Using HTML <head>

Let's learn how to make use of the HTML <head> tag in this lesson.

WE'LL COVER THE FOLLOWING

- Defining scripts
- Specifying metadata

The `<head>` tag is an important part of HTML. It is a container of non-content elements that belong to the whole page. It must contain the `<title>` element and may contain one of the following additional elements as well: `<style>`, `<link>`, `<script>`, `<noscript>`, `<meta>`, and `<base>`.

`<title>` is required as it provides the title of the page that is displayed in the browser toolbar, in favorite, provided the page is added to Favorites, and also used by search engines to display the headline when the page is in within the search results.

In Chapter 2: A Short Tour of HTML, CSS, and JavaScript, you already met the `<style>` element describing a style sheet. This element is also nested in `<head>`:

```
<head>
  <title>Table of Contents</title>
  <style>
    body {
      font-family: Verdana, Arial, sans-serif;
    }
  </style>
</head>
```

As you remember, when you moved a style sheet into a separate file, you tied that file to the page with the `<link>` tag:

```
<head>
  <title>Table of Contents</title>

  <link href="style.css" rel="stylesheet" />
</head>
```

In this definition, the `href` attribute names the document, while `rel` describes the relationship between the page and the linked document. The only clue to understand the role of the document is `rel`, this attribute is required.

Defining scripts

In Chapter 2, you also met with the `<script>` tag that defined a JavaScript code snippet; however, that time `<script>` was nested into the `<body>`.

You can add one or more `<script>` tags to the `<head>` and `<body>`. You should know that the browser processes a `<script>` section as soon as it is read. So, if you use `<script>` in `<head>`, the code within will be executed before any real content is read.

> 📜**NOTE #1**: The `<script>` tag allows you to specify the type attribute, which declares the MIME-type of the script. If you omit type, the `"text/javascript"` default **MIME-type** is used, so the script is taken into account as JavaScript. You can use other MIME-types, depending on what is supported by your browser.

> 📜**NOTE #2**: The `<script>` tag also lets you refer to external script files with the `src` (source) attribute. You will learn more about this tag later, in the chapters treating JavaScript.

The `<script>` tag has a pair, `<noscript>`, usually used only in tandem with `<script>`. When your browser does not support scripts—or scripts are disabled because of security reasons—, this tag defines the alternate content. For example, the following script displays a message when the page cannot use JavaScript:

```
<head>
```

```
  <script>
    document.write("Hello from JavaScript");
  </script>

  <noscript>Sorry, your browser does not support JavaScript</noscript>
</head>
```

Specifying metadata

Surfing the web is unimaginable without search. Search engines examine the content of a page, but also use metadata, such as keywords, the name of the page's author, and so on. You can assign metadata to your web pages with the `<meta>` tag. The following sample shows, how you can add additional information to your page that is extracted and used by search engines:

```
<head>
  <title>My book's Table of Contents</title>
  <meta name="description" content="This page provides you the TOC"/>
  <meta name="author" content="Istvan Novak"/>
  <meta name="keywords" content="html,css,javascript" />
</head>
```

You can provide name and content attribute pairs with `<meta>`, where name describes the type of metadata, and content provides the related value.

Using `<meta>`, you can also describe the character encoding of the page:

```
<meta charset="utf-8" />
```

Show Useful Info

Now that we've learned about the `<head>` tag, in the *next lesson*, we'll see how to add content to our HTML web page.