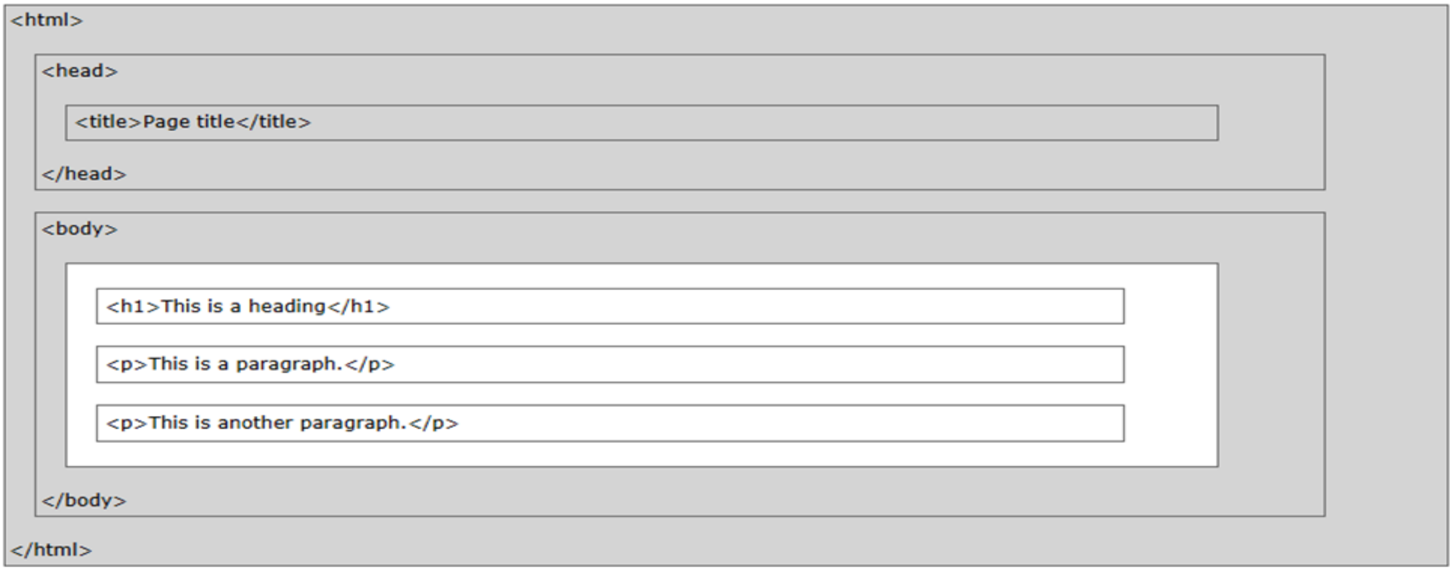
Html Basics

HTML is the standard markup language for creating Web pages.

- HTML stands for Hyper Text Markup Language
- HTML describes the structure of Web pages using markup
- HTML elements are the building blocks of HTML pages
- HTML elements are represented by tags
- HTML tags label pieces of content such as "heading", "paragraph", "table", and so on
- Browsers do not display the HTML tags, but use them to render the content of the page

HTML Page Structure

Below is a visualization of an HTML page structure:



```
<html>
  <head>
    <title>Page title</title>
  </head>
  <body>
      <h1>This is a heading</h1>
      <p>This is a paragraph.</p>
      <p>This is another paragraph.</p>
  </body>
</html>
```

The <!DOCTYPE> Declaration

- The `<!DOCTYPE>` declaration represents the document type, and helps browsers to display web pages correctly.
- It must only appear once, at the top of the page (before any HTML tags).
- The `<!DOCTYPE>` declaration is not case sensitive.
- The `<!DOCTYPE>` declaration for HTML5 is: `<!DOCTYPE html>`

HTML Tags

HTML tags are element names surrounded by angle brackets:

<tagname>content goes here...</tagname>

- HTML tags normally come **in pairs** like `<p>` and `</p>`
- The first tag in a pair is the **start tag,** the second tag is the **end tag**
- The end tag is written like the start tag, but with a **forward slash** inserted before the tag name

HTML Documents

All HTML documents must start with a document type declaration: `<!DOCTYPE html>`.

The HTML document itself begins with `<html>` and ends with `</html>`.

The visible part of the HTML document is between `<body>` and `</body>`.

HTML Elements

An HTML element usually consists of a **start** tag and **end** tag, with the content inserted in between:

```
<tagname>Content goes here...</tagname>
```

The HTML **element** is everything from the start tag to the end tag:

```
<p>My first paragraph.</p>
```

Nested HTML Elements

HTML elements can be nested (elements can contain elements).

All HTML documents consist of nested HTML elements.

This example contains four HTML elements:

```
<!DOCTYPE html>
<html>
<body>
<h1>My First Heading</h1>
<p>My first paragraph.</p>
</body>
</html>
```

Empty HTML Elements

HTML elements with no content are called empty elements.

`<br>` is an empty element without a closing tag (the `<br>` tag defines a line break):

```
<p>This is a <br> paragraph with a line break.</p>
```

The HTML <head> Element

- The `<head>` element is a container for metadata (data about data) and is placed between the `<html>` tag and the `<body>` tag.
- HTML metadata is data about the HTML document. Metadata is not displayed.

- Metadata typically define the document title, character set, styles, links, scripts, and other meta information.
- The following tags describe metadata: `<title>`, `<style>`, `<meta>`, `<link>`, `<script>`, and `<base>`.

The HTML <title> Element

The `<title>` element defines the title of the document, and is required in all HTML/XHTML documents.

The `<title>` element:

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

A simple HTML document:

```
<!DOCTYPE html>
<html>
<head>
  <title>Page Title</title>
</head>
<body>
The content of the document......
</body>
</html>
```

The HTML <style> Element

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

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

The HTML <link> Element

The `<link>` element is used to link to external style sheets:

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

HTML Links - Hyperlinks

HTML links are hyperlinks.

You can click on a link and jump to another document.

When you move the mouse over a link, the mouse arrow will turn into a little hand.

HTML Links - Syntax

In HTML, links are defined with the `<a>` tag:

```
<a href="url">link text</a>
```

Local Links

The example above used an absolute URL (a full web address).

A local link (link to the same web site) is specified with a relative URL (without https://www....).

```
<a href="html_images.asp">HTML Images</a>
```

HTML Link Colors

By default, a link will appear like this (in all browsers):

- An unvisited link is underlined and blue
- A visited link is underlined and purple
- An active link is underlined and red

You can change the default colors, by using CSS:

```
<style>
a:link {
  color: green;
  background-color: transparent;
  text-decoration: none;
}
a:visited {
  color: pink;
  background-color: transparent;
  text-decoration: none;
}
a:hover {
  color: red;
  background-color: transparent;
  text-decoration: underline;
}
a:active {
  color: yellow;
  background-color: transparent;
  text-decoration: underline;
}
</style>
```

HTML Links - The target Attribute

The `target` attribute specifies where to open the linked document.

The `target` attribute can have one of the following values:

- `_blank` - Opens the linked document in a new window or tab
- `_self` - Opens the linked document in the same window/tab as it was clicked (this is default)
- `_parent` - Opens the linked document in the parent frame
- `_top` - Opens the linked document in the full body of the window
- *framename* - Opens the linked document in a named frame

This example will open the linked document in a new browser window/tab:

```
<a href="https://www.google.com/" target="_blank">Visit Google!</a>
```

HTML Links - Image as Link

It is common to use images as links:

```
<a href="default.asp">
  <img src="smiley.gif" alt="HTML tutorial" style="width:42px;height:42px;border:0;">
</a>
```

The HTML <meta> Element

The `<meta>` element is used to specify which character set is used, page description, keywords, author, and other metadata.

Metadata is used by browsers (how to display content), by search engines (keywords), and other web services.

Define the character set used:

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

Define a description of your web page:

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

Define keywords for search engines:

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

Define the author of a page:

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

Refresh document every 30 seconds:

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

Example of `<meta>` tags:

```
<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">
```

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.

The HTML <script> Element

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

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

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

The 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.

To select an HTML element, JavaScript very often uses the `document.getElementById()` method.

This JavaScript example writes "Hello JavaScript!" into an HTML element with id="demo":

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

The HTML <noscript> Tag

The `<noscript>` tag is used to provide an alternate content for users that have disabled scripts in their browser or have a browser that doesn't support client-side scripts:

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

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

The HTML <base> Element

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

```
<base href="https://www.xyz.com/images/" target="_blank">
```