Overview

Welcome to JavaScript

JavaScript is one of the most popular programming languages on earth and is used to add interactivity to webpages, process data, as well as create various applications (mobile apps, desktop apps, games, and more)

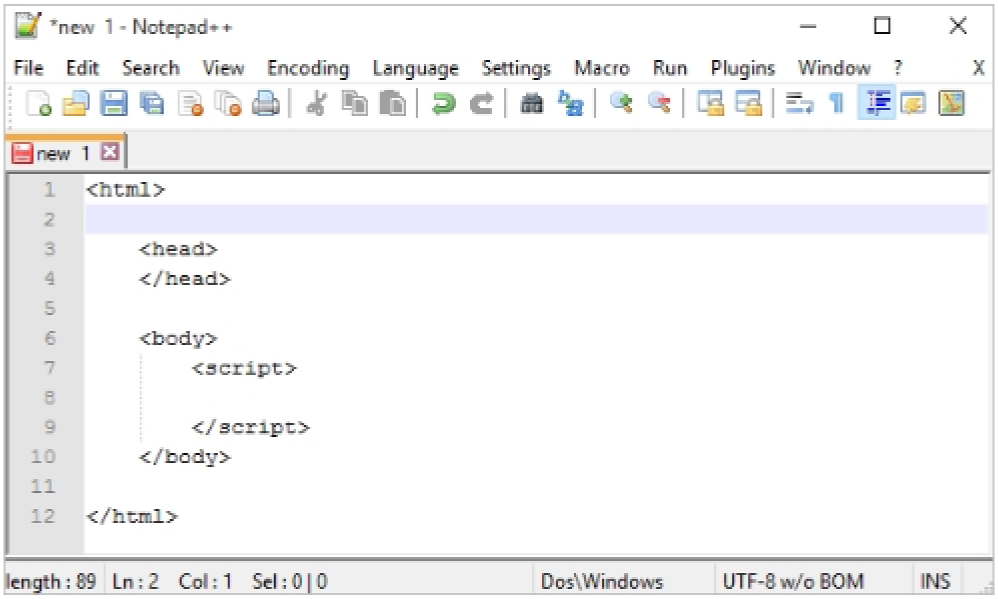
> Learning the fundamentals of a language will enable you to create the program you desire, whether client-side or server-side.

Your First JavaScript

Let's start with adding JavaScript to a webpage.
JavaScript on the web lives inside the **HTML** document.
In HTML, JavaScript code must be inserted between **<script>** and **</script>** tags:

```
<script>
  ...
</script>
```

JavaScript can be placed in the HTML page's **<body>** and **<head>** sections.
In the example below, we placed it within the **<body>** tag.



```
<html>

    <head>
    </head>

    <body>
        <script>

        </script>
    </body>

</html>
```

```
length : 89    Ln : 2    Col : 1    Sel : 0 | 0    Dos\Windows    UTF-8 w/o BOM    INS
```

> Remember that the script, which is placed in the head section, will be executed before the <body> is rendered. If you want to get elements in the body, it's a good idea to place your script at the end of the body tag.

Output

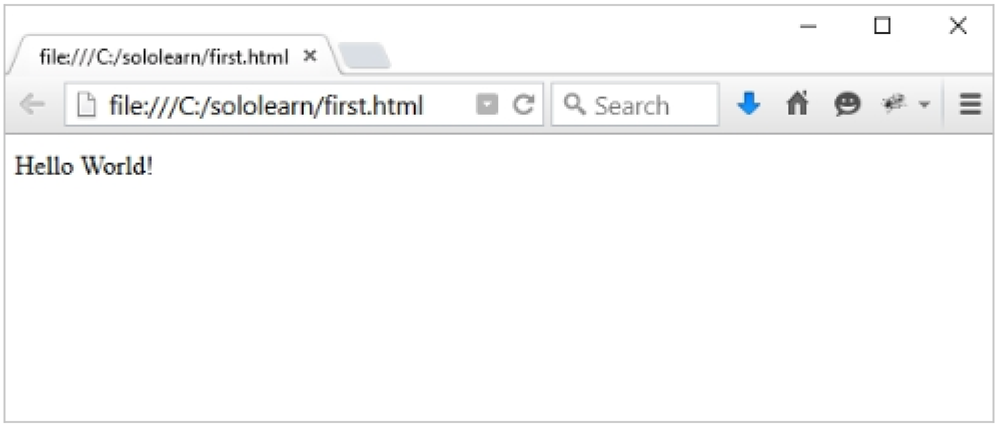
Let's use JavaScript to print "Hello World" to the browser.

```
<html>
  <head> </head>
  <body>
    <script>
      document.write("Hello World!");
    </script>
  </body>
</html>
```

Try It Yourself

The **document.write()** function writes a string into our HTML document. This function can be used to write text, HTML, or both.

The above code displays the following result:



file:///C:/sololearn/first.html ×
file:///C:/sololearn/first.html
Search

```
Hello World!
```

> The **document.write()** method should be used only for testing. Other output mechanisms appear in the upcoming lessons.

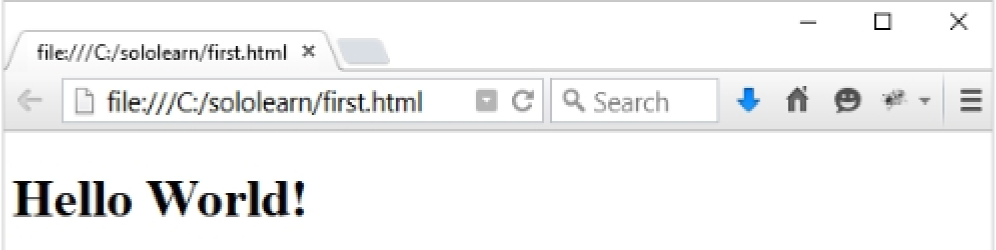
Formatting Text

Just like in HTML, we can use HTML tags to format text in JavaScript.
For example, we can output the text as a heading.

```
<html>
  <head> </head>
  <body>
    <script>
      document.write("<h1>Hello World!</h1>");
    </script>
  </body>
</html>
```

Try It Yourself

Result:



file:///C:/sololearn/first.html ×
file:///C:/sololearn/first.html
Search

Hello World!

> You can output almost everything in the webpage using JavaScript. Many JavaScript frameworks use this to create HTML pages.

JavaScript in <head>

You can place any number of scripts in an HTML document.
Typically, the script tag is placed in the head of the HTML document:

```
<html>
  <head>
    <script>
    </script>
  </head>
  <body>
  </body>
</html>
```

> There is also a <noscript> tag. Its content will be shown if the client's browser doesn't support JS scripts.

JavaScript in <body>

Alternatively, include the JavaScript in the <body> tag.

```
<html>
  <head> </head>
  <body>
    <script>
    </script>
  </body>
</html>
```

> It's a good idea to place scripts at the bottom of the <body> element.
> This can improve page load, because HTML display is not blocked by scripts loading.

The <script> Tag

The <script> tag can take two attributes, **language** and **type**, which specify the script's type:

```
<script language="javascript" type="text/javascript">

</script>
```

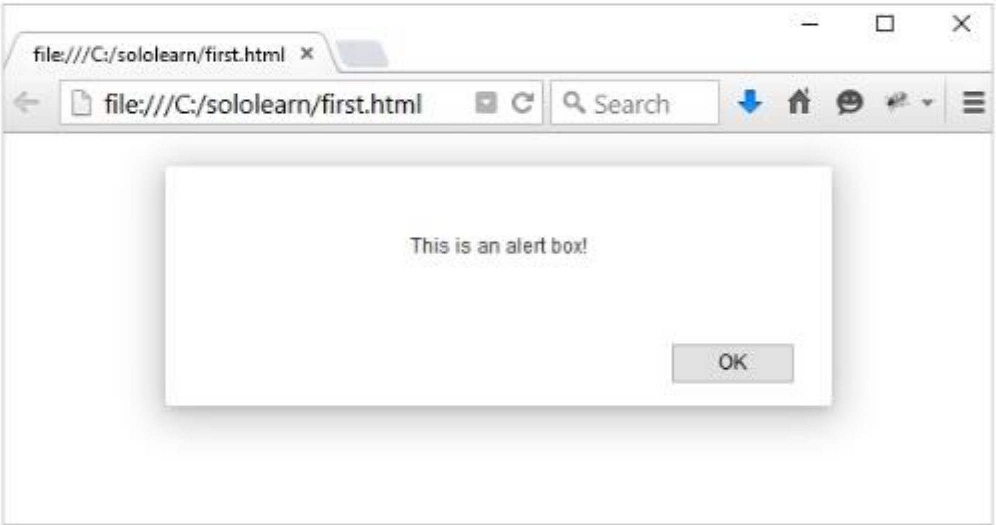
> The **language** attribute is deprecated, and should not be used.

In the example below, we created an alert box inside the script tag, using the **alert()** function.

```
<html>
  <head>
   <title></title>
   <script type="text/javascript">
     alert("This is an alert box!");
   </script>
  </head>
  <body>
  </body>
</html>
```

Try It Yourself

Result:


file:///C:/sololearn/first.html
file:///C:/sololearn/first.html
Search
OK

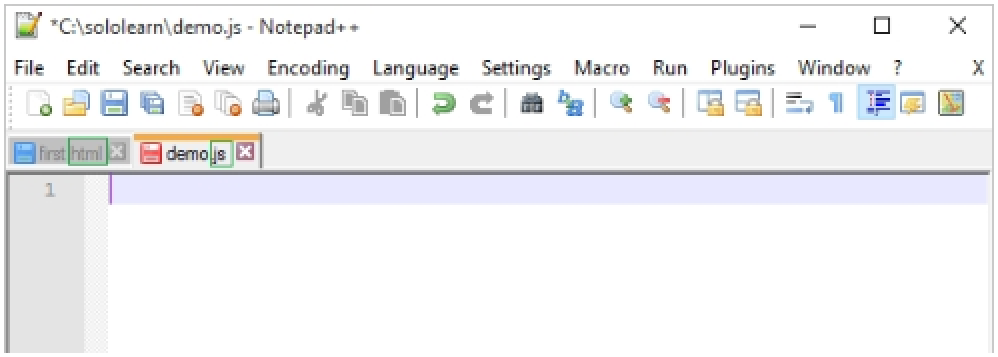
This is an alert box!

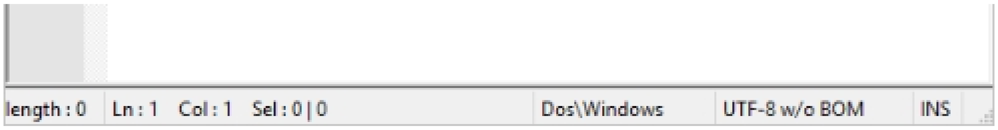
The **type** attribute: <script type="text/javascript"> is also no longer required, as JavaScript is the default HTML scripting language.

External JavaScript

Scripts can also be placed in **external files**.
External scripts are useful and practical when the same code is used in a number of different web pages.
JavaScript files have the **file extension .js**.

Below, we've created a new **text file**, and called it **demo.js**.


*C:\sololearn\demo.js - Notepad++
File Edit Search View Encoding Language Settings Macro Run Plugins Window ? X
first.html
demo.js



> Having JS scripts in separate files makes your code much readable and clearer.

External JavaScript

To use an external script, put the name of the script file in the **src** (source) attribute of the <script> tag.

Here is an example:

```
<html>
  <head>
   <title> </title>
   <script src="demo.js"></script>
  </head>
  <body>
  </body>
</html>
```

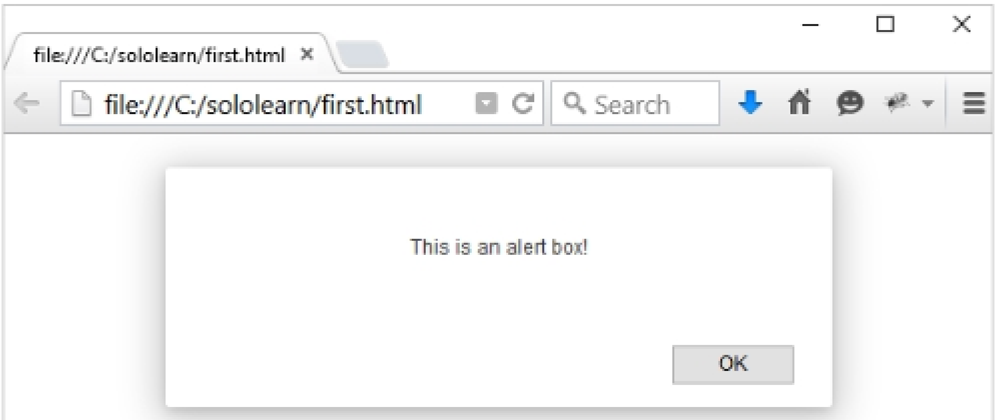
Your **demo.js** file includes the following JavaScript:

```
alert("This is an alert box!");
```

> External scripts cannot contain <script> tags.

External JavaScript

The result of the previous example will be identical to the result when we included the JavaScript within the HTML file.


file:///C:/sololearn/first.html
file:///C:/sololearn/first.html
Search
OK

This is an alert box!

You can place an external script reference in <head> or <body>, whichever you prefer.
The script will behave as if it were located exactly where the <script> tag is located.

> Placing a JavaScript in an external file has the following advantages:
> - It separates HTML and code.
> - It makes HTML and JavaScript easier to read and maintain.
> - Cached JavaScript files can speed up page loads.

JavaScript Comments

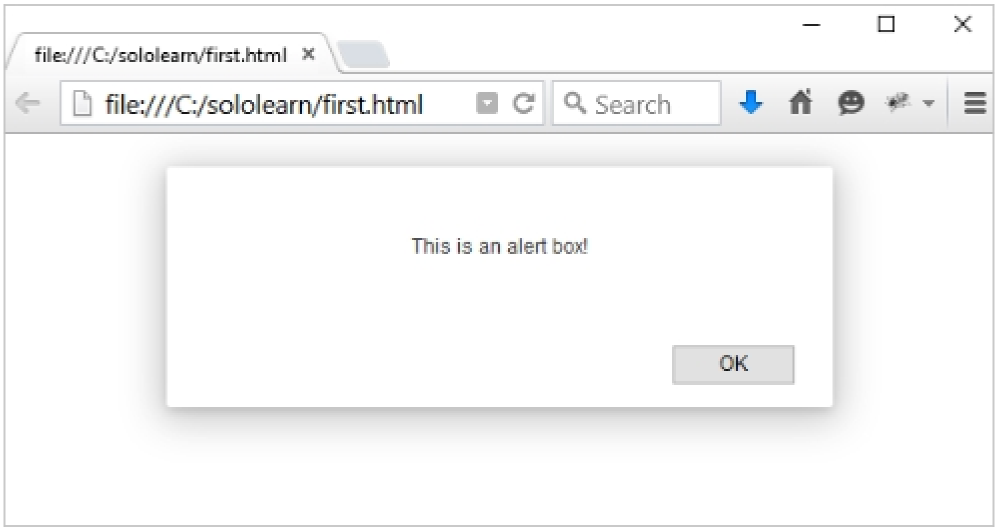
Not all JavaScript statements are "executed".
Code after a double slash //, or between /* and */, is treated as a **comment**.
Comments are ignored, and are not executed.

Single line comments use double slashes.

```
<script>
  // This is a single line comment
  alert("This is an alert box!");
</script>
```

Try It Yourself

Result:


file:///C:/sololearn/first.html
file:///C:/sololearn/first.html
Search
OK

This is an alert box!

> It's a good idea to make a comment about the logic of large functions to make your code more readable for others.

Multiple-Line Comments

Everything you write between /*and */ will be considered as a multi-line comment.

Here is an example.

```
<script>
  /* This code
  creates an
  alert box */
  alert("This is an alert box!");
</script>
```

Try It Yourself

> Comments are used to describe and explain what the code is doing.

End.