HTML <head> Tag

The <head> element is a container for all the head elements.

The <head> element can include a title for the document, scripts, styles, meta information, and more.

The following elements can go inside the <head> element:

* <title> (this element is required in an HTML document)
* <style>
* <base>
* <link>
* <meta>
* <script>
* <noscript>

# HTML <title> Tag

The <title> tag is required in all HTML documents and it defines the title of the document.

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

**Note:** You can NOT have more than one <title> element in an HTML document.

**Tip:** If you omit the <title> tag, the document will not validate as HTML.

<head>  
  <title>Title of the document</title>  
</head>

# HTML <style> Tag

The <style> tag is used to define style information for an HTML document.

Inside the <style> element you specify how HTML elements should render in a browser.

Each HTML document can contain multiple <style> tags.

**Tip:** To link to an external style sheet, use the <link> tag.

**Note:** If the "scoped" attribute is not used, each <style> tag must be located in the head section.

## Attributes

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Value** | **Description** |
| media | *media\_query* | Specifies what media/device the media resource is optimized for |
| Scoped  [HTML 5] | scoped | Specifies that the styles only apply to this element's parent element and that element's child elements |
| type | text/css | Specifies the media type of the <style> tag |

<html>  
<head>  
<style>  
h1 {color:red;}  
p {color:blue;}  
</style>  
</head>  
<body>  
<h1>A heading</h1>  
<p>A paragraph.</p>  
</body>  
</html>

# HTML <base> Tag

The <base> tag specifies the base URL/target for all relative URLs in a document.

There can be at maximum one <base> element in a document, and it must be inside the <head> element.

**Tip:** Put the <base> tag as the *first* element inside the <head> element, so that other elements in the head section use the information from the <base> element.

**Note:** If the <base> tag is present, it must have either a href attribute or a target attribute, or both.

## Attributes

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Value** | **Description** |
| href | *URL* | Specifies the base URL for all relative URLs in the page |
| target | \_blank \_parent \_self \_top *framename* | Specifies the default target for all hyperlinks and forms in the page |

<head>  
  <base href="http://www.wwebsite.com/images/" target="\_blank">  
</head>  
  
<body>  
<img src="stickman.gif" width="24" height="39" alt="Stickman">  
<a href="http://www.website.com">Website</a>  
</body>

# HTML <link> Tag

The <link> tag defines a link between a document and an external resource.

The <link> tag is used to link to external style sheets.

**Note:** The <link> element is an empty element, it contains attributes only.

**Note:** This element goes only in the head section, but it can appear any number of times.

## Attributes

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Value** | **Description** |
| charset | *char\_encoding* | Not supported in HTML5. Specifies the character encoding of the linked document |
| crossorigin  [HTML 5] | anonymous use-credentials | Specifies how the element handles cross-origin requests |
| href | *URL* | Specifies the location of the linked document |
| hreflang | *language\_code* | Specifies the language of the text in the linked document |
| media | media\_query | Specifies on what device the linked document will be displayed |
| rel | alternate author dns-prefetch help icon license next pingback preconnect prefetch preload prerender prev search stylesheet | Required. Specifies the relationship between the current document and the linked document |
| rev | *reversed relationship* | Not supported in HTML5. Specifies the relationship between the linked document and the current document |
| sizes  [HTML 5] | *Height*x*Width* any | Specifies the size of the linked resource. Only for rel="icon" |
| target | \_blank \_self \_top \_parent *frame\_name* | Not supported in HTML5. Specifies where the linked document is to be loaded |
| type | media\_type | Specifies the media type of the linked document |

<head>  
  <link rel="stylesheet" type="text/css" href="theme.css">  
</head>

# HTML <meta> Tag

Metadata is data (information) about data.

The <meta> tag provides metadata about the HTML document. Metadata will not be displayed on the page, but will be machine parsable.

Meta elements are typically used to specify page description, keywords, author of the document, last modified, and other metadata.

The metadata can be used by browsers (how to display content or reload page), search engines (keywords), or other web services.

<head>  
  <meta charset="UTF-8">  
  <meta name="description" content="Free Web tutorials">  
  <meta name="keywords" content="HTML,CSS,XML,JavaScript">  
  <meta name="author" content="John Doe">  
  <meta name="viewport" content="width=device-width, initial-scale=1.0">  
</head>

HTML5 introduced a method to let web designers take control over the viewport (the user's visible area of a web page), through the <meta> tag (See "Setting The Viewport" example below).

**Note:**<meta> tags always go inside the <head> element.

**Note:** Metadata is always passed as name/value pairs.

**Note:** The content attribute MUST be defined if the name or the http-equiv attribute is defined. If none of these are defined, the content attribute CANNOT be defined.

**Setting The Viewport**

HTML5 introduced a method to let web designers take control over the viewport, through the <meta> tag.

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

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

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

A <meta> viewport element 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**

## Attributes

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Value** | **Description** |
| charset  [HTML 5] | *character\_set* | Specifies the character encoding for the HTML document |
| content | *text* | Gives the value associated with the http-equiv or name attribute |
| http-equiv | content-type default-style refresh | Provides an HTTP header for the information/value of the content attribute |
| name | application-name author description generatorkeywords viewport | Specifies a name for the metadata |
| scheme | *format/URI* | Not supported in HTML5. Specifies a scheme to be used to interpret the value of the content attribute |

**Example 1 - Define keywords for search engines:**

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

**Example 2 - Define a description of your web page:**

<meta name="description" content="Free Web tutorials on HTML and CSS">

**Example 3 - Define the author of a page:**

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

**Example 4 - Refresh document every 30 seconds:**

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

**Example 5 - Setting the viewport to make your website look good on all devices:**

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

# HTML <script> Tag

The <script> tag is used to define a client-side script (JavaScript).

The <script> element either contains scripting statements, or it points to an external script file through the src attribute.

Common uses for JavaScript are image manipulation, form validation, and dynamic changes of content.

**Note:** If the "src" attribute is present, the <script> element must be empty.

**Tip:** Also look at the <noscript> element for users that have disabled scripts in their browser, or have a browser that doesn't support client-side scripting.

**Note:** There are several ways an external script can be executed:

* If async="async": The script is executed asynchronously with the rest of the page (the script will be executed while the page continues the parsing)
* If async is not present and defer="defer": The script is executed when the page has finished parsing
* If neither async or defer is present: The script is fetched and executed immediately, before the browser continues parsing the page

## Attributes

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Value** | **Description** |
| async  [HTML 5] | async | Specifies that the script is executed asynchronously (only for external scripts) |
| charset | *charset* | Specifies the character encoding used in an external script file |
| defer | defer | Specifies that the script is executed when the page has finished parsing (only for external scripts) |
| src | *URL* | Specifies the URL of an external script file |
| type | *media\_type* | Specifies the media type of the script |
| xml:space | preserve | Not supported in HTML5. Specifies whether whitespace in code should be preserved |

<script>  
document.getElementById("demo").innerHTML = "Hello JavaScript!";  
</script>

# HTML <noscript> Tag

The <noscript> tag defines an alternate content for users that have disabled scripts in their browser or have a browser that doesn't support script.

The <noscript> element can be used in both <head> and <body>.

When used inside the <head> element: <noscript> must contain only <link>, <style>, and <meta> elements.

The content inside the <noscript> element will be displayed if scripts are not supported, or are disabled in the user's browser.

<script>  
document.write("Hello World!")  
</script>  
<noscript>Your browser does not support JavaScript!</noscript>

**Tip:** It is also a good practice to use the comment tag to "hide" scripts from browsers without support for client-side scripts (so they don't show them as plain text):

<script>  
<!--  
function displayMsg() {  
    alert("Hello World!")  
}  
//-->  
</script>