JavaScript Basics For Absolute Beginners

Mohammed Abdulhady

April 14, 2017

Contents

Chapter 1

Introduction

1.1 History of JavaScript

JavaScript was created in 1995 by Brendan Eich who worked at Netscape. JavaScript was originally named Mocha by Marc Andreessen, the founder of Netscape. The name was changed to LiveScript and then to JavaScript. It was some kind of a marketing move because of the popularity of Java at that time.

1.2 What is JavaScript?

JavaScript is an object-oriented programming language that is used to make web pages interactive and creating web applications. JavaScript is an interpreted language, it needs an interpreter to translate the codes and executes the program. The interpreter, which is the browser in our case, executes instructions directly line by line without previously compiling it into machine language.

Many people confused between Java (developed in the 1990s at Sun Microsystems) and JavaScript. They are completely different. Only the names are similar.

1.3 Why JavaScript

JavaScript is a very simple, powerful and popular programming language. It has become an essential technology along with HTML and CSS.
You will need to learn JavaScript if you want to get into web development, if you are planning to be a front-end or you can even use it for back-end development.
As o this moment, JavaScript has extended to desktop apps development, mobile apps development and games development. It has become a very important and useful skill to master.

Chapter 2

Numbers

2.1 What Are Numbers?

Numbers are the values that you can use mathematical operations on. There is not any special syntax for numbers in JavaScript. You can just write them straight into JavaScript.

Example

```
123456;
```

2.2 Decimals and fractions

Numbers in JavaScript can be written with, or without, decimals. Unlike any other programming language, JavaScript does not distinguish between numbers. It has only one type of numbers. So, you can use whole number and decimals together without worrying or having to convert from one to another.

Example

```
1 + 2.345;
```

Output

```
3.345
```

2.3 Negative Numbers

If you want to make any number negative, you can do that by placing the `-` operator in front of it.

Example

```
-100;
```

Output

```
-100
```

Also, you can get a negative number by subtracting a big number from a smaller one.

Example

```
3 - 100;
```

Output

```
-97
```

2.4 Arithmetic Operators

Operators are those symbols that you use between two or more values to perform different operations such as addition, subtraction and more.
In this book, we will focus on the ones that we think you will see most often.

2.4.1 Addition

the `+` operator is used to add two or more numbers.

Example

```
3 + 2 + 1;
```

Output

```
6
```

2.4.2 Subtraction

the `-` operator is used to subtract numbers from each others.

Example

```
3 - 2 - 1;
```

Output

```
0
```

2.4.3 Multiplication

the `*` operator is used to multiply two or more numbers. notice that we use `*` instead of `×` symbol that is commonly used in math.

Example

```
3 * 2;
```

Output

```
6
```

2.4.4 Division

the `/` operator is used to divide numbers by numbers. notice that we use `/` instead of `÷` symbol that is commonly used in math.

Example

```
6 / 2;
```

Output

```
3
```

2.4.5 Modulus

the `%` operator is used to get the division remainder of two numbers.

Example

```
5 % 2;
```

Output

```
1
```

2.4.6 Increment

the `++` operator is used to increment the number by 1.

Example

```
5++;
5+1;
```

Output

```
6
6
```

2.4.7 Decrement

the `--` operator is used to decrement the number by 1.

Example

```
5--;
5-1;
```

Output

```
4
4
```

2.5 Ordering and Grouping

JavaScript expressions follow the **order of operations** , so even if the `+` and `-` operators come first in the following example, the `*` and `/` operators will be performed first between the second and third number.

Example

```
5 + 10 * 2;
5 - 10 / 2;
```

Output

```
25
0
```

However, you can have more control over the order of operations using the grouping operator `()`.
Using these operators to group other values and operations will force JavaScript to perform the grouped operations first, no matter what the ordering is.

Example

```
(5 + 10) * 2;
(5 - 10) / 2;
```

Output

```
30
-2.5
```

Chapter 3

Strings

3.1 What Are Strings ?

Strings are values made up with series of characters. They can be made of letters, symbols, numbers or anything.

A JavaScript String must be contained inside a pair of double quotation `" "` or ever single quotation `' '`.

Example

```
'This is our string.';
"This is our string.";
```

3.2 Enclosing quotation marks

Let us say you want to use the quotation marks inside your string. You can not do this normally like any other character, because quotation marks are one of the **special characters** family.
To solve that problem, you have to easy options.
First option, you will need to use opposite quotation marks inside and outside. This means for each string that include single quotes needs to use double quotes and vice versa.

Example

```
"I've eaten an apple.";
'I said "I have eaten an apple"';
```

Second option, you can use a backslash \ right before each quote inside the string. This makes the JavaScript know that you want to use a **special characters**.

Example

```
'I\'ve eaten an apple.';
"I said \"I have eaten an apple\"";
```

3.3 Methods and properties

Strings in JavaScript have their own predefined list of operations you can perform to string. They are call "methods and properties".
Here are some of the most helpful and commonly used :-

3.3.1 Length

The string length is a property that returns the number of characters that the string has.

Example

```
"What is my length?".length;
```

Output

```
18
```

3.3.2 toLowerCase

The String toLowerCase() Method returns a copy of the string but with all capital letters converted to small letters.

Example

```
"I aM A StRiNg !!".toLowerCase();
```

Output

```
"i am a string !!"
```

3.3.3 toUpperCase

The String toUpperCase() Method returns a copy of the string but with all small letters converted to capital letters.

Example

```
"I aM A StRiNg !!".toUpperCase();
```

Output

```
"I AM A STRING !!"
```

3.3.4 Trim

The trim() method cuts off any whitespaces at the begin and end of the string.

Example

```
"     Trim me, please  ".trim();
```

Output

```
"Trim me, please"
```

3.3.5 Replace

The String replace() method replaces a specified value with another value in a string and returns the result.

Example

```
"I love red oranges".replace("oranges", "apples");
```

Output

```
"I love red apples"
```

Chapter 4

Booleans

Booleans are the values that can only be one of two things: either `true` or `false`.
It's really useful to store booleans in **variables** to keep track of their value and change them as you like.

Example

```
var flag = true;

flag = false;

flag;
```

Output

```
false
```

Booleans are also essential for **conditional** work. We will discuss that later.