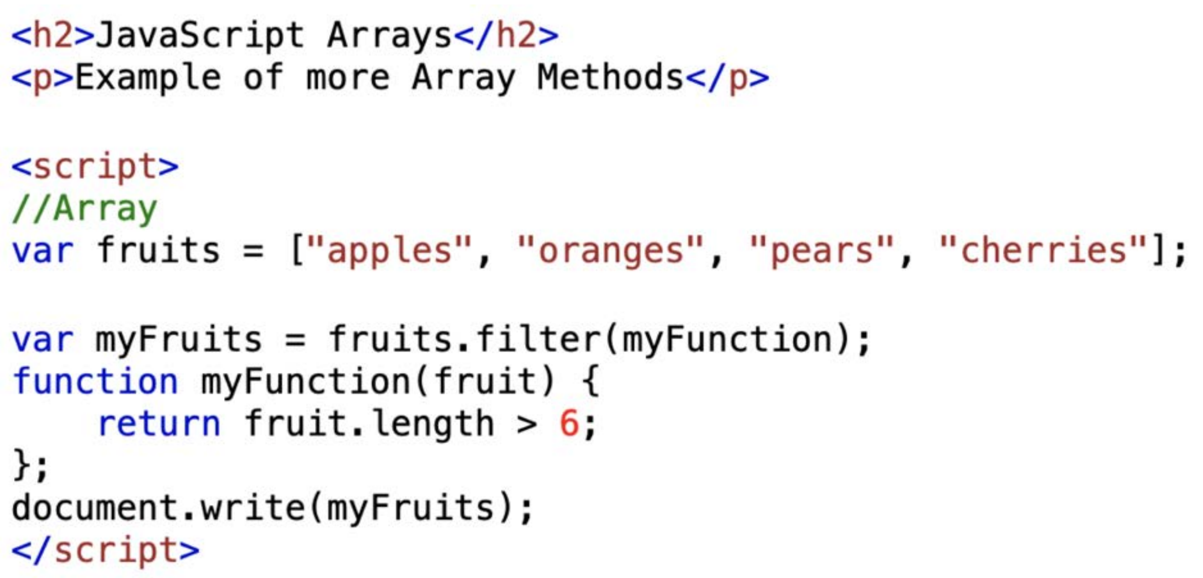
JavaScript Arrays

Example of more Array Methods

candy apples, candy oranges, candy pears, candy cherries

More Array Methods

- filter() Method
  - Creates a new array populated with elements that meet the filter criteria (provided as a function) of the parent array.
  - Example:

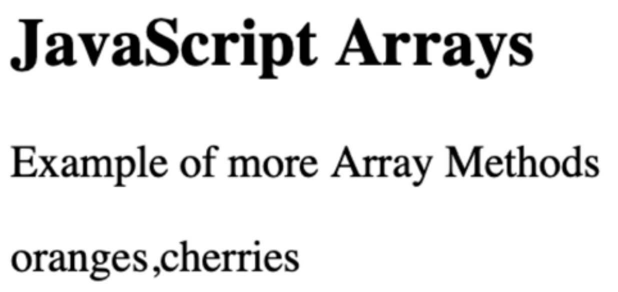


```
<h2>JavaScript Arrays</h2>
<p>Example of more Array Methods</p>

<script>
//Array
var fruits = ["apples", "oranges", "pears", "cherries"];

var myFruits = fruits.filter(myFunction);
function myFunction(fruit) {
    return fruit.length > 6;
};
document.write(myFruits);
</script>
```

View on Browser:



JavaScript Arrays

Example of more Array Methods

oranges,cherries

More Array Methods

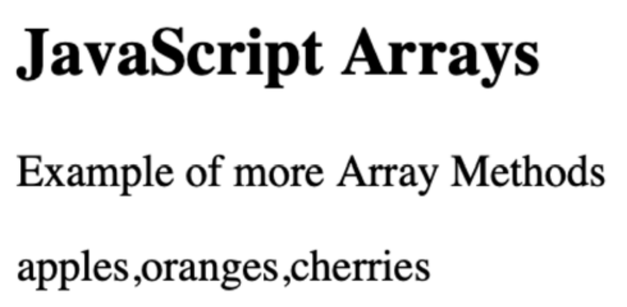
- filter() Method
  - Creates a new array populated with elements that meets the filter criteria (provided as a function) of the parent array.
  - Example:

```
<h2>JavaScript Arrays</h2>
<p>Example of more Array Methods</p>

<script>
//Array
var fruits = ["apples", "oranges", "pears", "cherries"];

var myFruits = fruits.filter(myFunction);
function myFunction(fruit) {
    //return fruit.length > 6;
    return fruit.includes("es");
};
document.write(myFruits);
</script>
```

View on Browser:



JavaScript Arrays

Example of more Array Methods

apples,oranges,cherries

More Array Methods

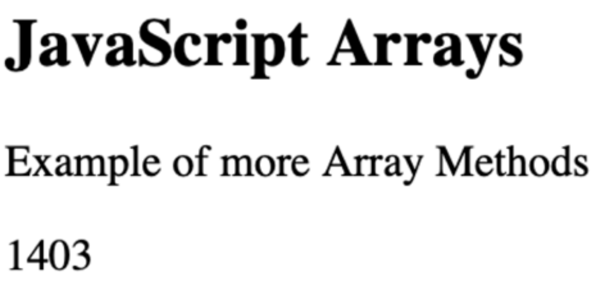
- reduce() Method
  - reduces the array to a single final value. It takes two arguments: *reducer* & *accumulator*. The *accumulator* accumulates a value based on the action (*reducer*) performs.
  - Example:

```
<h2>JavaScript Arrays</h2>
<p>Example of more Array Methods</p>

<script>
//Array
var daysales = [305, 432, 376, 290];

var weeklySales = daysales.reduce(myFunction);
function myFunction(accumTotal,curSales) {
    return accumTotal + curSales;
};
document.write(weeklySales);
</script>
```

View on Browser:



JavaScript Arrays

Example of more Array Methods

1403

Generate random numbers

- In JavaScript, it is possible to generate random numbers within a specified range of numbers so that number generated can be used within the program.
- This can be easily done using the Math.random() function.
- How to use:
  - Without a specified range of numbers:
    - Math.random() => This will generate a number from 0 (inclusive) to 1 (exclusive - up to 0.9999999999999999)
  - With a specified range of numbers:
    - Math.random()*max => This will generate a number from 0 to max
    - Math.random()*max+min => This will generate a number from min to max+min
    - Math.random()*max-min => This will generate a number from min to max-min

Will generate up to ?.9999999999999999
Example: Max=10 => 9. 9999999999999999

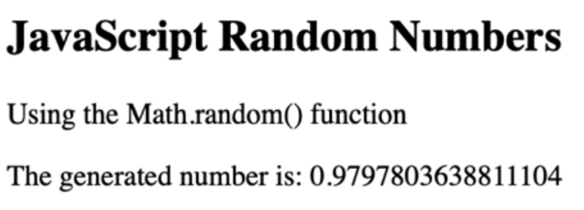
Generate random numbers

- Example 1 (without a specified range):

```
<h2>JavaScript Random Numbers</h2>
<p>Using the Math.random() function</p>

<script>
var myNumber = Math.random();
document.write("The generated number is: " + myNumber);
</script>
```

View on Browser:



JavaScript Random Numbers

Using the Math.random() function

The generated number is: 0.9797803638811104

Generate random numbers

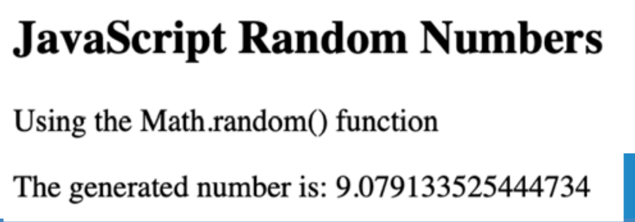
- Example 2A (with a specified range from 0 to 10):

```
<h2>JavaScript Random Numbers</h2>
<p>Using the Math.random() function</p>

<script>
var myNumber = Math.random()*10;
document.write("The generated number is: " + myNumber);
</script>
```

Note:
By default numbers are in decimal places but can be rounded down to their nearest integer using *Math.floor()*.

View on Browser:



JavaScript Random Numbers

Using the Math.random() function

The generated number is: 9.079133525444734

Generate random numbers

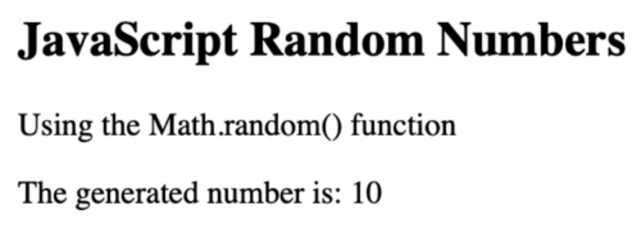
- Example 2B (with specified range: 0 to 11, round to nearest integer):

```
<h2>JavaScript Random Numbers</h2>
<p>Using the Math.random() function</p>

<script>
var myNumber = Math.floor(Math.random() * 11);
document.write("The generated number is: " + myNumber);
</script>
```

Note:
Since *Math.floor* round down (instead of up) to its nearest integer, the max number of 11 is used instead of 10.

View on Browser:



JavaScript Random Numbers

Using the Math.random() function

The generated number is: 10

Generate random numbers

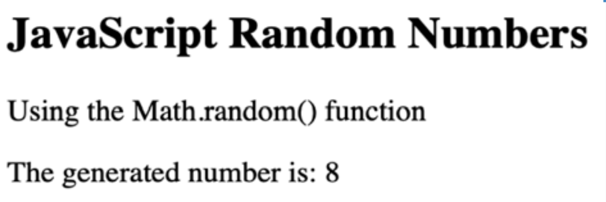
- Example 3 (with specified range: 1 to 11, round to nearest integer):

```
<h2>JavaScript Random Numbers</h2>
<p>Using the Math.random() function</p>

<script>
var myNumber = Math.floor(Math.random() * 10+1);
document.write("The generated number is: " + myNumber);
</script>
```

Note:
In this case, the range tested is actually between 1 and 11 ie. 10+1=11. But with Math.floor, it will round down to 10 if the generated number is 10.999999999999999

View on Browser:



JavaScript Random Numbers

Using the Math.random() function

The generated number is: 8

Generate random numbers

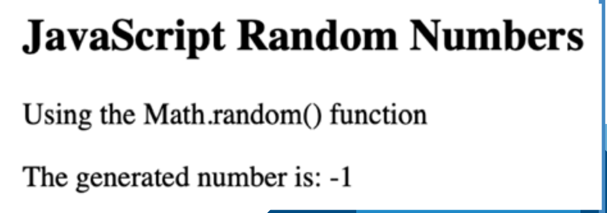
- Example 4 (with specified range: -1 to 9, round to nearest integer):

```
<h2>JavaScript Random Numbers</h2>
<p>Using the Math.random() function</p>

<script>
var myNumber = Math.floor(Math.random() * 10-1);
document.write("The generated number is: " + myNumber);
</script>
```

Note:
In this case, the range tested is actually between -1 and 9 ie. 10-1=9. But with Math.floor, it will round down to 8 if the generated number is 8.9999999999999999

View on Browser:



JavaScript Random Numbers

Using the Math.random() function

The generated number is: -1

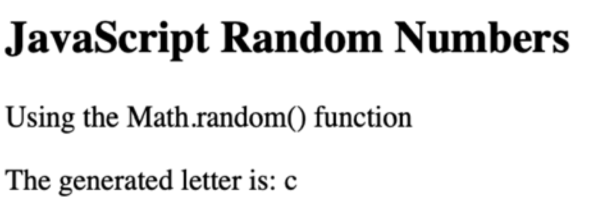
Generate random numbers

- Example:

```
<h2>JavaScript Random Numbers</h2>
<p>Using the Math.random() function</p>

<script>
var alphabet = "abcdefghijklmnopqrstuvwxyz";
var randomAlphabet = alphabet[Math.floor(Math.random() * alphabet.length)]
// example if var randomAlphabet = alphabet[12]  => this output the letter m
document.write("The generated letter is: " + randomAlphabet);
</script>
```

View on Browser:



JavaScript Random Numbers

Using the Math.random() function

The generated letter is: c

Resources

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

https://www.w3schools.com/jsref/jsref_sort.asp

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Math/random

Questions?