

HTML5

Notes for Professionals

Chapter 12: Linking Resources

Attribute	Details
charset	Specifies the character encoding of the linked document.
crossorigin	Specifies how the element handles cross origin requests.
href	Specifies the location of the linked document.
hreflang	Specifies the language of the text in the linked document.
media	Specifies on what device the linked document will be displayed, often used with based on the device in question.
rel	Specifies the relationship between the current document and the current doc based on the device in question.
required	Specifies the relationship between the linked document and the current doc based on the device in question.
rev	Specifies the size of the linked document. Only when rel="stylesheet".
size	Specifies where the linked document is to be loaded.
target	Specifies the media type of the linked document.
type	Specifies a base64 encoded hash (sha256, sha384, or sha512) of the linked document to verify its legitimacy.
integrity	

Section 12.1: JavaScript

Synchronous
`<script src="/path/to/js"></script>`
Standard practice is to place JavaScript `<script>` tags just before the closing `</body>` element to allow your site's visuals to show up more quickly and discourages your JavaScript elements that haven't loaded yet.

Asynchronous
`<script src="/path/to/js" async></script>`
Another alternative, when the JavaScript code being loaded is not necessary to fully parse, is to use the `async` attribute. Using `async` the browser will load the script once it is fully downloaded, without interrupting the HTML parsing in order to support elements that haven't loaded yet.

Deferred
`<script src="/path/to/js" defer></script>`
Deferred scripts are like `async` scripts, with the exception that the parsing will not be interrupted. Deferred scripts are guaranteed to be loaded in the order of the `<script>` tags.

`<script>` element defines content to be displayed if the user has scripts disabled or if the browser does not support using scripts. The `<noscript>` tag can be placed in either the `<head>` or the `<body>`.

Chapter 17: Input Control Elements

Parameter	Details
class	Indicates the Class of the input.
id	Indicates the ID of the input.
type	Identifies the type of input control to display. Acceptable values are hidden, text, text1, text2, password, date, time, number, range, color, checkbox, radio, file, select, image, reset, and button. Defaults to text if not specified, if the value is invalid, or if the browser does not support the type specified.
name	Indicates the name of the input.
disabled	Boolean value that indicates the input should be disabled. Disabled controls cannot be edited, are not sent on form submissions, and cannot receive focus.
checked	When the value of the type attribute is radio or checkbox, the presence of this Boolean attribute indicates that the control is selected by default; otherwise it is ignored.
multiple	HTML5 Indicates multiple files or values can be passed (Applies only to file and email type inputs)
placeholder	HTML5 A hint to the user of what can be entered in the control. The placeholder text must not contain carriage returns or line-feeds.
autocomplete	HTML5 Indicates whether the value of the control can be automatically completed by the browser.
readonly	Boolean value that indicates the input is not editable. Readonly controls are still sent on form submission, but will not receive focus. HTML5: This attribute is ignored when the value of type attribute is either set to hidden, range, color, checkbox, radio, file or button.
required	HTML5 Indicates a value must be present or the element must be checked in order for the form to be submitted.
alt	An alternative text for images, in case they are not displayed.
autofocus	The <code><input></code> element should get the focus when page loads.
value	Specifies the value of <code><input></code> element.
step	The <code>step</code> attribute specifies the legal number intervals. It works with the following input types: number, range, date, datetime-local, month, time and week.

A key component of interactive web systems, input tags are HTML elements designed to take a specific form of input from users. Different types of input elements can regulate the data entered to fit a specified format and provide security to password entry.

Section 17.1: Text

The most basic input type and the default input if no type is specified. This input type defines a single-line text field with line breaks automatically removed from the input value. All other characters can be entered into this. `<input type="text">` elements are used within a `<form>` element to declare input controls that allow users to input data.

Syntax

```
<input type="text">
```

or (without specifying a type, using the default attributes):

```
<input>
```

The default width of a text field input is 20 characters. This can be changed by specifying a value for the `width` attribute like this:

```
<input type="text" size="50">
```

HTML5 Notes for Professionals

Chapter 20: Sectioning Elements

Section 20.1: Nav Element

The `<nav>` element is primarily intended to be used for sections that contain main navigation blocks for the website; this can include links to other parts of the web page (e.g. anchors for a table of contents) or other pages entirely.

Inline items

The following will display an inline set of hyperlinks:

```
<nav>  
  <a href="/index.html">Home</a>  
  <a href="/about.html">About</a>  
</nav>
```

List items when needed

If the content represents a list of items, use a list item to show this and enhance the user experience. Note the `role="navigation"`, more on this below.

```
<nav role="navigation">  
  <ul>  
    <li><a href="/index.html">Home</li>  
    <li><a href="/about.html">About</li>  
  </ul>  
</nav>
```

Avoid unnecessary usage

`<footer>` elements may have a list of links to other parts of the site (FAQ, T&C, etc.). The footer element alone is sufficient in this case, you don't need to further wrap your links with a `<nav>` element in the `<footer>`.

```
<!-- the <nav> is not required in the <footer -->  
<footer>  
  <a href="#">...</a>  
</footer>
```

`<!-- the footer alone is sufficient -->`

```
<footer>  
  <a href="#">...</a>  
</footer>
```

Notes:

- `<main>` `<section>` descendants are not allowed within a `<nav>`.
- Adding a `role="navigation" data-role="navigation"` to the `<nav>` element is advised to aid user agents that don't support HTML5 and also to provide more context for those that do.

HTML5 Notes for Professionals

100+ pages
of professional hints and tricks

Contents

About	1
Chapter 1: Getting started with HTML	2
Section 1.1: Hello World	2
Chapter 2: Doctypes	5
Section 2.1: Adding the Doctype	5
Section 2.2: HTML 5 Doctype	5
Chapter 3: Headings	6
Section 3.1: Using Headings	6
Chapter 4: Paragraphs	7
Section 4.1: HTML Paragraphs	7
Chapter 5: Text Formatting	8
Section 5.1: Highlighting	8
Section 5.2: Bold, Italic, and Underline	8
Section 5.3: Abbreviation	9
Section 5.4: Inserted, Deleted, or Stricken	9
Section 5.5: Superscript and Subscript	9
Chapter 6: Anchors and Hyperlinks	11
Section 6.1: Link to another site	11
Section 6.2: Link to an anchor	12
Section 6.3: Link to a page on the same site	12
Section 6.4: Link that dials a number	12
Section 6.5: Open link in new tab/window	13
Section 6.6: Link that runs JavaScript	13
Section 6.7: Link that runs email client	14
Chapter 7: Lists	15
Section 7.1: Ordered List	15
Section 7.2: Unordered List	16
Section 7.3: Nested lists	17
Section 7.4: Description List	17
Chapter 8: Tables	19
Section 8.1: Simple Table	19
Section 8.2: Spanning columns or rows	19
Section 8.3: Column Groups	20
Section 8.4: Table with thead, tbody, tfoot, and caption	21
Section 8.5: Heading scope	22
Chapter 9: Comments	24
Section 9.1: Creating comments	24
Section 9.2: Commenting out whitespace between inline elements	24
Chapter 10: Classes and IDs	26
Section 10.1: Giving an element a class	26
Section 10.2: Giving an element an ID	27
Section 10.3: Acceptable Values	27
Section 10.4: Problems related to duplicated IDs	29
Chapter 11: Data Attributes	30
Section 11.1: Older browsers support	30
Section 11.2: Data Attribute Use	30

Chapter 12: Linking Resources	31
Section 12.1: JavaScript	31
Section 12.2: External CSS Stylesheet	32
Section 12.3: Favicon	32
Section 12.4: Alternative CSS	32
Section 12.5: Resource Hint: dns-prefetch, prefetch, prerender	33
Section 12.6: Link 'media' attribute	33
Section 12.7: Prev and Next	33
Section 12.8: Web Feed	33
Chapter 13: Include JavaScript Code in HTML	35
Section 13.1: Handling disabled Javascript	35
Section 13.2: Linking to an external JavaScript file	35
Section 13.3: Directly including JavaScript code	35
Section 13.4: Including a JavaScript file executing asynchronously	35
Chapter 14: Using HTML with CSS	36
Section 14.1: External Stylesheet Use	36
Section 14.2: Internal Stylesheet	36
Section 14.3: Inline Style	37
Section 14.4: Multiple Stylesheets	37
Chapter 15: Images	38
Section 15.1: Creating an image	38
Section 15.2: Choosing alt text	38
Section 15.3: Responsive image using the srcset attribute	39
Section 15.4: Responsive image using picture element	40
Chapter 16: Image Maps	41
Section 16.1: Introduction to Image Maps	41
Chapter 17: Input Control Elements	43
Section 17.1: Text	43
Section 17.2: Checkbox and Radio Buttons	44
Section 17.3: Input Validation	46
Section 17.4: Color	47
Section 17.5: Password	48
Section 17.6: File	48
Section 17.7: Button	49
Section 17.8: Submit	50
Section 17.9: Reset	50
Section 17.10: Hidden	50
Section 17.11: Tel	50
Section 17.12: Email	51
Section 17.13: Number	51
Section 17.14: Range	51
Section 17.15: Search	51
Section 17.16: Image	51
Section 17.17: Week	52
Section 17.18: Url	52
Section 17.19: DateTime-Local	52
Section 17.20: Month	52
Section 17.21: Time	52
Section 17.22: DateTime (Global)	53
Section 17.23: Date	53

Chapter 18: Forms	54
Section 18.1: Submitting	54
Section 18.2: Target attribute in form tag	55
Section 18.3: Uploading Files	55
Section 18.4: Grouping a few input fields	55
Chapter 19: Div Element	57
Section 19.1: Basic usage	57
Section 19.2: Nesting	57
Chapter 20: Sectioning Elements	59
Section 20.1: Nav Element	59
Section 20.2: Article Element	60
Section 20.3: Main Element	61
Section 20.4: Header Element	62
Section 20.5: Footer Element	63
Section 20.6: Section Element	63
Chapter 21: Navigation Bars	64
Section 21.1: Basic Navigation Bar	64
Section 21.2: HTML5 Navigation Bar	64
Chapter 22: Label Element	65
Section 22.1: About Label	65
Section 22.2: Basic Use	65
Chapter 23: Output Element	67
Section 23.1: Output Element Using For and Form Attributes	67
Section 23.2: Output Element with Attributes	67
Chapter 24: Void Elements	68
Section 24.1: Void elements	68
Chapter 25: Media Elements	69
Section 25.1: Audio	69
Section 25.2: Video	69
Section 25.3: Using ` <code><video></code> ` and ` <code><audio></code> ` element to display audio/video content	69
Section 25.4: Video header or background	70
Chapter 26: Progress Element	71
Section 26.1: Progress	71
Section 26.2: Changing the color of a progress bar	71
Section 26.3: HTML Fallback	72
Chapter 27: Selection Menu Controls	73
Section 27.1: Select Menu	73
Section 27.2: Options	73
Section 27.3: Option Groups	74
Section 27.4: Datalist	74
Chapter 28: Embed	76
Section 28.1: Basic usage	76
Section 28.2: Defining the MIME type	76
Chapter 29: IFrames	77
Section 29.1: Basics of an Inline Frame	77
Section 29.2: Sandboxing	77
Section 29.3: Setting the Frame Size	77
Section 29.4: Using the "srcdoc" Attribute	78

Section 29.5: Using Anchors with IFrames	78
Chapter 30: Content Languages	79
Section 30.1: Base Document Language	79
Section 30.2: Element Language	79
Section 30.3: Elements with Multiple Languages	79
Section 30.4: Regional URLs	79
Section 30.5: Handling Attributes with Different Languages	79
Chapter 31: SVG	81
Section 31.1: Inline SVG	81
Section 31.2: Embedding external SVG files in HTML	81
Section 31.3: Embedding SVG using CSS	82
Chapter 32: Canvas	83
Section 32.1: Basic Example	83
Section 32.2: Drawing two rectangles on a <canvas>	83
Chapter 33: Meta Information	85
Section 33.1: Page Information	85
Section 33.2: Character Encoding	85
Section 33.3: Robots	86
Section 33.4: Social Media	86
Section 33.5: Mobile Layout Control	87
Section 33.6: Automatic Refresh	88
Section 33.7: Phone Number Recognition	88
Section 33.8: Automatic redirect	88
Section 33.9: Web App	89
Chapter 34: Marking up computer code	90
Section 34.1: Block with <pre> and <code>	90
Section 34.2: Inline with <code>	90
Chapter 35: Marking-up Quotes	91
Section 35.1: Inline with <q>	91
Section 35.2: Block with <blockquote>	91
Chapter 36: Tabindex	93
Section 36.1: Add an element to the tabbing order	93
Section 36.2: Remove an element from the tabbing order	93
Section 36.3: Define a custom tabbing order (not recommended)	93
Chapter 37: Global Attributes	94
Section 37.1: Contenteditable Attribute	94
Chapter 38: HTML 5 Cache	95
Section 38.1: Basic Example of HTML5 cache	95
Chapter 39: HTML Event Attributes	96
Section 39.1: HTML Form Events	96
Section 39.2: Keyboard Events	96
Chapter 40: Character Entities	97
Section 40.1: Character Entities in HTML	97
Section 40.2: Common Special Characters	97
Chapter 41: ARIA	98
Section 41.1: role="presentation"	98
Section 41.2: role="alert"	98
Section 41.3: role="alertdialog"	98

[Click here to download full PDF material](#)