HTML 1: Overview

Chapter 2

Randy Connolly and Ricardo Hoar

Objectives

1. HTML Defined and its History
2. HTML Syntax
3. Semantic Markup
4. Structure of HTML
5. Quick Tour of HTML
6. HTML Semantic Elements

Section 1 of 6

HTML DEFINED + ITS HISTORY

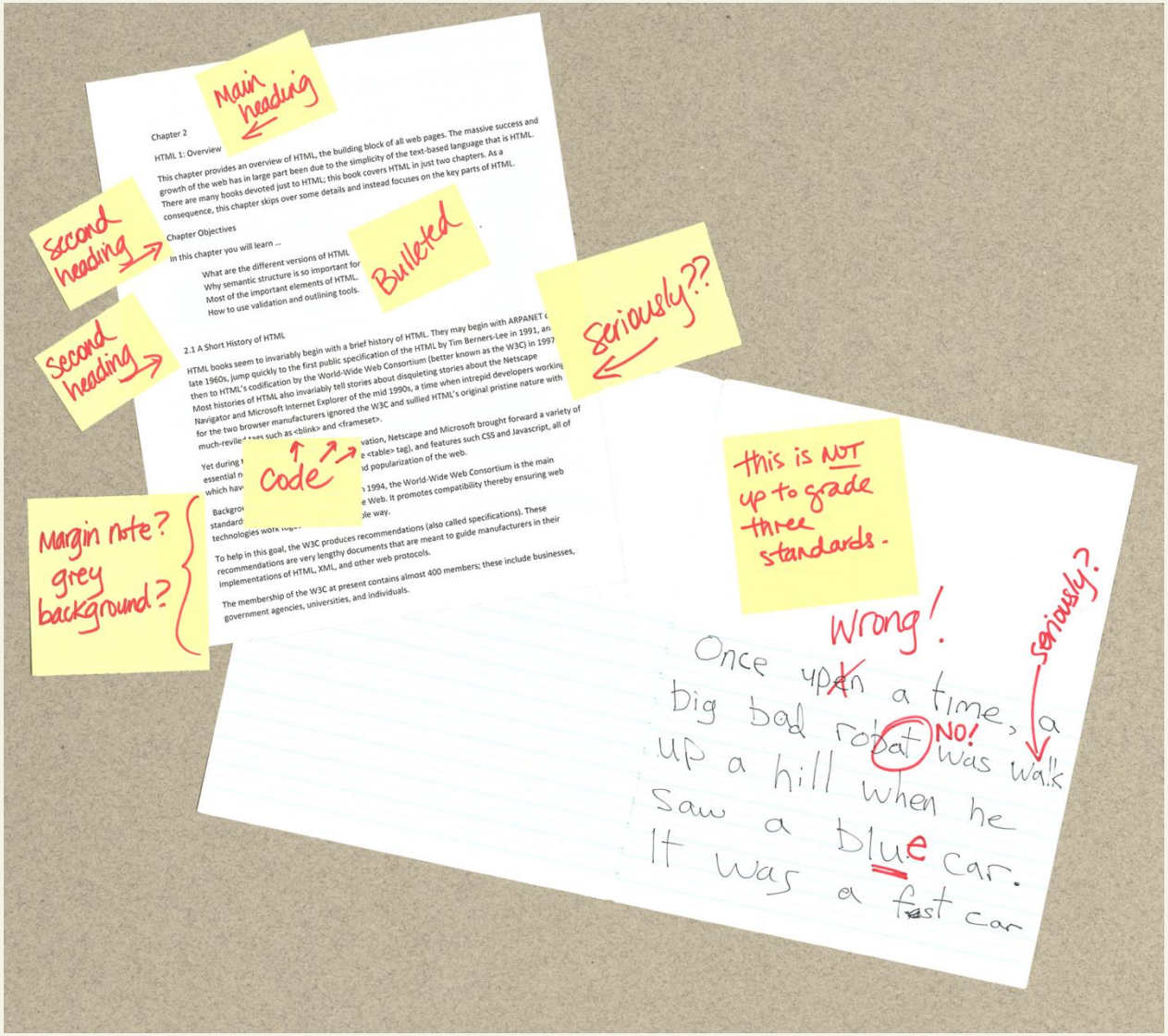
HTML Syntax

What is a markup language?

HTML is defined as a **markup language**.

- markup is a way of indicate information about the content.
- annotations distinct from the text being annotated.
- The term comes from the days of print, when editors would write instructions on manuscript pages that might be revision instructions to the author or copy editor.

Sample ad hoc markup



What is the W3C?

Standards

The W3C is the main standards organization for the World Wide Web.

To promotes compatibility the W3C produces **recommendations** (also called **specifications**).

In 1998, the W3C turned its attention to a new specification called XHTML 1.0, which was a version of HTML that used stricter XML (Extensible Markup Language) syntax rules.

XHTML

You too can be strict

The goal of XHTML with its strict rules was to make page rendering more predictable by forcing web authors to create web pages without syntax errors.

The XML-based syntax rules for XHTML are pretty easy to follow.

The main rules are:

- lowercase tag names,
- attributes always within quotes,
- and all elements must have a closing element (or be self-closing).

XHTML

Two versions

To help web authors, two versions of XHTML were created:

XHTML 1.0 Strict and **XHTML 1.0 Transitional**.

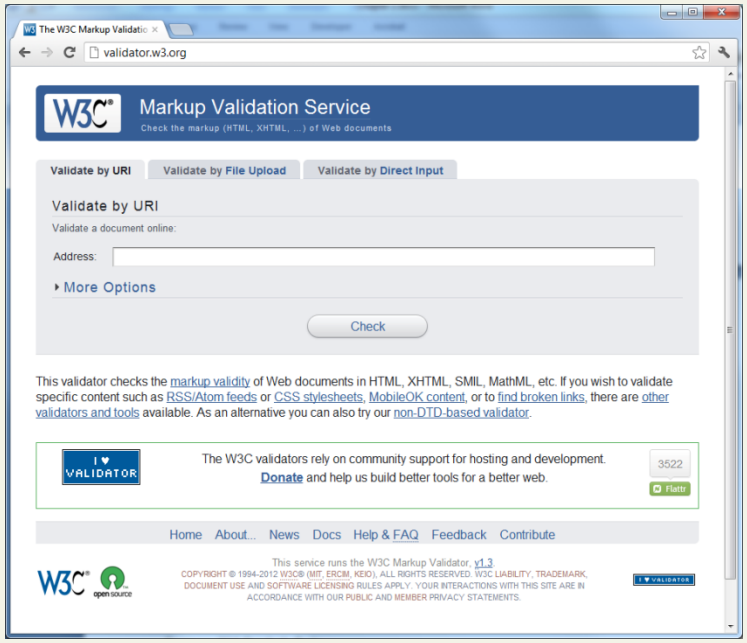
- The **strict** version was meant to be rendered by a browser using the strict syntax rules and tag support described by the W3C XHTML 1.0 Strict specification.
- The **transitional** recommendation is a more forgiving flavor of XHTML, and was meant to act as a temporary transition to the eventual global adoption of XHTML Strict.

During much of the 2000s, the focus in the professional web development community was on standards: that is, on limiting oneself to the W3C specification for XHTML.

Validators

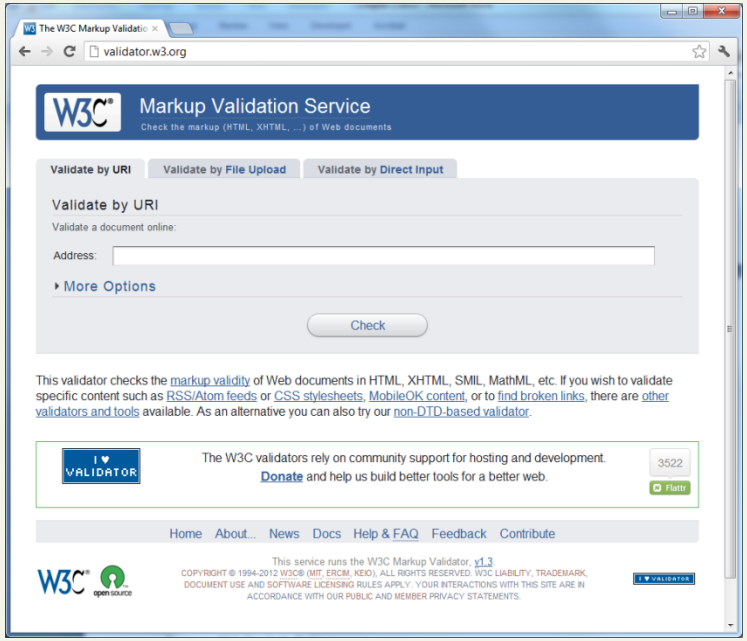
How to ensure your pages follow a standard

A key part of the standards movement in the web development community of the 2000s was the use of **HTML Validators** as a means of verifying that a web page's markup followed the rules for XHTML transitional or strict.



How about an example

Only if you have an internet connection



Open a web browser to the W3C validator and find a few websites to test.

Type the URL into the bar, and you can check if the home page is valid against various standards (or auto-detect)

XHTML 2.0 and WHATWG

Where did it go?

In the mid 2000s, XHTML 2.0 proposed a revolutionary and substantial change to HTML.

- backwards compatibility with HTML and XHTML 1.0 was dropped.
- Browsers would become significantly less forgiving of invalid markup.

At around the same time, a group of developers at Opera and Mozilla formed the **WHATWG** (Web Hypertext Application Technology Working Group) group within the W3C.

This group was not convinced that the W3C's embrace of XML and its abandonment of backwards-compatibility was the best way forward for the web.

HTML5

Three main aims

By 2009, the W3C stopped work on XHTML 2.0 and instead adopted the work done by WHATWG and named it HTML5.

There are three main aims to HTML5:

- Specify unambiguously how browsers should deal with invalid markup.
- Provide an open, non-proprietary programming framework (via Javascript) for creating rich web applications.
- Be backwards compatible with the existing web.

HTML5

It evolves

All of the major browser manufacturers have embraced HTML5.

Certainly not all browsers and all versions support every feature of HTML5.

This is in fact by design. HTML in HTML5 is now a living language: that is, it is a language that evolves and develops over time.

As such, every browser will support a gradually increasing subset of HTML5 capabilities

Section 2 of 6

HTML SYNTAX

HTML Syntax

Elements and Attributes

- **HTML documents** are composed of textual content and **HTML elements**
- **HTML element** encompasses
  - the **element name** within angle brackets (i.e., the **tag**) and
  - HTML elements can also contain **attributes**.
  - **the content** within the tag.

HTML Syntax

Elements and Attributes

An **empty element** does not contain any text content; instead, it is an instruction to the browser to do something.

- In XHTML, empty elements had to be terminated by a trailing slash.
- In HTML5, the trailing slash in empty elements is optional.

Nesting HTML elements

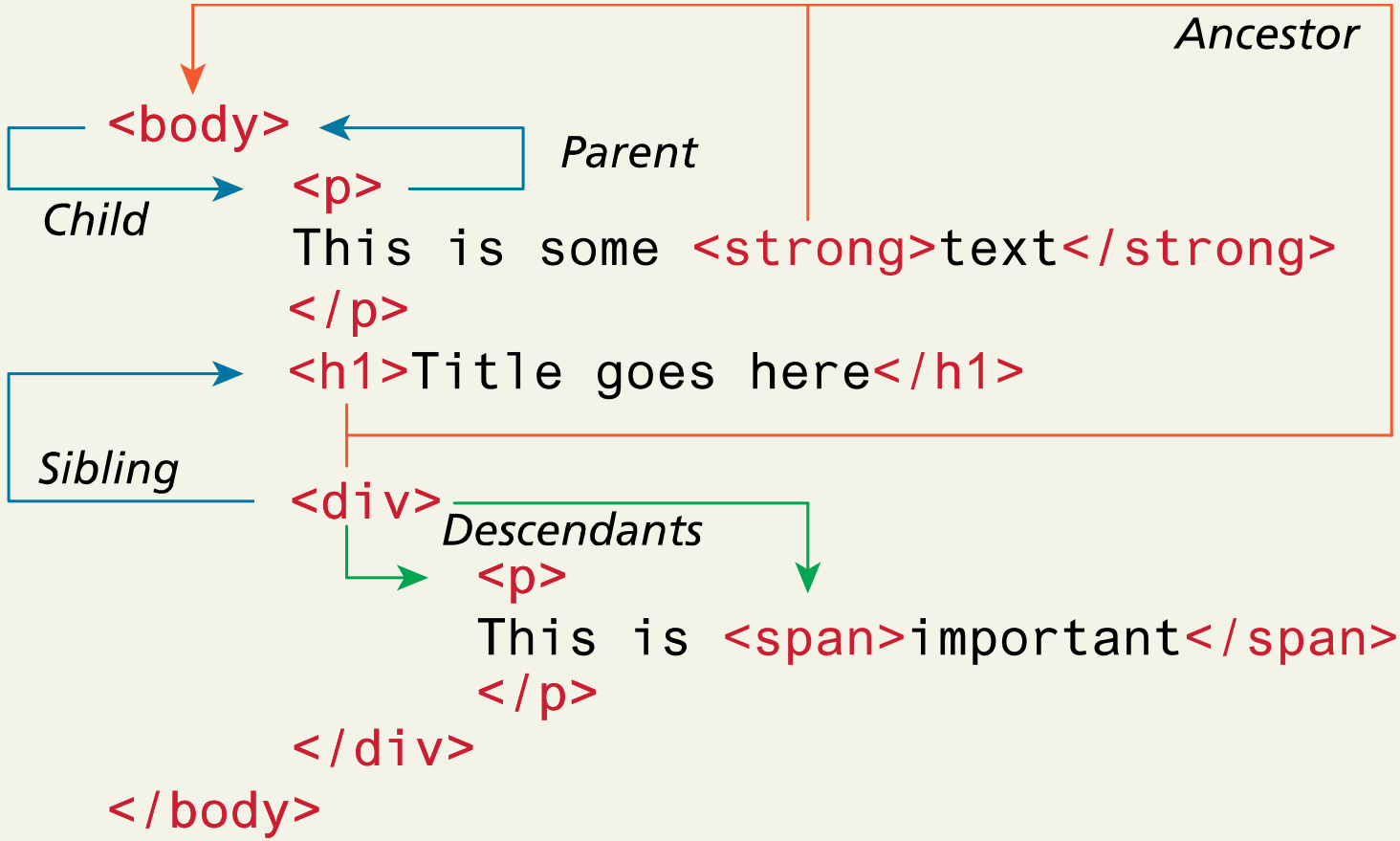
Often an HTML element will contain other HTML elements.

In such a case, the container element is said to be a parent of the contained, or child, element.

Any elements contained within the child are said to be **descendents** of the parent element; likewise, any given child element, may have a variety of **ancestors**.

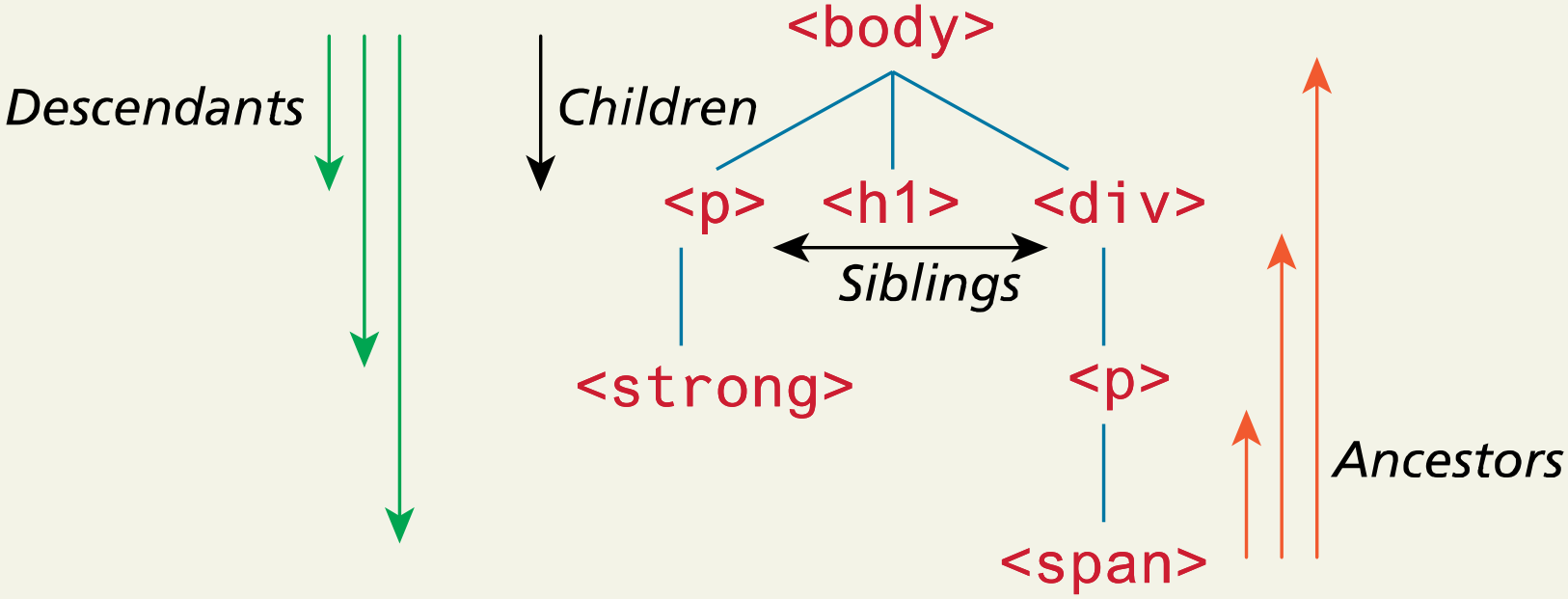
HTML Syntax

Nesting HTML Elements



HTML Syntax

Nesting HTML Elements

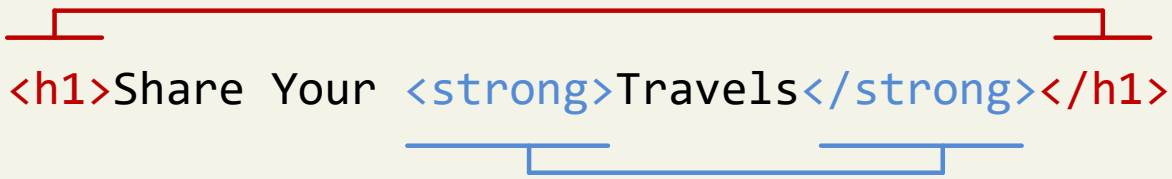


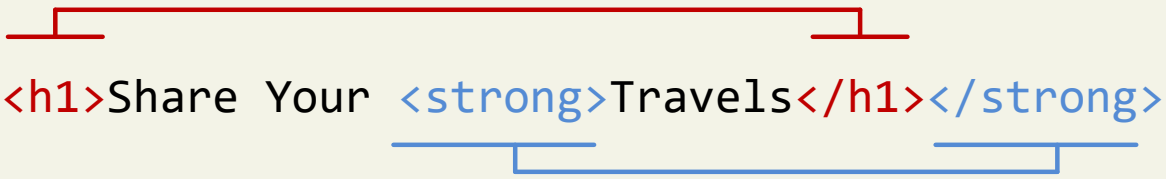
Nesting HTML elements

In order to properly construct a hierarchy of elements, your browser expects each HTML nested element to be properly nested.

That is, a child's ending tag must occur before its parent's ending tag.

Correct Nesting





Incorrect Nesting

Section 3 of 6

SEMANTIC MARKUP

Semantic Markup

What does it mean?

Over the past decades, a strong and broad consensus has grown around the belief that HTML documents should **only** focus on the structure of the document.

Information about how the content should look when it is displayed in the browser is best left to CSS (Cascading Style Sheets).

As a consequence, beginning HTML authors are often counseled to create **semantic HTML** documents.

That is, an HTML document should not describe how to visually present content, but only describe its content's structural semantics or meaning.

Structure

Structure is a vital way of communicating information in paper and electronic documents.

All of the tags that we will examine in this presentation are used to describe the basic structural information in a document, such as articles, headings, lists, paragraphs, links, images, navigation, footers, and so on.

Semantic Markup

Its advantages

Eliminating presentation-oriented markup and writing semantic HTML markup has a variety of important advantages:

Maintainability. Semantic markup is easier to update and change than web pages that contain a great deal of presentation markup.

Faster. Semantic web pages are typically quicker to author and faster to download.

Accessibility. Visiting a web page using voice reading software can be a very frustrating experience if the site does not use semantic markup.

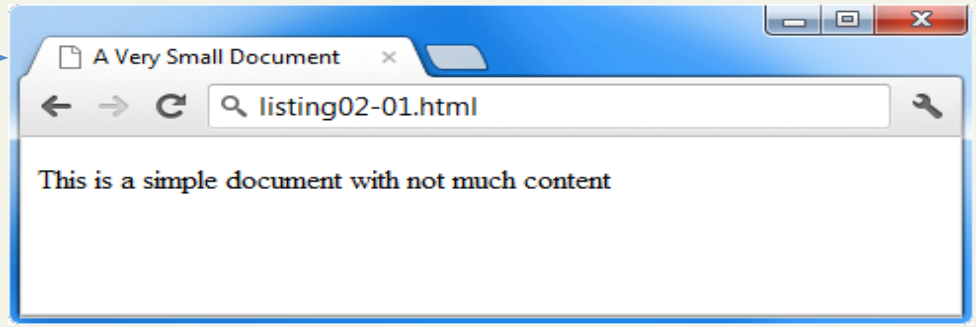
Search engine optimization. Semantic markup provides better instructions for search engines: it tells them what things are important content on the site.

Section 4 of 6

STRUCTURE OF HTML

Simplest HTML document

```
<!DOCTYPE html>
<title>A Very Small Document</title>
<p>This is a simple document with not much content</p>
```



DOCTYPE Short for **Document Type Definition** tells the browser what type of document it is about to process

The <title> element (Item 1) is used to provide a broad description of the content. The title is not displayed within the browser window. Instead, the title is typically displayed by the browser in its window and/or tab.

A more complete document

1 DOCTYPE

(short for **Document Type Definition**)

Tells the browser (or any other client software that is reading this HTML document) what type of document it is about to process.

Notice that it does not indicate what version of HTML is contained within the document: it only specifies that it contains HTML.

HTML, Head, and Body

HTML5 does not require the use of the **<html>**, **<head>**, and **<body>**.

However, in XHTML they were required, and most web authors continue to use them.

2 The **<html>** element is sometimes called the **root element** as it contains all the other HTML elements in the document.

Head and Body

HTML pages are divided into two sections: the **head** and the **body**, which correspond to the <head> and <body> elements.

3. The head contains descriptive elements *about* the document (title, style sheets, JavaScript files etc.)

4. The body contains content that will be displayed by the browser.

Inside the head

There are no brains

You will notice that the <head> element contains a variety of additional elements.

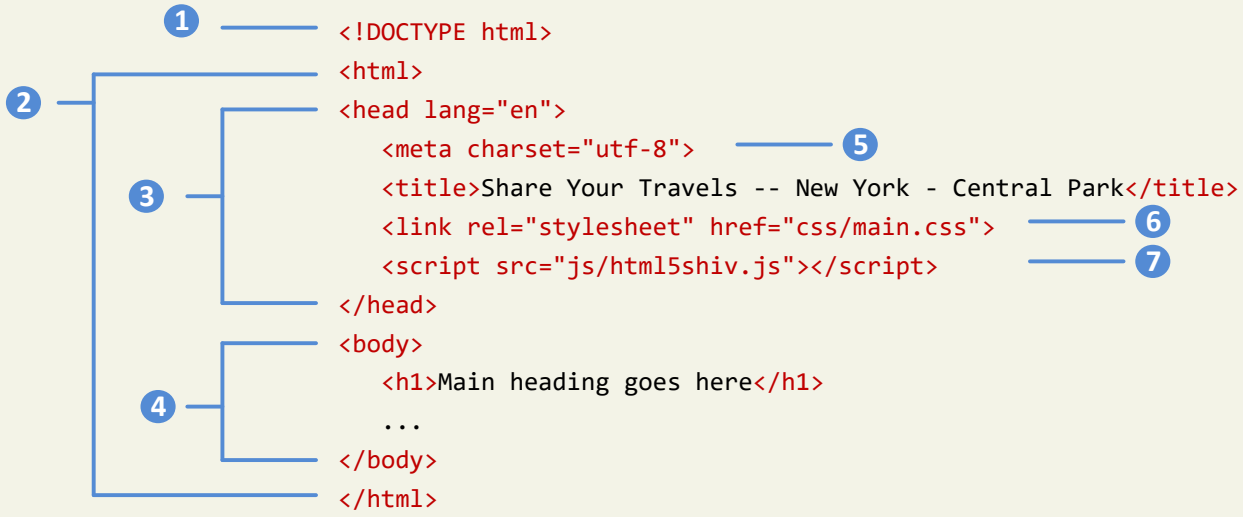
5. The first of these is the **<meta>** element. Our example declares that the character encoding for the document is UTF-8.

Inside the head

No brains but metas, styles and javascripts

6. Our example specifies an external CSS style sheet file that is used with this document.

7. It also references an external Javascript file.



Section 5 of 6

QUICK TOUR OF HTML

Why a quick tour?

HTML5 contains many structural elements – too many to completely cover in this presentation.

Rather than comprehensively cover all these elements, this presentation will provide a quick overview of the most common elements.

Sample Document

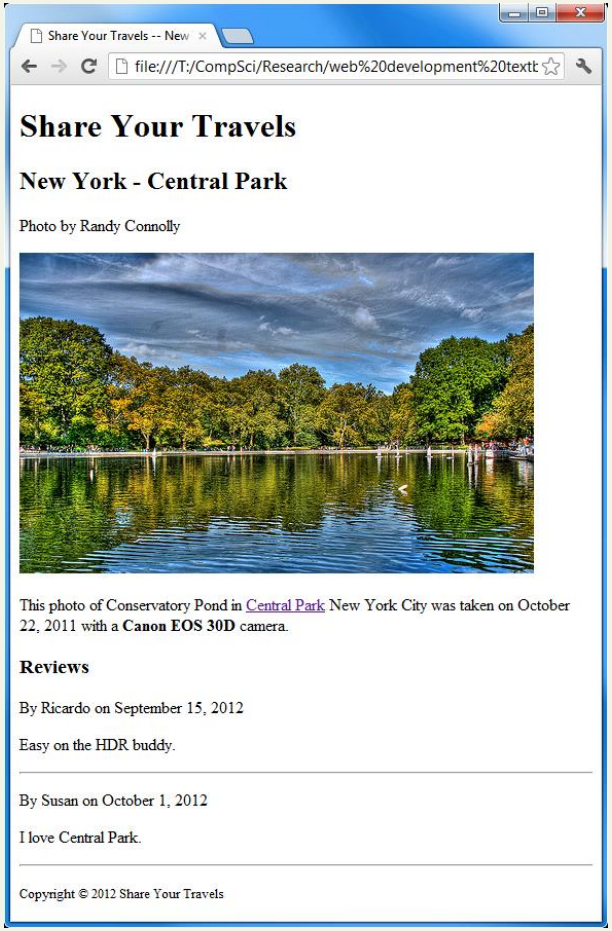
```
<body>
   <h1>Share Your Travels</h1>
   <h2>New York - Central Park</h2>
   <p>Photo by Randy Connolly</p>
   <p>This photo of Conservatory Pond in
      <a href="http://www.centralpark.com/">Central Park</a>
      New York City was taken on October 22, 2011 with a
      <strong>Canon EOS 30D</strong> camera.
   </p>
   <img src="images/central-park.jpg" alt="Central Park" />

   <h3>Reviews</h3>
   <div>
      <p>By Ricardo on <time>September 15, 2012</time></p>
      <p>Easy on the HDR buddy.</p>
   </div>

   <div>
      <p>By Susan on <time>October 1, 2012</time></p>
      <p>I love Central Park.</p>
   </div>

   <p><small>Copyright &copy; 2012 Share Your Travels</small></p>
</body>
```

1 2 3 4 5 6 7 8 9

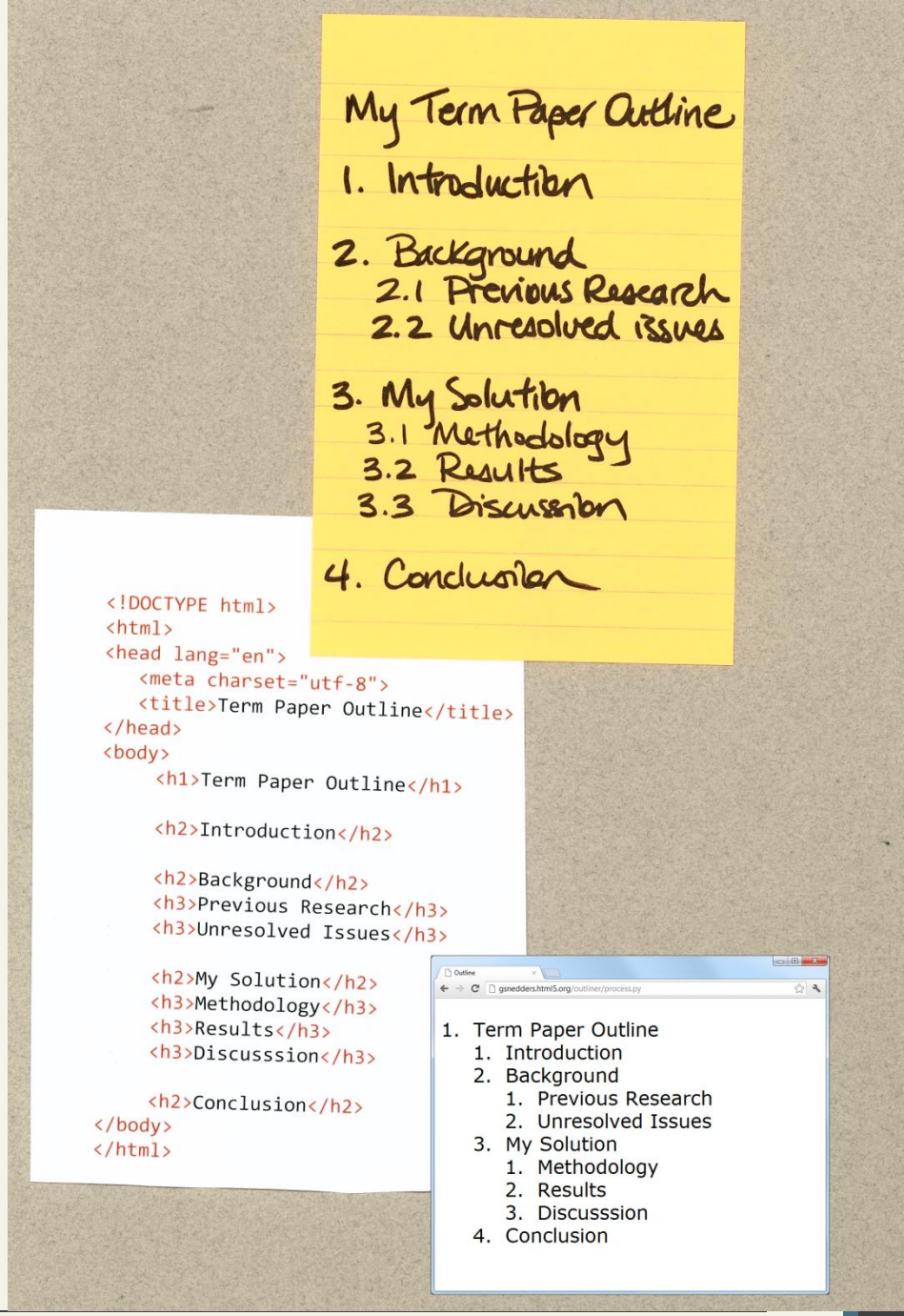


1 Headings

<h1>, <h2>, <h3>, etc

HTML provides six levels of heading (**h1, h2, h3**, …), with the higher heading number indicating a heading of less importance.

Headings are an essential way for document authors use to show their readers the structure of the document.



Headings

The browser has its own default styling for each heading level.

However, these are easily modified and customized via CSS.

Headings

Be semantically accurate

In practice, specify a heading level that is semantically accurate.

Do not choose a heading level because of its default presentation

- e.g., choosing <h3> because you want your text to be bold and 16pt

Rather, choose the heading level because it is appropriate

- e.g., choosing <h3> because it is a third level heading and not a primary or secondary heading

2 Paragraphs

<p>

Paragraphs are the most basic unit of text in an HTML document.

Notice that the **<p>** tag is a container and can contain HTML and other **inline HTML elements**

Inline HTML elements refers to HTML elements that do not cause a paragraph break but are part of the regular “flow” of the text.

6 Divisions

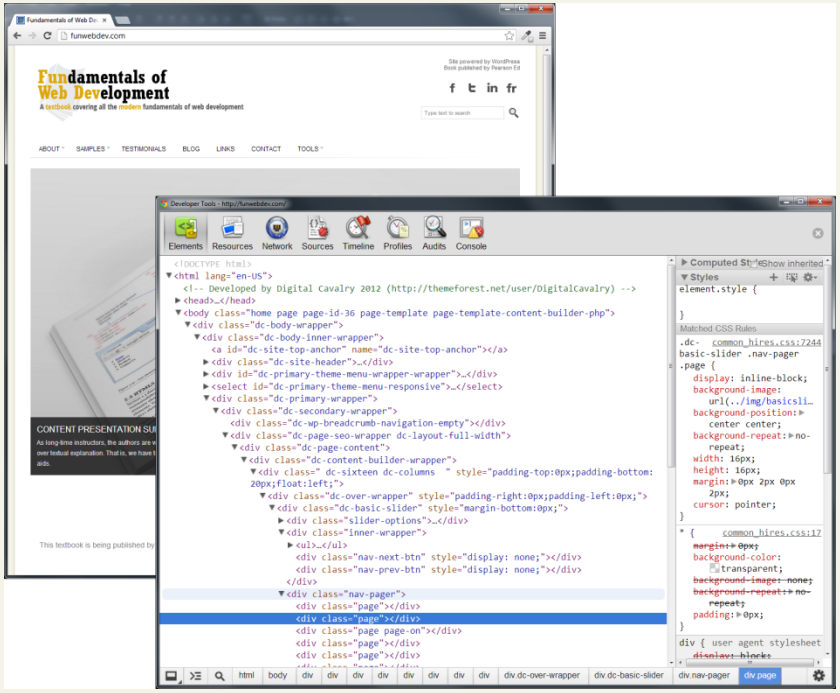
This ** ** tag is also a container element and is used to create a logical grouping of content

- The element has no intrinsic presentation.
- It is frequently used in contemporary CSS-based layouts to mark out sections.

Using div elements

Can you say "div-tastic"

HTML5 has a variety of new semantic elements (which we will examine later) that can be used to reduce somewhat the confusing mass of div within divs within divs that is so typical of contemporary web design.



3 Links

<a>

Links are created using the **<a>** element (the "a" stands for anchor).

A link has two main parts: the destination and the label.

```
<a href="http://www.centralpark.com">Central Park</a>
```

Destination

Label (text)

```
<a href="index.html"><img src="logo.gif" /></a>
```

Label (image)

Types of Links

You can use the anchor element to create a wide range of links:

- Links to external sites (or to individual resources such as images or movies on an external site).
- Links to other pages or resources within the current site.
- Links to other places within the current page.
- Links to particular locations on another page.
- Links that are instructions to the browser to start the user's email program.
- Links that are instructions to the browser to execute a Javascript function.

Quick Tour of HTML Elements

Links

Link to external site

```
<a href="http://www.centralpark.com">Central Park</a>
```

Link to resource on external site

```
<a href="http://www.centralpark.com/logo.gif">Central Park</a>
```

Link to another page on same site as this page

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

Link to another place on the same page

```
<a href="#top">Go to Top of Document</a>
...
<a name="top">
```

Defines anchor for a link to another place on same page

Quick Tour of HTML Elements

Links (continued)

Link to specific place on another page

```
<a href="productX.html#reviews">Reviews for product X</a>
```

Link to email

```
<a href="mailto:person@somewhere.com">Someone</a>
```

Link to JavaScript function

```
<a href="javascript:OpenAnnoyingPopup();">See This</a>
```

Link to telephone (automatically dials the number when user clicks on it using a smartphone browser)

```
<a href="tel:+18009220579">Call toll free (800) 922-0579</a>
```

Link Text

Some guidance … or … don't "Click Here"

Links with the label "Click Here" were once a staple of the web.

Today, such links are frowned upon, since:

- they do not tell users where the link will take them
- as a verb "click" is becoming increasingly inaccurate when one takes into account the growth of mobile browsers.

Instead, textual link labels should be descriptive.

~~"Click here to see the race results"~~

"Race Results" or **"See Race Results".**

Uniform Resource Identifier
(URI)

Each information resource on the Web has an address which can be codified by a URI.

A URI is defined as

<scheme>:<scheme-specific-part>

The <scheme> is the name of the mechanism used to access the resource (for instance, http protocol)

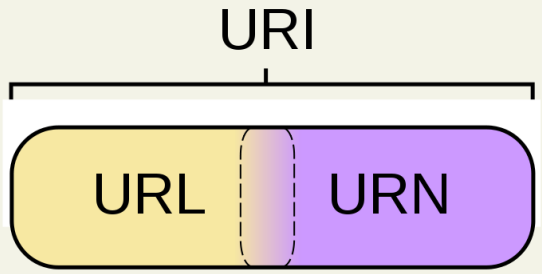
- A colon character (:)
- A scheme-specific part.
  - For instance
  - The name of the machine hosting the resource (for instance, www.w3.org)
  - The name of the resource itself, given as path

Uniform Resource Identifier
(URI)

A **URI** may be a locator (URL) or a name (URN), or both.

A Uniform Resource Locator (URL) is a URI that, in addition to identifying a resource, specifies the means of acting upon and obtaining the representation.

For example, the URL http://www.w3.org/TR



Uniform Resource Identifier
(URI)

A Uniform Resource Name (URN) is a URI that identifies a resource by name, in a particular namespace. The resource does not need to necessarily be network homed.

For example, the URN *urn:isbn:0-395-36341-1* is a URI that specifies the identifier system, i.e. International Standard Book Number (ISBN), as well as the unique reference within that system and allows one to talk about a book, but does not suggest where and how to obtain an actual copy of it.

URN - person's name

URL – person's street address.

URL Absolute Referencing

For external resources

When referencing a page or resource on an external site, a full **absolute reference** is required: that is,

- the protocol (typically, http://),
- the domain name,
- any paths, and then finally
- the file name of the desired resource.

URL Relative Referencing

An essential skill

We also need to be able to successfully reference files within our site.

This requires learning the syntax for so-called **relative referencing**.

When referencing a resource that is on the same server as your HTML document, then you can use briefer relative referencing.

If the URL does not include the "http://" then the browser will request the current server for the file.

URL Relative Referencing

If all the resources for the site reside within the same **directory** (also referred to as a **folder**), then you can reference those other resources simply via their filename.

However, most real-world sites contain too many files to put them all within a single directory.

For these situations, a relative pathname is required along with the filename.

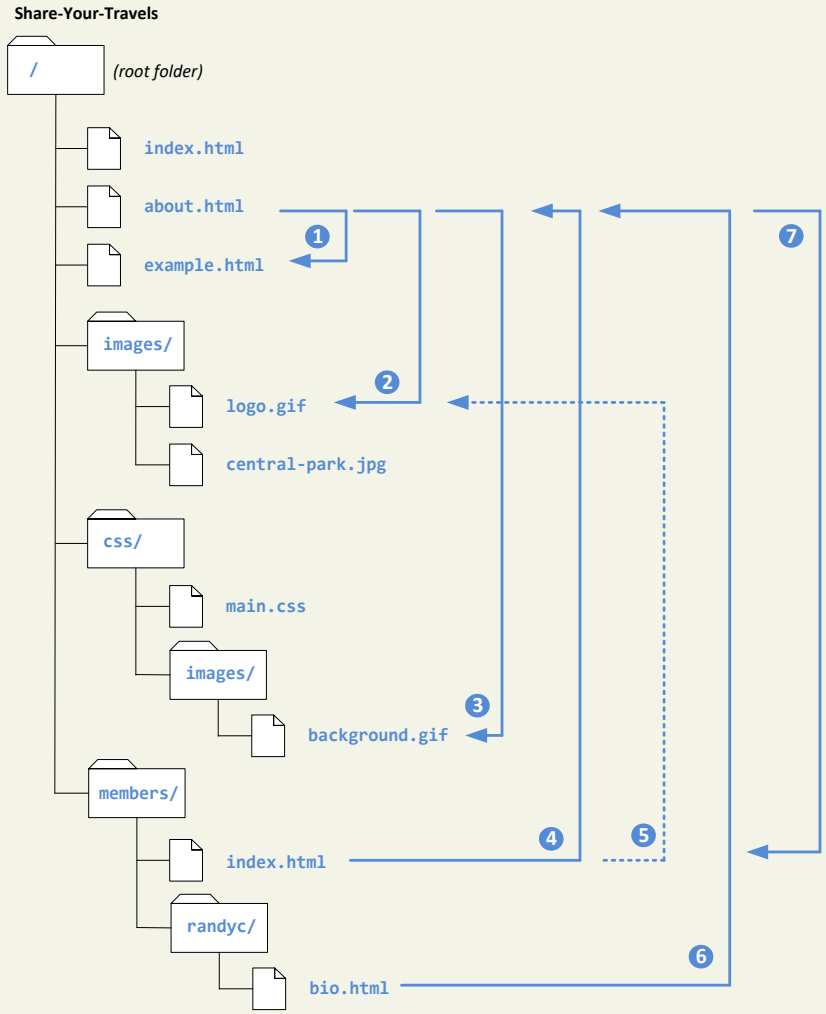
The **pathname** tells the browser where to locate the file on the server.

Pathnames

Pathnames on the web follow Unix conventions.

- Forward slashes (“/”) are used to separate directory names from each other and from file names.
- Double-periods (“..”) are used to reference a directory “above” the current one in the directory tree.

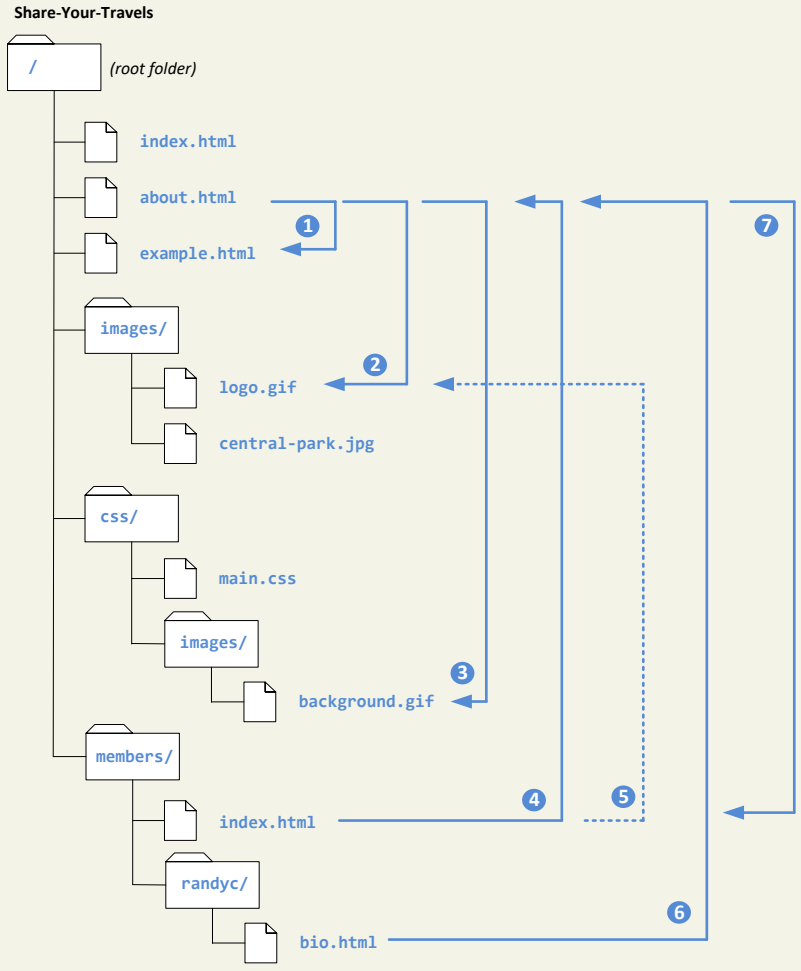
URL Relative Referencing



	Relative Link Type	Example
1	**Same Directory** To link to a file within the same folder, simply use the file name.	To link to example.html from about.html (in Figure 2.18), use: `<a href="example.html">`
2	**Child Directory** To link to a file within a subdirectory, use the name of the subdirectory and a slash before the file name.	To link to logo.gif from about.html, use: `<a href="images/logo.gif">`
3	**Grandchild/Descendant Directory** To link to a file that is multiple subdirectories *below* the current one, construct the full path by including each subdirectory name (separated by slashes) before the file name.	To link to background.gif from about.html, use: `<a href="css/images/background.gif">`

4	**Parent/Ancestor Directory**
	Use "../" to reference a folder *above* the current one. If trying to reference a file several levels above the current one, simply string together multiple "../".
		To link to about.html from index.html in members, use:
		`<a href="../about.html">`
		To link to about.html from bio.html, use:
		`<a href="../../about.html">`

URL Relative Referencing



5 Sibling Directory

Use "../"to move up to the appropriate level, and then use the same technique as for child or grandchild directories.

To link to logo.gif from index.html in members, use:

```
<a href="../images/logo.gif">
```

To link to background.gif from bio.html, use:

```
<a
href="../../css/images/background.gif">
```

6 Root Reference

An alternative approach for ancestor and sibling references is to use the so-called **root reference** approach. In this approach, begin the reference with the root reference (the "/") and then use the same technique as for child or grandchild directories. **Note that these will only work on the server! That is, they will not work when you test it out on your local machine.**

To link to about.html from bio.html, use:

```
<a href="/about.html">
```

To link to background.gif from bio.html, use:

```
<a href="/images/background.gif">
```

7 Default Document

Web servers allow references to directory names without file names. In such a case, the web server will serve the default document, which is usually a file called index.html (apache) or default.html (IIS). **Again, this will only generally work on the web server.**

To link to index.html in members from about.html, use either:

```
<a href="members">
```

Or

```
<a href="/members">
```

Inline Text Elements

Do not disrupt the flow

Inline elements do not disrupt the flow of text (i.e., go to a new block).

HTML5 defines over 30 of these elements.

- **<a>**
- **<abbr>**
- **
**
- **<cite>**
- **<code>**
- ****
- **<mark>**
- **<small>**
- ****
- ****
- **<time>**

Images

While the **<img>** tag is the oldest method for displaying an image, it is not the only way.

For purely decorative images, such as background gradients and patterns, logos, border art, and so on, it makes semantic sense to keep such images out of the markup and in CSS where they more rightly belong.

But when the images are **content**, such as in the images in a gallery or the image of a product in a product details page, then the <img> tag is the semantically appropriate approach.

Images

Specifies the URL of the image to display (note: uses standard relative referencing).

Text in `title` attribute will be displayed in a pop-up tool tip when user moves mouse over image.

```
<img src="images/central-park.jpg" alt="Central Park" title="Central Park" width="80" height="40" />
```

Text in `alt` attribute provides a brief description of image's content for users who are unable to see it.

Specifies the width and height of image in pixels

Lists

HTML provides three types of lists

Unordered lists. Collections of items in no particular order; these are by default rendered by the browser as a bulleted list.

Ordered lists. Collections of items that have a set order; these are by default rendered by the browser as a numbered list.

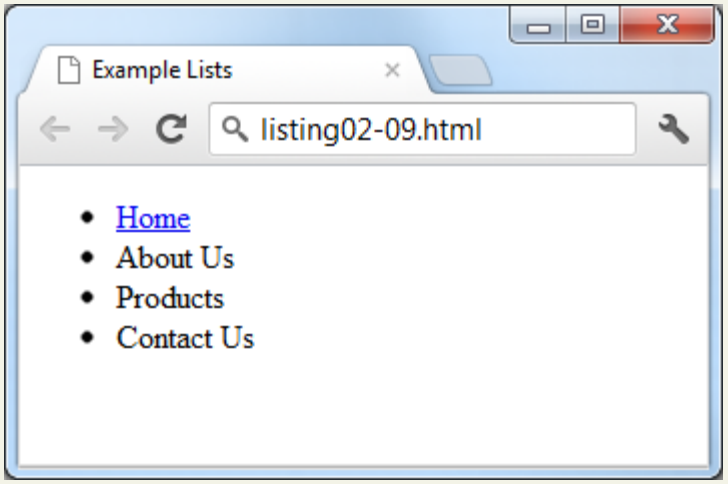
Definition lists. Collection of name and definition pairs. These tend to be used infrequently. Perhaps the most common example would be a FAQ list.

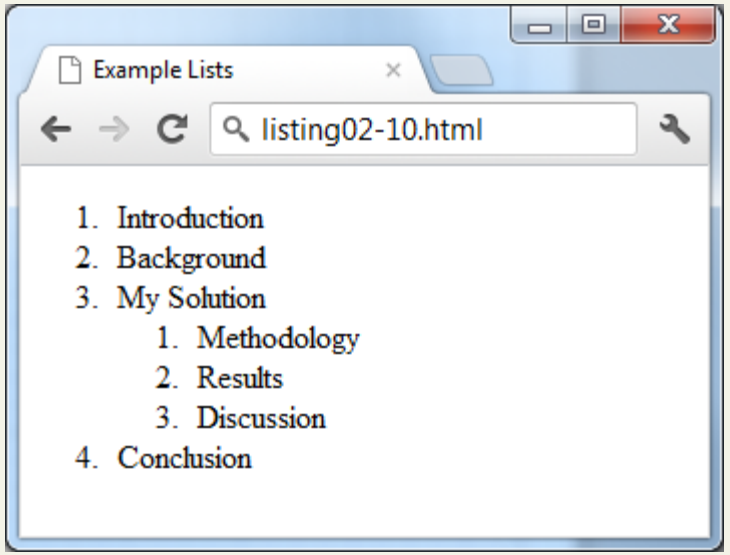
Lists

Notice that the list item element can contain other HTML elements

```
<ul>
   <li><a href="index.html">Home</a></li>
   <li>About Us</li>
   <li>Products</li>
   <li>Contact Us</li>
</ul>
```

```
<ol>
   <li>Introduction</li>
   <li>Background</li>
   <li>My Solution</li>
   <li>
      <ol>
         <li>Methodology</li>
         <li>Results</li>
         <li>Discussion</li>
      </ol>
   </li>
   <li>Conclusion</li>
</ol>
```





Character Entities

These are special characters for symbols for which there is either no way easy way to type in via a keyboard (such as the copyright symbol or accented characters) or which have a reserved meaning in HTML (for instance the “<” or “>” symbols).

They can be used in an HTML document by using the entity name or the entity number.

Entity	Description
	Nonbreakable space
&lt;	<
&gt;	>
&copy;	©
&trade;	™

Section 6 of 6

HTML SEMANTIC ELEMENTS

HTML5 Semantic Elements

Why are they needed?

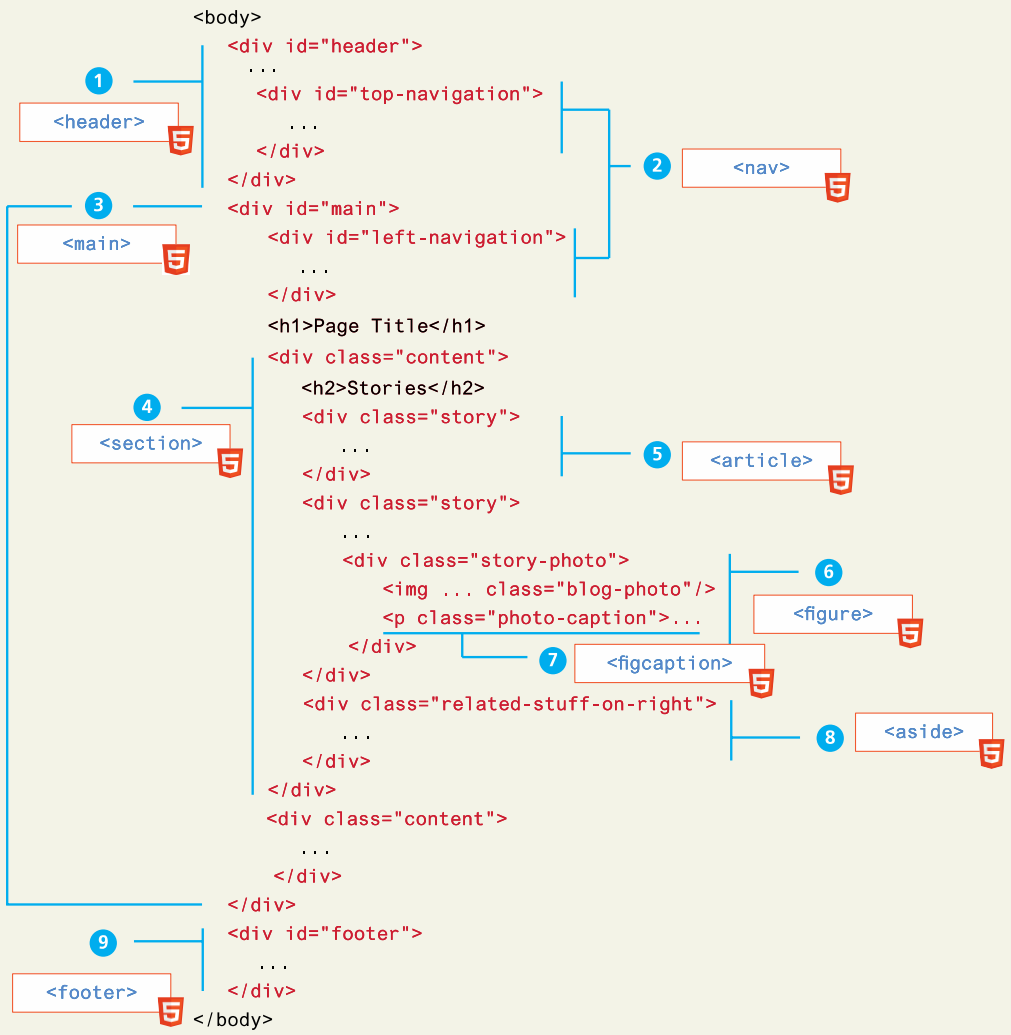
One substantial problem with modern, pre-HTML5 semantic markup:

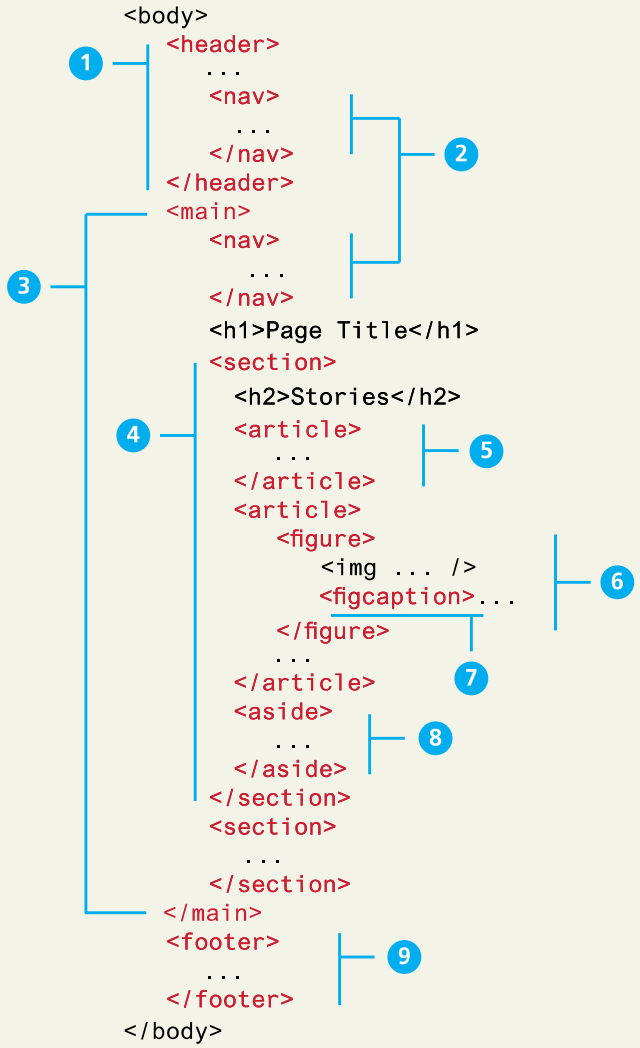
> most complex web sites are absolutely packed solid with elements.

Unfortunately, all these elements can make the resulting markup confusing and hard to modify.

Developers typically try to bring some sense and order to the chaos by using **id** or **class** names that provide some clue as to their meaning.

HTML5 Semantic Structure Elements





Header and Footer

<header> <footer>

Most web site pages have a recognizable header and footer section.

Typically, the **header** contains

- the site logo
- title (and perhaps additional subtitles or taglines)
- horizontal navigation links, and
- perhaps one or two horizontal banners.

Header and Footer

<header> <footer>

The typical footer contains less important material, such as

- smaller text versions of the navigation,
- copyright notices,
- information about the site's privacy policy, and
- perhaps twitter feeds or links to other social sites.

Header and Footer

Both the HTML5 <header> and <footer> element can be used not only for *page* headers and footers, they can also be used for header and footer elements within other HTML5 containers, such as <article> or <section>.

```
<header>
   <img src="logo.gif" alt="logo" />
   <h1>Fundamentals of Web Development</h1>
   ...
</header>
<article>
   <header>
       <h2>HTML5 Semantic Structure Elements </h2>
      <p>By <em>Randy Connolly</em></p>
      <p><time>September 30, 2012</time></p>
   </header>
   ...
</article>
```

Navigation

<nav>

The **<nav>** element represents a section of a page that contains links to other pages or to other parts within the same page.

Like the other HTML5 semantic elements, the browser does not apply any special presentation to the <nav> element.

The <nav> element was intended to be used for major navigation blocks, presumably the global and secondary navigation systems.

Navigation

```
<header>
   <img src="logo.gif" alt="logo" />
   <h1>Fundamentals of Web Development</h1>
   <nav>
      <ul>
         <li><a href="index.html">Home</a></li>
         <li><a href="about.html">About Us</a></li>
         <li><a href="browse.html">Browse</a></li>
      </ul>
   </nav>
</header>
```

HTML5 Semantic Structure Elements

Main

- **<main>** is meant to contain the main unique content of the document.
- <main> provides a semantic replacement for markup such as or
- The **<section>** element represents a section of a document, typically with a title or heading.
- The **<article>** element represents a section of content that forms an independent part of a document or site (could be read independently of other content); for example, a magazine or newspaper article, or a blog entry.

Sections versus Divs

How to decide which to use

The WHATWG specification warns readers that the <section> element is **not** a generic container element. HTML already has the element for such uses.

When an element is needed only for styling purposes or as a convenience for scripting, it makes sense to use the element instead.

Another way to help you decide whether or not to use the <section> element is to ask yourself if it is appropriate for the element's contents to be listed explicitly in the document's outline.

If so, then use a <section>, otherwise use a .

Figure and Figure Captions

<figure> <figcaption>

The W3C Recommendation indicates that the <figure> element can be used not just for images but for any type of *essential* content that could be moved to a different location in the page or document and the rest of the document would still make sense.

However...

The **<figure>** element should **not** be used to wrap every image.

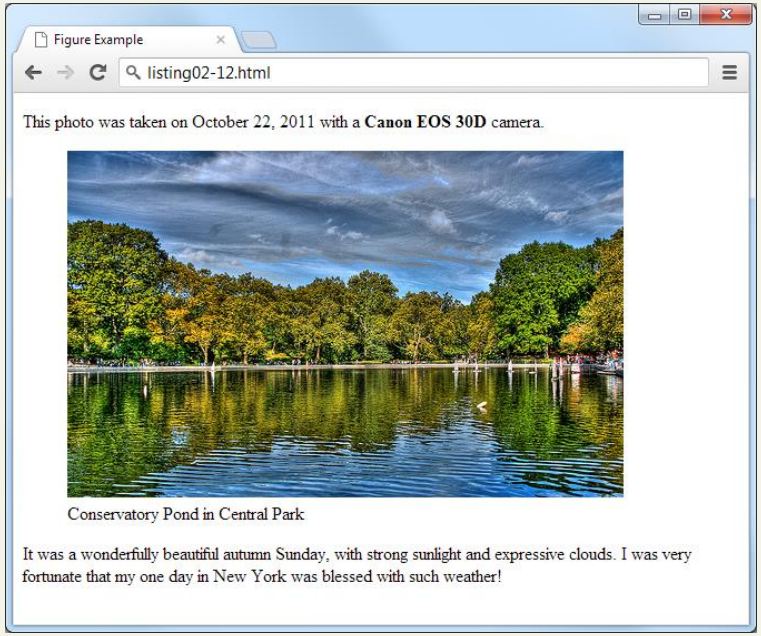
For instance, it makes no sense to wrap the site logo or non-essential images such as banner ads and graphical embellishments within <figure> elements.

Instead, only use the <figure> element for circumstances where the image (or other content) has a caption and where the figure is essential to the content but its position on the page is relatively unimportant.

Figure and Figure Captions

Figure could be moved to a different location in document …
But it has to exist in the document (i.e., the figure isn't optional)

```
<p>This photo was taken on October 22, 2011 with a Canon EOS 30D camera.</p>
<figure>
  <img src="images/central-park.jpg" alt="Central Park" /><br/>
  <figcaption>Conservatory Pond in Central Park</figcaption>
</figure>
<p>
It was a wonderfully beautiful autumn Sunday, with strong sunlight and
expressive clouds. I was very fortunate that my one day in New York was
blessed with such weather!
</p>
```



Aside

<aside>

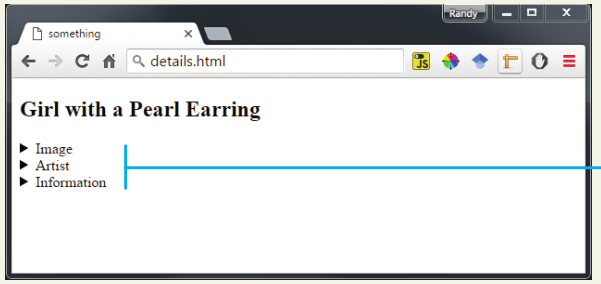
The **<aside>** element is similar to the **<figure>** element in that it is used for marking up content that is separate from the main content on the page.

But while the **<figure>** element was used to indicate important information whose location on the page is somewhat unimportant, the **<aside>** element "represents a section of a page that consists of content that is tangentially related to the content around the aside element."

The **<aside>** element could thus be used for sidebars, pull quotes, groups of advertising images, or any other grouping of non-essential elements.

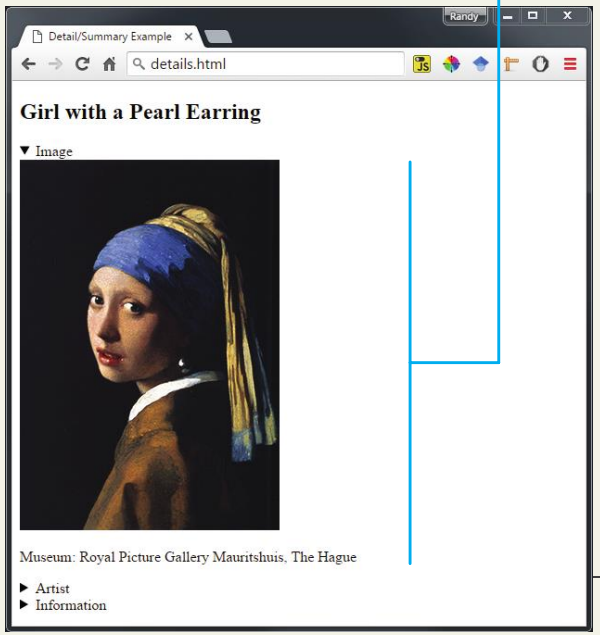
HTML5 Semantic Structure Elements

Details and Summary




```
<body>
   <h2>Girl with a Pearl Earring</h2>
   <details>
     <summary>Image</summary>
     <img src="images/106020.jpg"><br>
     <p>Museum: Royal Picture Gallery Mauritshuis ...
   </details>
   <details>
     <summary>Artist</summary>
     <p><strong>Jan Vermeer</strong> was a Dutch ...
   </details>
   <details>
     <summary>Information</summary>
     <p>
        Date: 1665<br>
        Medium: Oil on Canvas
     </p>
  </details>
</body>
```

Clicking on the summary label reveals the rest of the content with the <details> container



EXAMPLES

HTML 5.0 Document Structure

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <title> My first HTML document </title>
    <!-- A simple html document -->
  </head>
  <body>
    <p> Hello world! </p>
  </body>
</html>
```

Document Metadata

Meta Data -- information about a document

```
<head>
<meta charset="utf-8">
<meta name="author" content="Mario Rossi">
<meta name="description" content="A tutorial
about HTML">
<meta name="generator" content="Notepad++">
<meta name="keywords"
content="HTML,CSS,XML,JavaScript">
<title>HTML5 Example</title>
</head>
```

Possible values for name: application-name, author, description, generator, keywords

Document Metadata

When several meta elements provide language-dependent information, search engines may filter on the lang attribute to display search results using the language preferences of the user.

```
<!-- For speakers of US English -->
<meta name="keywords" lang="en-us"
content="vacation, Greece, sunshine">
<!-- For speakers of British English -->
<meta name="keywords" lang="en"
content="holiday, Greece, sunshine">
<!-- For speakers of French -->
<meta name="keywords" lang="fr"
content="vacances, Gr&egrave;ce, soleil">
```

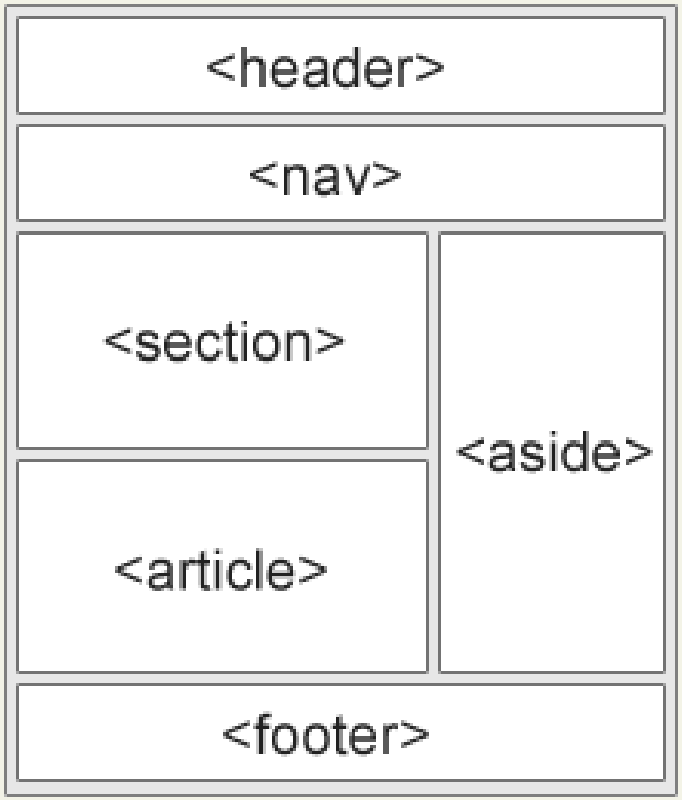
Document Metadata

The base element allows authors to specify the document base URL for the purposes of resolving relative URLs, and the name of the default browsing context for the purposes of following hyperlinks. The element does not represent any content beyond this information.

```
<!DOCTYPE html>
<html>   <head>
      <meta charset="utf-8">
       <title>This is an example for the &lt;base&gt; element</title>
       <base href="http://www.example.com/news/index.html">
   </head>
   <body>
       <p>Visit the <a href="archives.html">archives</a>.</p>
   </body>
</html>
```

The link in the above example would be a link to "http://www.example.com/news/archives.html".

Sections & Document Organization



Address element

Address Element

The address element represents the contact information for its nearest article or body element ancestor. If that is the body element, then the contact information applies to the document as a whole.

The address element must not be used to represent arbitrary addresses (e.g. postal addresses), unless those addresses are in fact the relevant contact information.

Typically, the address element would be included along with other information in a footer element.

```
<address>
  For more details, contact
  <a href="mailto:js@example.com">John Smith</a>.
 </address>
```

Example

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <title>HTML5</title>
 </head>
<body>
  <header>
     <h1>Header</h1>
     <h2>Subtitle</h2>
     <h4>HTML5 Rocks!</h4>
  </header>
```

Example

```
<div id="container">
      <nav>
       <h3>Nav</h3>
       <a href="">Link 1</a>
       <a href="">Link 2</a>
       <a href="">Link 3</a>
      </nav>

      <section>
         <article>
            <header>
               <h1>Article Header</h1>
            </header>
```

Example

```
        <p>HTML5: "Lorem ipsum dolor nunc aut nunquam sit amet,
        consectetur adipiscing elit. Vivamus at est eros, vel fringilla urna.
        Pellentesque odio</p>
       <footer>

          <h2>Article Footer</h2>

       </footer>

    </article>

</section>


<aside>

    <h3>Aside</h3>

    <p>HTML5: "Lorem ipsum dolor nunc aut nunquam sit amet,
    consectetur adipiscing elit. Vivamus at est eros, vel fringilla urna.
    Pellentesque odio rhoncus</p>

</aside>
```

Example

```
    <footer>
        <h2>Footer</h2>
            <address>
            For more details, contact  <a id="mail"
                        href="mailto:js@example.com">John Smith</a>.
            </address>
            <p><small>© copyright 2014 Example Corp.</small></p>
     </footer>
   </div>
</body>

</html>
```

Example with Styles

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8" />
  <title>HTML5</title>
<link rel="stylesheet" href="html5.css">

  </head>
   <body>
   <header>
     <h1>Header</h1>
     <h2>Subtitle</h2>
     <h4>HTML5 Rocks!</h4>
   </header>
```

Grouping content elements

Grouping Content

Element p

The **p** element represents a paragraph.

The p element should not be used when a more specific element is more appropriate.

```
<section>
 <!-- ... -->
 <p>Last modified: 2001-04-23</p>
 <p>Author: fred@example.com</p>
</section>
```

Technically correct, but not appropriate

```
<section>
 <!-- ... -->
 <footer>Last modified: 2001-04-23</footer>
 <address>Author: fred@example.com</address>
</section>
```

Better

Grouping Content

Element hr

The **hr** element represents a paragraph-level thematic break, e.g. a scene change in a story, or a transition to another topic within a section of a reference book

There is no need for an hr element between the sections themselves, since the section elements and the h1 elements imply thematic changes themselves.

Grouping Content

Element pre

The **pre** element represents a block of preformatted text, in which structure is represented by typographic conventions rather than by elements.

Some examples of cases where the pre element could be used:

- Including an e-mail, with paragraphs indicated by blank lines, lists indicated by lines prefixed with a bullet, and so on.
- Including fragments of computer code, with structure indicated according to the conventions of that language.
- Displaying ASCII art.

```
<p>This is the <code>Panel</code> constructor:</p>
<pre><code>function Panel(element, canClose, closeHandler) {
  this.element = element;
  this.canClose = canClose;
  this.closeHandler = function () { if (closeHandler) closeHandler() };
}</code></pre>
```

Grouping Content

Element blockquote

The **blockquote** element represents content that is quoted from another source, optionally with a citation which must be within a footer or cite element, and optionally with in-line changes such as annotations and abbreviations.

```
<blockquote>
 The people recognize themselves in their commodities; they find their
 soul in their automobile, hi-fi set, split-level home, kitchen equipment.
 — <cite><a href="http://en.wikipedia.org/wiki/Herbert_Marcuse">Herbert Marcuse</a></cite>
 </blockquote>
```

Grouping Content

Element figure

The **figure** element represents some flow content, optionally with a caption, that is self-contained (like a complete sentence) and is typically referenced as a single unit from the main flow of the document.

The element can thus be used to annotate illustrations, diagrams, photos, code listings, etc.

The first **figcaption** element child of the element, if any, represents the caption of the figure element's contents. If there is no child figcaption element, then there is no caption.

A figure element's contents are part of the surrounding flow. If the purpose of the page is to display the figure, for example a photograph on an image sharing site, the figure and figcaption elements can be used to explicitly provide a caption for that figure.

Grouping Content

Element figure

Some uses

```
<figure>
 <p>'Twas brillig, and the slithy toves<br>
 Did gyre and gimble in the wabe;<br>
 All mimsy were the borogoves,<br>
 And the mome raths outgrabe.</p>
 <figcaption><cite>Jabberwocky</cite> (first verse). Lewis Carroll, 1832-98</figcaption>
</figure>
```

```
<figure>
  <figcaption>Oil-based paint on canvas. Maria Towle, 1858.</figcaption>
  <img src="castle1858.jpeg" alt="The castle now has two towers and two walls.">
 </figure>
```

Grouping Content

Element div

The **div** element has no special meaning at all. It represents its children. It can be used with the **class**, **lang**, and other attributes to mark up semantics common to a group of consecutive elements.

Example: div used to set the language of two paragraphs at once

```
<div lang="en-GB">
  <p>My other cat, coloured black and white, is a sweetie. He followed
  us to the pool today, walking down the pavement with us. Yesterday
  he apparently visited our neighbours. I wonder if he recognises that
  their flat is a mirror image of ours.</p>
  <p>Hm, I just noticed that in the last paragraph I used British
  English. But I'm supposed to write in American English. So I
  shouldn't say "pavement" or "flat" or "colour"...</p>
 </div>
```

Grouping Content

Element main

The **main** element represents the main content of the body of a document or application. The main content area consists of content that is directly related to or expands upon the central topic of a document or central functionality of an application.

The main content area of a document includes content that is unique to that document and excludes content that is repeated across a set of documents such as site navigation links, copyright information, site logos and banners and search forms (unless the document or applications main function is that of a search form).

Authors must not include more than one main element in a document.

```
<main>
  <h1>Skateboards</h1>
  <p>The skateboard is the way cool kids get around</p>
  <article>  </article>
  <article>  </article>
</main>
```

Grouping Content

Unordered Lists

HTML supports **ordered**, **unordered** and **definition** lists.

Unordered lists

An unordered list starts with the <ul> tag.

Each list item starts with the <li> tag.

The list items are marked with bullets (typically small black circles).

```
<p>I have lived in the following countries:</p>
<ul>
 <li>Norway
 <li>Switzerland
 <li>United Kingdom
 <li>United States
</ul>
```

Grouping Content

Ordered Lists

Ordered lists

An ordered list starts with the <ol> tag.

Each list item starts with the <li> tag.

The list items are marked with numbers.

The li element has an ordinal value.

The value attribute, if present, must be a valid integer giving the ordinal value of the list item. If the attribute's value cannot be converted to a number, the attribute must be treated as if it was absent. The attribute has no default value.

Grouping Content

Ordered Lists

Ordered lists

The ol element has three specific attributes

reversed – number the list backwards

start – ordinal value of the first item

type – kind of list marker

The type attribute represents the state given in the cell in the second column of the row whose first cell matches the attribute's value; if none of the cells match, or if the attribute is omitted, then the attribute represents the decimal state

Keywords	State	Description
1 (U+0031)	decimal	Decimal numbers
a (U+0061)	lower-alpha	Lowercase latin alphabet
A (U+0041)	upper-alpha	Uppercase latin alphabet
i (U+0069)	lower-roman	Lowercase roman numerals
I (U+0049)	upper-roman	Uppercase roman numerals

Grouping Content

Ordered Lists

Ordered lists (example 1)

```
<figure>  <figcaption>The top 10 movies of all time</figcaption>
 <ol>
  <li value="10"><cite>Josie and the Pussycats</cite>, 2001</li>
  <li value="9"><cite lang="sh">Црна мачка, бели мачор</cite>, 1998</li>
  <li value="8"><cite>A Bug's Life</cite>, 1998</li>
  <li value="7"><cite>Toy Story</cite>, 1995</li>
  <li value="6"><cite>Monsters, Inc</cite>, 2001</li>
  <li value="5"><cite>Cars</cite>, 2006</li>
  <li value="4"><cite>Toy Story 2</cite>, 1999</li>
  <li value="3"><cite>Finding Nemo</cite>, 2003</li>
  <li value="2"><cite>The Incredibles</cite>, 2004</li>
  <li value="1"><cite>Ratatouille</cite>, 2007</li>
 </ol> </figure>
```

Grouping Content

Ordered Lists

Ordered lists (example 2)

```
<figure>  <figcaption>The top 10 movies of all time</figcaption>
 <ol reversed type="a">
  <li><cite>Josie and the Pussycats</cite>, 2001</li>
  <li><cite lang="sh">Црна мачка, бели мачор</cite>, 1998</li>
  <li><cite>A Bug's Life</cite>, 1998</li>
  <li><cite>Toy Story</cite>, 1995</li>
  <li><cite>Monsters, Inc</cite>, 2001</li>
  <li><cite>Cars</cite>, 2006</li>
  <li><cite>Toy Story 2</cite>, 1999</li>
  <li><cite>Finding Nemo</cite>, 2003</li>
  <li><cite>The Incredibles</cite>, 2004</li>
  <li><cite>Ratatouille</cite>, 2007</li>
</ol></figure>
```

Grouping Content

Ordered Lists

Ordered lists (example 2)

```
<figure>  <figcaption>The top 10 movies of all time</figcaption>
 <ol reversed type="a" start="10">
  <li><cite>Josie and the Pussycats</cite>, 2001</li>
  <li><cite lang="sh">Црна мачка, бели мачор</cite>, 1998</li>
  <li><cite>A Bug's Life</cite>, 1998</li>
  <li><cite>Toy Story</cite>, 1995</li>
  <li><cite>Monsters, Inc</cite>, 2001</li>
  <li><cite>Cars</cite>, 2006</li>
  <li><cite>Toy Story 2</cite>, 1999</li>
  <li><cite>Finding Nemo</cite>, 2003</li>
  <li><cite>The Incredibles</cite>, 2004</li>
  <li><cite>Ratatouille</cite>, 2007</li>
</ol></figure>
```

Grouping Content

Definition Lists

Definition lists

Lists of items (terms), with a description of each item (term).

A definition list starts with a **<dl>** tag (definition list).

Each term starts with a **<dt>** tag (definition term).

Each description starts with a **<dd>** tag (definition description).

Name-value groups may be terms and definitions, metadata topics and values, questions and answers, or any other groups of name-value data.

The values within a group are alternatives; multiple paragraphs forming part of the same value must all be given within the same dd element.

Grouping Content

Definition Lists

Definition lists (Example)

```
<dl>
 <dt lang="en-US"> <dfn>color</dfn> </dt>
 <dt lang="en-GB"> <dfn>colour</dfn> </dt>
 <dd> A sensation which (in humans) derives
from the ability of the fine structure of the eye
to distinguish three differently
 filtered analyses of a view. </dd>
</dl>
```

Text-level semantics

Text-level semantics

Element a

If the **a** element has an **href** attribute, then it represents a hyperlink labeled by its contents.

If the **a** element has no href attribute, then the element represents a placeholder for where a link might otherwise have been placed, if it had been relevant, consisting of just the element's contents.

Attributes

href - Address of the hyperlink

target - Default browsing context for hyperlink navigation and form submission

download - Whether to download the resource instead of navigating to it, and its file name if so

rel - Relationship between the document containing the hyperlink and the destination resource

hreflang - Language of the linked resource

type - Hint for the type of the referenced resource

Text-level semantics (Target)

Keyword	Ordinary effect	Effect in an iframe with...	
		sandbox=""	sandbox="allow-top-navigation"
none specified, for links and form submissions	current	current	current
empty string	current	current	current
_blank	new	maybe new	maybe new
_self	current	current	current
_parent if there isn't a parent	current	current	current
_parent if parent is also top	parent/top	none	parent/top
_parent if there is one and it's not top	parent	none	none
_top if top is current	current	current	current
_top if top is not current	top	none	top
name that doesn't exist	new	maybe new	maybe new
name that exists and is a descendant	specified descendant	specified descendant	specified descendant
name that exists and is current	current	current	current
name that exists and is an ancestor that is top	specified ancestor	none	specified ancestor/top
name that exists and is an ancestor that is not top	specified ancestor	none	none
other name that exists with common top	specified	none	none
name that exists with different top, if familiar and one permitted sandboxed navigator	specified	specified	specified
name that exists with different top, if familiar but not one permitted sandboxed navigator	specified	none	none
name that exists with different top, not familiar	new	maybe new	maybe new

Text-level semantics

Link type	Effect on...		Brief description
	`link`	`a` and `area`	
`alternate`	Hyperlink	Hyperlink	Gives alternate representations of the current document.
`author`	Hyperlink	Hyperlink	Gives a link to the author of the current document or artic Remove developer-view styles
`bookmark`	*not allowed*	Hyperlink	Gives the permalink for the nearest ancestor section.
`help`	Hyperlink	Hyperlink	Provides a link to context-sensitive help.
`icon`	External Resource	*not allowed*	Imports an icon to represent the current document.
`license`	Hyperlink	Hyperlink	Indicates that the main content of the current document is covered by the copyright license described by the referenced document.
`next`	Hyperlink	Hyperlink	Indicates that the current document is a part of a series, and that the next document in the series is the referenced document.
`nofollow`	*not allowed*	Annotation	Indicates that the current document's original author or publisher does not endorse the referenced document.
`noreferrer`	*not allowed*	Annotation	Requires that the user agent not send an HTTP `Referer` (sic) header if the user follows the hyperlink.
`prefetch`	External Resource	External Resource	Specifies that the target resource should be preemptively cached.
`prev`	Hyperlink	Hyperlink	Indicates that the current document is a part of a series, and that the previous document in the series is the referenced document.
`search`	Hyperlink	Hyperlink	Gives a link to a resource that can be used to search through the current document and its related pages.
`stylesheet`	External Resource	*not allowed*	Imports a stylesheet.
`tag`	*not allowed*	Hyperlink	Gives a tag (identified by the given address) that applies to the current document.

Text-level semantics

Element a

Example:

```
<nav>
 <ul>
  <li> <a href="http://www.unipi.it" target="_blank">New</a> </li>
  <li> <a href="http://www.unipi.it" target="_self">Self</a> </li>
  <li> <a href="http://www.unipi.it" target=”_top" rel="tag">Top</a> </li>
 </ul>
</nav>
```

Text-level semantics

Elements em, strong, small

The **em** element represents stress emphasis of its contents.

The **strong** element represents strong importance, seriousness, or urgency (contents that the user needs to see sooner than other parts of the document) for its contents.

The **small** element represents side comments such as small print.

Example

Example Corp today announced record profits for the second quarter <small>(Full Disclosure: Foo News is a subsidiary of Example Corp)</small>, leading to speculation about a third quarter merger with Demo Group .

Text-level semantics

Elements s and cite

The **s** element represents contents that are no longer accurate or no longer relevant.

```
<p>Buy our Iced Tea and Lemonade!</p>
<p><s>Recommended retail price: $3.99 per bottle</s></p>
<p><strong>Now selling for just $2.99 a bottle!</strong></p>
```

The **cite** element represents a reference to a creative work. It must include the title of the work or the name of the author(person, people or organization) or an URL reference, which may be in an abbreviated form as per the conventions used for the addition of citation metadata.

```
<p>In the words of <cite>Charles Bukowski</cite> -
```

Text-level semantics

Element q

The **q** element represents some phrasing content quoted from another source.

```
<p>The W3C page <cite>About W3C</cite> says the W3C's mission is <q cite="http://www.w3.org/Consortium/">To lead the World Wide Web to its full potential by developing protocols and guidelines that ensure long-term growth for the Web</q>. I disagree with this mission.</p>
```

Text-level semantics

Elements dfn and abbr

The **dfn** element represents the defining instance of a term. The paragraph, description list group, or section that is the nearest ancestor of the dfn element must also contain the definition(s) for the term given by the dfn element.

If the dfn element has a *title* attribute, then the exact value of that attribute is the term being defined.

The **abbr** element represents an abbreviation or acronym, optionally with its expansion. The *title* attribute may be used to provide an expansion of the abbreviation. The attribute, if specified, must contain an expansion of the abbreviation, and nothing else.

Text-level semantics

Elements dfn and abbr

Example:

```
<p>The <dfn id=gdo><abbr title="Garage Door Opener">GDO</abbr></dfn>
is a device that allows off-world teams to open the iris.</p>
<!-- ... later in the document: -->
<p>Teal'c activated his <a href=#gdo><abbr title="Garage Door Opener">GDO</abbr></a>
and so Hammond ordered the iris to be opened.</p>
```

Text-level semantics

Elem code, var, samp, kbd, sup, sub

The **code** element represents a fragment of computer code.

The **var** element represents a variable.

The **samp** element represents (sample) output from a program or computing system.

The **kbd** element represents user input (typically keyboard input, although it may also be used to represent other input, such as voice commands).

The **sup** element represents a superscript and the **sub** element represents a subscript.

```
<p>The coordinate of the <var>i</var>th point is
<code>(<var>x<sub><var>i</var></sub></var>,
<var>y<sub><var>i</var></sub></var>)</code>.
For example, the 10th point has coordinate
<code>(<var>x<sub>10</sub></var>,
<var>y<sub>10</sub></var>)</code>.</p>
```

Text-level semantics

Elements i, b and mark

The **i** element represents a span of text in an alternate voice or mood, or otherwise offset from the normal prose in a manner indicating a different quality of text, such as a taxonomic designation, a technical term, an idiomatic phrase from another language, transliteration, a thought, or a ship name in Western texts.

The **b** element represents a span of text to which attention is being drawn for utilitarian purposes without conveying any extra importance and with no implication of an alternate voice or mood, such as key words in a document abstract, product names in a review, actionable words in interactive text-driven software.

The **mark** element represents a run of text in one document marked or highlighted for reference purposes, due to its relevance in another context.

Text-level semantics

Elements span, br, wbr

The **span** element doesn't mean anything on its own, but can be useful when used together with the global attributes, e.g. class, lang, or dir. It represents its children.

The **br** element represents a line break.

The **wbr** element represents a line break opportunity.

So then he pointed at the tiger and screamed

"there<wbr>is<wbr>no<wbr>way<wbr>you<wbr>are<wbr>ever<wbr>going<wbr>to<wbr>catch<wbr>me"!

Edits

Edits

Elements ins, del

The **ins** element represents an addition to the document.

The **del** element represents a removal from the document.

The **cite** attribute may be used to specify the address of a document that explains the change.

The **datetime** attribute may be used to specify the time and date of the change.

```
<h1>List of <del>fruits</del><ins>colors</ins></h1>
<ul>
<li><del datetime="2009-10-10T23:38-07:00">Apple</del></li>
 <li>Orange</li>
 <li><del>Pear</del></li>
 <li><ins>Teal</ins></li>
 <li><del>Lemon</del><ins>Yellow</ins></li>
 <li><ins>Olive</ins></li>
</ul>
```

Embedded content

Embedded Content

The img element

An **img** element represents an image.

The <img> element is empty, which means that it contains attributes only and it has no closing tag.

To display an image on a page, you need to use the **src** attribute. The value of the src attribute is the URL of the image you want to display on your page.

The syntax of defining an image:

```
<img src="url" alt="sample.txt">
```

The browser puts the image where the image tag occurs in the document.

Except where otherwise specified, the alt attribute must be specified and its value must not be empty; the value must be an appropriate functional replacement for the image.

Embedded Content

The img element

```
<p> <img src= "tower.jpg" width="40" height="40"
alt="Tower of Pisa"> </p>

 <p> <img src= "tower.jpg" width="90"  height="90"
alt="Tower of Pisa"> </p>

   <p>
     You can make an image smaller or larger by changing the
     values of the "height" and "width" attributes.
   </p>
```

Embedded Content

The img element

An **img** is always in one of the following states:

Unavailable

The user agent has not obtained any image data.

Partially available

The user agent has obtained some of the image data.

Completely available

The user agent has obtained all of the image data and at least the image dimensions are available.

Broken

The user agent has obtained all of the image data that it can, but it cannot even decode the image enough to get the image dimensions (e.g. the image is corrupted, or the format is not supported, or no data could be obtained).

Embedded Content

The image map

Image maps allow authors to specify regions of an image and assign a specific action to each region (e.g., retrieve a document, run a program, etc.). When the region is activated by the user, the action is executed.

An image map is created by associating an image with a specification of sensitive geometric areas on the image.

The **map** element, in conjunction with an **img** element and any **area** element descendants, defines an image map. The element represents its children.

The **name** attribute gives the map a name so that it can be referenced. The attribute must be present and must have a non-empty value with no space characters.

Embedded Content

The image map

```
<p>Click on the sun or on one of the planets to get more information:</p>
<div>
<img src="planets.gif" width="145" height="126" alt="Planets" usemap="#planetmap">
<map name="planetmap">
  <area shape="rect" coords="0,0,82,126" alt="Sun" href="sun.html">
  <area shape="circle" coords="90,58,3" alt="Mercury" href="mercur.html">
  <area shape="circle" coords="124,58,8" alt="Venus" href="venus.html">
</map>
</div>
```

Embedded Content

The image map

The <area> tag defines an area inside an image-map (an image-map is an image with clickable areas).

The area element is always nested inside a <map> tag

Note: The **usemap** attribute in the <img> tag is associated with the map element's name attribute, and creates a relationship between the image and the map.

Embedded Content

The image map

Area attributes

Attribute	Value	Description
alt	*text*	Specifies an alternate text for an area
coords	*coordinates*	Specifies the coordinates of an area
href	*URL*	Specifies the destination of a link in an area
download	download	hyperlink is used for downloading a resource
shape	default rect circle poly	Specifies the shape of an area
target	_blank _parent _self _top	Specifies where to open the linked page specified in the href attribute

Embedded Content

The image map

Coords attribute

This attribute specifies the position and shape on the screen. **The number and order of values depends on the shape being defined**. Possible combinations:

rect: left-x, top-y, right-x, bottom-y.

circle: center-x, center-y, radius.

poly: x1, y1, x2, y2, ..., xN, yN.

The first x and y coordinate pair and the last should be the same to close the polygon. When these coordinate values are not the same, user agents should infer an additional coordinate pair to close the polygon.

Embedded Content

The iframe element

The **iframe** element represents a nested browsing context.

The **src** attribute gives the address of a document that the nested browsing context is to contain.

The **srcdoc** attribute gives the HTML content that the nested browsing context is to contain. The value of the attribute is the source of an iframe srcdoc document.

```
<iframe srcdoc="<p>Hello world!</p>" </iframe>
```

The name attribute, if present, must be a valid browsing context name.

> The given value is used to name the nested browsing context. When the browsing context is created, if the attribute is present, the browsing context name must be set to the value of this attribute; otherwise, the browsing context name must be set to the empty string.

Embedded Content

The iframe element

```
<body>
<iframe src="html25.html" name="iframe_a">
<p>Your browser does not support iframes.</p>
</iframe>
<a href= "html26.html"
target="iframe_a">Planets</a>
<p><b>Note:</b> Because the target of the link
matches the name of the iframe, the link will open
in the iframe.</p>
</body>
</html>
```

Embedded Content

The iframe element

The **sandbox** attribute, when specified, enables a set of extra restrictions on any content hosted by the iframe.

The **width** and **height** attributes determine the horizontal and vertical dimensions, respectively.

Example

<iframe sandbox="allow-same-origin allow-forms allow-scripts" src= "html26.html" width="400" height="500">

[Your user agent does not support frames or is currently configured not to display frames.

However, you may visit the related document.]

Embedded Content

The embed element

The **embed** element provides an integration point for an external (typically non-HTML) application or interactive content.

The **src** attribute gives the address of the resource being embedded. The attribute, if present, must contain a valid non-empty URL potentially surrounded by spaces.

The type attribute, if present, gives the MIME type by which the plugin to instantiate is selected.

```
<embed src="vowels.swf" width=“720" height=“500">
<embed src="movie.mp4" width="320" height="240" title="Title of my video">
```

Embedded Content

The object element

The **object** element can represent an external resource, which, depending on the type of the resource, will either be treated as an image, as a nested browsing context, or as an external resource to be processed by a plugin.

The **data** attribute, if present, specifies the address of the resource.

The **type** attribute, if present, specifies the type of the resource. If present, the attribute must be a valid MIME type.

The **typemustmatch** attribute is a boolean attribute whose presence indicates that the resource specified by the data attribute is only to be used if the value of the type attribute and the Content-Type of the aforementioned resource match.

```
<object id="vowels" type="application/x-shockwave-flash"
                        data="./wowels.swf" width="720" height="500">
        <param name="movie" value="./wowels.swf">
</object>

<object type="application/pdf"
   data="00-introduzione-al-corso.pdf"
   width="500"
   height="800">
</object>
```

Embedded Content

The param element

The **param** element defines parameters for plugins invoked by object elements. It does not represent anything on its own.

The **name** attribute gives the name of the parameter.

The **value** attribute gives the value of the parameter.

Both attributes must be present. They may have any value.

Embedded Content

The video element

A **video** element is used for playing videos or movies, and audio files with captions.

The video element is a media element whose media data is ostensibly video data, possibly with associated audio data.

```
<video width="320" height="240" src="movie.mp4"
type="video/mp4“ controls>
Your browser does not support the video tag.
</video>
```

Embedded Content

The video element

Attribute	Value	Description
autoplay	autoplay	Specifies that the video will start playing as soon as it is ready
controls	controls	Specifies that video controls should be displayed (such as a play/pause button etc).
height	pixels	Sets the height of the video player
loop	loop	Specifies that the video will start over again, every time it is finished
muted	muted	Specifies that the audio output of the video should be muted
poster	URL	Specifies an image to be shown while the video is downloading, or until the user hits the play button
preload	auto metadata none	Specifies if and how the author thinks the video should be loaded when the page loads. Metadata: only metadata. None: the browser should not load the video.
src	URL	Specifies the URL of the video file
width	pixels	Sets the width of the video player

Embedded Content

The source element

The **source** element allows authors to specify multiple alternative media resources for media elements. It does not represent anything on its own.

The **src** attribute gives the address of the media resource. The value must be a valid non-empty URL potentially surrounded by spaces. This attribute must be present.

```
<video width="320" height="240" controls>
  <source src="movie.mp4" type="video/mp4">
  <source src="movie.ogg" type="video/ogg">
  Your browser does not support the video tag.
</video>
```

Embedded Content

The audio element

An **audio** element represents a sound or audio stream.

Content may be provided inside the audio element. User agents should not show this content to the user; it is intended for older Web browsers which do not support audio, so that legacy audio plugins can be tried, or to show text to the users of these older browsers informing them of how to access the audio contents.

```
<audio controls>
  <source src= "applause.ogg" type="audio/ogg">
  <source src= "applause.mp3" type="audio/mpeg">
  Your browser does not support the audio element.
</audio>
```

Links

Example 1

```
<h1>Table of Contents</h1>
<p><a href="#section1">Introduction</a><br>
<a href="#section2">Some background</a><br>
<a href="#section2.1">On a more personal note</a><br>
...the rest of the table of contents... ...the document body...</p>
<h2><a id="section1">Introduction</a></h2>
<p>...section 1...</p>
<h2><a id="section2">Some background</a></h2>
<p>...section 2...</p>
<h3><a id="section2.1">On a more personal note</a></h3>
<p>...section 2.1...</p>
```

Links

Example 2

We may achieve the same effect by making the header elements themselves the anchors:

```
<h1> Table of Contents</h1>
   <p>  <a href="#section1">Introduction</a><br>
    <a href="#section2">Some background</a><br>
    <a href="#section2.1">On a more personal note</a><br>
    ...the rest of the table of contents... ...the document  body...
   </p>
   <h2 id="section1"> Introduction</h2>
         <p>...section 1...</p>
   <h2 id="section2">Some background</h2>
         <p>...section 2...</p>
   <h3 id="section2.1"> On a more personal note </h3>
         <p>...section 2.1...</p>
```

Links

Element <link>

Attribute	Value	Description
href	URL	Specifies the URL of the page the link goes to
hreflang	language_code	Specifies the language of the linked document
media	media_query	Specifies what media/device the linked document is optimized for
rel	alternate author bookmark help license next nofollow noreferrer prefetch prev search tag	Specifies the relationship between the current document and the linked document
sizes	value	Sizes of the icons (for rel="icon")
type	Mime_type	Specifies the Mime type of the linked document

Links

Example 3

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