```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>

<h1>This is a Heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

All browsers need the **doctype**. Without the **DOCTYPE** you are forcing the browsers to render in Quirks Mode. The **DOCTYPE** declaration is <! ... **DOCTYPEs** from earlier versions of **HTML** were longer because the **HTML** language was SGML-based and therefore **required** a reference to a DTD.

To maintain compatibility with the greatest possible number of web pages, modern web browsers are generally developed with multiple rendering **modes**: in "**standards mode**" pages are rendered according to the **HTML** and CSS specifications, while in "**quirks mode**" attempts are made to emulate the behavior of older browsers.

Quirks Mode and Standards Mode

In the old days of the web, pages were typically written in two versions: One for Netscape Navigator, and one for Microsoft Internet Explorer. When the web standards were made at W3C, browsers could not just start using them, as doing so would break most existing sites on the web. Browsers therefore introduced two modes to treat new standards compliant sites differently from old legacy sites.

There are now three modes used by the layout engines in web browsers: quirks mode, almost standards mode, and full standards mode. In **quirks mode**, layout emulates nonstandard behavior in Navigator 4 and Internet Explorer 5. This is essential in order to support websites that were built before the widespread adoption of web standards. In **full standards mode**, the behavior is (hopefully) the

behavior described by the HTML and CSS specifications. In **almost standards mode**, there are only a very small number of quirks implemented.

How do browsers determine which mode to use?

For HTML documents, browsers use a DOCTYPE in the beginning of the document to decide whether to handle it in quirks mode or standards mode. To ensure that your page uses full standards mode, make sure that your page has a DOCTYPE like in this example:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset=UTF-8>
    <title>Hello World!</title>
  </head>
  <body>
  </body>
</html>
```

The DOCTYPE shown in the example, <!DOCTYPE html>, is the simplest possible, and the one recommended by HTML5. Earlier versions of the HTML standard recommended other variants, but all existing browsers today will use full standards mode for this DOCTYPE, even the dated Internet Explorer 6. There are no valid reasons to use a more complicated DOCTYPE. If you do use another DOCTYPE, you may risk choosing one which triggers almost standards mode or quirks mode.

Make sure you put the DOCTYPE right at the beginning of your HTML document. Anything before the DOCTYPE, like a comment or an XML declaration will trigger quirks mode in Internet Explorer 9 and older.

In HTML5, the only purpose of the DOCTYPE is to activate full standards mode. Older versions of the HTML standard gave additional meaning to the DOCTYPE, but

no browser has ever used the DOCTYPE for anything other than switching between quirks mode and standards mode.

The Head Element

The HTML `<head>` element is a container for the following elements: `<title>`, `<style>`, `<meta>`, `<link>`, `<script>`, and `<base>`.

The HTML <title> Element

The `<title>` element defines the title of the document. The title must be text-only, and it is shown in the browser's title bar or in the page's tab.

The <title> tag is required in HTML documents!

The contents of a page title is very important for search engine optimization (SEO)! The page title is used by search engine algorithms to decide the order when listing pages in search results.

The `<title>` element:

- defines a title in the browser toolbar
- provides a title for the page when it is added to favorites
- displays a title for the page in search engine-results

So, try to make the title as accurate and meaningful as possible!

A simple HTML document:

The HTML <style> Element

The `<style>` element is used to define style information for a single HTML page:

Example

```
<style>
  body {background-color: powderblue;}
  h1 {color: red;}
  p {color: blue;}
</style>
```

The HTML <link> Element

The `<link>` element defines the relationship between the current document and an external resource.

The `<link>` tag is most often used to link to external style sheets:

Example

```
<link rel="stylesheet" href="mystyle.css">
```

The HTML <meta> Element

The `<meta>` element is typically used to specify the character set, page description, keywords, author of the document, and viewport settings.

The metadata will not be displayed on the page, but are used by browsers (how to display content or reload page), by search engines (keywords), and other web services.

Examples

Define the character set used:

```
<meta charset="UTF-8">
```

Define keywords for search engines:

```
<meta name="keywords" content="HTML, CSS, JavaScript">
```

Define a description of your web page:

```
<meta name="description" content="Free Web tutorials">
```

Define the author of a page:

```
<meta name="author" content="John Doe">
```

Refresh document every 30 seconds:

```
<meta http-equiv="refresh" content="30">
```

Setting the viewport to make your website look good on all devices:

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```
<meta charset="UTF-8">
<meta name="description" content="Free Web tutorials">
<meta name="keywords" content="HTML, CSS, JavaScript">
<meta name="author" content="John Doe">
```

Setting The Viewport

The viewport is the user's visible area of a web page. It varies with the device - it will be smaller on a mobile phone than on a computer screen.

You should include the following `<meta>` element in all your web pages:

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

This gives the browser instructions on how to control the page's dimensions and scaling.

The `width=device-width` part sets the width of the page to follow the screen-width of the device (which will vary depending on the device).

The `initial-scale=1.0` part sets the initial zoom level when the page is first loaded by the browser.

Here is an example of a web page *without* the viewport meta tag, and the same web page *with* the viewport meta tag:

Without the viewport meta tag

With the viewport meta tag

The HTML <script> Element

The `<script>` element is used to define client-side JavaScripts.

The following JavaScript writes "Hello JavaScript!" into an HTML element with id="demo":

Example

```
<script>
function myFunction() {
  document.getElementById("demo").innerHTML = "Hello JavaScript!";
}
</script>
```

The HTML <base> Element

The `<base>` element specifies the base URL and/or target for all relative URLs in a page.

The `<base>` tag must have href.

There can only be one single `<base>` element in a document!

Example

Specify a default URL and a default target for all links on a page:

```
<head>
<base href="https://www.mywebsite.com/">
</head>

<body>
<img src="images/stickman.gif" width="24" height="39" alt="Stickman">
<a href="tags/tag_base.asp">HTML base Tag</a>
</body>
```

- The `<head>` element is a container for metadata (data about data)
- The `<head>` element is placed between the `<html>` tag and the `<body>` tag
- The `<title>` element is required and it defines the title of the document
- The `<style>` element is used to define style information for a single documents
- The `<link>` tag is most often used to link to external style sheets
- The `<meta>` element is typically used to specify the character set, page description, keywords, author of the document, and viewport settings
- The `<script>` element is used to define client-side JavaScripts
- The `<base>` element specifies the base URL and/or target for all relative URLs in a page