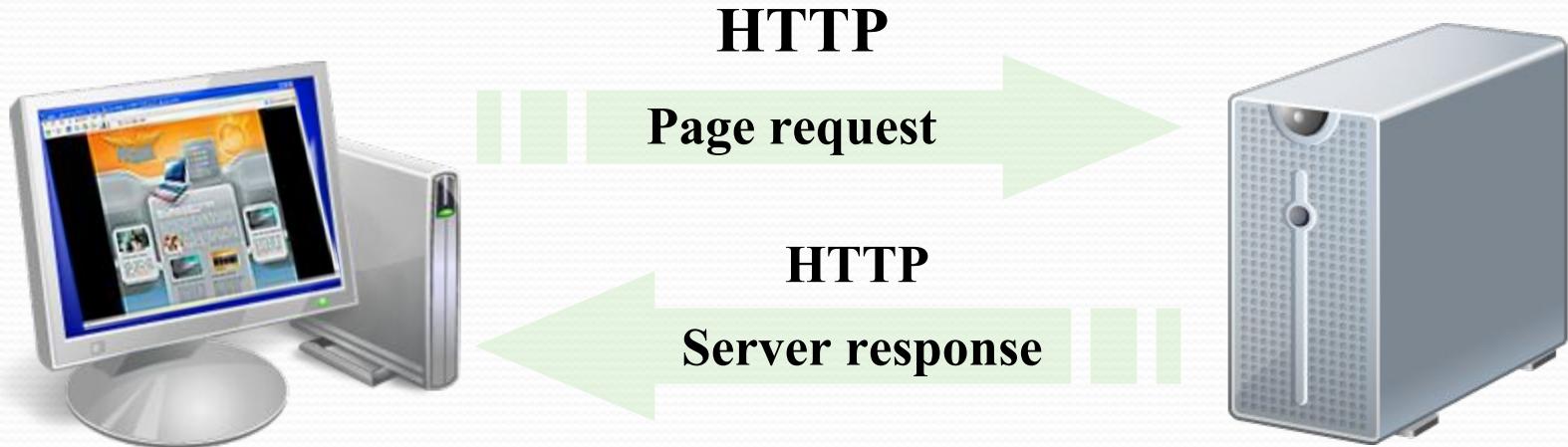


The background features a light gray surface with a subtle grid pattern. Overlaid on this are several thin, wavy lines in shades of teal, blue, and green, some with a slight glow. A dense cluster of small, semi-transparent white dots is located in the upper right quadrant.

HTML

How the Web Works?

- WWW use classical client / server architecture
 - HTTP is text-based request-response protocol



**Client running a
Web Browser**

**Server running
Web Server
Software (IIS,
Apache, etc.)**

What is a Web Page?

- Web pages are text files containing HTML
- HTML – Hyper Text Markup Language
 - A notation for describing
 - document structure (semantic markup)
 - formatting (presentation markup)
 - Looks (looked?) like:
 - A Microsoft Word document
- The markup tags provide information about the page content structure
- A web page is a text file in which a hypertext language is written according to HTML grammar.
- This HTML code is displayed by the browser converting it to a web page.

- **HTML:**
- "**Hypertext**" refers to links that connect web pages to one another, either within a single website or between websites.
- Links are a fundamental aspect of the Web.
- HTML uses "**markup**" to annotate text, images, and other content for display in a Web browser.
- HTML markup includes special "elements" such as <head>, <title>, <body>, <header>, <footer>, <article> etc.
- An HTML element are "tags", which consist of the element name surrounded by "<" and ">".
- The name of an element inside a tag is case insensitive.
- **HTML is a markup language that defines the structure of your content.**
- **HTML consists of a series of elements, which you use to enclose, or wrap, different parts of the content to make it appear a certain way, or act a certain way.**

Basic of HTML

● Tag, Element ,Attribute

● HTML Tags:

● Tags are used to mark up the start of an HTML element and they are usually enclosed in angle brackets.

● Most tags must be opened `<tagname>` and closed `</tagname>` in order to function.

● Example:

● ` `

● **HTML elements:** Elements enclose the contents in between the tags. They consist of some kind of structure or expression. It generally consists of a start tag, content and an end tag.

● Example:

● `This is the content.`

● HTML Attributes:

● Attributes contain additional pieces of information.

● Attributes take the form of an opening tag and additional info is placed inside.

● It is used to define the property of an HTML element. I

● t always placed in the opening tag of an element. It generally provides additional styling (attribute) to the element.

● An example of an attribute is:

● ``

- The majority of tags must be opened (`<tag>`) and closed (`</tag>`) with the element information such as a title or text resting between the tags.
- When using multiple tags, the tags must be closed in the order in which they were opened. For example:
- Example: `This is really important!`

HTML Tags

HTML tags are used to hold the HTML element.

HTML tag starts with < and ends with >

HTML tags are almost like keywords where every single tag has unique meaning.

HTML Elements

HTML element holds the content.

Whatever written within a HTML tag are HTML elements.

HTML elements specifies the general content.

HTML Attributes

HTML attributes are used to describe the characteristic of an HTML element in detail.

HTML attributes are found only in the starting tag.

HTML attributes specify various additional properties to the existing HTML element.

Creating HTML Pages

- An HTML file must have an `.htm` or `.html` file extension
- HTML files can be created with text editors:
 - NotePad, NotePad ++, PSPad
- Or HTML editors (WYSIWYG Editors):
 - Microsoft FrontPage
 - Macromedia Dreamweaver
 - Netscape Composer
 - Microsoft Word
 - Visual Studio

HTML

- HTML is the standard markup language for creating Web pages.
 - HTML stands for Hyper Text Markup Language
 - HTML is the standard markup language for creating Web pages
 - HTML describes the structure of a Web page
 - HTML consists of a series of elements
 - HTML elements tell the browser how to display the content
 - HTML elements label pieces of content such as "this is a heading", "this is a paragraph", "this is a link", etc.

History of HTML

1991	HTML first published	
1995	HTML 2.0	
1997	HTML 3.2	After HTML 4.01 was released, focus shifted to XHTML and its stricter standards.
1999	HTML 4.01	XHTML 2.0 had even stricter standards than 1.0, rejecting web pages that did not comply. It fell out of favor gradually and was abandoned completely in 2009.
2000	XHTML 1.0	
2002	XHTML 2.0	
2009		HTML5 is much more tolerant and can handle markup from all the prior versions.
2012	HTML5	Though HTML5 was published officially in 2012, it has been in development since 2004.
2014		<u>W3C Recommendation: HTML5</u>
2016		W3C Candidate Recommendation: HTML 5.1
2017		<u>W3C Recommendation: HTML5.1 2nd Edition</u>
2017		<u>W3C Recommendation: HTML5.2</u>

Type of content	HTML 1.2	HTML 4.01	HTML5	Purpose
Heading	Yes	Yes	Yes	Organize page content by adding headings and subheadings to the top of each section of the page
Paragraph	Yes	Yes	Yes	Identify paragraphs of text
Address	Yes	Yes	Yes	Identify a block of text that contains contact information
Anchor	Yes	Yes	Yes	Link to other web content
List	Yes	Yes	Yes	Organize items into a list
Image	Yes	Yes	Yes	Embed a photograph or drawing into a web page
Table	No	Yes	Yes	Organize data into rows and columns
Style	No	Yes	Yes	Add CSS to control how objects on a web page are presented
Script	No	Yes	Yes	Add Javascript to make pages respond to user behaviors (more interactive)
Audio	No	No	Yes	Add audio to a web page with a single tag
Video	No	No	Yes	Add video to a web page with a single tag
Canvas	No	No	Yes	Add an invisible drawing pad to a web page, on which you can add drawings (animations, games, and other interactive features) using Javascript

What is HTML5?

- HTML5 is the newest version of HTML, only recently gaining partial support by the makers of web browsers.
- It incorporates all features from earlier versions of HTML, including the stricter XHTML.
- It adds a diverse set of new tools for the web developer to use.

Goals of HTML5

- Support all existing web pages.
- Reduce the need for external plugins and scripts to show website content.
- Improve the semantic definition (i.e. meaning and purpose) of page elements.
- Make the rendering of web content universal and independent of the device being used.
- Handle web documents errors in a better and more consistent fashion.

New Elements in HTML5

<article>	<figcaption>	<progress>
<aside>	<footer>	<section>
<audio>	<header>	<source>
<canvas>	<hgroup>	<svg>
<datalist>	<mark>	<time>
<figure>	<nav>	<video>

These are just some of the new elements introduced in HTML5. We will be exploring each of these during this course.

Other New Features in HTML5

- Built-in audio and video support (without plugins)
- Enhanced form controls and attributes
- The Canvas (a way to draw directly on a web page)
- Drag and Drop functionality
- Support for CSS3 (the newer and more powerful version of CSS)
- More advanced features for web developers, such as data storage and offline applications.

First Look at HTML5

Remember the DOCTYPE declaration from XHTML?

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

In HTML5, there is just one possible DOCTYPE declaration and it is simpler:

```
<!DOCTYPE html>
```

Just 15 characters!

The DOCTYPE tells the browser which type and version of document to expect. This should be the last time the DOCTYPE is ever changed. From now on, all future versions of HTML will use this same simplified declaration.

The <html> Element

This is what the <html> element looked like in XHTML:

```
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en"
      lang="en">
```

Again, HTML5 simplifies this line:

```
<html lang="en">
```

The **lang** attribute in the <html> element declares which language the page content is in. Though not strictly required, it should always be specified, as it can assist search engines and screen readers.

Each of the world's major languages has a two-character code, e.g. Spanish = "es", French = "fr", German = "de", Chinese = "zh", Arabic = "ar".

The <head> Section

Here is a typical XHTML <head> section:

```
<head>
  <meta http-equiv="Content-type" content="text/html; charset=UTF-8" />
  <title>My First XHTML Page</title>
  <link rel="stylesheet" type="text/css" href="style.css" />
</head>
```

And the HTML5 version:

```
<head>
  <meta charset="utf-8">
  <title>My First HTML5 Page</title>
  <link rel="stylesheet" href="style.css">
</head>
```

Notice the simplified character set declaration, the shorter CSS stylesheet link text, and the removal of the trailing slashes for these two lines.

Basic HTML5 Web Page

Putting the prior sections together, and now adding the <body> section and closing tags, we have our first complete web page in HTML5:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>My First HTML5 Page</title>
  <link rel="stylesheet" href="style.css">
</head>
<body>
  <p>HTML5 is fun!</p>
</body>
</html>
```

- **HTML <meta> Tag**
- Describe metadata within an HTML document:
- The <meta> tag defines metadata about an HTML document.
- Metadata is data (information) about data.
- <meta> tags always go inside the <head> element, and are typically used to specify character set, page description, keywords, author of the document, and viewport settings.
- Metadata will not be displayed on the page, but is machine parsable.
- Metadata is used by browsers (how to display content or reload page), search engines (keywords), and other web services.
- There is 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).

- A Simple HTML Document Structure

- <!DOCTYPE html>

- <html>

- <head>

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

- </head>

- <body>

- The content of the document.....

- </body>

- </html>

- <!DOCTYPE>
- All HTML documents must start with a <!DOCTYPE> declaration.
- The declaration is not an HTML tag.
- It is an "information" to the browser about what document type to expect.
- In HTML 5, the declaration is simple:
- <!DOCTYPE html>

- The `<!DOCTYPE html>` declaration defines that this document is an HTML5 document
- **<!DOCTYPE html>**

- The `<html>` element is the root element of an HTML page
- The `<html>` tag is the container for all other HTML elements
`<html lang="en">`
The `<head>` element contains meta information about the HTML page

- The `<title>` element specifies a title for the HTML page (which is shown in the browser's title bar or in the page's tab)
- The `<body>` element defines the document's body, and is a container for all the visible contents, such as headings, paragraphs, images, hyperlinks, tables, lists, etc.

- The <head> element contains meta information about the HTML page
- The <head> element is a container for metadata (data about data) and is placed between the <html> tag and the <body> tag.
- Metadata is data about the HTML document.
- Metadata is not displayed.
- Metadata typically define the document title, character set, styles, scripts, and other meta information.
- The following elements can go inside the <head> element:
 - <title> (required in every HTML document)
 - <style>
 - <base>
 - <link>
 - <meta>
 - <script>
 - <noscript>

- **Title tag:**
- The <title> tag defines the title of the document.
- The title must be text-only, and it is shown in the browser's title bar or in the page's tab.
- The <title> tag is required in HTML documents!

The <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

- Body tag:
- The <body> tag defines the document's body.
- The <body> element contains all the contents of an HTML document, such as headings, paragraphs, images, hyperlinks, tables, lists, etc.
- Note: There can only be one <body> element in an HTML document.

```
<html>
<head>
<style>
body {
    background-image: url(w3s.png);
}
</style>
</head>
<body>

<h1>Hello world!</h1>
<p><a href="https://www.xyz.com">html tutorial</a></p>

</body>
```

HTML Structure

- HTML is comprised of “elements” and “tags”
 - Begins with `<html>` and ends with `</html>`
- Elements (tags) are nested one inside another:

```
<html> <head></head> <body></body> </html>
```
- Tags have attributes:

```

```
- HTML describes structure using two main sections:`<head>` and `<body>`

HTML Code Formatting

- The HTML source code should be formatted to increase readability and facilitate debugging.
 - Every block element should start on a new line.
 - Every nested (block) element should be indented.
 - Browsers ignore multiple whitespaces in the page source, so formatting is harmless.

First HTML Page

```
<!DOCTYPE HTML>
<html>
  <head>
    <title>My First HTML Page</title>
  </head>
  <body>
    <p>This is some text...</p>
  </body>
</html>
```



First HTML Page: Tags

```
<!DOCTYPE HTML>
<html>
  <head>
    <title>My First HTML Page</title>
  </head>
  <body>
    <p>This is some text...</p>
  </body>
</html>
```

Opening tag

Closing tag

An HTML element consists of an opening tag, a closing tag and the content inside.

First HTML Page: Header

```
<!DOCTYPE HTML>
<html>
  <head>
    <title>My First HTML Page</title>
  </head>
  <body>
    <p>This is some text...</p>
  </body>
</html>
```

HTML

header

First HTML Page: Body

```
<!DOCTYPE HTML>  
<html>  
  <head>  
    <title>My First HTML Page</title>  
  </head>  
  <body>  
    <p>This is some text...</p>  
  </body>  
</html>
```

HTML body

Some Simple Tags

- Hyperlink Tags

```
<a href="http://www.telerik.com/"  
    title="Telerik">Link to Telerik Web site</a>
```

- Image Tags

```

```

- Text formatting tags

This text is ***emphasized.***

**
new line
**

This one is ****more emphasized.****

Some Simple Tags – Example

some-tags.html

```
<!DOCTYPE HTML>
<html>
<head>
  <title>Simple Tags Demo</title>
</head>
<body>
<a href="http://www.telerik.com/" title=
  "Telerik site">This is a link.</a>
<br />

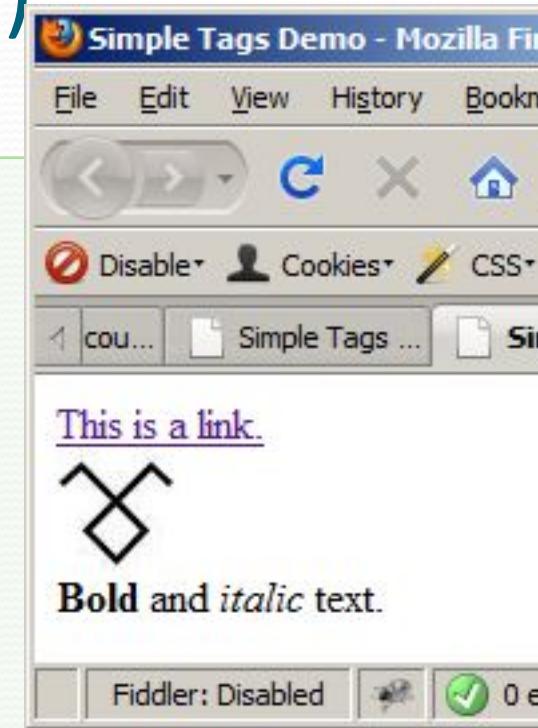
<br />
<strong>Bold</strong> and <em>italic</em> text.
</body>
</html>
```

Some Simple Tags – Example (2)

some-tags.html

```
<!DOCTYPE HTML>
<html>
<head>
  <title>Simple Tags Demo</title>
</head>
<body>
<a href="http://www.telerik.com/" title=
  "Telerik site">This is a link.</a>
<br />

<br />
<strong>Bold</strong> and <em>italic</em> text.
</body>
</html>
```



Tags Attributes

- Tags can have **attributes**

Attribute **alt** with value
"logo"

- Attributes specify properties and behavior
- Example:

```

```

- Few attributes can apply to every element:
 - **id, style, class, title**
 - The **id** is unique in the document
 - Content of **title** attribute is displayed as hint when the element is hovered with the mouse
 - Some elements have obligatory attributes

Headings and Paragraphs

```
<h1>Heading 1</h1>
<h2>Sub heading 2</h2>
<h3>Sub heading 3</h3>
```

- Heading Tags (h1 – h6)

```
<p>This is my first paragraph</p>
<p>This is my second paragraph</p>
```

- Paragraph Tags

```
<div style="background: skyblue;">
    This is a div</div>
```

- Sections: div and span

Headings and Paragraphs – Example

headings.html

```
<!DOCTYPE HTML>
<html>
  <head><title>Headings and paragraphs</title></head>
  <body>
    <h1>Heading 1</h1>
    <h2>Sub heading 2</h2>
    <h3>Sub heading 3</h3>

    <p>This is my first paragraph</p>
    <p>This is my second paragraph</p>

    <div style="background:skyblue">
      This is a div</div>
  </body>
</html>
```

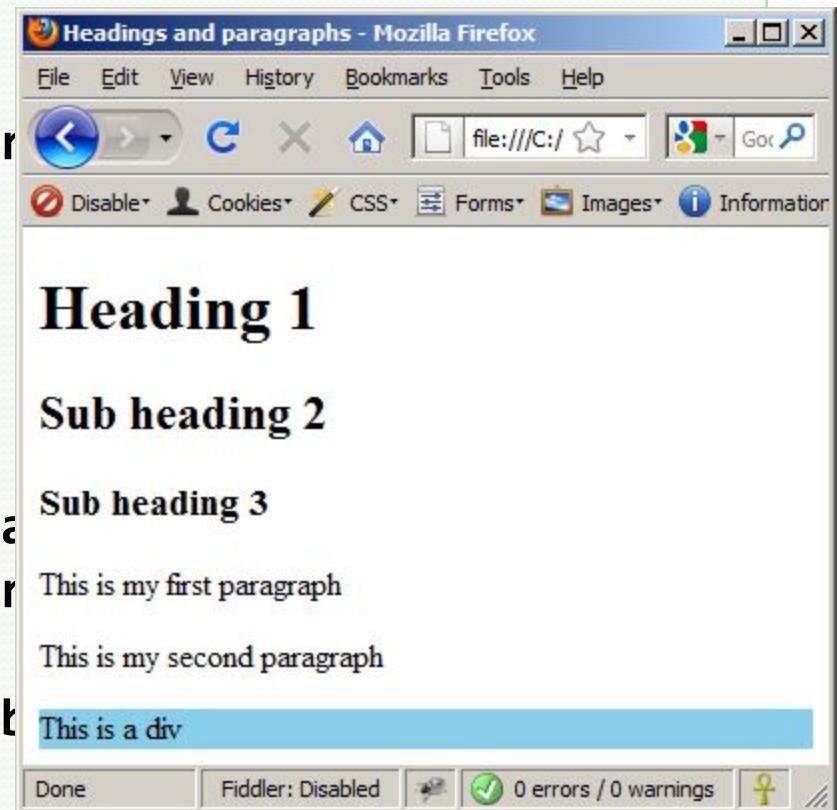
Headings and Paragraphs – Example (2)

headings.html

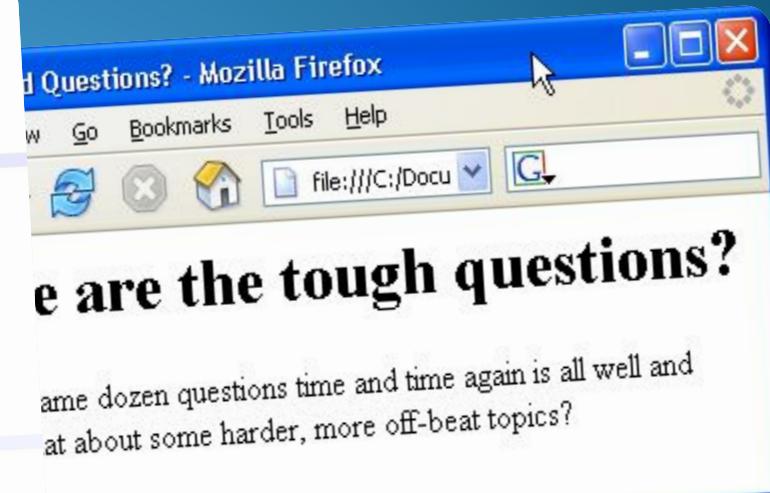
```
<!DOCTYPE HTML>
<html>
  <head><title>Headings and par
  <body>
    <h1>Heading 1</h1>
    <h2>Sub heading 2</h2>
    <h3>Sub heading 3</h3>

    <p>This is my first paragraph</p>
    <p>This is my second paragraph</p>

    <div style="background:skyblue">
      This is a div</div>
  </body>
</html>
```



```
1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
2   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
3 <html xmlns="http://www.w3.org/1999/xhtml">
4   <head>
5     <title>Tabview - Demo</title>
6     <script src="prototype.js" type="text/javascript"></script>
7     <script src="tabview.js" type="text/javascript"></script>
8     <link href="tabview.css" rel="stylesheet" type="text/css" />
9   </head>
10  <body id="body">
11    <ul class="tab-collection">
12      <li class="tab" title="Tab1">
13        <h1>Tab 1</h1>
14        
15      </li>
16      <li class="tab" title="Tab2">
17        <h1>Tab 2</h1>
18        
19      </li>
20      <li class="tab" title="Tab3">
21        <h1>Tab 3</h1>
22        
23      </li>
24    </ul>
25    <script type="text/javascript">
26      UI.Tabview.init('body', { width: '500px' });
27    </script>
28  </body>
29 </html>
```



ame dozen questions time and time again is all well and at about some harder, more off-beat topics?

Introduction to HTML

HTML Document Structure in Depth

The <head> Section

- Contains information that doesn't show directly on the viewable page
- Starts after the <!doctype> declaration
- Begins with <head> and ends with </head>
- Contains mandatory single <title> tag
- Can contain some other tags, e.g.
 - <meta>
 - <script>
 - <style>
 - <!-- comments -->

<head> Section: <title> tag

- Title should be placed between <head> and </head> tags

```
<title>Telerik Academy - Winter Season 2009/2010</title>
```



- Used to specify a title in the window title bar
- Search engines and people rely on titles

<head> Section: <meta>

- Meta tags additionally describe the content contained within the page

```
<meta name="description" content="HTML  
tutorial" />
```

```
<meta name="keywords" content="html, web  
design, styles" />
```

```
<meta name="author" content="Chris Brewer" />
```

```
<meta http-equiv="refresh" content="5;  
url=http://www.telerik.com" />
```

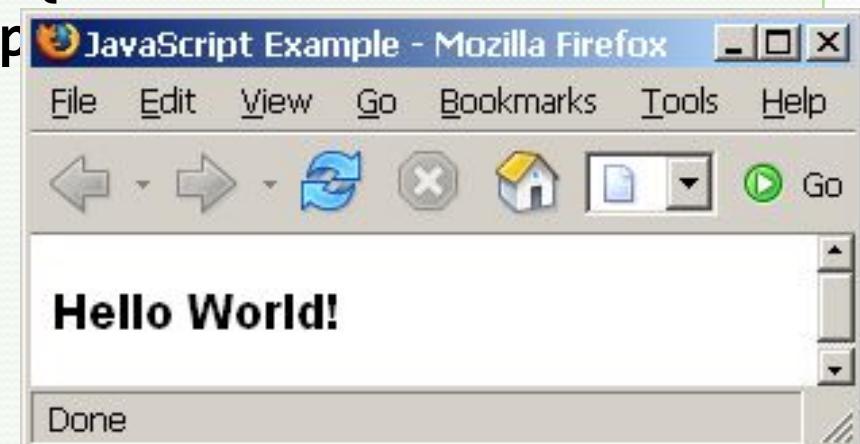
<head> Section: <script>

- The <script> element is used to embed scripts into an HTML document
 - Scripts are executed in the client's Web browser
 - Scripts can live in the <head> and in the <body> sections
- Supported client-side scripting languages:
 - JavaScript (it is not Java!)
 - VBScript
 - JScript

The <script> Tag – Example

scripts-example.html

```
<!DOCTYPE HTML>
<html>
  <head>
    <title>JavaScript Example</title>
    <script type="text/javascript">
      function sayHello() {
        document.write("<p>Hello World!</p>")
      }
    </script>
  </head>
  <body>
    <script type=
      "text/javascript">
      sayHello();
    </script>
  </body>
</html>
```

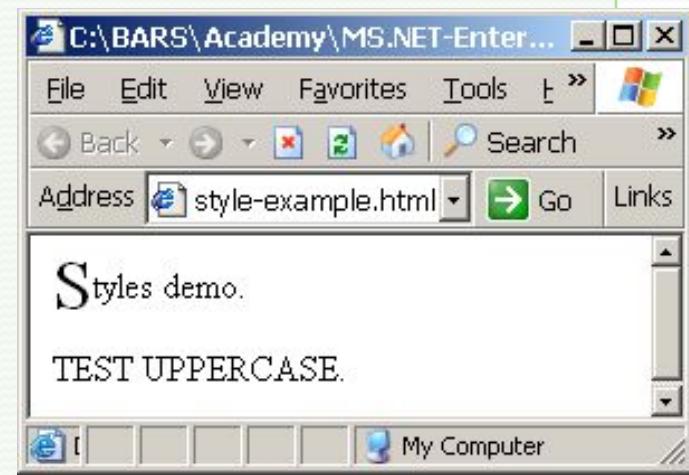


<head> Section: <style>

- The <style> element embeds formatting information (CSS styles) into an HTML page

```
<!DOCTYPE HTML>
<head>
  <style type="text/css">
    p { font-size: 12pt; line-height: 12pt; }
    p:first-letter { font-size: 200%; }
    span { text-transform: uppercase; }
  </style>
</head>
<body>
  <p>Styles demo.<br />
    <span>Test uppercase</span>.
  </p>
</body>
</html>
```

style-example.html



Comments: <!-- --> Tag

- Comments can exist anywhere between the `<html></html>` tags
- Comments start with `<!--` and end with `-->`

```
<!-- Telerik Logo (a JPG file) -->

<!-- Hyperlink to the web site -->
<a href="http://telerik.com/">Telerik</a>
<!-- Show the news table -->
<table class="newstable">
    ...
```

<body> Section: Introduction

- The <body> section describes the viewable portion of the page
- Starts after the <head> </head> section
- Begins with <body> and ends with </body>

```
<html>
  <head><title>Test page</title></head>
  <body>
    <!-- This is the Web page body -->
  </body>
</html>
```

Text Formatting

- Text formatting tags modify the text between the opening tag and the closing tag
 - Ex. **Hello** makes “Hello” bold

	bold
<i></i>	<i>italicized</i>
<u></u>	<u>underlined</u>
<sup></sup>	Sample ^{superscript}
<sub></sub>	Sample _{subscript}
	strong
	<i>emphasized</i>
<pre></pre>	Preformatted text
<blockquote></blockquote>	Quoted text block
	Deleted text – strike through

Text Formatting – Example

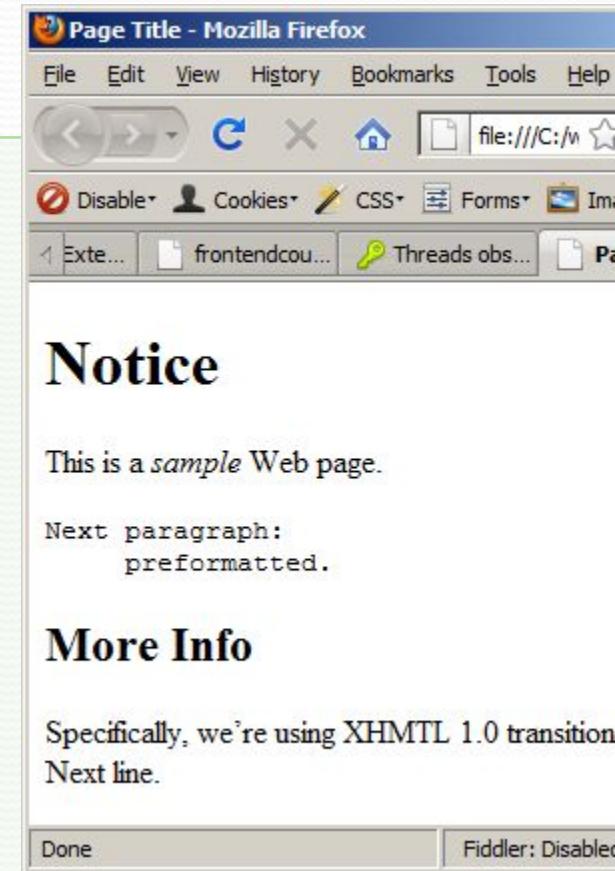
text-formatting.html

```
<!DOCTYPE HTML>
<html>
  <head>
    <title>Page Title</title>
  </head>
  <body>
    <h1>Notice</h1>
    <p>This is a <em>sample</em> Web page.</p>
    <p><pre>Next paragraph:
      preformatted.</pre></p>
    <h2>More Info</h2>
    <p>Specifically, we're using XHMTL 1.0 transitional.<br />
      Next line.</p>
  </body>
</html>
```

Text Formatting – Example (2)

text-formatting.html

```
<!DOCTYPE HTML>
<html>
  <head>
    <title>Page Title</title>
  </head>
  <body>
    <h1>Notice</h1>
    <p>This is a <em>sample</em> Web page.</p>
    <p><pre>Next paragraph:
      preformatted.</pre></p>
    <h2>More Info</h2>
    <p>Specifically, we're using HMTL />
      Next line.</p>
  </body>
</html>
```



Hyperlinks: <a> Tag

- Link to a document called `form.html` on the same server in the same directory:

```
<a href="form.html">Fill Our Form</a>
```

- Link to a document called `parent.html` on the same server in the parent directory:

```
<a href="../parent.html">Parent</a>
```

- Link to a document called `cat.html` on the same server in the subdirectory `stuff`:

```
<a href="stuff/cat.html">Catalog</a>
```

Hyperlinks: <a> Tag (2)

- Link to an external Web site:

```
<a href="http://www.devbg.org" target="_blank">BASD</a>
```

- Always use a full URL, including "http://", not just "www.somesite.com"
- Using the `target="_blank"` attribute opens the link in a new window
- Link to an e-mail address:

```
<a href="mailto:bugs@example.com?subject=Bug+Report">  
Please report bugs here (by e-mail only)</a>
```

Hyperlinks: <a> Tag (3)

- Link to a document called `apply-now.html`
 - On the same server, in same directory
 - Using an image as a link button:

```
<a href="apply-now.html"></a>
```

- Link to a document called `index.html`
 - On the same server, in the subdirectory `english` of the parent directory:

```
<a href="../english/index.html">Switch to  
English version</a>
```

Hyperlinks and Sections

- Link to another location in the same document:

```
<a href="#section1">Go to Introduction</a>
```

...

- Link to a specific location in another document:

```
<a href="chapter3.html#section3.1.1">Go to Section  
3.1.1</a>
```

```
<!-- In chapter3.html -->
```

...

```
<div id="section3.1.1">  
  <h3>3.1.1. Technical Background</h3>  
</div>
```

Hyperlinks – Example

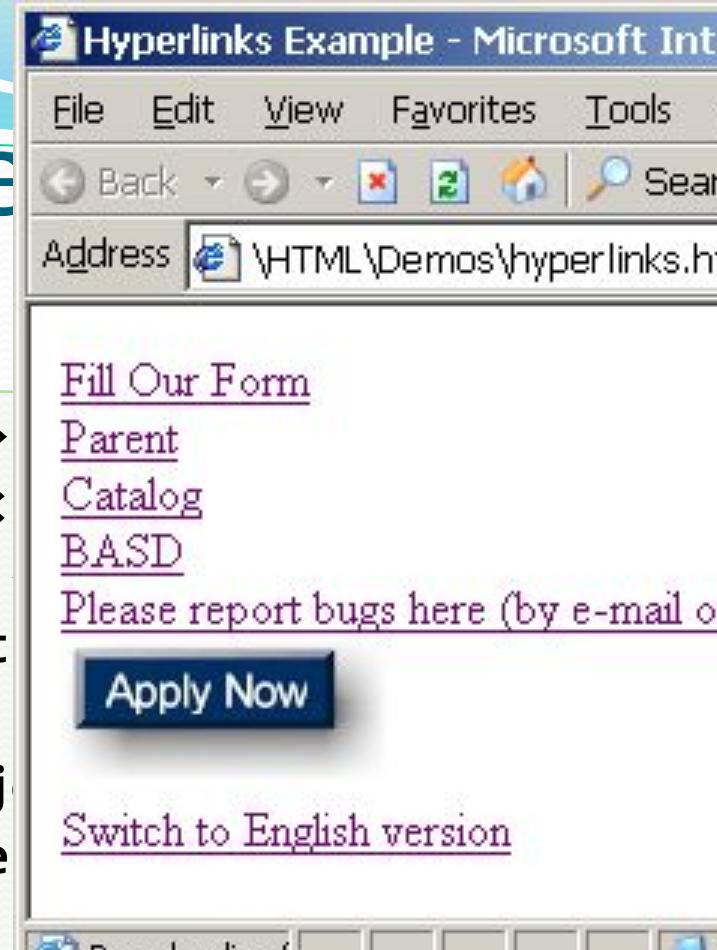
hyperlinks.html

```
<a href="form.html">Fill Our Form</a> <br />
<a href="../../parent.html">Parent</a> <br />
<a href="stuff/cat.html">Catalog</a> <br />
<a href="http://www.devbg.org" target="_blank">BASD</a>
<br />
<a href="mailto:bugs@example.com?subject=Bug
Report">Please report bugs here (by e-mail only)</a>
<br />
<a href="apply-now.html"></a> <br />
<a href="../../english/index.html">Switch to English
version</a> <br />
```

Hyperlinks – Example

hyperlinks.html

```
<a href="form.html">Fill Our Form</a>
<a href="../parent.html">Parent</a> <
<a href="stuff/cat.html">Catalog</a>
<a href="http://www.devbg.org" target
<br />
<a href="mailto:bugs@example.com?subj
Report">Please report bugs here (by e
<br />
<a href="apply-now.html">Switch to English
version</a> <br />
```



Links to the Same Document – Example

links-to-same-document.html

```
<h1>Table of Contents</h1>

<p><a href="#section1">Introduction</a><br />
<a href="#section2">Some background</A><br />
<a href="#section2.1">Project History</a><br />
...the rest of the table of contents...

<!-- The document text follows here -->

<h2 id="section1">Introduction</h2>
... Section 1 follows here ...
<h2 id="section2">Some background</h2>
... Section 2 follows here ...
<h3 id="section2.1">Project History</h3>
... Section 2.1 follows here ...
```

Links to the Same Document – Example

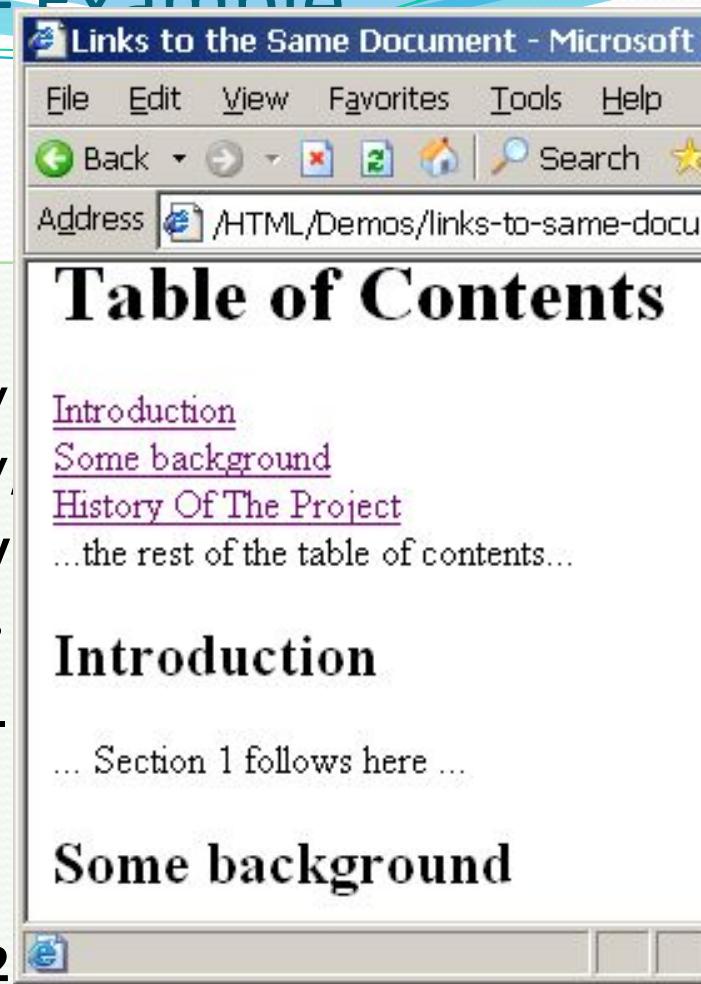
(2)

links-to-same-document.html

```
<h1>Table of Contents</h1>

<p><a href="#section1">Introduction</a>
<a href="#section2">Some background</a>
<a href="#section2.1">Project History</a>
...the rest of the table of contents.

<!-- The document text follows here --
<h2 id="section1">Introduction</h2>
... Section 1 follows here ...
<h2 id="section2">Some background</h2>
... Section 2 follows here ...
<h3 id="section2.1">Project History</h3>
... Section 2.1 follows here ...
```



Images: tag

◆ Inserting an image with tag:

```

```

◆ Image attributes:

<code>src</code>	Location of image file (relative or absolute)
<code>alt</code>	Substitute text for display (e.g. in text mode)
<code>height</code>	Number of pixels of the height
<code>width</code>	Number of pixels of the width
<code>border</code>	Size of border, 0 for no border

```

```

◆ Example:

- **HTML Basics:**
- **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>`.**

- **The major points of HTML are given below:**
- HTML stands for HyperText Markup Language.
- HTML is used to create web pages and web applications.
- HTML is widely used language on the web.
- We can create a static website by HTML only.
- Technically, HTML is a Markup language rather than a programming language.

```
<!DOCTYPE html>
<html>
<body>
<h1>My First Heading</h1>
<p>My first 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 Headings**
- HTML headings are defined with the <h1> to <h6> tags.
- <h1> defines the most important heading. <h6> defines the least important heading:
- **Example**
- <h1>This is heading 1</h1>
- <h2>This is heading 2</h2>
- <h3>This is heading 3</h3>

- An HTML element is defined by a start tag, some content, and an end tag.
- HTML Elements
- The HTML element is everything from the start tag to the end tag:
- <tagname>Content goes here...</tagname>
- Examples of some HTML elements:
- <h1>My First Heading</h1>
- <p>My first paragraph.</p>

Basic HTML Template

```
<!DOCTYPE html>

<html>
  <head>
    <title> HTML Template </title>
  </head>
  <body>
    <h1> Main Heading Here </h1>
    <p> Paragraph here. </p>
  </body>
</html>
```

- Copy this code into any text editor
- Save it as first.html
- (or give it any other name.html)
- Double click the file to open it in your browser

Tag	Description	
<u><html></u>	Defines the root of an HTML document	
<u><body></u>	Defines the document's body	
<u><h1> to <h6></u>	Defines HTML headings	
Start tag	Element content	End tag
<h1>	My First Heading	</h1>
<p>	My first paragraph.	</p>
 	<i>none</i>	<i>none</i>

- **Nested HTML Elements**
- HTML elements can be nested (this means that elements can contain other elements).
- All HTML documents consist of nested HTML elements.
- The following example contains four HTML elements (<html>, <body>, <h1> and <p>):

- **Example**

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

- **HTML Attributes**
- HTML attributes provide additional information about HTML elements.

- **HTML Attributes**
- All HTML elements can have attributes
- Attributes provide additional information about elements
- Attributes are always specified in the start tag
- Attributes usually come in name/value pairs like: name="value"

Hello, World!

```
<!DOCTYPE html>

<html lang="us-en">
  <head>
    <title> Hello, World! </title>
  </head>
  <body>
    <h1> Hello, World! </h1>
    <p> My first HTML web page.</p>
  </body>
</html>
```

- Change the title and body text from the previous template
- Add the language attribute
- Add additional paragraphs
- Save it with a different filename.html

Headings

```
<body>  
  
  <h1> This is Heading 1 </h1>  
  <h2> This is Heading 2 </h2>  
  <h3> This is Heading 3 </h3>  
  <h4> This is Heading 4 </h4>  
  <h5> This is Heading 5 </h5>  
  <h6> This is Heading 6 </h6>  
  
</body>
```

- There are six heading levels `<h1>` to `<h6>` each with a smaller font size
- The levels represent a semantic hierarchy indicating the relative importance of each heading
- `<h1>` to `<h3>` are most common
- Formatting is the domain of CSS
- Create a page with different heading levels to see the difference

Summary

```
<!DOCTYPE html>
<html> </html>
```

`lang="en-us"`

```
<head> </head>
```

```
<title> </title>
<body> </body>
```

```
<h1> </h1>
```

```
<p> </p>
```

- Tells the browser this is an HTML5 document
- Delimits the HTML document
- Adds the language attribute
- Delimits an area for information
- Creates a title in the browser title bar
- Establishes the content to display
- Declares a heading
- Declares a paragraph

Lists

1. Unordered lists
2. Ordered lists
3. Definition lists

● Unordered Lists

- An unordered list is a bulleted list
- Define Unordered List - ... Use the tags to define the start and end of an unordered list.
- A number of list items (li elements) will go within the ul tags.
- Unordered List Item - some item Add the text for each item in between some and tags.
- Each list item must have its own li tags.
- Bullet Type <ul type="disc | circle | square">
- By default a browser will show a round bullet.
- This can be changed by using the type attribute of the ul tag, which will change the bullet type for the entire list.
Item Type <li type="?">
- You can set the type of bullet for an item in the middle of the list by setting the type attribute of an li tag.

Ordered Lists

- This list is used to create an indexed list, such as a numbered or alphabetical list.
- Define Ordered List - ... Use the tags to set the start and end of the list.
- A number of list items will then go between the ordered list tags.
- Ordered List Item - an item Each item must use the tags the same as with an unordered list. But this time the browser will number each item automatically, instead of showing bullets.
- List Type <ol type="A | a | I | i | 1"> Set the type of list index by using the type="?" attribute. The default style is numeric, and you can also choose from upper or lowercase, alphabetic or roman numerals.
- List Starting Position <ol start="?"> Set the starting number (or letter) if you don't want the list to start at 1 or A.
- Item Value <li value="?"> You can set the value of an item in the middle of the list manually, if you do not want it to follow the previous letter or number.
- Simply set the value attribute of the item you wish to change. Note: subsequent items will follow the new value.

● Definition Lists

- This type of list is used to define and describe terms, much like a dictionary.
- Typically an entry in the list consists of a term, and a definition of that term.
- A browser will usually bold the term, and indent the definition.
- Define a Definition List - <dl> </dl>Set the start and end of a definition list. All entries go within the dl tags.
- Each entry will usually consist of one dt and one dd element.
Definition Title - <dt> </dt>The title of a term being defined.
- Note: you may have a term with no definition, or multiple terms with the same definition.
- Definition Description - <dd> </dd>The definition of a term. Note: you can have multiple definitions for a single term.

Unordered Lists

- For unordered list:

** **

- Start each item of the list with the ****, or list item, tag.
- The item will be preceded by a "bullet".
- Items in the list can contain more than one paragraph. For that use the **<P>** tag.
- Attribute: TYPE=disc/square/circle.

Ordered Lists

- Use the tag to start an ordered list of items, and use the to end the ordered list.
- Each item in an ordered list is indented, just as they are with unordered lists.
- Also, each list item in an ordered list is preceded with a number, in order, instead of a "bullet".

Nested Lists

- You can create nested lists in HTML by simply including the list tags inside a list item tag.

Definition Lists

- The definition list tag, or <DL>, can be used to define a definition.
- It is commonly used to describe items found in a glossary.
- The definition list is made up of a term, <DT>, and followed by one or more definition descriptions <DD> items.
- <DT> **Definition Term**
- <DD> **Definition Data / Description**
- (It is preferable to use
 instead of multiple <DD> tags within a <DT>.)

Unordered Lists

```
<ul>  
  <li> Item </li>  
  <li> Item </li>  
  <li> Item </li>  
</ul>
```

- **** tags define a non-~~list~~ list (an “unordered” list)
- **** tags define a list item within the unordered list

Ordered Lists

```
<ol>
```

```
  <li> Item 1 </li>
  <li> Item 2 </li>
  <li> Item 3 </li>
```

```
</ol>
```

- **** tags define a sequential numbered list (an “ordered” list)
- **** tags define a list item within the ordered list

Type Attribute

```
<ul type='circle'>  
  <li> Item </li>  
  <li> Item </li>  
  <li> Item </li>  
</ul>
```

- Lists items are best styled in CSS but the `type` attribute can change the ~~default~~ style
- `type="disc"` can be:
`type="square"`
`type="circle"`

Type Attribute

```
<ol type='A'>  
  <li> Item 1 </li>  
  <li> Item 2 </li>  
  <li> Item 3 </li>  
</ol>
```

- Numbers can be:
 - type = type = type = type = “i”
 - type = type = type = roman
 - type = numerals
 - etc

Nested Lists

```
<ul>
  <li> Item 1 </li>
  <li> Item 2 </li>
    <ol>
      <li> Sub Item 1 </li>
      <li> Sub Item 2 </li>
    </ol>
  <li> List Item 3 </li>
</ul>
```

- Nested lists create a list within a list
- Copy the nested code opposite
- Switch the ordered and unordered lists
- Add the type attribute to each list

Definition Lists

```
<dl>  
  <dt> Term 1 </dt>  
    <dd> description </dd>  
  <dt> Term 2 </dt>  
    <dd> description </dd>  
</dl>
```

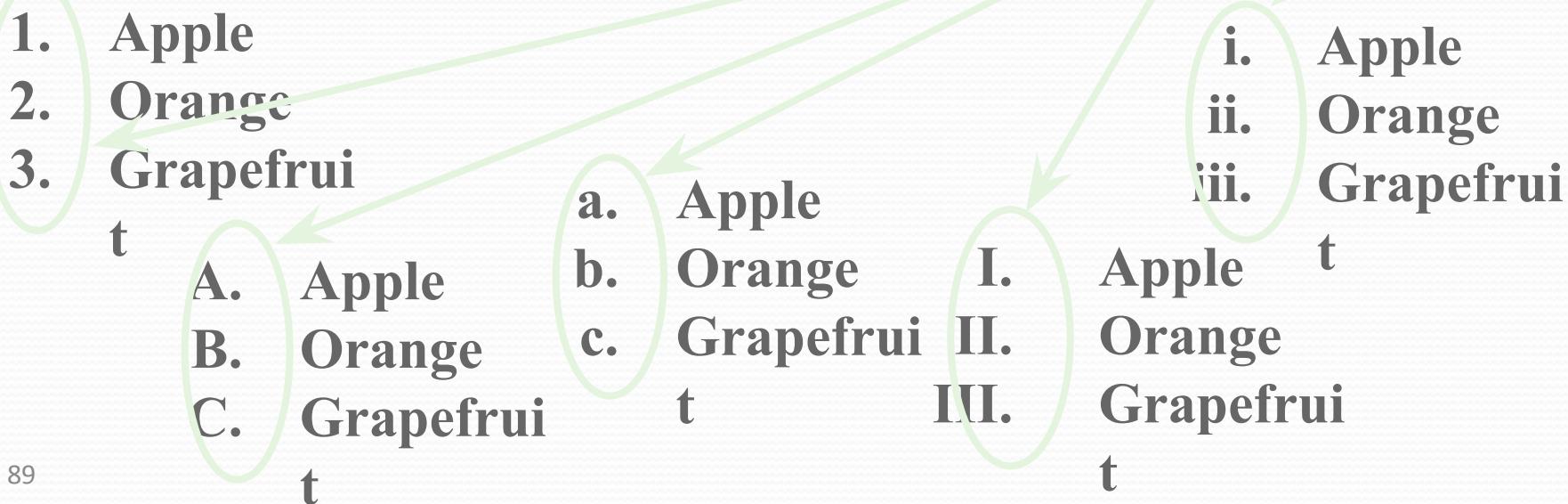
- **<dl>** tags delimit definition list
- **<dt>** tags delimit the term
- **<dd>** tags delimit the definition
- Create a definition list with several terms and definition / descriptions below each term

Ordered Lists: Tag

- Create an Ordered List using :

```
<ol type="1">
  <li>Apple</li>
  <li>Orange</li>
  <li>Grapefruit</li>
</ol>
```

- Attribute values for type are 1, A, a, I, or i



Unordered Lists: Tag

- Create an Unordered List using :

```
<ul type="disk">
  <li>Apple</li>
  <li>Orange</li>
  <li>Grapefruit</li>
</ul>
```

- Attribute values for type are:

- disc, circle or square

- Apple
- Orange
- Pear

- Apple
- Orange
- Pear

- Apple
- Orange
- Pear

Definition lists: <dl> tag

- Create definition lists using <dl>
 - Pairs of text and associated definition; text is in <dt> tag, definition in <dd> tag

```
<dl>
  <dt>HTML</dt>
  <dd>A markup language ...</dd>
  <dt>CSS</dt>
  <dd>Language used to ...</dd>
</dl>
```

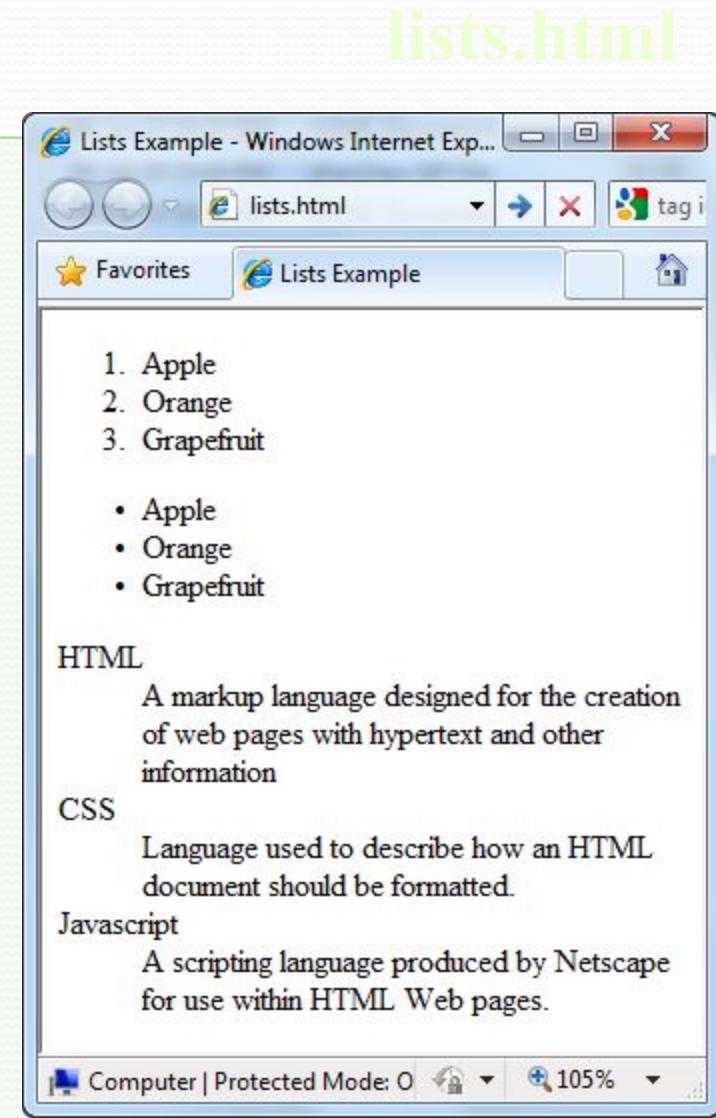
- Renders without bullets
- Definition is indented

Lists – Example

```
<ol type="1">
  <li>Apple</li>
  <li>Orange</li>
    <li>Grapefruit</li>
</ol>

<ul type="disc">
  <li>Apple</li>
  <li>Orange</li>
    <li>Grapefruit</li>
</ul>

<dl>
  <dt>HTML</dt>
  <dd>A markup lang...</dd>
</dl>
```



Summary

```
<ul> </ul>
<ol> </ol>
<li> </li>
<ul type="circle">
<dl> </dl>
<dt> </dt>
<dd> </dd>
```

- tags define an unordered list
- tags define an ordered list
- tags separate list items
- The “type” attribute specifies the marker
- <dl> tags define a “definition” list
- <dt> tags specify the term
- <dd> tags indent the description below the term

Anchor Tags - HyperLinks

```
<a href="url"> Link Text </a>
```

- Links are created using the anchor **<a>** tags
- The link address is specified in the **href** attribute
- **Link Text** becomes the underlined link the user clicks on to activate the link

Anchor Tags - HyperLinks

```
<a href="http://www.google.com">  
Google </a>
```

```
<a href="first.html">  
My First Web Page </a>
```

```
<a href="#top"> Top of Page </a>
```

- Link to an external site
- Add several external links to a page

- Link to a local file
- Create a list of the web pages you saved
and add local links (menu)

- Link to an internal page marker
- See next slide for more information

Internal Links

```
<body id="top">
```

```
<h1> Internal Link </h1>
```

```
<p> . . . </p>
```

```
<a href="#top"> Top of Page </a>
```

```
</body>
```

- To make internal links work, you need a long document
- Copy lots of text and add paragraph tags as needed
- Add an **id** attribute to the body tag
- Create an internal link at the bottom of the page using the id as the href

Image Tag

```

```

- tags have no closing tag
- A self-closing tag (as shown) can be used but is optional in HTML5
- Attributes can be written as a single line of text or with whitespace for added readability
 - The browser ignores whitespace
 - The **src** attribute is the image path
 - The **alt** attribute is for assistive devices
 - ~~efficiency~~ **width** and **height** for browser

Images as Links

```
<a href="http://www.website.com">  
    
</a>
```

- Create a clickable image
- Wrap an `` tag inside an anchor `<a>` tag

Summary

`<a> `

`href="url"`

`Link Text`

- Anchor `<a>` tags define a hyperlink
- The `href` attribute identifies the target of the link
- Link Text (or image) creates the clickable link

``

`src="path"`

`alt="text"`

`width="size"`

`height="size"`

- `` tags are used to place an image on the page
- The `src` attribute states the file location (path) for the image
- The `alt` attribute provides alternative text for the image
- The `width` and `height` attributes state the dimensions of the image

- **The href Attribute**
- The `<a>` tag defines a hyperlink. The href attribute specifies the URL of the page the link goes to:
- **Example:**
`Visit W3Schools`
- **The src Attribute:**
- The `` tag is used to embed an image in an HTML page. The src attribute specifies the path to the image to be displayed:
- **Example**
``

Miscellaneous Tags

- <hr />: Draws a horizontal rule (line):

```
<hr size="5" width="70%" />
```

- <center></center>: Deprecated!

```
<center>Hello World!</center>
```

- : Deprecated!

```
<font size="3" color="blue">Font3</font>
<font size="+4" color="blue">Font+4</font>
```

Inline vs. Block

- **Inline** elements do not create line breaks
- They stay within the normal flow of the container tag
- Links and images are inline elements

- **Block** elements begin and end with line breaks
- They break out of the flow of the parent container – and add whitespace
- Paragraphs and headings are block elements

Nested Containers

- Tags create containers
- Containers can contain other (nested) tags
 - The `<head>` and `<body>` tags are nested in the `<html>` container tag
 - The `` tags are nested in the `` and `` container tags
- Block elements can have inline elements nested within their normal flow

Generic Containers

- **<div>** and **** tags are generic containers
- The **<div>** tag is a block level container that creates a section (division) and holds other elements – typically for styling, layout, or effects
- The **** tag is an inline container used to mark text for special treatment – for styling or effects

Container Attributes

- Attributes are used to identify containers
- The attributes **id=“name”** and **class=“name”** provide script and styling references
- The **id** attribute name must be unique on the page
- The **class** attribute name can repeat on a page

Division Tags

```
<div id="div1" class="preamble">  
    <!-- other tags go here -->  
</div>
```

- **<div>** tags create a division within the page and group related elements
- The **id** and **class** attributes are used for references for scripts and CSS styles
- The **id** and **class** attributes are optional and/or used independently
- **<div>** tags are generic and have no semantic meaning

Span Tags

```
<p> Normal text starts here . . .

<span id="special" class="red">
    <!-- special text here -->
</span>

. . . normal text ends here. </p>
```

- **** tags mark text for special treatment within the normal flow (without adding whitespace)
- The **id** and **class** attributes are used as references for scripts and CSS styles
- The **id** and **class** attributes are entirely optional and/or used independently

Tables

```
<table>
  <tr>
    <td> Row 1 Col 1 </td>
    <td> Row 1 Col 2 </td>
    <td> Row 1 Col 3 </td>
  </tr>
  <tr>
    <td> Row 2 Col 1 </td>
    <td> Row 2 Col 2 </td>
    <td> Row 2 Col 3 </td>
  </tr>
</table>
```

- **<table>** tags define a data table
- **<tr>** tags define a row with line breaks
- **<td>** tags define a data cell, and display inline, horizontally, in a row

- Copy this code to create a table with 2 rows with 3 columns
- Add additional rows and columns

Table Headings

```
<table>
  <tr>
    <th> Col 1 Heading </th>
    <th> Col 2 Heading </th>
    <th> Col 3 Heading </th>
  </tr>
  <tr>
    <td> Row 1 Col 1 </td>
    <td> Row 1 Col 2 </td>
    <td> Row 1 Col 3 </td>
  </tr>
</table>
```

- **<th>** tags are inline elements that can be used to define column headings
- **<th>** tags replace **<td>** tags in the top row for column headings
- <th> tags replace the first <td> tag of every new row to create row labels

Row Span and Column Span

```
<th colspan="3"> Heading </th>
<td colspan="2"> Wide Data </td>

<td rowspan="2"> Long Data </tr>
```

- The **colspan** attribute merges multiple cells into a single cell that spans more than 1 column
- The **attribute merges cells vertically to span more than 1 row
 - The number indicates how many rows to merge (span)**
- Merged plus all other cells must equal the number of cells in the original row and column count

Caption

```
<table>  
  <caption>  
    <p> Title Goes Here  
    <p> This table shows . . .  
  </caption>  
  ...  
</table>
```

- The **<caption>** tag is a container for a table title and/or description
- The **<caption>** tag must directly follow the **<table>** tag
- Long captions can also be done with **<table>** tag standard html tags placed before the

Summary 5

<table> </table>

<tr> </tr>

<td> </td>

<th> </th>

colspan="2"

rowspan="2"

<caption> </caption>

- <table> tag defines a table container
- <tr> tags define a table row
- <td> tags define a cell for table data
- <th> tags define a table heading
- **colspan** specifies a number of columns to span
- **rowspan** specifies a number of rows to span
- <caption> provide for a title and brief description

Text Formatting

- CSS is the preferred way to change the browser's display style
- The separation of style and markup is a recommended best practice
- However, there are some basic text formatting tags that can be used in HTML
 - These are inline tags used to mark text within the flow of another tag
 - Those included here are generic tags – without any semantic meaning

Text Markup

- | | |
|---|---------------------------------|
| ● <code> ... </code> | ● □ Renders as bold text |
| ● <code><i> ... </i></code> | ● □ Renders as italicized text |
| ● <code><small> ... </small></code> | ● □ Sets a smaller font size |
| ● <code><sub> ... </sub></code> | ● □ Creates subscripted text |
| ● <code><sup> ... </sup></code> | ● □ Creates superscripted text |
| ● <code><ins> ... </ins></code> | ● □ Marks as inserted text |
| ● <code> ... </code> | ● □ Marks as deleted text |
| ● <code><mark> ... </mark></code> | ● □ Renders as highlighted text |

Quotations

<blockquote>

Four score and seven years ago
our fathers brought forth, upon
this continent, a new nation,
conceived in liberty, and
dedicated to the proposition that
all men are created equal.

</blockquote>

- <q> tags are used for short inline quotations with quotation marks
- He said, <q> I will return </q>
- The <blockquote> tag is a block level element used for longer passages and displays with indented text
- A cite attribute and a cite element are also available

Line Breaks

< p >

Without line break tags **
**
the browser would display **
**
this as one continuous paragraph.

</ p >

- Web browsers ignore whitespace
- The **
** tag can be used to force a linebreak

Preformatted Text

<pre>

This paragraph will render
with the whitespace - both

Vertical Horizontal

It does not need
any additional tags
to force these line breaks

</pre>

- Web browsers ignore whitespace
- The <pre> tag preserves whitespace both vertically and horizontally
- Use where whitespace is significant

Comments and Horizontal Rule

```
<!-- comment -->
```

- Add comments to html code
where needed

```
<hr>
```

- The **<hr>** tag adds a horizontal rule
as separator line across the
screen

- An **HTML element** is defined by a starting tag. If the element contains other content, it ends with a closing tag, where the element name is preceded by a forward slash as shown below with few tags –

Start Tag	Content	End Tag
<p>	This is paragraph content.	</p>
<h1>	This is heading content.	</h1>
<div>	This is division content.	</div>

- So here <p>....</p> is an HTML element, <h1>...</h1> is another HTML element.
- There are some HTML elements which don't need to be closed, such as <img.../>, <hr /> and
 elements. These are known as **void elements**.

● HTML Tag vs. Element

- An HTML element is defined by a *starting tag*. If the element contains other content, it ends with a *closing tag*.
- For example, **<p>** is starting tag of a paragraph and **</p>** is closing tag of the same paragraph but **<p>This is paragraph</p>** is a paragraph element.
- Nested HTML Elements
- It is very much allowed to keep one HTML element inside another HTML element –

```
<!DOCTYPE html>
<html>
  <head>
    <title>Nested Elements Example</title>
  </head>
  <body>
    <h1>This is <i>italic</i> heading</h1>
    <p>This is <u>underlined</u> paragraph</p>
  </body>
</html>
```

All Attributes of the anchor Element

All Attributes of the **anchor** Element

Attribute name	Values	Notes
target	_blank _parent _self _top frame name	Specifies the context in which the linked resource will open.
title	text	Defines the title of a link, which appears to the user as a tooltip.
href	url	Specifies the linked document, resource, or location.

- **HTML Image**
- HTML img tag is used to display image on the web page.
HTML img tag is an empty tag that contains attributes only,
closing tags are not used in HTML image element.
- <h2>HTML Image Example</h2>
-

```

```

- **Attributes of HTML img tag**
 - The src and alt are important attributes of HTML img tag.
 - All attributes of HTML image tag are given below.
- 1) **src**
 - It is a necessary attribute that describes the source or path of the image. It instructs the browser where to look for the image on the server.
 - The location of image may be on the same directory or another server.
 - 2) **alt**
 - The alt attribute defines an alternate text for the image, if it can't be displayed.
 - The value of the alt attribute describe the image in words. The alt attribute is considered good for SEO prospective.
 - 3) **width**
 - It is an optional attribute which is used to specify the width to display the image. It is not recommended now.
 - You should apply CSS in place of width attribute.
 - 4) **height**
 - The HTML height attribute also supports iframe, image and object elements.
 - It is not recommended now. You should apply CSS in place of height attribute.

Use of alt attribute:

- We can use alt attribute with `tag`.
- It will display an alternative text in case if image cannot be displayed on browser.

Following is the example for alt attribute:

- To insert an image in your web, that image must be present in your same folder where you have put the HTML file.
- But if in some case image is available in some other directory then you can access the image like this:

```

```

- **Use `img` tag as a link**
- We can also link an image with other page or we can use an image as a link.
- To do this, put `img` tag inside the `a` tag.

Example:

```
<a href="https://www.javatpoint.com/what-is-  
robotics"></a>
```

Block and Inline Elements

- Block elements add a line break before and after them
 - `<div>` is a block element
 - Other block elements are `<table>`, `<hr>`, headings, lists, `<p>` and etc.
- Inline elements don't break the text before and after them
 - `` is an inline element
 - Most HTML elements are inline, e.g. `<a>`

The <div> Tag

- <div> creates logical divisions w/ content
- Block style element
- Used with CSS
- Example:



div-and-span.html

```
<div style="font-size:24px; color:red">DIV  
example</div>  
  
<p>This one is <span style="color:red;  
font-weight:bold">only a test</span>.</p>
```

The Tag

- Inline style element
- Useful for modifying a specific portion
 - Don't create a separate area (part of the document)
- Very useful with CSS

span.html

```
<p>This one is <span style="color:red;  
font-weight:bold">only a test</span>.</p>  
  
<p>This one is another <span style="font-size:32px;  
font-weight:bold">TEST</span>.</p>
```



HTML Tables

Title	Title	Title	Title	Title	Title
Data	Data	Data	Data	Data	Data
Data	Data	Data	Data	Data	Data
Data	Data	Data	Data	Data	Data
Data	Data	Data	Data	Data	Data
Data	Data	Data	Data	Data	Data

```
<html>
<head>
<title>How To Create HTML Tables</title>
</head>
<body>
<table border=1 cellspacing=0 cellpadding=0>
<tr>
<td width=110 valign=top>
<br>upper left corner
</td>
<td width=110 valign=top>
<br>upper right corner
</td>
</tr>
<tr>
<td width=110 valign=top>
<br>left center cell
</td>
<td width=110 valign=top>
<br>right center cell
</td>
</tr>
<tr>
<td width=110 valign=top>
<br>lower left corner
</td>
<td width=110 valign=top>
<br>lower right corner
</td>
</tr>
</table>
</body>
</html>
```

HTML Tables

- Tables represent tabular data
 - A table consists of one or several rows
 - Each row has one or more columns
- Tables comprised of several core tags:
`<table></table>`: begin / end the table
`<tr></tr>`: create a table row
`<td></td>`: create tabular data (cell)
- Tables should not be used for layout. Use CSS floats and positioning styles instead

HTML Tables

- Start and end of a table

```
<table> ... </table>
```

- Start and end of a row

```
<tr> ... </tr>
```

- Start and end of a cell in a row

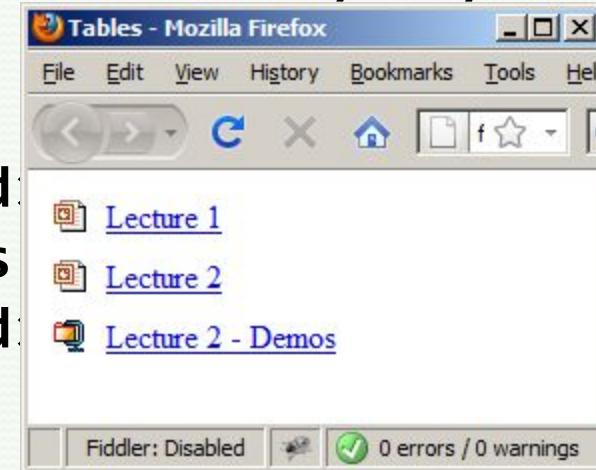
```
<td> ... </td>
```

Simple HTML Tables – Example

```
<table cellspacing="0" cellpadding="5">
  <tr>
    <td></td>
    <td><a href="lecture1.ppt">Lecture 1</a></td>
  </tr>
  <tr>
    <td></td>
    <td><a href="lecture2.ppt">Lecture 2</a></td>
  </tr>
  <tr>
    <td></td>
    <td><a href="lecture2-demos.zip">
      Lecture 2 - Demos</a></td>
  </tr>
</table>
```

Simple HTML Tables – Example (2)

```
<table cellspacing="0" cellpadding="5">
  <tr>
    <td></td>
    <td><a href="lecture1.ppt">Lecture 1</a></td>
  </tr>
  <tr>
    <td></td>
    <td><a href="lecture2.ppt">Lecture 2</a></td>
  </tr>
  <tr>
    <td></td>
    <td><a href="lecture2-demos">
      Lecture 2 - Demos</a></td>
  </tr>
</table>
```



Complete HTML Tables

- Table rows split into three semantic sections: header, body and footer
 - <thead> denotes table header and contains <th> elements, instead of <td> elements
 - <tbody> denotes collection of table rows that contain the very data
 - <tfoot> denotes table footer but comes BEFORE the <tbody> tag
 - <colgroup> and <col> define columns (most often used to set column widths)

Complete HTML Table: Example

```
<table>
  <colgroup>
    <col style="width:100px" /><col />
  </colgroup>
  <thead>
    <tr><th>Column 1</th><th>Column 2</th></tr>
  </thead>
  <tfoot>
    <tr><td>Footer 1</td><td>Footer 2</td>
  </tfoot>
  <tbody>
    <tr><td>Cell 1.1</td><td>Cell 1.2</td></tr>
    <tr><td>Cell 2.1</td><td>Cell 2.2</td></tr>
  </tbody>
```

columns

header

th

footer

Last comes the body (data)

Complete HTML Table:

Example

```
<table>
```

```
  <colgroup>
```

```
    <col style="width:150px;" />
```

```
  </colgroup>
```

```
  <thead>
```

```
    <tr><th>Column 1</th></tr>
```

```
  </thead>
```

```
  <tfoot>
```

```
    <tr><td>Footer 1</td><td>Footer 2</td></tr>
```

```
  </tfoot>
```

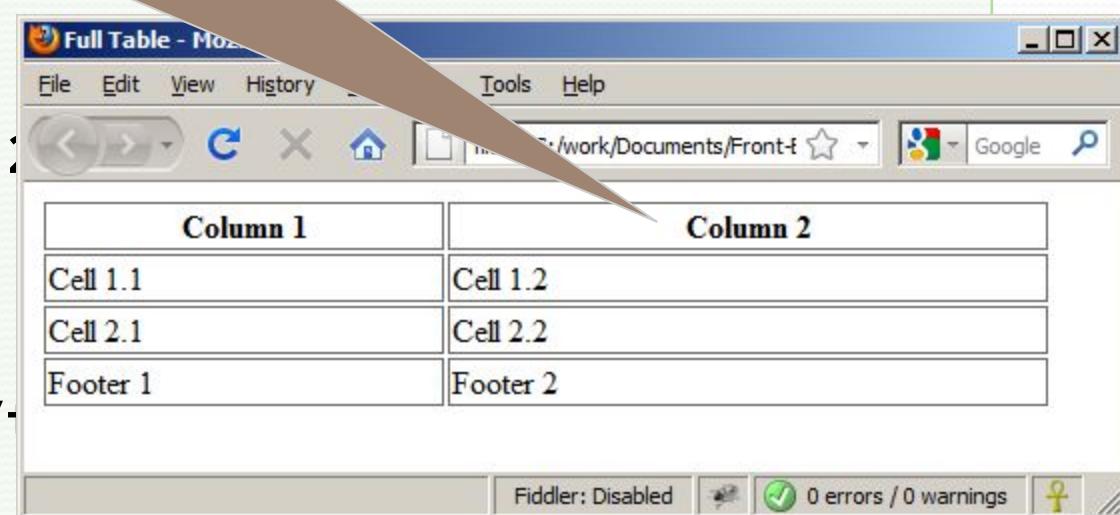
```
  <tbody>
```

```
    <tr><td>Cell 1.1</td><td>Cell 1.2</td></tr>
```

```
    <tr><td>Cell 2.1</td><td>Cell 2.2</td></tr>
```

By default, header text
is bold and centered.

table-full.html



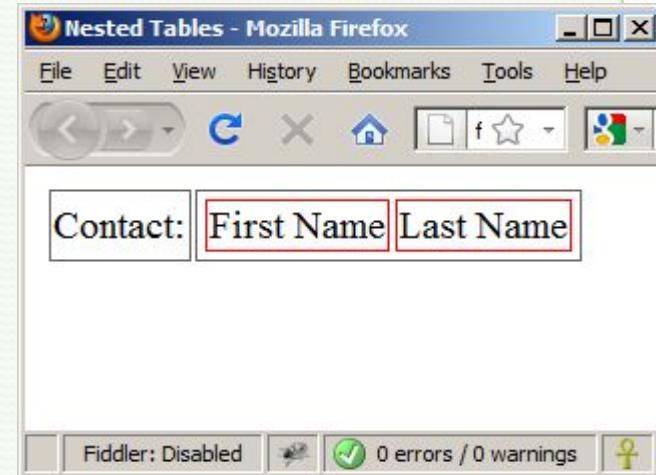
Although the footer is
before the data in the
code, it is displayed last

Nested Tables

- Table data “cells” (`<td>`) can contain nested tables (tables within tables):

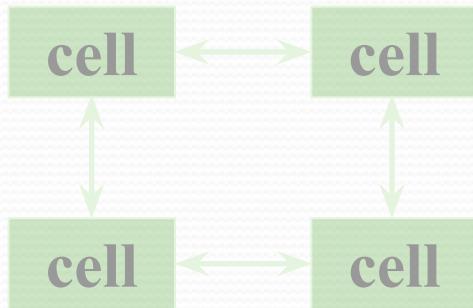
`nested-tables.html`

```
<table>
  <tr>
    <td>Contact:</td>
    <td>
      <table>
        <tr>
          <td>First Name</td>
          <td>Last Name</td>
        </tr>
      </table>
    </td>
  </tr>
</table>
```



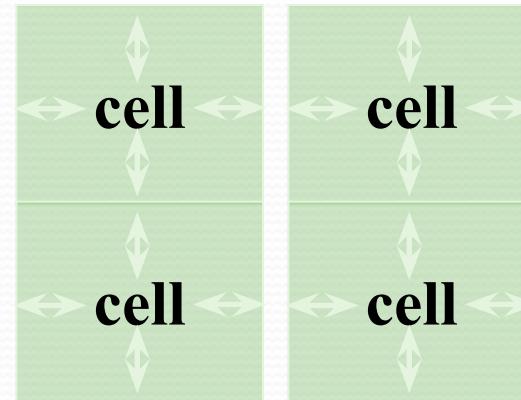
- Tables have two important attributes:
Cell Spacing and Padding

- ◆ **cellspacing**



- ◆ **Defines the empty space between cells**

- ◆ **cellpadding**



- ◆ **Defines the empty space around the cell content**

Cell Spacing and Padding – Example

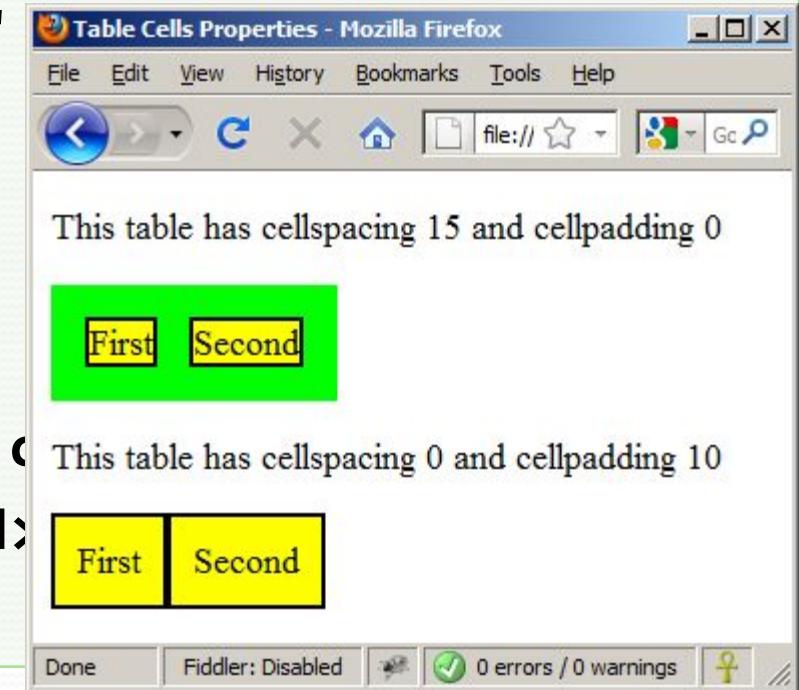
table-cells.html

```
<html>
  <head><title>Table Cells</title></head>
  <body>
    <table cellspacing="15" cellpadding="0">
      <tr><td>First</td>
      <td>Second</td></tr>
    </table>
    <br/>
    <table cellspacing="0" cellpadding="10">
      <tr><td>First</td><td>Second</td></tr>
    </table>
  </body>
</html>
```

Cell Spacing and Padding – Example (2)

table-cells.html

```
<html>
  <head><title>Table Cells</title></head>
  <body>
    <table cellspacing="15"
      <tr><td>First</td>
      <td>Second</td></tr>
    </table>
    <br/>
    <table cellspacing="0" <
      <tr><td>First</td><td>
    </table>
  </body>
</html>
```



Column and Row Span

- Table cells have two important attributes:

- ◆ **colspan**

colspan="1"

cell[1,1]

colspan="1"

cell[1,2]

cell[2,1]

colspan="2"

rowspan="2"

cell[1,1]

rowspan="1"

cell[1,2]

cell[2,1]

rowspan="1"

- ◆ Defines how many columns the cell

- ◆ **rowspan**

- ◆ Defines how many rows the

table-colspan-rowspan.html

Column and Row Span – Example

```
<table cellspacing="0">
  <tr class="1"><td>Cell[1,1]</td>
    <td colspan="2">Cell[2,1]</td></tr>
  <tr class="2"><td>Cell[1,2]</td>
    <td rowspan="2">Cell[2,2]</td>
    <td>Cell[3,2]</td></tr>
  <tr class="3"><td>Cell[1,3]</td>
    <td>Cell[2,3]</td></tr>
</table>
```

Column and Row Span –

Example (2)

table-colspan-rowspan.html

```
<table cellspacing="0">
    <tr class="1"><td>Cell[1,1]</td>
        <td colspan="2">Cell[2,1]</td></tr>
    <tr class="2"><td>Cell[1,2]</td>
        <td rowspan="2">Cell[1,3]</td>
        <td>Cell[2,2]</td>
    <tr class="3">
        <td>Cell[1,2]</td>
        <td>Cell[2,2]</td>
        <td>Cell[3,2]</td>
    </tr>
</table>
```

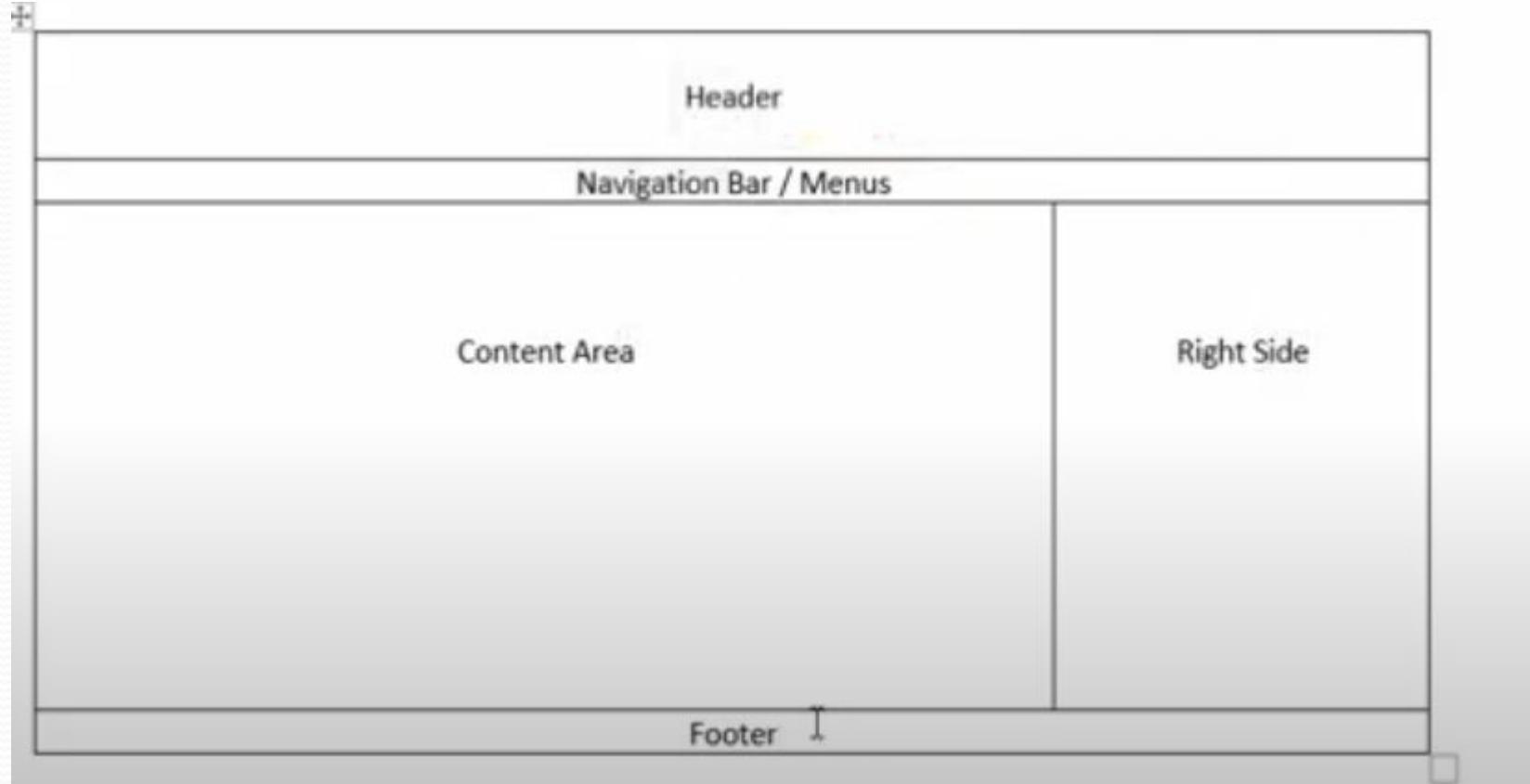
Cell[1,1]	Cell[2,1]	
Cell[1,2]	Cell[2,2]	Cell[3,2]
Cell[1,3]		Cell[2,3]

ID	Name	Email	Pic
01	Programmer	admin@onlineittuts.com	 GitHub Tutorial For Beginners

Copyright by OnlineITtuts

[OnlineITtuts](#)

```
<table border="02px" width="600" bgcolor="yellow">
<tr align="center">
<th> ID </th>
<th> Name</th>
<th> Email</th>
<th> Pic</th>
</tr>
<tr align="center" bgcolor="gray">
<td> 01 </td>
<td> Programmer</td>
<td> admin@onlineittuts.com</td>
<td></td>
</tr>
<tr align="center">
<td colspan="4">Copyright by OnlineITtuts</td>
</tr>
<tr align="center">
<td colspan="4"><a href="https://www.onlineittuts.com">OnlineITtu
</tr>
</table>
```



```
<!DOCTYPE html>
<html>
  <head>
    <title>Div Tag in HTML5</title>
  </head>
  <body>

    <!--Main Div-->
    <div class="main">

      <!--Header-->
      <div class="header">
        <h2>Header Div</h2>
      </div>

      <!--Navigation-->
      <div class="navigation">
        <h2>Navigation Div</h2>
      </div>
    
```

```
<style>

    .main
    {
        width: 980px;
        height: 900px;
    }

    .header
    {
        |
    }

</style>
```



HTML Forms

Entering User Data from a Web Page

Send me an application now!

First Name:

Last Name:

Age: 1-17 yrs 18 yrs and over

I would like to learn to work with:

watercolors
 acrylics
 pastels

I am interested in art lessons because:

Registration Form - Mozilla Fi...

User name:

Password:

Gender: Male Female

Click to accept our terms:

Done

Done

HTML Forms

- Forms are the primary method for gathering data from site visitors
- Create a form block

```
<form></form>
```

The “method” attribute tells how the form data should be sent – via GET or POST request

- Example:

```
<form name="myForm" method="post"  
action="path/to/some-script.php">
```

```
...
```

```
</form>
```

The "action" attribute tells where the form data should be sent

Form Fields

- Single-line text input fields:

```
<input type="text" name="FirstName" value="This  
is a text field" />
```

- Multi-line textarea fields:

```
<textarea name="Comments">This is a multi-line  
text field</textarea>
```

- Hidden fields contain data not shown to the user:

```
<input type="hidden" name="Account" value="This  
is a hidden text field" />
```

- Often used by JavaScript code

Fieldsets

- Fieldsets are used to enclose a group of related form fields:

```
<form method="post" action="form.aspx">
  <fieldset>
    <legend>Client Details</legend>
    <input type="text" id="Name" />
    <input type="text" id="Phone" />
  </fieldset>
  <fieldset>
    <legend>Order Details</legend>
    <input type="text" id="Quantity" />
    <textarea cols="40" rows="10"
      id="Remarks"></textarea>
  </fieldset>
</form>
```

- The `<legend>` is the fieldset's title.

Form Input Controls

- Checkboxes:

```
<input type="checkbox" name="fruit"  
value="apple" />
```

- Radio buttons:

```
<input type="radio" name="title" value="Mr." />
```

- Radio buttons can be grouped, allowing only one to be selected from a group:

```
<input type="radio" name="city" value="Lom" />  
<input type="radio" name="city" value="Ruse" />
```

Other Form Controls

- Dropdown menus:

```
<select name="gender">  
    <option value="Value 1"  
        selected="selected">Male</option>  
    <option value="Value 2">Female</option>  
    <option value="Value 3">Other</option>  
</select>
```

- Submit button:

```
<input type="submit" name="submitBtn"  
value="Apply Now" />
```

Other Form Controls (2)

- Reset button – brings the form to its initial state

```
<input type="reset" name="resetBtn"  
value="Reset the form" />
```

- Image button – acts like submit but image is displayed and click coordinates are sent

```
<input type="image" src="submit.gif"  
name="submitBtn" alt="Submit" />
```

- Ordinary button – used for Javascript, no default action

```
<input type="button" value="click me" />
```

Other Form Controls (3)

- Password input – a text field which masks the entered text with * signs

```
<input type="password" name="pass" />
```

- Multiple select field – displays the list of items in multiple lines, instead of one

```
<select name="products" multiple="multiple">
  <option value="Value 1"
    selected="selected">keyboard</option>
  <option value="Value 2">mouse</option>
  <option value="Value 3">speakers</option>
</select>
```

Other Form Controls (4)

- File input – a field used for uploading files

```
<input type="file" name="photo" />
```

- When used, it requires the form element to have a specific attribute:

```
<form enctype="multipart/form-data">  
...  
  <input type="file" name="photo" />  
...  

```

Labels

- Form labels are used to associate an explanatory text to a form field using the field's ID.

```
<label for="fn">First Name</label>
<input type="text" id="fn" />
```

- Clicking on a label focuses its associated field (checkboxes are toggled, radio buttons are checked)
- Labels are both a usability and accessibility feature and are required in order to pass accessibility validation.

HTML Forms – Example

```
<form method="post" action="apply-now.php">
<input name="subject" type="hidden" value="Class" />
<fieldset><legend>Academic information</legend>
  <label for="degree">Degree</label>
  <select name="degree" id="degree">
    <option value="BA">Bachelor of Art</option>
    <option value="BS">Bachelor of Science</option>
    <option value="MBA" selected="selected">Master of
      Business Administration</option>
  </select>
  <br />
  <label for="studentid">Student ID</label>
  <input type="password" name="studentid" />
</fieldset>
<fieldset><legend>Personal Details</legend>
  <label for="fname">First Name</label>
  <input type="text" name="fname" id="fname" />
  <br />
  <label for="lname">Last Name</label>
  <input type="text" name="lname" id="lname" />
```

HTML Forms – Example (2)

```
<br />
Gender:
<input name="gender" type="radio" id="gm" value="m" />
<label for="gm">Male</label>
<input name="gender" type="radio" id="gf" value="f" />
<label for="gf">Female</label>
<br />
<label for="email">Email</label>
<input type="text" name="email" id="email" />
</fieldset>
<p>
<textarea name="terms" cols="30" rows="4"
readonly="readonly">TERMS AND CONDITIONS...</textarea>
</p>
<p>
<input type="submit" name="submit" value="Send Form" />
```

HTML F

(3)

HTML Forms Example - Mozilla Firefox

File Edit View History Bookmarks Tools Help

file:///C:/work/Di

Academic information

Degree Master of Business Administration

Student ID

Geography
Mathematics

Classes attended English

Personal Details

First Name

Last Name

Gender: Male Female

Email

TERMS AND CONDITIONS...

Send Form Clear Form

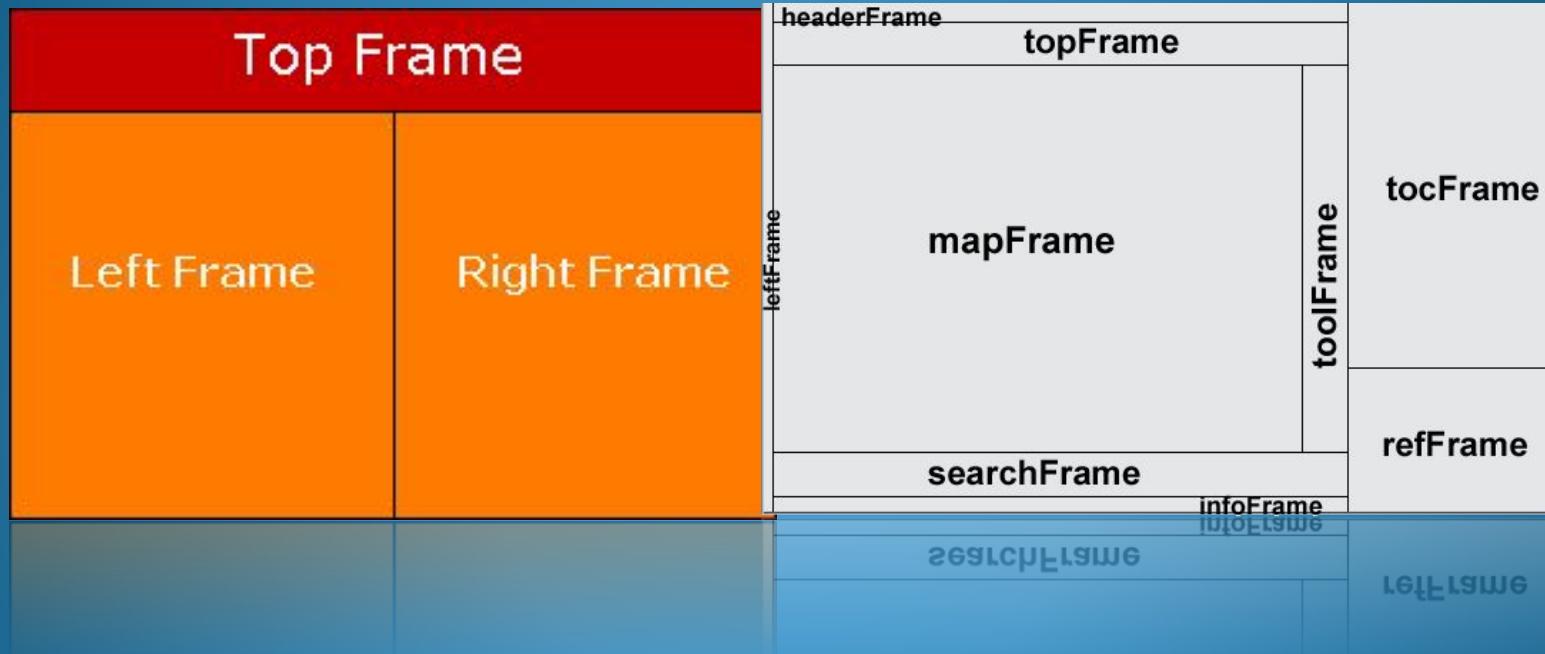
Done Fiddler: Disabled 0 errors / 0 warnings

This screenshot shows a Mozilla Firefox browser window displaying an HTML form titled "HTML Forms Example". The form is divided into sections: "Academic information" and "Personal Details". In the "Academic information" section, there is a dropdown menu set to "Master of Business Administration", a dropdown menu for "Student ID" containing "Geography" and "Mathematics", and a dropdown menu for "Classes attended" containing "English". In the "Personal Details" section, there are input fields for "First Name" and "Last Name", and a gender selection field with "Male" and "Female" options, where "Female" is selected. There is also an input field for "Email". Below the form, a large text area contains the placeholder text "TERMS AND CONDITIONS...". At the bottom of the form, there are two buttons: "Send Form" and "Clear Form". The Firefox status bar at the bottom indicates "Fiddler: Disabled" and shows 0 errors and 0 warnings.

TabIndex

- The tabindex HTML attribute controls the order in which form fields and hyperlinks are focused when repeatedly pressing the TAB key
 - tabindex="0" (zero) - "natural" order
 - If X > Y, then elements with tabindex="X" are iterated before elements with tabindex="Y"
 - Elements with negative tabindex are skipped, however, this is not defined in the standard

```
<input type="text" tabindex="10" />
```



HTML Frames

<frameset>, <frame> and <iframe>

HTML Frames

- Frames provide a way to show multiple HTML documents in a single Web page
- The page can be split into separate views (frames) horizontally and vertically
- Frames were popular in the early ages of HTML development, but now their usage is rejected
- Frames are not supported by all user agents (browsers, search engines, etc.)
 - A <noframes> element is used to provide content for non-compatible agents.

HTML Frames – Demo

frames.html

```
<html>

  <head><title>Frames Example</title></head>

  <frameset cols="180px,* ,150px">
    <frame src="left.html" />
    <frame src="middle.html" />
    <frame src="right.html" />
  </frameset>

</html>
```

- ◆ Note the target attribute applied to the elements in the left frame.

Inline Frames: <iframe>

- Inline frames provide a way to show one website inside another website:

[iframe-demo.html](#)

```
<iframe name="iframeGoogle" width="600" height="400"  
src="http://www.google.com" frameborder="yes"  
scrolling="yes"></iframe>
```

HTML5 Tags/Elements

Tag	Description
<a>	Defines a hyperlink.
<abbr>	Defines an abbreviated form of a longer word or phrase.
<acronym>	Obsolete Defines an acronym. Use <abbr> instead.
<address>	Specifies the author's contact information.
<applet>	Obsolete Embeds a Java applet (mini Java applications) on the page. Use <object> instead.
<area>	Defines a specific area within an image map.
<article> S	Defines an article.
<aside> S	Defines some content loosely related to the page content.
<audio> S	Embeds a sound, or an audio stream in an HTML document.
	Displays text in a bold style.
<base>	Defines the base URL for all relative URLs in a document.

<basefont>	Obsolete Specifies the base font for a page. Use CSS instead.
<bdi> 5	Represents text that is isolated from its surrounding for the purposes of bidirectional text formatting.
<bdo>	Overrides the current text direction.
<big>	Obsolete Displays text in a large size. Use CSS instead.
<blockquote>	Represents a section that is quoted from another source.
<body>	Defines the document's body.
 	Produces a single line break.
<button>	Creates a clickable button.
<canvas> 5	Defines a region in the document, which can be used to draw graphics on the fly via scripting (usually JavaScript).
<caption>	Defines the caption or title of the table.

`<center>` Obsolete Align contents in the center. Use CSS instead.

<code><cite></code>	Indicates a citation or reference to another source.
<code><code></code>	Specifies text as computer code.
<code><col></code>	Defines attribute values for one or more columns in a table.
<code><colgroup></code>	Specifies attributes for multiple columns in a table.
<code><data></code> 5	Links a piece of content with a machine-readable translation.
<code><datalist></code> 5	Represents a set of pre-defined options for an <code><input></code> element.
<code><dd></code>	Specifies a description, or value for the term (<code><dt></code>) in a description list (<code><dl></code>).
<code></code>	Represents text that has been deleted from the document.
<code><details></code> 5	Represents a widget from which the user can obtain additional information or controls on-demand.
<code><dfn></code>	Specifies a definition.
<code><dialog></code> 5	Defines a dialog box or subwindow.

<dir>	Obsolete Defines a directory list. Use instead.
<div>	Specifies a division or a section in a document.
<dl>	Defines a description list.
<dt>	Defines a term (an item) in a description list.
	Defines emphasized text.
<embed> 5	Embeds external application, typically multimedia content like audio or video into an HTML document.
<fieldset>	Specifies a set of related form fields.
<figcaption> 5	Defines a caption or legend for a figure.
<figure> 5	Represents a figure illustrated as part of the document.
	Obsolete Defines font, color, and size for text. Use CSS instead.
<footer> 5	Represents the footer of a document or a section.
<form>	Defines an HTML form for user input.

<frame>	Obsolete	Defines a single frame within a frameset.
<frameset>	Obsolete	Defines a collection of frames or other frameset.
<head>		Defines the head portion of the document that contains information about the document such as title.
<header> 5		Represents the header of a document or a section.
<hgroup> 5		Defines a group of headings.
<h1> to <h6>		Defines HTML headings.
<hr>		Produce a horizontal line.
<html>		Defines the root of an HTML document.
<i>		Displays text in an italic style.
<iframe>		Displays a URL in an inline frame.
		Represents an image.
<input>		Defines an input control.

<ins>	Defines a block of text that has been inserted into a document.
<kbd>	Specifies text as keyboard input.
<keygen> 	Represents a control for generating a public-private key pair.
<label>	Defines a label for an <input> control.
<legend>	Defines a caption for a <fieldset> element.
	Defines a list item.
<link>	Defines the relationship between the current document and an external resource.
<main> 	Represents the main or dominant content of the document.
<map>	Defines a client-side image-map.
<mark> 	Represents text highlighted for reference purposes.
<menu>	Represents a list of commands.
<menuitem> 	Defines a list (or menuitem) of commands that a user can perform.

<meta>	Provides structured metadata about the document content.
<meter> 5	Represents a scalar measurement within a known range.
<nav> 5	Defines a section of navigation links.
<noframes>	Obsolete Defines an alternate content that displays in browsers that do not support frames.
<noscript>	Defines alternative content to display when the browser doesn't support scripting.
<object>	Defines an embedded object.
	Defines an ordered list.
<optgroup>	Defines a group of related options in a selection list.
<option>	Defines an option in a selection list.
<output> 5	Represents the result of a calculation.
<p>	Defines a paragraph.
<param>	Defines a parameter for an object or applet element.

<picture> 	Defines a container for multiple image sources.
<pre>	Defines a block of preformatted text.
<progress> 	Represents the completion progress of a task.
<q>	Defines a short inline quotation.
<rp> 	Provides fall-back parenthesis for browsers that don't support ruby annotations.
<rt> 	Defines the pronunciation of character presented in a ruby annotations.
<ruby> 	Represents a ruby annotation.
<s>	Represents contents that are no longer accurate or no longer relevant.
<samp>	Specifies text as sample output from a computer program.
<script>	Places script in the document for client-side processing.
<section> 	Defines a section of a document, such as header, footer etc.
<select>	Defines a selection list within a form.

<select>	Defines a selection list within a form.
<small>	Displays text in a smaller size.
<source> <small>5</small>	Defines alternative media resources for the media elements like <audio> or <video> .
	Defines an inline styleless section in a document.
<strike>	<small>Obsolete</small> Displays text in strikethrough style.
	Indicate strongly emphasized text.
<style>	Inserts style information (commonly CSS) into the head of a document.
<sub>	Defines subscripted text.
<summary> <small>5</small>	Defines a summary for the <details> element.
<sup>	Defines superscripted text.
<svg> <small>5</small>	Embed SVG (Scalable Vector Graphics) content in an HTML document.
<table>	Defines a data table.

<tbody>	Groups a set of rows defining the main body of the table data.
<td>	Defines a cell in a table.
<template> 5	Defines the fragments of HTML that should be hidden when the page is loaded, but can be cloned and inserted in the document by JavaScript.
<textarea>	Defines a multi-line text input control (text area).
<tfoot>	Groups a set of rows summarizing the columns of the table.
<th>	Defines a header cell in a table.
<thead>	Groups a set of rows that describes the column labels of a table.
<time> 5	Represents a time and/or date.
<title>	Defines a title for the document.
<tr>	Defines a row of cells in a table.
<track> 5	Defines text tracks for the media elements like <audio> or <video> .
<tt>	Obsolete Displays text in a teletype style.

<u>	Displays text with an underline.
	Defines an unordered list.
<var>	Defines a variable.
<video> 	Embeds video content in an HTML document.
<wbr> 	Represents a line break opportunity.

- New Elements in HTML5
- New Semantic/Structural Elements
- HTML5 offers new elements for better document structure:

Tag	Description
<u><article></u>	Defines an article in a document
<u><aside></u>	Defines content aside from the page content
<u><bdi></u>	Isolates a part of text that might be formatted in a different direction from other text outside it
<u><details></u>	Defines additional details that the user can view or hide
<u><dialog></u>	Defines a dialog box or window
<u><figcaption></u>	Defines a caption for a <figure> element
<u><figure></u>	Defines self-contained content
<u><footer></u>	Defines a footer for a document or section
<u><header></u>	Defines a header for a document or section
<u><main></u>	Defines the main content of a document
<u><mark></u>	Defines marked/highlighted text
<u><meter></u>	Defines a scalar measurement within a known range (a gauge)
<u><nav></u>	Defines navigation links

<u><progress></u>	Represents the progress of a task
<u><rp></u>	Defines what to show in browsers that do not support ruby annotations
<u><rt></u>	Defines an explanation/pronunciation of characters (for East Asian typography)
<u><ruby></u>	Defines a ruby annotation (for East Asian typography)
<u><section></u>	Defines a section in a document
<u><summary></u>	Defines a visible heading for a <details> element
<u><time></u>	Defines a date/time
<u><wbr></u>	Defines a possible line-break

Progress:

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

Downloading progress:

```
<progress value="22" max="100">
```

```
</progress>
```

<p>Note: The progress tag is not supported in Internet Explorer 9 and earlier versions.</p>

```
</body>
```

```
</html>
```

Output

Nav Tag:Navigation

```
<!DOCTYPE html>
<html>
<body>
<nav>
<a href="../html/index.html">HTML</a> |
<a href="../css/index.html">CSS</a> |
<a href="../js/index.html">JavaScript</a> |
<a href="../jquery/index.html">jQuery</a>
</nav>
<p><strong>Note:</strong> The nav tag is not supported in Internet Explorer 8 and
earlier versions.</p>
</body>
</html>
```

Output

- **HTML5 <figcaption> Tag**
- The <figcaption> element defines a caption or legend for a figure.
- Syntax
- The basic syntax of the <figcaption> tag is given with:

- HTML / XHTML: <figcaption> ... </figcaption>

```
<figure>
  
  <figcaption>NASA - Space Shuttle
  Discovery</figcaption>
</figure>
```

New Form Elements

Tag	Description
<u><datalist></u>	Specifies a list of pre-defined options for input controls
<u><output></u>	Defines the result of a calculation

```
<input list="browsers">  
  
<datalist id="browsers">  
  <option value="Internet  
Explorer">  
  <option value="Firefox">  
  <option value="Chrome">  
  <option value="Opera">  
  <option value="Safari">  
</datalist>
```

```
<form oninput="x.value=parseInt(a.v  
alue)+parseInt(b.value)">0  
  <input type="range" id="a" value=  
"50">100  
  
  +<input type="number" id="b" value=  
"50">  
  =<output name="x" for="a  
b"></output>  
</form>
```

- Datalist:
- The <datalist> element specifies a set of pre-defined options for an <input> element.
- It can be used to provide the quick choices for an input field like an "autocomplete" feature.
- It not only saves the time of a user but also reduces errors, because the user has less likelihood of making a spelling mistake.
- The list attribute of the input element is used to bind it together with a datalist element.

- **Syntax**
- **The basic syntax of the <datalist> tag is given with:**
- `<datalist> ... </datalist>`

- **Example:**

```
<p>Enter your favorite browser name:</p>
<input type="text" list="browsers">
<datalist id="browsers">
    <option value="Firefox">
    <option value="Chrome">
    <option value="Internet Explorer">
    <option value="Opera">
    <option value="Safari">
</datalist>
```

● New Input Types

New Input Types

- color
- date
- datetime
- datetime-local
- email
- month
- number
- range
- search
- tel
- time
- url
- week

New Input Attributes

- autocomplete
- autofocus
- form
- formaction
- formenctype
- formmethod
- formnovalidate
- formtarget
- height and width
- list
- min and max
- multiple
- pattern (regexp)
- placeholder
- required
- step

- Input Type Date
- The `<input type="date">` is used for input fields that should contain a date.
- `<form>`

Birthday:

```
<input type="date" name="bday">  
</form>
```

Example:

date output

- You can also use the min and max attributes to add restrictions to dates:
- <form>

Enter a date before 1980-01-01:

```
<input type="date" name="bday" max="1979-12-31">  
<br>
```

Enter a date after 2000-01-01:

```
<input type="date" name="bday" min="2000-01-02">  
<br>  
</form>
```

- Input Type Datetime-local
- The `<input type="datetime-local">` specifies a date and time input field, with no time zone.
- **<form>**

Birthday (date and time):

```
<input type="datetime-local" name="bdytime">  
</form>
```

Birthday (date and time): 

- Input Type Number
- The `<input type="number">` defines a numeric input field.
- `<form>`

Quantity (between 1 and 5):

```
<input type="number" name="quantity" min  
="1" max="5">  
</form>
```

Input Number

Output

Input Restrictions

Here is a list of some common input restrictions (some are new in HTML5):

Attribute	Description
disabled	Specifies that an input field should be disabled
max	Specifies the maximum value for an input field
maxlength	Specifies the maximum number of character for an input field
min	Specifies the minimum value for an input field
pattern	Specifies a regular expression to check the input value against
readonly	Specifies that an input field is read only (cannot be changed)
required	Specifies that an input field is required (must be filled out)
size	Specifies the width (in characters) of an input field
step	Specifies the legal number intervals for an input field
value	Specifies the default value for an input field

● HTML5 Graphics

Tag	Description
<u><canvas></u>	Draw graphics, on the fly, via scripting (usually JavaScript)
<u><svg></u>	Draw scalable vector graphics

HTML5 <canvas> Tag

- **Description**
- The <canvas> element defines a region in the document, which can be used to draw graphics on the fly via scripting (usually JavaScript).
- For example, it can be used to draw path and shapes, graphs or even perform animations.

● **Syntax**

- The basic syntax of the <canvas> tag is given with:
- HTML / XHTML: <canvas> ... </canvas>
- Example:

Canvas Output

```
<canvas id="myCanvas" width="300" height="200"></canvas>
<script type="text/javascript">
window.onload = function(){
    var canvas = document.getElementById("myCanvas");
    var context = canvas.getContext("2d");
    context.moveTo(50, 150);
    context.lineTo(250, 50);
    context.stroke();
}
</script>
```

- Tag-Specific Attributes
- The following table shows the attributes that are specific to the <canvas> tag.

Attribute	Value	Description
<code>width</code>	<i>pixels</i>	Sets the width of the canvas.
<code>height</code>	<i>pixels</i>	Sets the height of the canvas.

The HTML <svg> Element

- What is SVG?
- SVG stands for Scalable Vector Graphics
- SVG is used to define graphics for the Web
- SVG is a W3C recommendation
- The HTML <svg> Element
- The HTML <svg> element is a container for SVG graphics.
- SVG has several methods for drawing paths, boxes, circles, text, and graphic images.

```
<!DOCTYPE html>
<html>
<body>
<svg width="100" height="100">
  <circle cx="50" cy="50" r="40" stroke="green"
stroke-width="4" fill="yellow" />
</svg>
</body>
</html>
```

● New Media Elements

Tag	Description
<u><audio></u>	Defines sound content
<u><embed></u>	Defines a container for an external (non-HTML) application
<u><source></u>	Defines multiple media resources for media elements (<video> and <audio>)
<u><track></u>	Defines text tracks for media elements (<video> and <audio>)
<u><video></u>	Defines video or movie

```
<audio controls>
  <source src="horse.ogg" type="audio/ogg">
  <source src="horse.mp3" type="audio/mpeg">
  Your browser does not support the audio tag.
</audio>
<embed src="helloworld.swf">
<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>
```

HTML5 <audio> Tag

● Description

- The <audio> element is used to embed audio content in an HTML document without requiring any additional plug-in like Flash player.

● Syntax

- The basic syntax of the <audio> tag is given with:

- HTML / XHTML: <audio> ... </audio>

- The example below shows the <audio> tag in action.

```
<audio controls="controls" src="birds.mp3">
```

Your browser does not support the HTML5 Audio element.

```
</audio>
```

```
<audio controls="controls">  
  <source src="birds.mp3" type="audio/mpeg">  
  <source src="birds.ogg" type="audio/ogg">  
    Your browser does not support the HTML5 Audio  
    element.  
</audio>
```

Tag-Specific Attributes

Attribute	Value	Description
autoplay	autoplay	This Boolean attribute specifies that the audio will automatically start playing as soon as it can do so without stopping to finish loading the data.
controls	controls	If specified, the browsers will display controls to allow the user to control audio playback, such as play/pause, volume, etc.
loop	loop	This Boolean attribute specifies that the audio will automatically start over again, upon reaching the end.
muted	muted	This Boolean attribute specifies whether the audio will be initially silenced. The default value is false, meaning that the audio will be played.
preload	auto metadata none	Provides a hint to the browser about whether to download of the audio itself or its metadata. The autoplay attribute can override this attribute, because if you want to automatically play a audio, the browser will obviously need to download it.
src	URL	Specifies the location of the audio file. Alternatively, you can use the preferred <code><source></code> tag as it allows for multiple options.

- HTML5 <embed> Tag
- **Description**
- The <embed> element can be used to embed external application, typically multimedia content like audio or video into an HTML document.
- Syntax
- The basic syntax of the <embed> tag is given with:
 - <embed>XHTML: <embed />
 - <embed src="blur.swf" width="400px" height="200px">

- **Tag-Specific Attributes**
- The following table shows the attributes that are specific to the <embed> tag.

Attribute	Value	Description
width	<i>pixels</i>	Sets the width of the embedded content.
height	<i>pixels</i>	Sets the height of the embedded content.
src	<i>URL</i>	Specifies the URL of the resource being embedded.
type	<i>content-type</i>	Specifies the type of the resource.

HTML5 <article> Tag

● Description

- The <article> element represents a section of content that forms an independent part of a document, such as a blog post, article, or other self-contained unit of information, that may be linked to or included in some other content body.

● Syntax

- The basic syntax of the <article> tag is given with:
- HTML / XHTML: <article> ... </article>

<article>

 <h1>Introduction to HTML</h1>

 <p>HTML is a markup language that is used for creating
 web pages.</p>

</article>

- HTML5 <aside> Tag
- **Description**
- The <aside> element represents a section of the web page that encloses content which is tangentially related to the content around it.
- **Syntax**
- The basic syntax of the <aside> tag is given with:
- HTML / XHTML: <aside> ... </aside>
- The example below shows the <aside> tag in action.

```
<aside>
  <h1>Apollo 13 Facts</h1>
  <p>Apollo 13 was the seventh manned mission in the American
  Apollo space program and the third intended to land on the
  Moon.</p>
</aside>
```

HTML5 <audio> Tag

● Description

- The <audio> element is used to embed audio content in an HTML document without requiring any additional plug-in like Flash player.

● Syntax

- The basic syntax of the <audio> tag is given with:

- HTML / XHTML: <audio> ... </audio>

- The example below shows the <audio> tag in action.

```
<audio controls="controls" src="birds.mp3">
```

Your browser does not support the HTML5 Audio element.

```
</audio>
```

```
<audio controls="controls">  
  <source src="birds.mp3" type="audio/mpeg">  
  <source src="birds.ogg" type="audio/ogg">  
    Your browser does not support the HTML5 Audio  
    element.  
</audio>
```

Tag-Specific Attributes

Attribute	Value	Description
autoplay	autoplay	This Boolean attribute specifies that the audio will automatically start playing as soon as it can do so without stopping to finish loading the data.
controls	controls	If specified, the browsers will display controls to allow the user to control audio playback, such as play/pause, volume, etc.
loop	loop	This Boolean attribute specifies that the audio will automatically start over again, upon reaching the end.
muted	muted	This Boolean attribute specifies whether the audio will be initially silenced. The default value is false, meaning that the audio will be played.
preload	auto metadata none	Provides a hint to the browser about whether to download of the audio itself or its metadata. The autoplay attribute can override this attribute, because if you want to automatically play a audio, the browser will obviously need to download it.
src	URL	Specifies the location of the audio file. Alternatively, you can use the preferred <code><source></code> tag as it allows for multiple options.

● **HTML5 <data> Tag:**

● **Description:**

- The <data> element links a piece of content with a machine-readable translation.
- If the content is time or date related, the more specific <time> element should be used instead.

● **Syntax**

- The basic syntax of the <data> tag is given with:

- <data value="machine-readable"> ... </data>

● **Example:**

```
<p>New Products</p>
```

```
<ul>
```

```
    <li><data value="204">Green Tea</data></li>
```

```
    <li><data value="205">Tomato Ketchup</data></li>
```

```
    <li><data value="206">Roasted Coffee Beans</data></li>
```

```
</ul>
```

● Tag-Specific Attributes

- The following table shows the attributes that are specific to the <data> tag.
- Attribute Value Description
- Value: machine-readable Specifies the machine-readable translation of the content of the element.

HTML5 <details> Tag

● Description

- The <details> element represents a control from which the user can obtain additional information on-demand.
- It can be used to create an interactive widget that the user can show or hide (like expand and collapse) to retrieve the additional information or controls.
- Any sort of element can be placed inside the details element.

● Syntax

- The basic syntax of the <details> tag is given with:
- <details> ... </details>

<details>

 <summary>What is HTML?</summary>

 <p>HTML is a markup language for describing the
structure of web pages.</p>

</details>

- HTML5 <dialog> Tag
- **Description**
- The <dialog> element defines a dialog box or other interactive component on a web page that a user can interact with to perform a task, such as a dismissible alert.
- Syntax
- The basic syntax of the <dialog> tag is given with:

- <dialog> ... </dialog>

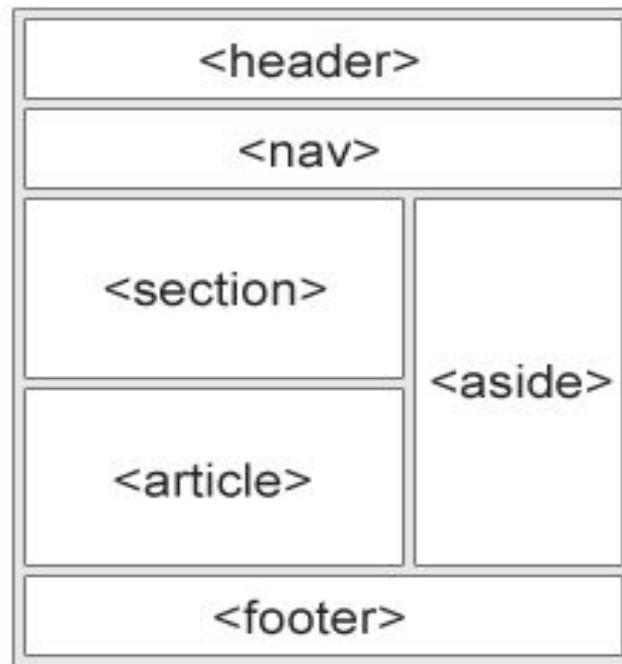
```
<dialog open>
  <p>Are you sure you want to remove this item?</p>
  <button type="button">Yes</button>
  <button type="button">No</button>
</dialog>
```

Semantic Tag:

- **What are Semantic Elements?**
- **A semantic element clearly describes its meaning to both the browser and the developer.**
- **Examples of non-semantic elements: <div> and - Tells nothing about its content.**
- **Examples of semantic elements: <form>, <table>, and <article> - Clearly defines its content.**

- Semantic Elements in HTML
- Many web sites contain HTML code like: <div id="nav"> <div class="header"> <div id="footer"> to indicate navigation, header, and footer.
- In HTML there are some semantic elements that can be used to define different parts of a web page:

- <article>
- <aside>
- <details>
- <figcaption>
- <figure>
- <footer>
- <header>
- <main>
- <mark>
- <nav>
- <section>
- <summary>
- <time>



Tag	Description
<u><article></u>	Defines independent, self-contained content
<u><aside></u>	Defines content aside from the page content
<u><details></u>	Defines additional details that the user can view or hide
<u><figcaption></u>	Defines a caption for a <figure> element
<u><figure></u>	Specifies self-contained content, like illustrations, diagrams, photos, code listings, etc.
<u><footer></u>	Defines a footer for a document or section
<u><header></u>	Specifies a header for a document or section
<u><main></u>	Specifies the main content of a document
<u><mark></u>	Defines marked/highlighted text
<u><nav></u>	Defines navigation links
<u><section></u>	Defines a section in a document
<u><summary></u>	Defines a visible heading for a <details> element
<u><time></u>	Defines a date/time

● Basic HTML page template, written in non-semantic HTML

```
<html>
  <head>
    <title>Example</title>
  </head>
  <body>
    <div id="header">
      Here goes logo, navigation, etc.
    </div>
    <div id="main-content">
      A place for website's main content
    </div>
    <div id="footer">
      Footer information, links, etc.
    </div>
  </body>
</html>
```

- **Example of semantic HTML shown below:**

```
<html>
  <head>
    <title>Example</title>
  </head>
  <body>
    <header>
      Here goes logo, navigation, etc.
    </header>
    <main>
      A place for website's main content
    </main>
    <footer>
      Footer information, links, etc.
    </footer>
  </body>
</html>
```

- **The main difference: we have replaced div tags with 3 new tags:**

- header, main, and footer. header, main, and footer tags are semantic because they are used to represent different sections on an HTML page.
- These are more descriptive than div tags which make partitioning webpages into tangible sections difficult.

- **Navigation**

- In HTML5, there is the nav tag, div, to wrap links that make a navigation menu.
- For instance, the navigation menu can be inserted within the header section:

```
<header>
  
  <nav>
    <a href="index.html">Home</a>
    <a href="services.html">Services</a>
    <a href="contact.html">Contact</a>
    <a href="about.html">About Us</a>
  </nav>
</header>
```

but, it also can be added after the `_header_` section:

```
```HTML
<header>

</header>
<nav>
 Home
 Services
 Contact
 About Us
</nav>
```

## Main Content

- To add some content into the main section, we can use new HTML5 tags, such as article and section.
- These tags simplify the structure of main, making it look like:

```
<main>
 <article>
 <h1>JavaScript Basics</h1>
 <p>JavaScript is a rich and expressive language...</p>
 <section>
 <h2>Syntax Basics</h2>
 <p>Understanding statements, variable naming, whitespace...</p>
 </section>
 <section>
 <h2>Operators</h2>
 <p>Operators allow you to manipulate values...</p>
 </section>
 <section>
 <h2>Conditional Code</h2>
 <p>Sometimes you only want to run a block of code under certain conditions...</p>
 </section>
 </article>
</main>
```

- The article tag is used for wrapping an autonomous content on a page.
- A content is autonomous if it can be removed from the page and put on some another page.
- The article tag can contain several section tags inside it, like in our example.
- An article is actually an autonomous section.
- The section tag is similar to the div tag, but it is more meaningful since it wraps logical groups of related content (e.g. a chapter of an article).
- The section tag can be also used to wrap the article itself, but we have the article tag for that purpose.

## Additional Content

### Aside section

An additional content, unimportant for understanding an article, but related to the article, can be inserted into the aside section.

```
<main>
```

```
 <article>
```

```
 <h1>JavaScript Basics</h1>
```

```
 <p>JavaScript is a rich and expressive language...</p>
```

```
 <section>
```

```
 <h2>Syntax Basics</h2>
```

```
 <p>Understanding statements, variable naming, whitespace...</p>
```

```
 </section>
```

```
 <section>
```

```
 <h2>Operators</h2>
```

```
 <p>Operators allow you to manipulate values...</p>
```

```
 </section>
```

```
 <section>
```

```
 <h2>Conditional Code</h2>
```

```
 <p>Sometimes you only want to run a block of code under certain conditions...</p>
```

```
 </section>
```

```
 <aside>
```

```
 <p>Viewed by 1503 people</p>
```

```
 <p>Author: John Smith</p>
```

```
 </aside>
```

```
 </article>
```

```
</main>
```

- HTML <section> Element
- The <section> element defines a section in a document.
- According to W3C's HTML documentation: "A section is a thematic grouping of content, typically with a heading."
- Examples of where a <section> element can be used:
  - Chapters
  - Introduction
  - News items
  - Contact information

- <https://htmlreference.io/forms/>
- [https://developer.mozilla.org/en-US/docs/Learn/Forms/HTML5\\_input\\_types](https://developer.mozilla.org/en-US/docs/Learn/Forms/HTML5_input_types)