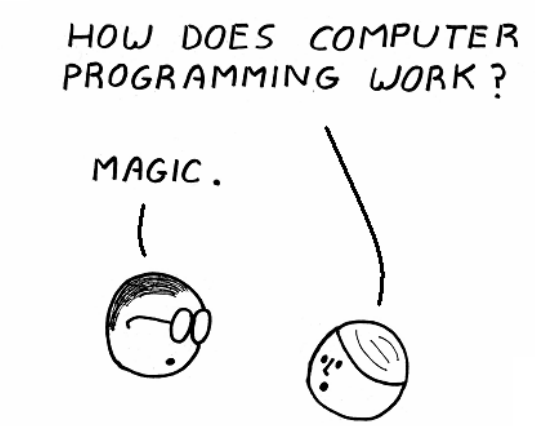
Introduction

This book will teach you the basics of programming and JavaScript. Whether you are an experienced programmer or not, this book is intended for everyone who wishes to learn the JavaScript programming language.



JavaScript (*JS for short*) is the programming language that is used to create dynamic interaction while developing webpages, games, applications, and even servers.JavaScript started at Netscape, a web browser developed in the 1990s, and is today one of the most famous and used programming languages.

Initially, it was created for making webpages alive and was able to run on a browser only. Now, it runs on any device that supports the JavaScript engine. Standard objects such as `Array`, `Date`, and `Math` are available in JavaScript, as well as operators, control structures, and statements. *Client-side JavaScript* and *Server-side JavaScript* are the extended versions of Core JavaScript.

- *Client-side JavaScript* enables the enhancement and manipulation of web pages and client browsers. Responses to user events such as mouse clicks, form input, and page navigation are some of its examples.
- *Server-side JavaScript* enables access to servers, databases, and file systems.

JavaScript is an interpreted language. While running Javascript an interpreter interprets each line and runs it. The modern browser uses Just In Time (JIT) technology for compilation, which compiles JavaScript into executable bytecode.

> "LiveScript" was the initial name given to JavaScript.

Basics

In this first chapter, we'll learn the basics of programming and the Javascript language.

Programming means writing code. A book is made up of chapters, paragraphs, sentences, phrases, words, and finally punctuation and letters, likewise a program can be broken down into smaller and smaller components. For now, the most important is a statement. A statement is analogous to a sentence in a book. On its own, it has structure and purpose, but without the context of the other statements around it, it isn't that meaningful.

A statement is more casually (and commonly) known as a *line of code*. That's because statements tend to be written on individual lines. As such, programs are read from top to bottom, left to right. You might be wondering what code (also called source code) is. That happens to be a broad term which can refer to the whole of the program or the smallest part. Therefore, a line of code is simply a line of your program.

Here is a simple example:

```
let hello = "Hello";
let world = "World";

// Message equals "Hello World"
let message = hello + " " + world;
```

This code can be executed by another program called an *interpreter* that will read the code, and execute all the statements in the right order.

Comments

Comments are statements that will not be executed by the interpreter, comments are used to mark annotations for other programmers or small descriptions of what code does, thus making it easier for others to understand what your code does.

In JavaScript, comments can be written in 2 different ways:

- Line starting with `//`:

```
// This is a comment, it will be ignored by the interpreter
let a = "this is a variable defined in a statement";
```

- Section of code starting with `/*` and ending with `*/`, this method is used for multi-line comments:

```
/*
This is a multi-line comment,
it will be ignored by the interpreter
*/
let a = "this is a variable defined in a statement";
```

Variables

The first step towards really understanding programming is looking back at algebra. If you remember it from school, algebra starts with writing terms such as the following.

```
3 + 5 = 8
```

You start performing calculations when you introduce an unknown, for example, x below:

```
3 + x = 8
```

Shifting those around you can determine x:

```
x = 8 - 3
-> x = 5
```

When you introduce more than one you make your terms more flexible - you are using variables:

```
x + y = 8
```

You can change the values of x and y and the formula can still be true:

```
x = 4
y = 4
```

or

```
x = 3
y = 5
```

The same is true for programming languages. In programming, variables are containers for values that change. Variables can hold all kinds of values and also the results of computations. Variables have a name and a value separated by an equals sign (=). Variable names can be any letter or word but bear in mind that there are restrictions from language to language of what you can use, as some words are reserved for other functionality.

Let's check out how it works in Javascript, The following code defines two variables, computes the result of adding the two, and defines this result as a value of a third variable.

```
let x = 5;
let y = 6;
let result = x + y;
```

ES6 Version

ECMAScript 2015 or ES2015 also known as E6 is a significant update to the JavaScript programming language since 2009. In ES6 we have three ways of declaring variables.

```
var x = 5;
const y = 'Test';
let z = true;
```

The types of declaration depend upon the scope. Unlike the `var` keyword, which defines a variable globally or locally to an entire function regardless of block scope, `let` allows you to declare variables that are limited in scope to the block, statement, or expression in which they are used. For example.

```
function varTest(){
    var x=1;
    if(true){
        var x=2; // same variable
        console.log(x); //2
    }
    console.log(x); //2
}

function letTest(){
    let x=1;
    if(true){
        let x=2;
        console.log(x); // 2
    }
    console.log(x); // 1
}
```

`const` variables are immutable - they are not allowed to be re-assigned.

```
const x = "hi!";
x = "bye"; // this will occurs an error
```