# 6 KEY JAVASCRIPT FEATURES

top developers learn

BitDegree

# INTRODUCTION

In this ebook, I will teach 6 key elements that you'll need while using Javascript.

Whether you're a fresh beginner and the word "string" invokes images of a guitar rather than JavaScript in your head, or you're a veteran of Java - these key elements are mandatory knowledge for everyone using this programming language. Even the experts could use a handy cheat sheet every once in a while!

So then, without further ado, let's jump right into it!

## TABLE OF CONTENTS

| | |
|---|---|
| 1. Onclick Events | 4 |
| 2. Array Sort Method | 6 |
| 3. Onload Event | 8 |
| 4. Input Text Value Property | 10 |
| 5. String Replace() Method | 12 |
| 6. String Split Method | 14 |

# 1 ONCLICK EVENTS

An onclick event occurs when the user clicks on a certain element. When clicked on an HTML object, it runs a specified line of code that has the “onclick” attribute. The event can be triggered by commands like “object.onclick” or “object.addEventListener”. One more thing - it is worth mentioning that the “addEventListener” command is not supported by earlier versions of Internet Explorer (8 and below).

## SYNTAX

Now that we’ve gotten the basics out of the way, this is how a JavaScript syntax looks like:

## EXAMPLES

The example below shows the current date when clicked on the text:

```
<button onclick=”getElementById(‘test’).innerHTML =
 Date()”>This will show date when clicked.</button>
```

This example changes the text color when clicked:

```
function myFunction() {
    document.getElementById(“test”).style.color = “red”;
}
```

This example also changes the text color, but it does so using a slightly different method:

```
function myFunction(elmnt,clr) {
   elmnt.style.color = clr;
}
```

The example below copies a text on click:

```
function myFunction() {
  docu lementById(“field1”).value;
}
```

This example changes the color of the background when clicked:

```
window.onclick = myFunction;
function myFunction() {
  document.getElementsByTagName(“BODY”)[0].style.
backgroundColor = “lime”;
}
```

## 2 ARRAY SORT METHOD

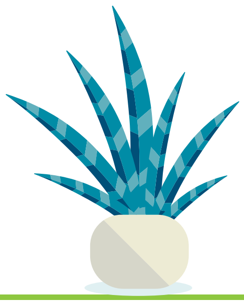
This method is used to sort array items. The order in which items are sorted may be either alphabetic or ascending, and descending or ascending.

By default, this method will sort the items as strings in an ascending and alphabetical order.

For string containing nothing but text, this works well ("Alex" goes before "Ben"). But sorting numbers as strings can be problematic.
For example, "32" is seen as bigger than "123", since "3", the first character in this number string, is bigger than "1".

For this reason, sort() will generally produce incorrect results when dealing with numbers that have more than one digit.

This can be fixed by providing a "compare function" (look into the parameter values).



## SYNTAX

The syntax of an “array sort” command looks like this:

**array.sort(compareFunction)**

## EXAMPLES

```
var sampleA2rray = ['Mercedes-Benz', 'Nissan', 'Audi', 'BMW']
var cars = sampleArray.sort();
 // contents of sortedArray now look like this: ['Audi', 'BMW',
'Mercedes-Benz', 'Nissan']
```

### JavaScript Array Sort Method Parameter Values and Technical Details

**Parameter:** compareFunction

**Parameter description:** Optional. A function used to provide an alternative way to sort the array. This function is supposed to return zero, a negative, or positive value, which depends on the arguments, for example:

**function(a, b){return a-b}**

Upon comparing the two values, the sort() method will send the values to the compare function, and sort the values based on the returned (zero, negative, positive) value.

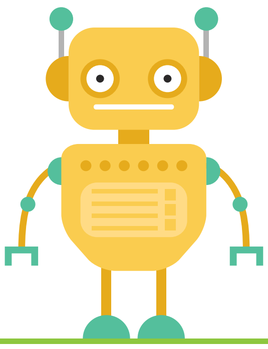
**Example:** When 52 and 102 are compared, the sort() method will call the compare function(52, 102).

The function will calculate 52-102, and return -50 (a negative numeric value).

Thus, sort() now sorts 52 as a number lower than 102.

**Return value:** The original array object with its items sorted according to the parameters.

**JavaScript version:** ECMAScript1



## 3 ONLOAD EVENT

The onload event transpires when an object is loaded. This event is most frequently used within the <body> element to perform a script once a website has fully loaded all content (including script fileas, images, CSS files, etc.).

The onload event can also be used to verify the visitor’s type of **browser** and **version** of the browser, and load the individual web page version based on the information. It can also deal with cookies.

Here is an example of an onload event:

```
<body onload=”sampleFunction()”>
```

## SYNTAX

In HTML:

```
<element onload="sampleScript"></element>
```

In JavaScript:

```
object.onload = function  () {"myScript"};
```

In JavaScript, with the “addEventListener()” method:

```
object.addEventListener("load", sampleScript);
```

## EXAMPLES

In this example, an <img> element is signed onload. Alert “Image is successfully loaded” is displayed after an image is loaded:

```
<img src="https://cdn.rawgit.com/bitdegree/banners/84706be6/
learn/pom-laptop.png?raw=true" onload="ImageLoad()"
width="120" height="120">
```

In this example, we deal with cookies by using the onload event:

```
<body onload="CookiesCheck()">
```

## 4 INPUT TEXT VALUE PROPERTY

This property is used to set or return the value of a text input field. The value property contains either the default value that is present upon loading the element OR the value entered by the user OR the value assigned by the script.

**Example:**

```
document.getElementById(“sampleText”).value = “John Bonham”;
```

**Returning the value of the property:**
textObject.value

**Setting the value of the property:**
textObject.value = text

**JavaScript Input Text Value Property Values and Technical Details:**

**Value:** Description
**Text:** Specify the value of the text input field
**Return value:** A String, which represents the value of the text field

## EXAMPLES

```
var sampleList = document.getElementById("sampleList");
document.getElementById("favoriteElem").value = sampleList.
  options[sampleList.selectedIndex].text;
```

```
var none = document.getElementById("none");
var option = none.options[none.selectedIndex].text;
var sampleText = document.getElementById("resultElem").value;
sampleText = sampleText + option;
document.getElementById("resultElem").value = txt;
```

## 5 STRING REPLACE() METHOD

The replace() method searches a string for a regular expression for a specified value and returns a string with replaced specified values. Use the global (g) modifier to replace all value occurrences. The original string is left not changed by u sing this method.

**Example:**

```
var string = “This is bitdegree.org!”;
var result = string.replace(“bitdegree.org”, “Doggo”);
```

**Result:**

*This is Doggo!*

### SYNTAX

The JavaScript String replace() Method syntax looks like this:

string.replace(searchvalue, newvalue)

### PARAMETER VALUES

**Parameter:** Description

**Searchvalue:** Needed. The regular expression or value, that will be replaced

**Newvalue:** Needed. Replace the search value

## EXAMPLES

**In the example below, we complete a global replacement:**

```
function LearnFunction() {
   var string = document.getElementById("learn").innerHTML;
   var result = string.replace(/white/g, "black");
   document.getElementById("learn").innerHTML = result;
}
```

**Result:**

*Ms White has a black bed and a black table.*

**In this example, we complete a case-insensitive, global replacement:**

```
function LearnFunction() {
    var string = document.getElementById("learn").innerHTML;
    var result = string.replace(/white/gi, "black");
    document.getElementById("learn").innerHTML = result;
}
```

**Result:**

*Ms black has a black bed and a black table.*

**In this example, we use a function for returning the replaced text:**

```
function LearnFunction() {
    var string = document.getElementById("learn").innerHTML;
    var result = string.replace(/white|bed|table/gi, function
  myFunction(z){return z.toUpperCase();});
    document.getElementById("learn").innerHTML = result;
}
```

**Result:**

*Ms BLACK has a BLACK bed and a BLACK table.*

# 6 STRING SPLIT METHOD

The split() string method splits a string into a array of substrings and returns the new array. The string is split between every character if the separator is (“”). The original string is left unchanged after this method.

**Example:**

```
var string = “Whats your name?”;
var result = string.split(“ “);
```

**Result array:**

*Whats,your,name?*

## SYNTAX

**string.split(separator, limit)**

## PARAMETER VALUES

**Parameter:** Description

**Separator:** Not required. Specifies the regular expression of the character, for splitting the string. The entire string will be returned if omitted

**Limit:** Not required. An integer that specifies the splits number, after this limit items will not be included

## EXAMPLES

**In this example, separator operator is omitted:**

```
function LearnFunction() {
    var string = "Whats your name?";
    var result = string.split();
    document.getElementById("learn").innerHTML = result;
}
```

**Result array:**

*Whats your name?*

**In the example below, every symbol, even blank spaces os separated:**

```
function LearnFunction() {
    var string = "Whats your name?";
    var result = string.split("");
    document.getElementById("learn").innerHTML = result;
}
```

**Result array:**

*W,h,a,t,s, ,y,o,u,r, ,n,a,m,e,?*

**In this example, array is limited for 2 values:**

```
function LearnFunction() {
    var string = "Whats your name?";
    var res = string.split(" ", 2);
    document.getElementById("learn").innerHTML = res;
}
```

**Result array:**

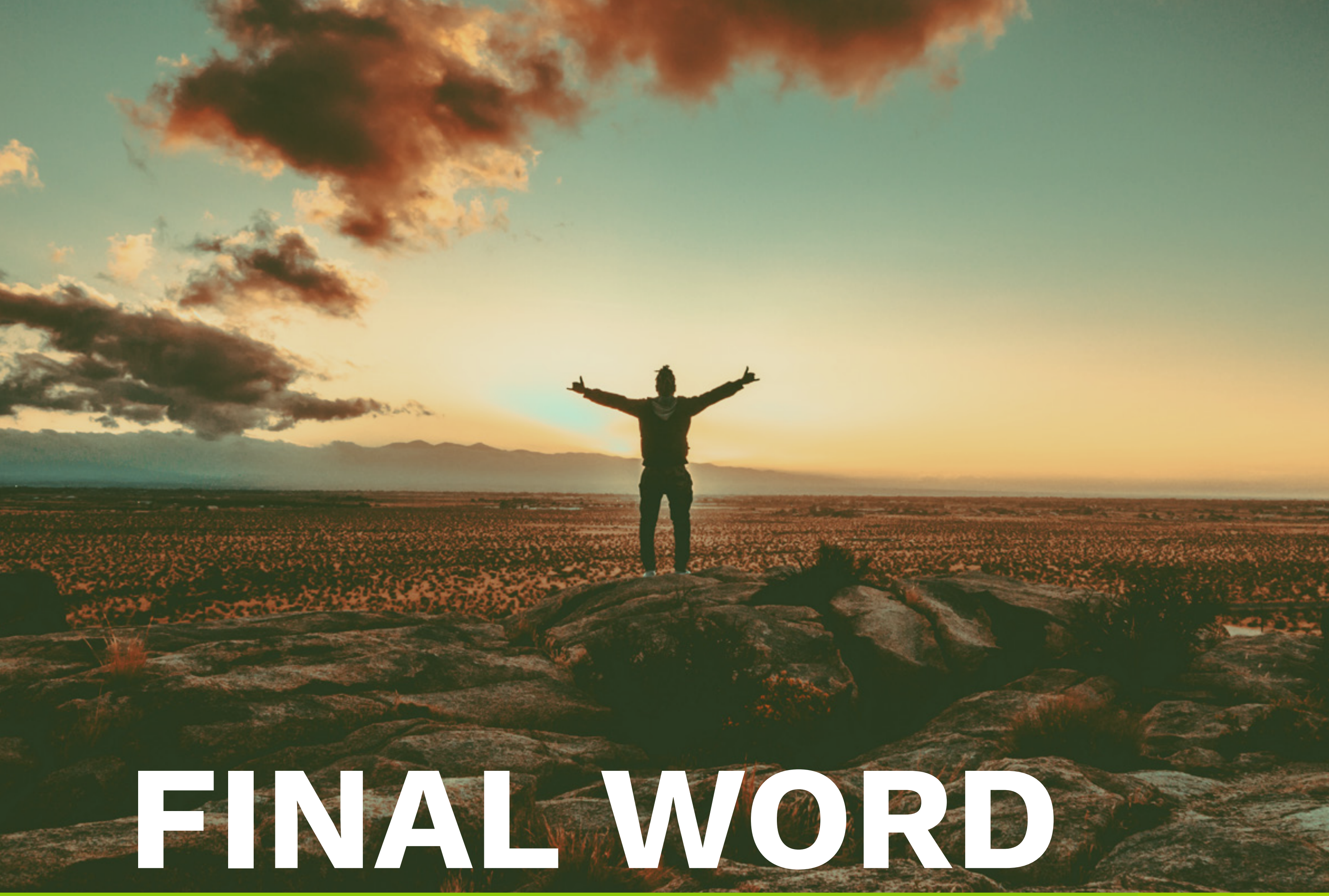
*Whats,your*

**In this example, letter “a” is used to separate the string:**

```
function LearnFunction() {
    var string = "Whats your name?";
    var result = string.split("a");
    document.getElementById("learn").innerHTML = result;
}
```

**Result array:**

*Wh,ts your n,me?*



# FINAL WORD

So, that's it - these were the 6 key features to learn for JavaScript! If you followed along - awesome, hope this ebook was helpful! Do yourself a favor and always keep this ebook close, so when your brain suddenly stops working the next time around, you could easily cheat sheet your way out of that situation!

*The strength of JavaScript is that you can do anything. The weakness is that you will.*

- Reg Braithwaite