

Web Fundamentals (CSS): Intro to JS

Submitted By:

Karen C. Escabarte

B.S. Computer Engineering/ OJT Trainee

January 10, 2022

Intro to JS

Estimated Time: 1.5 - 3 hours

TIME DURATION: 1hr

ASSIGNMENT LINK: <https://www.hackerhero.com/web-fundamentals/intro-to-js>

GOOGLE DOCS LINK:  **ESCABARTE - Intro to JS**

Objectives:

- To practice how to model the blocks in given image
- To apply basics properties
- To avoid advance positioning concepts

Clone the image below using only what you've learned about HTML and CSS. Use display: inline-block/inline/block to do all of your assignments. For a better view, you can download the image here: [JavaScript Tutorial](#)

You know, when beginning, it's better to type the code rather than a copy and paste or use shortcut tools that auto-generate. Typing manually helps your brain to remember more of what you code.

Before you submit your work:

- Remember that you're here to learn: Don't try to find the original source code of these images.
- Plan first which parts are Headings, Paragraphs, Links, etc. Use proper HTML tags based on the sole purpose.
- Observe proper indentation not only on HTML but also your CSS codes.
- Validate your HTML if it fits to the best practices.
- Do NOT make this responsive. Just make this work for a fixed width (set the width to be 970px). The responsive design course is added at the next chapter of the Web Fundamentals track, and you can do this later in your training.

Chapter

An Introduction

Lesson navigation

What is JavaScript?

What can in-browser JavaScript do?

What CAN'T in-browser JavaScript do?

What makes JavaScript unique?

Languages "over" JavaScript

Summary

Comments

An Introduction to JavaScript

Let's see what's so special about JavaScript, what we can achieve with it, and what other technologies play well with it.

What is JavaScript?

JavaScript was initially created to "make web pages alive".

The programs in this language are called **scripts**. They can be written right in a web page's HTML and run automatically as the page loads.

Scripts are provided and executed as plain text. They don't need special preparation or compilation to run.

In this aspect, JavaScript is very different from another language called [Java](#).

When JavaScript was created, it initially had another name: "LiveScript". But Java was very popular at that time, so it was decided that positioning a new language as a "younger brother" of Java would help.

But as it evolved, JavaScript became a fully independent language with its own specification called [ECMAScript](#), and now it has no relation to Java at all.

Today, JavaScript can execute not only in the browser, but also on the server, or actually on any device that has a special program called [the JavaScript engine](#).

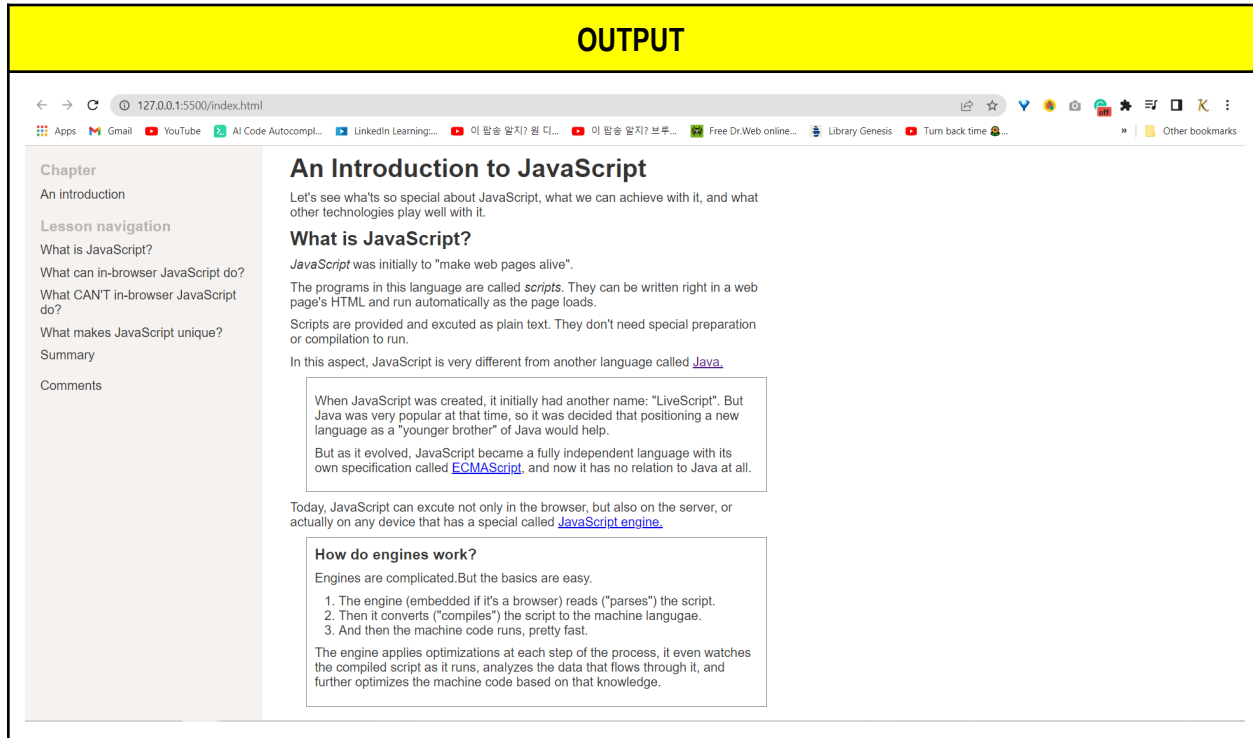
How do engines work?

Engines are complicated. But the basics are easy.

1. The engine (embedded if it's a browser) reads ("parses") the script.
2. Then it converts ("compiles") the script to the machine language.
3. And then the machine code runs, pretty fast.

The engine applies optimizations at each step of the process. It even watches the compiled script as it runs, analyzes the data that flows through it, and further optimizes the machine code based on that knowledge.

OUTPUT



HTML CODE

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <link rel="stylesheet" href="style.css" />
    <title>Intro to JS</title>
  </head>
  <body>
    <div class="container">
      <div class="navbar">
        <div class="chapter">
          <h3>Chapter</h3>
          <p>An introduction</p>
        </div>
        <div class="lesson-nav">
          <h3>Lesson navigation</h3>
          <ul>
            <li>What is JavaScript?</li>
            <li>What can in-browser JavaScript do?</li>
            <li>What CAN'T in-browser JavaScript do?</li>
            <li>What makes JavaScript unique?</li>
            <li>Summary</li>
          </ul>
        </div>
        <p class="comments">Comments</p>
      </div>
    </div>
  </body>
</html>
```

```

<div class="main">
  <div class="wrapper">
    <div id="introduction">
      <h1>An Introduction to JavaScript</h1>
      <p>
        Let's see what's so special about JavaScript, what we can achieve
        with it, and what other technologies play well with it.
      </p>
    </div>

    <div id="content">
      <h2>What is JavaScript?</h2>
      <p><em>JavaScript</em> was initially to "make web pages alive".</p>
      <p>
        The programs in this language are called <em>scripts</em>. They
        can be written right in a web page's HTML and run automatically as
        the page loads.
      </p>
      <p>
        Scripts are provided and executed as plain text. They don't need
        special preparation or compilation to run.
      </p>
      <p>
        In this aspect, JavaScript is very different from another language
        called <a href="https://javascript.info/">Java.</a>
      </p>
      <div class="box1">
        <p>
          When JavaScript was created, it initially had another name:
          "LiveScript". But Java was very popular at that time, so it was
          decided that positioning a new language as a "younger brother"
          of Java would help.
        </p>
        <p>
          But as it evolved, JavaScript became a fully independent
          language with its own specification called
          <a href="#">ECMAScript</a>, and now it has no relation to Java
          at all.
        </p>
      </div>
      <p>
        Today, JavaScript can execute not only in the browser, but also on
        the server, or actually on any device that has a special called
        <a href="#">JavaScript engine.</a>
      </p>
      <div class="box2">
        <h3>How do engines work?</h3>
        <p>Engines are complicated. But the basics are easy.</p>
        <ol>
          <li>
            The engine (embedded if it's a browser) reads ("parses") the
            script.
          </li>
          <li>
            Then it converts ("compiles") the script to the machine
            language.
          </li>
          <li>And then the machine code runs, pretty fast.</li>
        </ol>
      </div>
    </div>
  </div>
</div>

```

```

</ol>
<p>
    The engine applies optimizations at each step of the process, it
    even watches the compiled script as it runs, analyzes the data
    that flows through it, and further optimizes the machine code
    based on that knowledge.
</p>
</div>
</div>
</div>
</div>
</div>
</body>
</html>

```

CSS CODE

```

* {
    margin: 0;
    padding: 0;
    box-sizing: border-box;
}

body {
    font-family: Arial, Helvetica, sans-serif;
    color: #333;
}

.container {
    max-width: 970px;
}

.comments {
    padding: 20px;
}

.navbar {
    width: 300px;
    height: 100vh;
    background-color: #F5F2F0;
    display: inline-block;
    vertical-align: top;
}

.chapter {
    padding: 20px;
}

.chapter h3 {
    color: #b7b4b1;
}

```

```
}

.chapter p {
    color: #333;
    margin-top: 10px;
}

.lesson-nav {
    padding: 0 20px;
}

.lesson-nav h3{
    color:#b7b4b1;
}

.lesson-nav ul li {
    margin-top: 10px;
    list-style: none;
}

.main {
    width: 660px;
    height: 100vh;
    background-color: #FFFFFF;
    display: inline-block;
}

.wrapper {
    padding: 10px 30px;
}

.wrapper #introduction p {
    margin: 10px 0;
}

.wrapper #content p {
    margin: 10px 0;
}

.box1 {
    border: 1px solid gray;
    padding: 10px;
    margin-left: 20px;
}

.box2 {
    border: 1px solid gray;
    padding: 10px;
    margin-left: 20px;
}
```

```
}
```

```
.box2 ol li {  
  margin-left: 30px;  
}
```