



Learn HTML From Scratch

...with iFarouq Tech

Learn HTML from scratch

Fastest and easiest way to learn HTML code and start to build skills on web development.

iFarouq Tech

This document is not for sale, it was made for newbie student, those that just started to learn web design.

This was version published on 28-12-2020

Support us

Please support iFarouq Tech by spreading the word about this document on different platforms and Follow us on our accounts:

Subscribe to our channel @YouTube: [iFarouq Tech](#)

Like our page @Facebook: [iFarouq Tech](#)

Follow our page @Twitter: [iFarouq Tech](#)

Follow our page @Instagram: [iFarouq Tech](#)

You can follow us on any of the account above. We highly recommend you to subscribe to our YouTube Channel to learn more about Web Development in easiest way, also you will be notifying on every update on our new tutorials.

Thank You.!

Table Contents

CHAPTER ONE

Introduction to HTML	11
What is HTML?	11
HTML Tags	12
Web Browsers	12
HTML Page Structures	13
HTML <!DOCTYPE> Declaration	13
Common Declarations	14
HTML Versions	14

CHAPTER TWO: Editors

Write HTML Using Notepad or TextEdit	15
--	----

CHAPTER THREE: HTML Basics

HTML Documents	18
HTML Headings	18
HTML Paragraphs	19
HTML Links	19
HTML Images	20

CHAPTER FOUR: Elements

HTML Elements	21
Nested HTML Elements	21
Don't forget the End Tag	22
Empty HTML Elements	23
HTML Tip: Use Lower Tags	23

CHAPTER FIVE: ATTRIBUTES

HTML Attributes	24
Lang Attribute	24
The title Attribute	24
The href Attribute	25
Size Attribute	26
The alt Attributes	27
Single and Double Quotes	28
Chapter Summary	29

CHAPTER SIX: HEADINGS

HTML Headings	30
Headings are Important	30
HTML Horizontal Rules	31
HTML <head> Element	31
HTML <title> Element	32
HTML <meta> Element	32
More Meta Elements	32
HTML tip-how to view HTML source code	32

CHAPTER SEVEN: PARAGRAPHS

HTML Paragraphs	34
HTML Display	34
Don't forget the End Tag	35
HTML Line Breaks	36

The Poem Problem	36
HTML <pre> Elements	37
CHAPTER EIGHT: STYLES	
HTML Styling	38
HTML Style Attribute	39
HTML Text Color	39
HTML Text Fonts	39
HTML Text Size	40
HTML Text Alignment	40
Chapter Summary	41
CHAPTER NINE: TEXT FORMATTING ELEMTS	
HTML Bold and Strong Formatting	42
HTML <i>Italic</i> and <i>Emphasized</i> Formatting	43
HTML <small>small</small> Formatting	44
HTML Marked Formatting	44
HTML Deleted Formatting	45
HTML <ins>Inserted</ins> Formatting	45
HTML _{Subscript} Formatting	46
HTML ^{Superscript} Formatting	46
CHAPTER TEN: QOUTATION AND CITATION ELEMENTS	
HTML <q> for Short Quotations	47
HTML <blockquote> for Long Quotations	47
HTML <abbr> for Abbreviations	48
HTML <dfn> for Definitions	49
HTML <address> for Contact Information	50
HTML <cite> for Work Title	51
HTML <bdo> for Bi-Directional Override	52
CHAPTER ELEVEN: COMMENTS	
HTML Comment Tags	53
Condition Comments	54
Software Program Tags	54
CHAPTER TWELVE: STYLE CSS	
CSS = Style and Colors	55
Style HTML with CSS	56
CSS Syntax	56
Inline Styling (Inline CSS)	56
Internal Styling (Internal CSS)	57
External Styling (External CSS)	58
CSS Fonts	58
The CSS Box Model	59
The id Attribute	61
The Class Attribute	62
The Deprecated Tags and Attributes in HTML5	63
Chapter Summary	64
CHAPTER THIRTEEN: LINKS	
HTML Links	65
HTML Links-Syntax	65
HTML Links-Colors and Icons	67
HTML Links-Target Attributes	68
HTML Links-Image as Link	70

HTML Links-The id Attribute	70
Chapter Summary	73
CHAPTER FOURTEEN: IMAGES	
HTML Images Syntax	74
The alt Attribute	74
HTML Screen Readers	75
Width and Height or Style?	76
Images in another Folder	77
Images in another Server	77
Animated Images	78
Using an Images as a Link	78
Image Maps	79
Image Floating	80
Chapter Summary	80
CHAPTER FIFTEEN: TABLES	
Definition of HTML Tables	82
An HTML Table with a border attribute.....	83
An HTML Table with Collapsed borders	85
An HTML Table with cell Padding	85
HTML Table Headings	86
An HTML Table with Border Spacing	88
Table Cells that span many Columns	89
Table Cells that span many Rows	90
An HTML Table with Caption	92
Different Styles for Different Tables	93
Chapter Summary	98
CHAPTER SIXTEEN: LIST	
Unordered HTML List	100
Ordered HTML List	103
HTML Description List	105
Nested HTML List	106
Horizontal List	106
Chapter Summary	107
CHAPTER SEVENTEEN: BLOCK ELEMENTS	
HTML Block Elements and Inline Elements	109
The HTML <div> Element	110
The HTML Element	110
CHAPTER EIGHTEEN: CLASSES	
Classes Block Elements	113
Classes Inline Elements	115
CHAPTER NINETEEN: LAYOUT	
HTML Layout using <div> Elements	116
Website Layout using HTML5	119
HTML Layout using Tables	121
CHAPTER TWENTY: RESPONSIVE WEB DESIGN	
What is responsive web design?	124
Create your responsive Design	124
Using Bootstrap	125
CHAPTER TWENTY-ONE: iFRAME	
Definition	127

iFrame Syntax	127
Iframe set-Height and Width	127
Using iframe as a Target for a link	127
CHAPTER TWENTY-TWO: COLOR NAMES	
140 Color names are Supported by all Browsers	129
Sorted by color names	129
CHAPTER TWENTY-THREE: COLOR VALUES	
HTML Color Values	131
Color Supported by HEX Value	132
CHAPTER TWENTY-FOUR: COLOR SHADES	
Shade grey	135
16 Million Different Colors	135
In the stone age	137
CHAPTER TWENTY-FIVE: HTML SCRIPT	
Script Definition	139
Noscript Definition	139
The HTML <script> Tag	139
The HTML <noscript> Tag	140
A Taste of JavaScript	141
CHAPTER TWENTY-SIX: HTML HEAD	
The HTML <head> Element	144
Omitting <head> and <body>	144
Omitting <head>	145
The HTML <title> Element	145
The HTML <style> Element	146
The HTML <link> Element	147
The HTML <meta> Element	148
The HTML <script> Element	149
The HTML <base> Element	150
CHAPTER TWENTY-SEVEN: HTML ENTITIES	
HTML Entities	152
Non-Breaking Space	152
Combining the Diacritical marks	153
CHAPTER TWENTY-EIGHT: HTML SYMBOLS	
HTML Symbols Entities	155
Some Mathematical Symbols Supported by HTML	156
Some Greek Letters Supported by HTML	158
Some Other Entities Supported by HTML	160
CHAPTER TWENTY-NINE: HTML ENCODING (CHARACTER SET)	
What is Character Encoding?	168
The HTML Charset Attribute	168
Differences between Character sets	169
ASCII Character Set	171
The ANSI Character Set (windows-1252)	171
The ISO-8859-1 Character Set	172
The UTF-8 Character Set	172
CHAPTER THIRTY: HTML UNIFORM RESOURCE LOCATOR	
URL - Uniform Resource Locator	173
Common URL Schemes	174
URL Encoding	174

ASCII Encoding Examples	174
CHAPTER THIRTY-ONE: HTML AND XHTML	
What is XHTML?	176
Why XHTML?	176
The Most Important Differences from HTML	177
<!DOCTYPE ...> is mandatory	177
XHTML Elements Must Be Properly Nested	178
XHTML Elements Must Always Be Closed	178
Empty Elements Must Also Be Closed	179
XHTML Elements Must Be In Lower Case	179
XHTML Attribute Names Must Be In Lower Case	179
Attribute Values Must Be Quoted	180
Attribute Minimization Is Forbidden	180
How to Convert from HTML to XHTML	180
CHAPTER THIRTY-TWO: HTML FORM	
HTML <form> Elements	182
The <input> Elements	182
Text input	183
Radio Button	183
Submit Button	184
The Action Attribute	185
The method Attribute	185
When to use GET?	186
When to use POST?	186
The name attribute	186
Grouping form data with <fieldset>	187
HTML form attribute	188
CHAPTER THIRTY-THREE: FORM ELEMENTS	
The <input> Elements	190
The <input> Element (Drop-down List)	190
The <textarea> Element	192
The <button> Element	192
HTML5 Form Element	193
CHAPTER THIRTY-FOUR: HTML INPUT TYPES	
Input type: text	198
Input type: password	199
Input type: submit	199
Input type: radio	200
Input type: checkbox	201
Input type: button	202
HTML5 input Types	202
CHAPTER THIRTY-FIVE: HTML INPUT ATTRIBUTES	
The value attribute	217
The readonly attribute	217
The disabled attribute	218
The size attribute	218
The maxlength attribute	219
HTML5 Attributes	220
CHAPTER THIRTY-SIX: HTML MULTIMEDIA	
What is multimedia?	237

Browser support	237
Multimedia Formats	237
Sound Formats	238
CHAPTER THIRTY-SEVEN: HTML VIDEO	
Playing Videos in HTML	240
The HTML <video> Element.....	241
The HTML <video> autoplay	242
HTML Video - Methods, Properties, and Events	243
CHAPTER THIRTY-EIGHT: HTML5 AUDIO	
Audio on web	246
The HTML <audio> Element	246
HTML audio-how it works	247
HTML Audio - Methods, Properties, and Events	248
CHAPTER FOURTY: NINE PLUG-INS	
HTML helpers (plug-ins)	249
The <object> Element	249
The <embed> Element	250
CHAPTER FOURTY: HTML YOUTUBE VIDEOS	
Struggling with Video Formats	253
Playing a YouTube Video in HTML	253

CHAPTER ONE

1.0 INTRODUCTION TO HTML

1.1 What is HTML?

HTML is a **markup** language for **describing** web documents (web pages).

- HTML stands for **Hyper Text Markup Language**
- A markup language is a set of **markup tags**
- HTML documents are described by **HTML tags**
- Each HTML tag **describes** different document content

1.2 HTML Example

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

Example Explained

- The **DOCTYPE** declaration defines the document type to be HTML
- The text between **<html>** and **</html>** describes an HTML document
- The text between **<head>** and **</head>** provides information about the document
- The text between **<title>** and **</title>** provides a title for the document
- The text between **<body>** and **</body>** describes the visible page content
- The text between **<h1>** and **</h1>** describes a heading

- The text between `<p>` and `</p>` describes paragraph

Using this description, a web browser can display a document with a heading and a paragraph.

1.3 HTML Tags

HTML tags are **keywords** (tag names) surrounded by **angle brackets**:

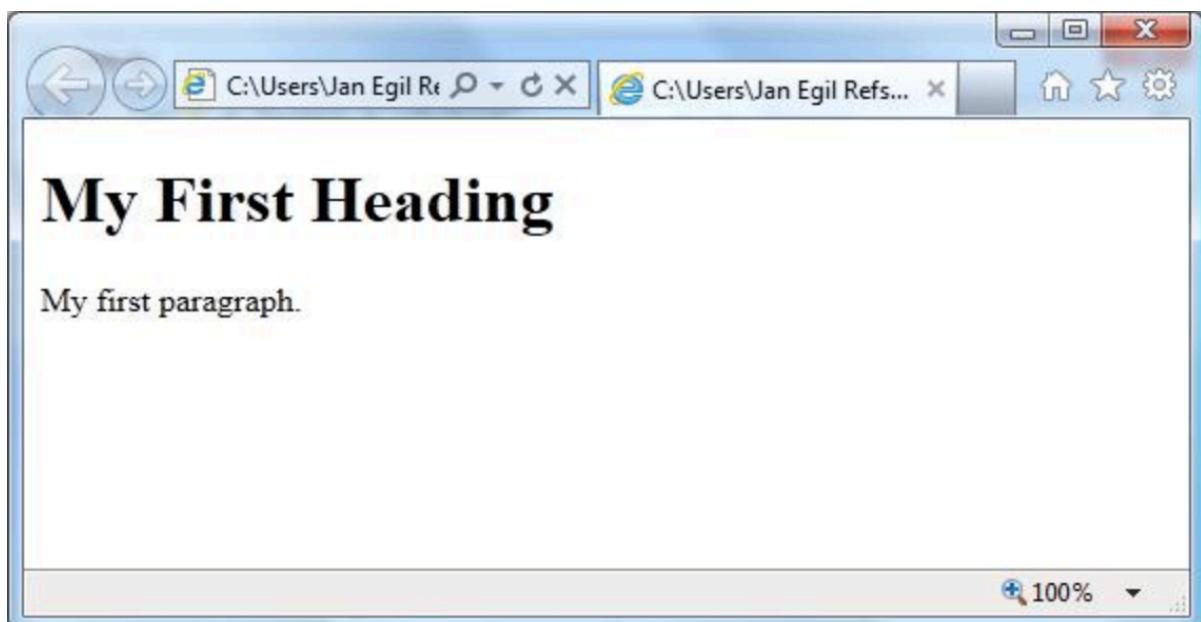
```
<tagname>content</tagname>
```

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

The start tag is often called the **opening tag**. The end tag is often called the **closing tag**.

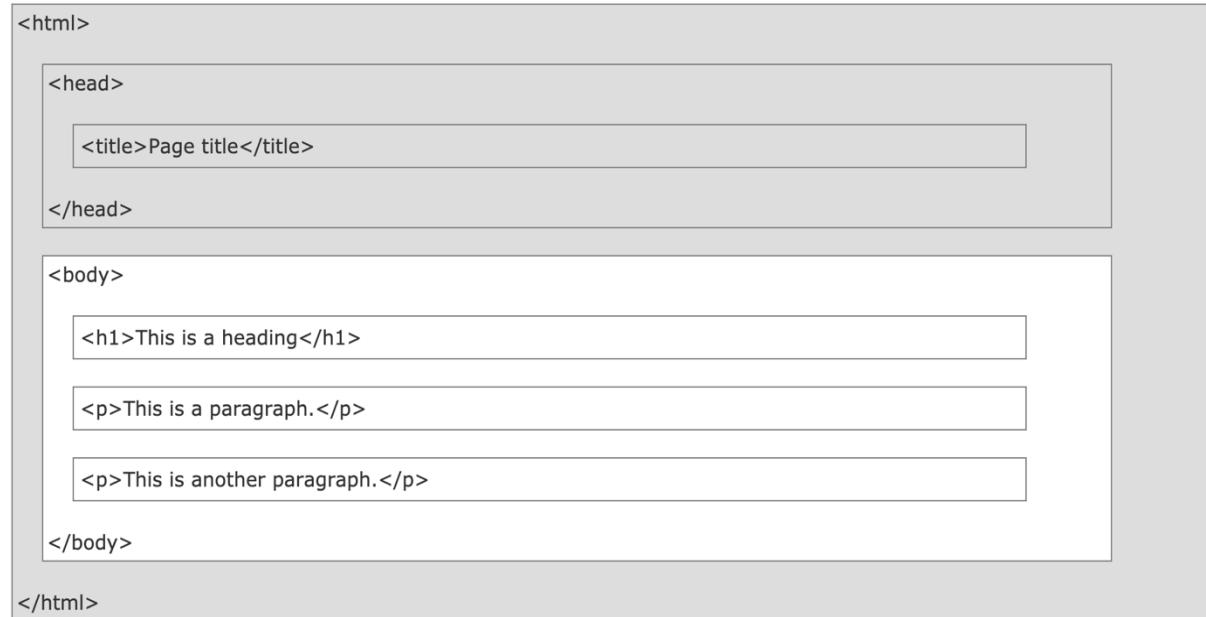
1.4 Web Browsers

The purpose of a web browser (Chrome, IE, Firefox, Safari) is to read HTML documents and display them. The browser does not display the HTML tags, but uses them to determine how to display the document:



1.5 HTML Page Structure

Below is a visualization of an HTML page structure:



1.6 The <!DOCTYPE> Declaration

The <!DOCTYPE> declaration helps the browser to display a web page correctly. There are different document types on the web. To display a document correctly, the browser must know both type and version. The doctype declaration is not case sensitive. All cases are acceptable:

```
<!DOCTYPE html>
<!DOCTYPE HTML>
<!doctype html>
<!Doctype Html>
```

1.7 Common Declarations

HTML5

```
<!DOCTYPE html>
```

HTML 4.01

```
<!DOCTYPE HTML PUBLIC "-//iFarouq_Tech//DTD HTML 4.01 Transitional//EN"  
"http://www.example.org/TR/html4/loose.dtd">
```

XHTML 1.0

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

All tutorials and examples at iFarouq Tech use HTML5.

1.8 HTML Versions

Since the early days of the web, there have been many versions of HTML:

Version	Year
HTML	1991
HTML 2.0	1995
HTML 3.2	1997
HTML 4.01	1999
XHTML	2000
HTML5	2012

CHAPTER TWO

2.0 HTML EDITORS

2.1 Write HTML Using Notepad or TextEdit

HTML can be edited by using a professional HTML editor like:

- Adobe Dreamweaver
- Microsoft Expression Web
- Coffee Cup HTML Editor

However, for learning HTML we recommend a text editor like Notepad (PC) or TextEdit (Mac). We believe using a simple text editor is a good way to learn HTML.

Follow the 4 steps below to create your first web page with Notepad.

Step 1: Open Notepad

To open Notepad in Windows 7 or earlier:

Click **Start** (bottom left on your screen). Click **All Programs**. Click **Accessories**.
Click **Notepad**.

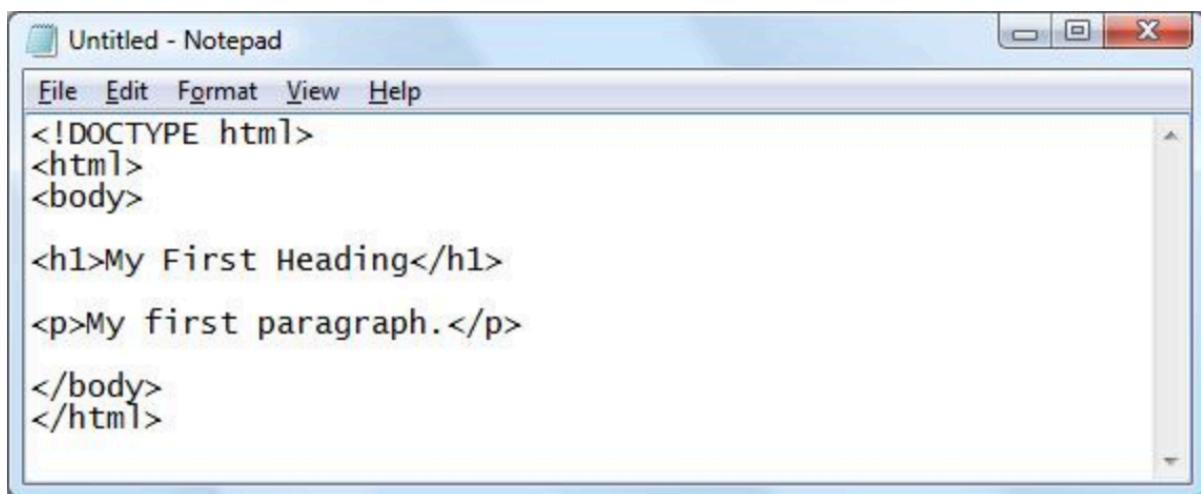
To open Notepad in Windows 8 or later: Open the **Start Screen** (the window symbol at the bottom left on your screen). Type **Notepad**.

Step 2: Write Some HTML

Write or copy some HTML into Notepad.

Example

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



Step 3: Save the HTML Page

Save the file on your computer. Select **File > Save as** in the Notepad menu. You can use either .htm or .html as file extension. There is no difference, it is up to you.

Step 4: View HTML Page in Your Browser

Double-click your saved HTML file, and the result will look much like this:



CHAPTER THREE

3.0 HTML BASICS

Don't worry if these examples use tags you have not learned. You will learn them in the next chapters.

3.1 HTML Documents

All HTML documents must start with a 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>`.

Example

```
<!DOCTYPE html>

<html>

<body>

<h1>My First Heading</h1>

<p>My first paragraph.</p>

</body>

</html>
```

3.2 HTML Headings

HTML headings are defined with the `<h1>` to `<h6>` tags:

Example

```
<!DOCTYPE html>

<html>

<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>  
</html>
```

3.3 HTML Paragraphs

HTML paragraphs are defined with the `<p>` tag:

Example

```
<!DOCTYPE html>  
<html>  
  <body>  
    <p>This is a paragraph.</p>  
    <p>This is a paragraph.</p>  
    <p>This is a paragraph.</p>  
  </body>  
</html>
```

3.4 HTML Links

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

Example

```
<!DOCTYPE html>  
<html>  
  <body>
```

```
<a href="../index.html">This is a link</a>
</body>
</html>
```

The link address is specified in the **href attribute**.

Attributes are used to provide additional information about HTML elements.

3.5 HTML Images

HTML images are defined with the **** tag.

The source file (**src**), alternative text (**alt**), and size (**width** and **height**) are provided as **attributes**:

Example

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

CHAPTER FOUR

4.0 HTML ELEMENTS

HTML **documents** are made up by HTML **elements**.

4.1 HTML Elements

HTML elements are written with a **start** tag, with an **end** tag, with the **content** in between:

```
<tagname>content</tagname>
```

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

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

Start tag	Element content	End tag
<h1>	My First Heading	</h1>
<p>	My first paragraph.	</p>

4.2 Nested HTML Elements

HTML elements can be nested (elements can contain elements). All HTML documents consist of nested HTML elements.

This example contains 4 HTML elements:

Example

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

HTML Example Explained

The `<html>` element defines the **whole document**. It has a **start** tag `<html>` and an **end** tag `</html>`.

The element **content** is another HTML element (the `<body>` element).

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

The `<body>` element defines the **document body**. It has a **start** tag `<body>` and an **end** tag `</body>`.

The element **content** is two other HTML elements (`<h1>` and `<p>`).

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

The `<h1>` element defines a **heading**. It has a **start** tag `<h1>` and an **end** tag `</h1>`. The element **content** is: My First Heading.

```
<h1>My First Heading</h1>
```

The `<p>` element defines a **paragraph**. It has a **start** tag `<p>` and an **end** tag `</p>`. The element **content** is: My first paragraph.

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

4.3 Don't Forget the End Tag

Some HTML elements will display correctly, even if you forget the end tag:

```
<!DOCTYPE html>
<html>
  <body>
    <p>This is a paragraph.
    <p>This is a paragraph.
  </body>
</html>
```

The example above works in all browsers, because the closing tag is considered optional. Never rely on this. It might produce unexpected results and/or errors if you forget the end tag.

4.4 Empty HTML Elements

HTML elements with no content are called empty elements.

`
` is an empty element without a closing tag (the `
` tag defines a line break).

Empty element can be "closed" in the opening tag like this: `
`.

HTML5 does not require empty elements to be closed. But if you need stricter validation, and make your document readable by XML parsers, please close all HTML elements.

4.5 HTML Tip: Use Lowercase Tags

HTML tags are not case sensitive: `<P>` means the same as `<p>`. The HTML5 standard does not require lowercase tags, but iFarouq Tech **recommends** lowercase in HTML4, and **demands** lowercase for stricter document types like XHTML.

CHAPTER FIVE

5.0 HTML ATTRIBUTES

Attributes provide additional information about HTML elements.

5.1 HTML Attributes

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

5.2 The lang Attribute

The document language can be declared in the `<html>` tag. The language is declared in the **lang** attribute. Declaring a language is important for accessibility applications (screen readers) and search engines:

Example

```
<!DOCTYPE html>
<html lang="en-US">
<body>
<h1>My First Heading</h1>
<p>My first paragraph.</p>
</body>
</html>
```

The first two letters specify the language (en). If there is a dialect, use two more letters (US).

5.3 The title Attribute

HTML paragraphs are defined with the `<p>` tag.

In this example, the `<p>` element has a **title** attribute. The value of the attribute is "About iFarouq Tech":

Example

```
<!DOCTYPE html>
<html>
  <body>
    <h1>About iFarouq Tech</h1>
    <p title = "About iFarouq Tech">
      iFarouq Tech is a web developer's site.
      It provides tutorials and references covering
      many aspects of web programming,
      including HTML, CSS, JavaScript, XML, SQL, PHP, ASP, etc.
    </p>
    <p><b>
      If you move the mouse over the paragraph above, the title will
      display as a tooltip.
    </b></p>
  </body>
</html>
```

When you move the mouse over the element, the title will be displayed as a tooltip.

5.4 The href Attribute

HTML links are defined with the `<a>` tag. The link address is specified in the **href** attribute:

Example

```
<!DOCTYPE html>
<html>
  <body>
    <a href="../index.html">This is a link</a>
  </body>
</html>
```

You will learn more about links and the `<a>` tag later in this tutorial.

5.5 Size Attributes

HTML images are defined with the `` tag. The filename of the source (`src`), and the size of the image (`width` and `height`) are all provided as **attributes**:

Example

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

The image size is specified in pixels: `width="104"` means 104 screen pixels wide.

You will learn more about images and the `` tag later in this tutorial.

5.6 The alt Attribute

The `alt` attribute specifies an alternative text to be used, when an HTML element cannot be displayed. The value of the attribute can be read by "screen readers". This way, someone "listening" to the webpage, i.e. a blind person, can "hear" the element.

Example

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

We Suggest: Always Use Lowercase Attributes

The HTML5 standard does not require lower case attribute names. The title attribute can be written with upper or lower case like **Title** and/or **TITLE**.

iFarouq Tech **recommends** lowercase in HTML4, and **demands** lowercase for stricter document types like XHTML.

Lower case is the most common. Lower case is easier to type.

At iFarouq Tech we always use lower case attribute names.

We Suggest: Always Quote Attribute Values

The HTML5 standard does not require quotes around attribute values.

The **href** attribute, demonstrated above, can be written as:

Example

```
<!DOCTYPE html>
<html>
  <body>
    <a href=../index.html>This is a link</a>
  </body>
</html>
```

iFarouq Tech **recommends** quotes in HTML4, and **demands** quotes for stricter document types like XHTML.

Sometimes it is **necessary** to use quotes. This will not display correctly, because it contains a space:

Example

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

```

<body>
    <h1>About iFarouq Tech</h1>
    <p title=About iFarouq Tech>
        You cannot omit quotes around an attribute value
        if the value contains spaces.
    </p>
    <p><b>
        If you move the mouse over the paragraph above,
        your browser will only display the first word from the title.
    </b></p>
</body>
</html>

```

Using quotes are the most common. Omitting quotes can produce errors.

At iFarouq Tech we always use quotes around attribute values.

5.7 Single or Double Quotes?

Double style quotes are the most common in HTML, but single style can also be used.

In some situations, when the attribute value itself contains double quotes, it is necessary to use single quotes:

Example

```
<p title='John "ShotGun" Nelson'>
```

Or vice versa:

Example

```
<p title="John 'ShotGun' Nelson">
```

Chapter Summary

- All HTML elements can have **attributes**
- The HTML **title** attribute provides additional "tool-tip" information
- The HTML **href** attribute provides address information for links
- The HTML **width** and **height** attributes provide size information for images
- The HTML **alt** attribute provides text for screen readers
- At iFarouq Tech we always use **lowercase** HTML attribute names
- At iFarouq Tech we always **quote** attributes with double quotes

HTML Attributes

Below is an alphabetical list of some attributes often used in HTML:

Attribute	Description
alt	Specifies an alternative text for an image
disabled	Specifies that an input element should be disabled
href	Specifies the URL (web address) for a link
id	Specifies a unique id for an element
src	Specifies the URL (web address) for an image
style	Specifies an inline CSS style for an element
title	Specifies extra information about an element (displayed as a tool tip)
value	Specifies the value (text content) for an input element.

A complete list, of all legal attributes for each HTML element, is listed in our: HTML Tag Reference.

CHAPTER SIX

6.0 HTML HEADINGS

Headings are important in HTML documents.

6.1 HTML Headings

Headings are defined with the `<h1>` to `<h6>` tags. `<h1>` defines the most important heading.

`<h6>` defines the least important heading.

Example

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

Note: Browsers automatically add some empty space (a margin) before and after each heading.

6.2 Headings Are Important

Use HTML headings for headings only. Don't use headings to make text **BIG** or **bold**. Search engines use your headings to index the structure and content of your web pages. Users skim your pages by its headings. It is important to use headings to show the document structure. h1 headings should be main headings, followed by h2 headings, then the less important h3, and so on.

6.3 HTML Horizontal Rules

The **<hr>** tag creates a horizontal line in an HTML page. The hr element can be used to separate content:

Example

```
<!DOCTYPE html>
<html>
  <body>
    <p>The hr tag defines a horizontal rule:</p>
    <hr>
    <p>This is a paragraph.</p>
    <hr>
    <p>This is a paragraph.</p>
    <hr>
    <p>This is a paragraph.</p>
  </body>
</html>
```

6.4 The HTML **<head>** Element

The HTML **<head>** element has nothing to do with HTML headings. The HTML **<head>** element contains **meta data**. Meta data are not displayed. The HTML **<head>** element is placed between the **<html>** tag and the **<body>** tag:

Example

```
<!DOCTYPE html>
<html>

  <head>
    <title>My First HTML</title>
    <meta charset="UTF-8">
  </head>
```

```
<body>
    <p>The HTML head element contains meta data.</p>
    <p>Meta data is data about the HTML document.</p>
</body>
</html>
```

Meta data means data **about** data. HTML meta data is data **about** the HTML document.

6.5 The HTML `<title>` Element

The HTML `<title>` element is meta data. It defines the HTML document's title. The title will not be displayed in the document, but might be displayed in the browser tab.

6.6 The HTML `<meta>` Element

The HTML `<meta>` element is also meta data. It can be used to define the character set, and other information about the HTML document.

6.7 More Meta Elements

In the chapter about HTML styles you discover more meta elements:

The HTML `<style>` element is used to define internal CSS style sheets.

The HTML `<link>` element is used to define external CSS style sheets.

6.8 HTML Tip - How to View HTML Source

Have you ever seen a Web page and wondered "Hey! How did they do that?"

To find out, right-click in the page and select "View Page Source" (in Chrome) or "View Source" (in IE), or similar in another browser. This will open a window containing the HTML code of the page.

HTML Tag Reference

iFarouq Tech's tag reference contains additional information about these tags and their attributes.

You will learn more about HTML tags and attributes in the next chapters of this tutorial.

Tag	Description
<code><html></code>	Defines an HTML document
<code><body></code>	Defines the document's body
<code><head></code>	Defines the document's head element
<code><h1> to <h6></code>	Defines HTML headings
<code><hr></code>	Defines a horizontal line

CHAPTER SEVEN

7.0 HTML PARAGRPHS

HTML documents are divided into paragraphs.

7.1 HTML Paragraphs

The HTML **< p >** element defines a **paragraph**.

Example

```
<!DOCTYPE html>
<html>
  <body>
    <p>This is a paragraph.</p>
    <p>This is a paragraph.</p>
    <p>This is a paragraph.</p>
  </body>
</html>
```

Browsers automatically add an empty line before and after a paragraph.

7.2 HTML Display

You cannot be sure how HTML will be displayed. Large or small screens, and resized windows will create different results. With HTML, you cannot change the output by adding extra spaces or extra lines in your HTML code.

The browser will remove extra spaces and extra lines when the page is displayed. Any number of spaces, and any number of new lines, count as **only one space**.

Example

```
<!DOCTYPE html>
<html>
  <body>
```

```
<p>
    This paragraph contains a lot of lines in the source code, but the
    browser ignores it.

</p>
<p>
    This paragraph contains    a lot of spaces in the source   code,
    but the   browser  ignores it.

</p>
<p>
    The number of lines in a paragraph depends on the size of the
    browser window. If you resize the browser window, the
    number of lines in this paragraph will change.

</p>
</body>

</html>
```

7.3 Don't Forget the End Tag

Most browsers will display HTML correctly even if you forget the end tag:

```
<!DOCTYPE html>
<html>
    <body>
        <p>This is a paragraph.
        <p>This is a paragraph.
        <p>This is a paragraph.
        <p>Don't forget to close your HTML tags!</p>
    </body>
</html>
```

The example above will work in most browsers, but don't rely on it.

Forgetting the end tag can produce unexpected results or errors.

Stricter versions of HTML, like XHTML, do not allow you to skip the end tag.

7.4 HTML Line Breaks

The HTML `
` element defines a **line break**. Use `
` if you want a line break (a new line) without starting a new paragraph:

Example

```
<!DOCTYPE html>
<html>
  <body>
    <p>This is<br>a para<br>graph with line breaks</p>
  </body>
</html>
```

The `
` element is an empty HTML element. It has no end tag.

7.5 The Poem Problem

Example

```
<!DOCTYPE html>
<html>
  <body>
    <p>In HTML, spaces and new lines are ignored:</p>
    <p>
      My Bonnie lies over the ocean.
      My Bonnie lies over the sea.
      My Bonnie lies over the ocean.
      Oh, bring back my Bonnie to me.
    </p>
  </body>
</html>
```

7.6 The HTML <pre> Element

The HTML <pre> element defines a block of **pre-formatted** text, with structured spaces and lines. To display anything, with right spacing and line-breaks, you must wrap the text in a <pre> element:

Example

```
<!DOCTYPE html>

<html>

<body>

<p>The pre tag is needed for displaying poems:</p>

<pre>

    My Bonnie lies over the ocean.

    My Bonnie lies over the sea.

    My Bonnie lies over the ocean.

    Oh, bring back my Bonnie to me.

</pre>

</body>

</html>
```

CHAPTER EIGHT

8.0 HTML STYLE

```
<!DOCTYPE html>

<html>

    <body>

        <h2 style="color:red">I am Red</h2>

        <h2 style="color:blue">I am Blue</h2>

    </body>

</html>
```

8.1 HTML Styling

Every HTML element has a default style (background color is white, text color is black, text-size is 12px ...). Changing the default style of an HTML element, can be done with the style attribute.

This example changes the default background color from white to lightgrey:

Example

```
<!DOCTYPE html>

<html>

    <body style="background-color:lightgrey">

        <h1>This is a heading</h1>

        <p>This is a paragraph.</p>

    </body>

</html>
```

The bgcolor attribute, supported in older versions of HTML, is not valid in HTML5.

8.2 The HTML Style Attribute

The HTML style attribute has the following **syntax**:

```
style="property:value"
```

The **property** is a CSS property. The **value** is a CSS value. You will learn more about CSS later in this tutorial.

8.3 HTML Text Color

The **color** property defines the text color to be used for an HTML element:

Example

```
<!DOCTYPE html>
<html>
  <body>
    <h1 style="color:blue">This is a heading</h1>
    <p style="color:red">This is a paragraph.</p>
  </body>
</html>
```

8.4 HTML Text Fonts

The **font-family** property defines the font to be used for an HTML element:

```
<!DOCTYPE html>
<html>
  <body>
    <h1 style="font-family:verdana">
      This is a heading
    </h1>
    <p style="font-family:courier">
      This is a paragraph.
    </p>
```

```
</body>  
</html>
```

The `` tag, supported in older versions of HTML, is not valid in HTML5.

8.5 HTML Text Size

The **font-size** property defines the text size to be used for an HTML element:

```
<!DOCTYPE html>  
<html>  
  <body>  
    <h1 style="font-size:300%">  
      This is a heading  
    </h1>  
    <p style="font-size:160%">  
      This is a paragraph.  
    </p>  
  </body>  
</html>
```

8.6 HTML Text Alignment

The **text-align** property defines the horizontal text alignment for an HTML element:

Example

```
<!DOCTYPE html>  
<html>  
  <body>  
    <h1 style="text-align:center">  
      Centered heading  
    </h1>  
    <p>This is a paragraph.  
    </p>  
  </body>
```

</html>

The <center> tag, supported in older versions of HTML, is not valid in HTML5.

Chapter Summary

- Use the **style** attribute for styling HTML elements
- Use **background-color** for background color
- Use **color** for text colors
- Use **font-family** for text fonts
- Use **font-size** for text sizes
- Use **text-align** for text alignment
-

CHAPTER NINE

9.0 HTML TEXT FORMATTING ELEMENTs

Text Formatting

This text is bold

This text is italic

This is ^{superscript}

In the previous chapter, you learned about HTML **styling**, using the HTML **style attribute**.

HTML also defines **special elements**, for defining text with a special **meaning**. HTML uses elements like **** and **<i>** for formatting output, like **bold** or *italic* text.

Formatting elements were designed to display special **types of text**:

- Bold text
- Important text
- Italic text
- Emphasized text
- Marked text
- Small text
- Deleted text
- Inserted text
- Subscripts
- Superscripts

9.1 HTML Bold and Strong Formatting

The HTML **** element defines **bold** text, without any extra importance.

Example

```
<!DOCTYPE html>
<html>
  <body>
    <p>This text is normal.</p>
    <p><b>This text is bold.</b></p>
  </body>
</html>
```

The HTML **** element defines **strong** text, with added semantic "strong" importance.

Example

```
<!DOCTYPE html>
<html>
  <body>
    <p>This text is normal.</p>
    <p><strong>This text is strong.</strong></p>
  </body>
</html>
```

9.2 HTML *Italic* and *Emphasized* Formatting

The HTML **<i>** element defines *italic* text, without any extra importance.

Example

```
<!DOCTYPE html>
<html>
  <body>
    <p>This text is normal.</p>
    <p><i>This text is italic.</i></p>
  </body>
</html>
```

The HTML **** element defines *emphasized* text, with added semantic importance.

Example

```
<!DOCTYPE html>
<html>
  <body>
    <p>This text is normal.</p>
    <p><em>This text is emphasized.</em></p>
  </body>
</html>
```

Browsers display `` as ``, and `` as `<i>`.

However, there is a difference in the meaning of these tags: `` and `<i>` defines bold and italic text,

but `` and `` means that the text is "important".

9.2 HTML Small Formatting

The HTML `<small>` element defines **small** text:

Example

```
<!DOCTYPE html>
<html>
  <body>
    <h2>HTML <small>Small</small> Formatting</h2>
  </body>
</html>
```

9.4 HTML Marked Formatting

The HTML `<mark>` element defines **marked** or highlighted text:

Example

The HTML `<mark>` element defines **marked** or highlighted text:

Example

```
<!DOCTYPE html>
<html>
  <body>
    <h2>HTML <mark>Marked</mark> Formatting</h2>
  </body>
</html>
```

9.5 HTML DELETED Formatting

The HTML `` element defines **deleted** (removed) of text.

Example

```
<!DOCTYPE html>
<html>
  <body>
    The del element represent deleted (removed) text.
    <p>My favorite color is <del>blue</del> red.</p>
  </body>
</html>
```

9.6 HTML Inserted Formatting

The HTML `<ins>` element defines **inserted** (added) text.

Example

```
<!DOCTYPE html>
<html>
  <body>
    The ins element represent inserted (added) text.
    <p>My favorite <ins>color</ins> is red.</p>
  </body>
</html>
```

9.7 HTML Subscript Formatting

The HTML `<sub>` element defines **subscripted** text.

Example

```
<!DOCTYPE html>
<html>
  <body>
    <p>This is <sub>subscripted</sub> text.</p>
  </body>
</html>
```

9.8 HTML Superscript Formatting

The HTML `<sup>` element defines **superscripted** text.

Example

```
<!DOCTYPE html>
<html>
  <body>
    <p>This is <sup>superscripted</sup> text.</p>
  </body>
</html>
```

HTML Text Formatting Elements

Tag	Description
<code></code>	Defines bold text
<code></code>	Defines emphasized text
<code><i></code>	Defines italic text
<code><small></code>	Defines smaller text
<code></code>	Defines important text
<code><sub></code>	Defines subscripted text
<code><sup></code>	Defines superscripted text
<code><ins></code>	Defines inserted text
<code></code>	Defines deleted text
<code><mark></code>	Defines marked/highlighted text

CHAPTER TEN

10.0 HTML QUOTATION AND CITATION ELEMENTS

Quotation

Here is a quote from WWF's website:

For 50 years, WWF has been protecting the future of nature. The world's leading conservation organization, WWF works in 100 countries and is supported by 1.2 million members in the United States and close to 5 million globally.

10.1 HTML <q> for Short Quotations

The HTML <q> element defines a **short quotation**. Browsers usually insert **quotation marks** around the <q> element.

Example

```
<!DOCTYPE html>
<html>
  <body>
    <p>Browsers usually insert quotation marks around the q element.</p>
    <p>WWF's goal is to: <q>Build a future where people live in harmony
       with nature.</q></p>
  </body>
</html>
```

10.2 HTML <blockquote> for Long Quotations

The HTML <blockquote> element defines a quoted section. Browsers usually **indent** <blockquote> elements.

Example

```
<!DOCTYPE html>
<html>
  <body>
    <p>Browsers usually indent blockquote elements.</p>
    <blockquote cite="http://www.worldwildlife.org/who/index.html">
      For 50 years, WWF has been protecting the future of nature.
      The world's leading conservation organization, WWF works in
      100 countries and is supported by 1.2 million members in the
      United States and close to 5 million globally.
    </blockquote>
  </body>
</html>
```

10.3 HTML <abbr> for Abbreviations

The HTML <abbr> element defines an **abbreviation** or an acronym. Marking abbreviations can give useful information to browsers, translation systems and search-engines.

Example

```
<!DOCTYPE html>
<html>
  <body>
    <p>The
      <abbr title="World Health Organization">WHO</abbr>
      was founded in 1948.
    </p>
    <p>
      Marking up abbreviations can give useful information to
      browsers, translation systems and search-engines.
    </p>
  </body>
</html>
```

10.4 HTML <dfn> for Definitions

The HTML <dfn> element defines the **definition** of a term or an abbreviation.

The usage of <dfn>, as described in the HTML5 standard, is complicated:

1. If the title attribute of the <dfn> element is present, it defines the term:

Example

```
<!DOCTYPE html>
<html>
  <body>
    <p>
      The <dfn title="World Health Organization">WHO</dfn> was
      founded in 1948.
    </p>
    <p>
      Marking up definitions can give useful information to
      browsers, translation systems and search-engines.
    </p>
  </body>
</html>
```

2. If the <dfn> element contains an <abbr> element with a title, then that title defines the term:

Example

```
<!DOCTYPE html>
<html>
  <body>
    <p>The
```

```

<dfn><abbr title="World Health Organization">WHO</abbr></dfn>
    was founded in 1948.

</p>
<p>
    Marking up definitions can give useful information to
    browsers, translation systems and search-engines.</p>
</body>
</html>

```

3. Otherwise, the <dfn> text content is the term, and the parent element contains the definition.

Example

```

<!DOCTYPE html>
<html>
    <body>
        <p>The
            <dfn>WHO</dfn> World Health Organization was founded in 1948.
        </p>
        <p>
            Marking up definitions can give useful information to browsers,
            translation systems and search-engines.</p>
    </body>
</html>

```

If you want to keep it simple, use the first form, or use <abbr> instead.

10.5 HTML <address> for Contact Information

The HTML <address> element defines contact information (author/owner) of a document or article. The element is usually displayed in **italic**. Most browsers will add a line break before and after the element.

Example

```
<!DOCTYPE html>
<html>
  <body>
    <p>
      The HTML address element defines contact information
      (author/owner) of a document or article.
    </p>
    <address>
      Written by Jon Doe.<br>
      Visit us at:<br>
      Example.com<br>
      Box 564, Disneyland<br>
      USA
    </address>
  </body>
</html>
```

10.6 HTML <cite> for Work Title

The HTML <cite> element defines the **title of a work**. Browsers usually displays <cite> elements in *italic*.

Example

```
<!DOCTYPE html>
<html>
  <body>
    <p>The HTML cite element defines the title of a work.</p>
    <p>Browsers usually displays cite elements in italic.</p>
    
    <p><cite>The Scream</cite> by Edward Munch. Painted in 1893.</p>
  </body>
</html>
```

10.7 HTML <bdo> for Bi-Directional Override

The HTML <bdo> element defines **bi-directional override**.

If your browser supports bdo, this text will be written from right to left:

Example

```
<!DOCTYPE html>
<html>
  <body>
    <p>
      If your browser supports bi-directional override (bdo), the next
      line will be written from right to left (rtl):
    </p>
    <bdo dir="rtl">
      This line will be written from right to left
    </bdo>
  </body>
</html>
```

HTML Quotations, Citations, and Definition Elements

Tag	Description
<u><abbr></u>	Defines an abbreviation or acronym
<u><address></u>	Defines contact information for the author/owner of a document
<u><bdo></u>	Defines the text direction
<u><blockquote></u>	Defines a section that is quoted from another source
<u><dfn></u>	Defines the definition of a term or an abbreviation.
<u><q></u>	Defines a short inline quotation
<u><cite></u>	Defines the title of a work

CHAPTER ELEVEN

11.0 HTML COMMENTS

Comment tags <!-- and --> are used to insert comments in HTML.

11.1 HTML Comment Tags

You can add comments to your HTML source by using the following syntax:

Example

```
<!-- Write your comments here -->
```

Note: There is an exclamation point (!) in the opening tag, but not in the closing tag.

Comments are not displayed by the browser, but they can help document your HTML.

With comments you can place notifications and reminders in your HTML:

Example

```
<!DOCTYPE html>

<html>

<body>

<!-- This is a comment -->

<p>This is a paragraph.</p>

<!-- Comments are not displayed in the browser -->

</body>

</html>
```

Comments are also great for debugging HTML, because you can comment out HTML lines of code, one at a time, to search for errors:

Example

```
<!DOCTYPE html>

<html>

<body>

<!-- Do not display this at the moment


```

```
-->  
</body>  
</html>
```

11.2 Conditional Comments

You might stumble upon conditional comments in HTML:

```
<!--[if IE 8]>  
    .... some HTML here ....  
<![endif]-->
```

Conditional comments defines HTML tags to be executed by Internet Explorer only.

11.3 Software Program Tags

HTML comments tags can also be generated by various HTML software programs.

For example <!--webbot bot--> tags wrapped inside HTML comments by FrontPage and Expression Web.

As a rule, let these tags stay, to help support the software that created them.

CHAPTER TWELVE

12.0 HTML STYLE-CSS

12.1 CSS = Styles and Colors

Manipulate Text

Colors, Boxes

Example

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      body {
        background-color:lightgrey
      }
      h1 {
        color:blue
      }
      p {
        color:green
      }
    </style>
  </head>
  <body>
    <h1>This is a heading</h1>
    <p>This is a paragraph.</p>
  </body>
</html>
```

12.2 Styling HTML with CSS

CSS stands for Cascading Style Sheets

Styling can be added to HTML elements in 3 ways:

- Inline - using a **style attribute** in HTML elements
- Internal - using a **<style> element** in the HTML <head> section
- External - using one or more **external CSS files**

The most common way to add styling, is to keep CSS syntax in separate CSS files. But, in this tutorial, we use internal styling, because it is easier to demonstrate, and easier for you to try it yourself.

You can learn much more about CSS in our [CSS Tutorial](#).

12.3 CSS Syntax

CSS styling has the following **syntax**:

element { property:value ; property:value }

The **element** is an HTML element name. The **property** is a CSS property. The **value** is a CSS value.

Multiple styles are separated with semicolon.

12.4 Inline Styling (Inline CSS)

Inline styling is useful for applying a unique style to a single HTML element:

Inline styling uses the **style attribute**.

This inline styling changes the text color of a single heading:

Example

```
<!DOCTYPE html>
<html>
  <body>
    <h1 style="color:blue">This is a Blue Heading</h1>
  </body>
</html>
```

12.5 Internal Styling (Internal CSS)

An internal style sheet can be used to define a common style for all HTML elements on a page.

Internal styling is defined in the **<head>** section of an HTML page, using a **<style>** element:

Example

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      body {
        background-color:lightgrey
      }
      h1 {
        color:blue
      }
      p {
        color:green
      }
    </style>
  </head>
```

```
<body>  
    <h1>This is a heading</h1>  
    <p>This is a paragraph.</p>  
</body>  
</html>
```

12.6 External Styling (External CSS)

External style sheet are ideal when the style is applied to many pages. With external style sheets, you can change the look of an entire site by changing one file.

External styles are defined in the `<head>` section of an HTML page, in the `<link>` element:

Example

```
<!DOCTYPE html>  
<html>  
    <head>  
        <link rel="stylesheet" href="styles.css">  
    </head>  
    <body>  
        <h1>This is a heading</h1>  
        <p>This is a paragraph.</p>  
    </body>  
</html>
```

12.7 CSS Fonts

The CSS property **color** defines the text color to be used for an HTML element. The CSS property **font-family** defines the font to be used for an HTML element. The CSS property **font-size** defines the text size to be used for an HTML element.

Example

```

<!DOCTYPE html>
<html>
    <head>
        <style>
            h1 {
                color:blue;
                font-family:verdana;
                font-size:300%;
            }
            p {
                color:red;
                font-family:courier;
                font-size:160%;
            }
        </style>
    </head>
    <body>
        <h1>This is a heading</h1>
        <p>This is a paragraph.</p>
    </body>
</html>

```

The `` tag, supported in older versions of HTML, is not valid in HTML5.

12.8 The CSS Box Model

Every visible HTML element has a box around it, even if you cannot see it.

The CSS **border** property defines a visible border around an HTML element:

Example

```

<!DOCTYPE html>
<html>
    <head>
        <style>

```

```

    p {
        border:1px solid grey;
    }
</style>

</head>

<body>

    <h1>This is a heading</h1>
    <p>This is a paragraph.</p>
    <p>This is a paragraph.</p>
    <p>This is a paragraph.</p>

</body>

</html>

```

The CSS **padding** property defines a padding (space) inside the border:

Example

```

<!DOCTYPE html>
<html>
    <head>
        <style>
            p {
                border:1px solid grey;
                padding:10px;
            }
        </style>
    </head>
    <body>
        <h1>This is a heading</h1>
        <p>This is a paragraph.</p>
        <p>This is a paragraph.</p>
        <p>This is a paragraph.</p>

    </body>
</html>

```

The CSS **margin** property defines a margin (space) outside the border:

Example

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      p {
        border:1px solid grey;
        padding:10px;
        margin:30px;
      }
    </style>
  </head>
  <body>
    <h1>This is a heading</h1>
    <p>This is a paragraph.</p>
    <p>This is a paragraph.</p>
    <p>This is a paragraph.</p>
  </body>
</html>
```

The CSS examples above use px to define sizes in pixels (screen pixels).

12.9 The id Attribute

All the examples above use CSS to style HTML elements in a general way. The CSS styles define an equal style for all equal elements.

To define a special style for a special element, first add an id attribute to the element:

Example

```
<p id="p01">I am different</p>
```

then define a different style for the (identified) element:

Example

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      p#p01 {
        color: blue;
      }
    </style>
  </head>
  <body>
    <p>This is a paragraph.</p>
    <p>This is a paragraph.</p>
    <p>This is a paragraph.</p>
    <p id="p01">I am different.</p>
  </body>
</html>
```

12.10 The class Attribute

To define a style for a special type (class) of elements, add a class attribute to the element:

Example

```
<p class="error">I am different</p>
```

Now you can define a different style for this type (class) of element:

Example

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      p.error {
```

```

        color:red;
    }
</style>
</head>
<body>
<p>This is a paragraph.</p>
<p>This is a paragraph.</p>
<p class="error">I am different.</p>
<p>This is a paragraph.</p>
<p class="error">I am different too.</p>
</body>
</html>

```

Use **id** to address **single** elements. Use **class** to address **groups** of elements.

12.11 Deprecated Tags and Attributes in HTML5

In older HTML versions, several tags and attributes were used to style documents. These tags are not supported in HTML5. Avoid using the **elements**: , <center> and <strike>. Avoid using the **attributes**: color and bgcolor.

Chapter Summary

- Use the HTML **style** attribute for inline styling
- Use the HTML style element to define internal CSS
- Use the HTML link element to define external CSS
- Use the HTML head element to store style and link elements
- Use the CSS **color** property for text colors
- Use the CSS **font-family** property for text fonts
- Use the CSS **font-size** property for text sizes

- Use the CSS **border** property for visible element borders
- Use the CSS **padding** property for space inside the border
- Use the CSS **margin** property for space outside the border

HTML Style Tags

Tag	Description
<code><style></code>	Defines style information for a document
<code><link></code>	Defines a link between a document and an external resource

CHAPTER THIRTEEN

13.0 HTML LINKS

Links are found in nearly all web pages. Links allow users to click their way from page to page.

13.1 HTML Links - Hyperlinks

HTML links are hyperlinks. A hyperlink is an element, a text, or an image that you can click on, and jump to another document.

13.2 HTML Links - Syntax

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

Link Syntax:

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

Example:

```
<!DOCTYPE html>

<html>

  <body>

    <p>
      <a href="index.html">Visit our HTML tutorial</a>
    </p>

  </body>

</html>
```

The **href** attribute specifies the destination address (<http://www.google.com/>)

The **link text** is the visible part (Visit our HTML tutorial).

Clicking on the link text, will send you to the specified address.

The link text does not have to be text. It can be an HTML image or any other HTML element.

Local Links

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

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

Example:

```
<!DOCTYPE html>
<html>
  <body>
    <p>
      <a href="html_images.html">HTML Images</a> is a link to a
      page on this website.

    </p>
    <p>
      <a href="http://www.example.org/">iFarouq Tech</a> is a link
      to a website on the World Wide Web.

    </p>
  </body>
</html>
```

13.3 HTML Links - Colors and Icons

When you move the mouse cursor over a link, two things will normally happen:

- The mouse arrow will turn into a little hand
- The color of the link element will change

By default, links will appear as this in all browsers:

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

You can change the defaults, using styles:

Example

```
<!DOCTYPE html>

<html>

<head>

<style>

a:link {

    color:#000000;

    background-color:transparent;

    text-decoration:none;

}

a:visited {

    color:#000000;

    background-color:transparent;
```

```

        text-decoration:none;

    }

a:hover {
    color:#ff0000;
    background-color:transparent;
    text-decoration:underline;
}

a:active {
    color:#ff0000;
    background-color:transparent;
    text-decoration:underline;
}

</style>

</head>

<body>

<p>You can change the default colors of links</p>

<a href="html_images.html" target="_blank">HTML Images</a>

</body>

</html>

```

13.4 HTML Links - The target Attribute

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

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

Example

```

<!DOCTYPE html>

<html>

    <body>

        <a href="index.html" target="_blank">Visit our HTML tutorial!</a>

        <p>If you set the target attribute to "_blank", the link will open in a new browser window or tab.</p>

    </body>

</html>

```

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

If your webpage is locked in a frame, you can use target="_top" to break out of the frame:

Example

```

<!DOCTYPE html>

<html>

    <body>

        <p>

            Locked in a frame? <a href="index.html" target="_top">Click here!</a>

        </p>

    </body>

</html>

```

13.5 HTML Links - Image as Link

It is common to use images as links:

Example

```
<!DOCTYPE html>

<html>

<body>

<p>The image is a link. You can click on it.</p>

<a href="default.html">

</a>

<p>

    We have added "border:0" to prevent IE9 (and earlier) from
    displaying a border around the image.

</p>

</body>

</html>
```

border:0 is added to prevent IE9 (and earlier) from displaying a border around the image.

13.6 HTML Links - The id Attribute

The **id** attribute can be used to create bookmarks inside HTML documents. Bookmarks are not displayed in any special way. They are invisible to the reader.

Example

Add an id attribute to any element:

```
<a id="tips">Useful Tips Section</a>
```

Then create a link to the `<a>` element (Useful Tips Section):

```
<a href="#tips">Visit the Useful Tips Section</a>
```

Or, create a link to the `<a>` element (Useful Tips Section) from another page:

```
<!DOCTYPE html>

<html>

<body>

<p>

    <a href="#C4">See also Chapter 4.</a>

</p>

<h2>Chapter 1</h2>

<p>This chapter explains ba bla bla</p>

<h2>Chapter 2</h2>

<p>This chapter explains ba bla bla</p>

<h2>Chapter 3</h2>

<p>This chapter explains ba bla bla</p>

<h2><a id="C4">Chapter 4</a></h2>

<p>This chapter explains ba bla bla</p>

<h2>Chapter 5</h2>

<p>This chapter explains ba bla bla</p>

<h2>Chapter 6</h2>

<p>This chapter explains ba bla bla</p>

<h2>Chapter 7</h2>

<p>This chapter explains ba bla bla</p>
```

```
<h2>Chapter 8</h2>  
<p>This chapter explains ba bla bla</p>  
  
<h2>Chapter 9</h2>  
<p>This chapter explains ba bla bla</p>  
  
<h2>Chapter 10</h2>  
<p>This chapter explains ba bla bla</p>  
  
<h2>Chapter 11</h2>  
<p>This chapter explains ba bla bla</p>  
  
<h2>Chapter 12</h2>  
<p>This chapter explains ba bla bla</p>  
  
<h2>Chapter 13</h2>  
<p>This chapter explains ba bla bla</p>  
  
<h2>Chapter 14</h2>  
<p>This chapter explains ba bla bla</p>  
  
<h2>Chapter 15</h2>  
<p>This chapter explains ba bla bla</p>  
  
<h2>Chapter 16</h2>  
<p>This chapter explains ba bla bla</p>  
  
<h2>Chapter 17</h2>  
<p>This chapter explains ba bla bla</p>  
  
</body>  
</html>
```

Without a trailing slash on subfolder addresses, you might generate two requests to the server.

Many servers will automatically add a slash to the address, and then create a new request.

Chapter Summary

- Use the HTML `<a>` element to define a link
- Use the HTML `href` attribute to define the link address
- Use the HTML `target` attribute to define where to open the linked document
- Use the HTML `` element (inside `<a>`) to use an image as a link
- Use the HTML `id` attribute (`id=value`) to define bookmarks in a page
- Use the HTML `href` attribute (`href="#value"`) to address the bookmark

HTML Link Tags

Tag	Description
<code><a></code>	Defines a hyperlink

CHAPTER FOURTEEN

HTML IMAGES

Example

```
<!DOCTYPE html>

<html>

    <body>

        <h2>Spectacular Mountains</h2>

    </body>

</html>
```

14.1 HTML Images Syntax

In HTML, images are defined with the `` tag. The `` tag is empty, it contains attributes only, and does not have a closing tag. The `src` attribute defines the url (web address) of the image:

Always specify image size. If the size is unknown, the page will flicker while the image loads.

```

```

14.2 The alt Attribute

The `alt` attribute specifies an alternate text for the image, if it cannot be displayed.

The value of the alt attribute should describe the image in words:

Example

```

```

The alt attribute is **required**. A web page will not validate correctly without it.

14.3 HTML Screen Readers

Screen readers are software programs that can read what is displayed on a screen. Used on the web, screen readers can "reproduce" HTML as text-to-speech, sound icons, or braille output.

Screen readers are used by people who are blind, visually impaired, or learning disabled.

Screen readers can read the **alt** attribute. **Image Size - Width and Height** You can use the **style** attribute to specify the **width** and **height** of an image.

The values are specified in pixels (use px after the value):

Example

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

Alternatively, you can use width and height **attributes**.

The values are specified in pixels (without px after the value):

Example

```
<!DOCTYPE html>
<html>
  <body>
```

```
  
</body>  
</html>
```

14.4 Width and Height or Style?

Both the width, the height, and the style attributes, are valid in the latest HTML5 standard. We suggest you use the style attribute. It prevents styles sheets from changing the default size of images:

Example

```
<!DOCTYPE html>  
<html>  
  <head>  
    <style>  
      img {  
        width:100%;  
      }  
    </style>  
  </head>  
  <body>  
    <p>It is better to use the style attribute (instead of the width and height attributes), because it prevents  
internal or external styles sheets to change the default size of an  
image:</p>  
      
      
  </body>  
</html>
```

At iFarouq Tech we prefer to use the style attribute.

14.5 Images in Another Folder

If not specified, the browser expects to find the image in the same folder as the web page.

However, it is common on the web, to store images in a sub-folder, and refer to the folder in the image name:

Example

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

If a browser cannot find an image, it will display a broken link icon:

Example

```
<!DOCTYPE html>
<html>
  <body>
    <p>
      If a browser cannot find an image, it will display a broken link
      icon.
    </p>
    
  </body>
</html>
```

14.6 Images on Another Server

Some web sites store their images on image servers. Actually, you can access images from any web address in the world:

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

14.7 Animated Images

The GIF standard allows animated images:

Example

```
<!DOCTYPE html>
<html>
  <body>
    <p>
      The GIF standard allows moving images.
    </p>
    
  </body>
</html>
```

Note that the syntax of inserting animated images is no different from non-animated images.

14.8 Using an Image as a Link

It is common to use images as links:

Example

```
<!DOCTYPE html>
<html>
  <body>
    <p>The image is a link. You can click on it.</p>
```

```

<a href="default.html">
    
</a>
<p>
    We have added "border:0" to prevent IE9 (and earlier) from
    displaying a border around the image.
</p>
</body>
</html>

```

We have added border:0 to prevent IE9 (and earlier) from displaying a border around the image.

14.9 Image Maps

For an image, you can create an image map, with clickable areas:

Example

```

<!DOCTYPE html>
<html>
    <body>
        <p>Click on the sun or on one of the planets to watch it closer:</p>
        
        <map name="planetmap">
            <area shape="rect" coords="0,0,82,126" alt="Sun"
                href="sun.html">
            <area shape="circle" coords="90,58,3" alt="Mercury"
                href="mercur.html">
            <area shape="circle" coords="124,58,8" alt="Venus"
                href="venus.html">
        </map>
    </body>
</html>

```

14.10 Image Floating

You can let an image float to the left or right of a paragraph:

Example

```
<!DOCTYPE html>
<html>
  <body>
    <p>
      
      A paragraph with an image. A paragraph with an image.
      A paragraph with an image. A paragraph with an image.
      A paragraph with an image. A paragraph with an image.
    </p>
    <p>The image floats to the left of the text.</p>
    <p>
      Please use the CSS float property. The align attribute is
      deprecated in HTML 4, and not supported in HTML5.
    </p>
  </body>
</html>
```

Chapter Summary

- Use the HTML **** element to define images
- Use the HTML **src** attribute to define the image file name
- Use the HTML **alt** attribute to define an alternative text
- Use the HTML **width** and **height** attributes to define the image size
- Use the CSS **width** and **height** properties to define the image size (alternatively)
- Use the CSS **float** property to define image floating
- Use the HTML **usemap** attribute to point to an image map

- Use the HTML <**map**> element to define an image map
- Use the HTML <**area**> element to define image map areas

Loading images takes time. Large images can slow down your page. Use images carefully.

HTML Image Tags

Tag	Description
	Defines an image
<map>	Defines an image-map
<area>	Defines a clickable area inside an image-map

CHAPTER FIFTEEN

15.0 HTML TABLES

HTML Table Example

Number	First Name	Last Name	Points
1	Eve	Jackson	94
2	John	Doe	80
3	Adam	Johnson	67
4	Jill	Smith	50

15.1 Defining HTML Tables

The `<table>` tag defines an HTML table. An HTML table consists of the `<table>` element and one or more `<tr>`, `<th>`, and `<td>` elements. The `<tr>` element defines a table row, the `<th>` element defines a table header, and the `<td>` element defines a table cell. A more complex HTML table may also include `<caption>`, `<col>`, `<colgroup>`, `<thead>`, `<tfoot>`, and `<tbody>` elements.

Example

```
<!DOCTYPE html>
<html>
  <body>
    <table style="width:100%">
      <tr>
        <td>Jill</td>
        <td>Smith</td>
        <td>50</td>
      </tr>
      <tr>
        <td>Eve</td>
        <td>Jackson</td>
        <td>94</td>
```

```

</tr>
<tr>
    <td>John</td>
    <td>Doe</td>
    <td>80</td>
</tr>
</table>
</body>
</html>

```

Example explained:

- Tables are defined with the **<table>** tag.
- Tables are divided into **table rows** with the **<tr>** tag.
- Table rows are divided into **table data** with the **<td>** tag.
- A table row can also be divided into **table headings** with the **<th>** tag.

Table data **<td>** are the data containers of the table. They can contain all sorts of HTML elements like text, images, lists, other tables, etc.

15.2 An HTML Table with a Border Attribute

If you do not specify a border for the table, it will be displayed without borders. A border can be added using the border attribute:

Example

```

<!DOCTYPE html>
<html>
    <body>
        <table border="1" style="width:100%">
            <tr>
                <td>Jill</td>
                <td>Smith</td>
                <td>50</td>

```

```

</tr>
<tr>
    <td>Eve</td>
    <td>Jackson</td>
    <td>94</td>
</tr>
<tr>
    <td>John</td>
    <td>Doe</td>
    <td>80</td>
</tr>
</table>
</body>
</html>

```

The border attribute is on its way out of the HTML standard! It is better to use CSS.

To add borders, use the **CSS border** property:

Example

```

<!DOCTYPE html>
<html>
    <head>
        <style>
            table, th, td {
                border: 1px solid black;
            }
        </style>
    </head>
    <body>
        <table style="width:100%">
            <tr>
                <td>Jill</td>
                <td>Smith</td>
                <td>50</td>
            </tr>
            <tr>
                <td>Eve</td>
                <td>Jackson</td>
                <td>94</td>
            </tr>
            <tr>
                <td>John</td>
                <td>Doe</td>
                <td>80</td>
            </tr>
        </table>
    </body>

```

```
</table>
</body>
</html>
```

Remember to define borders for both the table and the table cells.

15.3 An HTML Table with Collapsed Borders

If you want the borders to collapse into one border, add **CSS border-collapse**:

Example

```
<!DOCTYPE html>
<html>
<head>
<style>
    table, th, td {
        border: 1px solid black;
        border-collapse: collapse;
    }
</style>
</head>
<body>
<table style="width:100%">
    <tr>
        <td>
    </td>
    </tr>
    <tr>
        <td>Eve</td>
        <td>Jackson</td>
        <td>94</td>
    </tr>
    <tr>
        <td>John</td>
        <td>Doe</td>
        <td>80</td>
    </tr>
</table>
</body>
</html>
```

15.4 An HTML Table with Cell Padding

Cell padding specifies the space between the cell content and its borders. If you do not specify a padding, the table cells will be displayed without padding. To set the padding, use the **CSS**

padding property:

Example

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      table, th, td {
        border: 1px solid black;
        border-collapse: collapse;
      }
      th, td {
        padding: 15px;
      }
    </style>
  </head>
  <body>
    <table style="width:100%">
      <tr>
        <td>Jill</td>
        <td>Smith</td>
        <td>50</td>
      </tr>
      <tr>
        <td>Eve</td>
        <td>Jackson</td>
        <td>94</td>
      </tr>
      <tr>
        <td>John</td>
        <td>Doe</td>
        <td>80</td>
      </tr>
    </table>
    <p>Try to change the padding to 5px.</p>
  </body>
</html>
```

15.5 HTML Table Headings

Table headings are defined with the **<th>** tag. By default, all major browsers display table headings as bold and centered:

Example

```

<!DOCTYPE html>
<html>
  <head>
    <style>
      table, th, td {
        border: 1px solid black;
        border-collapse: collapse;
      }
      th, td {
        padding: 5px;
      }
    </style>
  </head>
  <body>
    <table style="width:100%">
      <tr>
        <th>Firstname</th>
        <th>Lastname</th>
        <th>Points</th>
      </tr>
      <tr>
        <td>Jill</td>
        <td>Smith</td>
        <td>50</td>
      </tr>
      <tr>
        <td>Eve</td>
        <td>Jackson</td>
        <td>94</td>
      </tr>
      <tr>
        <td>John</td>
        <td>Doe</td>
        <td>80</td>
      </tr>
    </table>
  </body>
</html>

```

To left-align the table headings, use the **CSS text-align** property:

Example

```

<!DOCTYPE html>
<html>
  <head>
    <style>
      table, th, td {
        border: 1px solid black;
        border-collapse: collapse;
      }
      th {
        text-align: left;
      }
      th, td {
        padding: 5px;
      }
    </style>
  </head>
  <body>
    <table style="width:100%">
      <tr>
        <th>Firstname</th>
        <th>Lastname</th>
        <th>Points</th>
      </tr>
      <tr>
        <td>Jill</td>
        <td>Smith</td>
        <td>50</td>
      </tr>
      <tr>
        <td>Eve</td>
        <td>Jackson</td>
        <td>94</td>
      </tr>
      <tr>
        <td>John</td>
        <td>Doe</td>
        <td>80</td>
      </tr>
    </table>
  </body>
</html>

```

```

        }
    th, td {
        padding: 5px;
    }
    th {
        text-align: left;
    }
</style>
</head>

<body>
    <table style="width:100%">
        <tr>
            <th>Firstname</th>
            <th>Lastname</th>
            <th>Points</th>
        </tr>
        <tr>
            <td>Jill</td>
            <td>Smith</td>
            <td>50</td>
        </tr>
        <tr>
            <td>Eve</td>
            <td>Jackson</td>
            <td>94</td>
        </tr>
        <tr>
            <td>John</td>
            <td>Doe</td>
            <td>80</td>
        </tr>
    </table>
</body>
</html>

```

15.6 An HTML Table with Border Spacing

Border spacing specifies the space between the cells. To set the border spacing for a table, use

the **CSS border-spacing** property:

Example

```

<!DOCTYPE html>
<html>
    <head>
        <style>

```

```

        table, th, td {
            border: 1px solid black;
            padding: 5px;
        }
        table {
            border-spacing: 15px;
        }
    </style>
</head>
<body>
    <table style="width:100%">
        <tr>
            <td>Jill</td>
            <td>Smith</td>
            <td>50</td>
        </tr>
        <tr>
            <td>Eve</td>
            <td>Jackson</td>
            <td>94</td>
        </tr>
        <tr>
            <td>John</td>
            <td>Doe</td>
            <td>80</td>
        </tr>
    </table>
    <p>Try to change the border-spacing to 5px.</p>
</body>
</html>

```

If the table has collapsed borders, border-spacing has no effect.

15.7 Table Cells that Span Many Columns

To make a cell span more than one column, use the **colspan** attribute:

Example

```

<!DOCTYPE html>
<html>
    <head>
        <style>
            table, th, td {
                border: 1px solid black;

```

```

        border-collapse: collapse;
    }

    th, td {
        padding: 5px;
        text-align: left;
    }

</style>

</head>

<body>

<h2>Cell that spans two columns:</h2>

<table style="width:100%">

    <tr>

        <th>Name</th>
        <th colspan="2">Telephone</th>

    </tr>

    <tr>

        <td>Bill Gates</td>
        <td>555 77 854</td>
        <td>555 77 855</td>

    </tr>

</table>

</body>

</html>

```

15.8 Table Cells that Span Many Rows

To make a cell span more than one row, use the **rowspan** attribute:

Example

```

<!DOCTYPE html>
<html>
    <head>

```

```

<style>
    table, th, td {
        border: 1px solid black;
        border-collapse: collapse;
    }
    th, td {
        padding: 5px;
        text-align: left;
    }
</style>

</head>
<body>
    <h2>Cell that spans two rows:</h2>
    <table style="width:100%">
        <tr>
            <th>Name:</th>
            <td>Bill Gates</td>
        </tr>
        <tr>
            <th rowspan="2">Telephone:</th>
            <td>555 77 854</td>
        </tr>
        <tr>
            <td>555 77 855</td>
        </tr>
    </table>
</body>
</html>

```

15.9 An HTML Table with a Caption

To add a caption to a table, use the `<caption>` tag:

Example

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      table, th, td {
        border: 1px solid black;
        border-collapse: collapse;
      }
      th, td {
        padding: 5px;
        text-align: left;
      }
    </style>
  </head>
  <body>
    <table style="width:100%">
      <caption>Monthly savings</caption>
      <tr>
        <th>Month</th>
        <th>Savings</th>
      </tr>
      <tr>
        <td>January</td>
        <td>$100</td>
      </tr>
      <tr>
```

```

<td>February</td>
<td>$50</td>
</tr>
</table>
</body>
</html>

```

The <caption> tag must be inserted immediately after the <table> tag.

15.10 Different Styles for Different Tables

Most of the examples above use a style attribute (width="100%") to define the width of each table. This makes it easy to define different widths for different tables. The styles in the <head> section, however, define a style for all tables in a page.

To define a special style for a special table, add an **id attribute** to the table:

Example

```

<table id="t01">
<tr>
    <th>Firstname</th>
    <th>Lastname</th>
    <th>Points</th>
</tr>
<tr>
    <td>Eve</td>
    <td>Jackson</td>
    <td>94</td>
</tr>
</table>

```

Now you can define a different style for this table:

```

<!DOCTYPE html>
<html>
    <head>
        <style>
            table, th, td {

```

```

        border: 1px solid black;
        border-collapse: collapse;
    }

    th, td {
        padding: 5px;
        text-align: left;
    }

    table#t01 {
        width: 100%;
        background-color: #f1f1c1;
    }

```

</style>

</head>

<body>

First Name	Last Name	Points
Jill	Smith	50
Eve	Jackson	94

```

<tr>
    <td>John</td>
    <td>Doe</td>
    <td>80</td>
</tr>
</table>
<br>
<table id="t01">
<tr>
    <th>First Name</th>
    <th>Last Name</th>
    <th>Points</th>
</tr>
<tr>
    <td>Jill</td>
    <td>Smith</td>
    <td>50</td>
</tr>
<tr>
    <td>Eve</td>
    <td>Jackson</td>
    <td>94</td>
</tr>
<tr>
    <td>John</td>
    <td>Doe</td>
    <td>80</td>
</tr>
</table>
</body>

```

```
</html>
```

And add more styles:

```
<!DOCTYPE html>
<html>
    <head>
        <style>
            table {
                width:100%;
            }
            table, th, td {
                border: 1px solid black;
                border-collapse: collapse;
            }
            th, td {
                padding: 5px;
                text-align: left;
            }
            table#t01 tr:nth-child(even) {
                background-color: #eee;
            }
            table#t01 tr:nth-child(odd) {
                background-color:#fff;
            }
            table#t01 th {
                background-color: black;
                color: white;
            }
        </style>
    </head>
    <body>
```

```

<table>
  <tr>
    <th>First Name</th>
    <th>Last Name</th>
    <th>Points</th>
  </tr>
  <tr>
    <td>Jill</td>
    <td>Smith</td>
    <td>50</td>
  </tr>
  <tr>
    <td>Eve</td>
    <td>Jackson</td>
    <td>94</td>
  </tr>
  <tr>
    <td>John</td>
    <td>Doe</td>
    <td>80</td>
  </tr>
</table>
<br>
<table id="t01">
  <tr>
    <th>First Name</th>
    <th>Last Name</th>
    <th>Points</th>
  </tr>
  <tr>

```

```

<td>Jill</td>
<td>Smith</td>
<td>50</td>
</tr>
<tr>
<td>Eve</td>
<td>Jackson</td>
<td>94</td>
</tr>
<tr>
<td>John</td>
<td>Doe</td>
<td>80</td>
</tr>
</table>
</body>
</html>

```

Chapter Summary

- Use the HTML **<table>** element to define a table
- Use the HTML **<tr>** element to define a table row
- Use the HTML **<td>** element to define a table data
- Use the HTML **<th>** element to define a table heading
- Use the HTML **<caption>** element to define a table caption
- Use the CSS **border** property to define a border
- Use the CSS **border-collapse** property to collapse cell borders
- Use the CSS **padding** property to add padding to cells
- Use the CSS **text-align** property to align cell text

- Use the CSS **border-spacing** property to set the spacing between cells
- Use the **colspan** attribute to make a cell span many columns
- Use the **rowspan** attribute to make a cell span many rows
- Use the **id** attribute to uniquely define one table

HTML Table Tags

Tag	Description
<u><table></u>	Defines a table
<u><th></u>	Defines a header cell in a table
<u><tr></u>	Defines a row in a table
<u><td></u>	Defines a cell in a table
<u><caption></u>	Defines a table caption
<u><colgroup></u>	Specifies a group of one or more columns in a table for formatting
<u><col></u>	Specifies column properties for each column within a <colgroup> element
<u><thead></u>	Groups the header content in a table
<u><tbody></u>	Groups the body content in a table
<u><tfoot></u>	Groups the footer content in a table

CHAPTER SIXTEEN

16.0 HTML LISTS

HTML can have Unordered Lists, Ordered Lists, or Description Lists:

Unordered HTML List

- The first item
- The second item
- The third item
- The fourth item

Ordered HTML List

1. The first item
2. The second item
3. The third item
4. The fourth item

HTML Description List

The first item

Description of item

The second item

Description of item

16.1 Unordered HTML Lists

An unordered list starts with the `` tag. Each list item starts with the `` tag. The list items will be marked with bullets (small black circles).

16.1.1 Unordered List:

```
<!DOCTYPE html>
<html>
  <body>
    <h2>Unordered List with Default Bullets</h2>
    <ul>
      <li>Apples</li>
      <li>Bananas</li>
      <li>Lemons</li>
      <li>Oranges</li>
    </ul>
  </body>
</html>
```

16.1.2 Unordered HTML Lists - The Style Attribute

A **style** attribute can be added to an **unordered list**, to define the style of the marker:

Style	Description
list-style-type:disc	The list items will be marked with bullets (default)
list-style-type:circle	The list items will be marked with circles
list-style-type:square	The list items will be marked with squares
list-style-type:none	The list items will not be marked

16.1.2.1 Disc:

```
<!DOCTYPE html>
<html>
  <body>
    <h2>Unordered List with Disc Bullets</h2>
    <ul style="list-style-type:disc">
      <li>Apples</li>
      <li>Bananas</li>
      <li>Lemons</li>
      <li>Oranges</li>
    </ul>
  </body>
</html>
```

16.1.2.2 Circle:

```
<!DOCTYPE html>
<html>
  <body>
```

```

<h2>Unordered List with Circle Bullets</h2>
<ul style="list-style-type:circle">
    <li>Apples</li>
    <li>Bananas</li>
    <li>Lemons</li>
    <li>Oranges</li>
</ul>
</body>
</html>

```

16.1.2.3 Square:

```

<!DOCTYPE html>
<html>
    <body>
        <h2>Unordered List with Square Bullets</h2>
        <ul style="list-style-type:square">
            <li>Apples</li>
            <li>Bananas</li>
            <li>Lemons</li>
            <li>Oranges</li>
        </ul>
    </body>
</html>

```

16.1.2.4 None:

```

<!DOCTYPE html>
<html>
    <body>
        <h2>Unordered List without Bullets</h2>
        <ul style="list-style-type:none">
            <li>Apples</li>
            <li>Bananas</li>
            <li>Lemons</li>
            <li>Oranges</li>
        </ul>
    </body>
</html>

```

Using a type attribute `<ul type="disc">`, instead of `<ul style="list-style-type:disc">`, also works. But in HTML5, the type attribute is not valid in unordered lists, only in ordered list.

16.2 Ordered HTML Lists

An ordered list starts with the `` tag. Each list item starts with the `` tag. The list items will be marked with numbers.

16.2.1 Ordered List:

```
<!DOCTYPE html>
<html>
  <body>
    <h2>Ordered List</h2>
    <ol>
      <li>Apples</li>
      <li>Bananas</li>
      <li>Lemons</li>
      <li>Oranges</li>
    </ol>
  </body>
</html>
```

16.2.3 Ordered HTML Lists - The Type Attribute

A **type** attribute can be added to an **ordered list**, to define the type of the marker

Type	Description
<code>type="1"</code>	The list items will be numbered with numbers (default)
<code>type="A"</code>	The list items will be numbered with uppercase letters
<code>type="a"</code>	The list items will be numbered with lowercase letters
<code>type="I"</code>	The list items will be numbered with uppercase roman numbers
<code>type="i"</code>	The list items will be numbered with lowercase roman numbers

16.2.3.1 Numbers:

```
<!DOCTYPE html>
<html>
  <body>
    <h2>Ordered List with Numbers</h2>
    <ol type="1">
      <li>Apples</li>
      <li>Bananas</li>
      <li>Lemons</li>
      <li>Oranges</li>
    </ol>
  </body>
</html>
```

```
</body>
</html>
```

16.2.3.2 **Upper Case:**

```
<!DOCTYPE html>
<html>
<body>
    <h2>Ordered List with Letters</h2>
    <ol type="A">
        <li>Apples</li>
        <li>Bananas</li>
        <li>Lemons</li>
        <li>Oranges</li>
    </ol>
</body>
</html>
```

16.2.3.3 **Lower Case:**

```
<!DOCTYPE html>
<html>
    <body>
        <h2>Ordered List with Lowercase Letters</h2>
        <ol type="a">
            <li>Apples</li>
            <li>Bananas</li>
            <li>Lemons</li>
            <li>Oranges</li>
        </ol>
    </body>
</html>
```

16.2.3.4 **Roman Upper Case:**

```
<!DOCTYPE html>
<html>
<body>
    <h2>Ordered List with Roman Numbers</h2>
```

```

<ol type="I">
    <li>Apples</li>
    <li>Bananas</li>
    <li>Lemons</li>
    <li>Oranges</li>
</ol>
</body>
</html>

```

16.2.3.5 Roman Lower Case:

```

<!DOCTYPE html>
<html>
    <body>
        <h2>Ordered List with Lowercase Roman Letters</h2>
        <ol type="i">
            <li>Apples</li>
            <li>Bananas</li>
            <li>Lemons</li>
            <li>Oranges</li>
        </ol>
    </body>
</html>

```

16.3 HTML Description Lists

A description list, is a list of terms, with a description of each term. The **<dl>** tag defines a description list. The **<dt>** tag defines the term (name), and the **<dd>** tag defines the data (description).

Description List:

```

<!DOCTYPE html>
<html>
    <body>
        <h2>A Description List</h2>
        <dl>
            <dt>Coffee</dt>

```

```

<dd>- black hot drink</dd>
<dt>Milk</dt>
<dd>- white cold drink</dd>
</dl>
</body>
</html>

```

16.4 Nested HTML Lists

List can be nested (lists inside lists).

Nested Lists:

```

<!DOCTYPE html>
<html>
  <body>
    <h2>A Nested List</h2>
    <ul>
      <li>Coffee</li>
      <li>Tea
        <ul>
          <li>Black tea</li>
          <li>Green tea</li>
        </ul>
      </li>
      <li>Milk</li>
    </ul>
  </body>
</html>

```

List items can contain new list, and other HTML elements, like images and links, etc.

16.5 Horizontal Lists

HTML lists can be styled in many different ways with CSS. One popular way, is to style a list to display horizontally:

Horizontal List:

```

<!DOCTYPE html>
<html>
  <head>
    <style>
      ul#menu li {
        display:inline;

```

```

        }
    </style>
</head>
<body>
    <h2>Horizontal List</h2>
    <ul id="menu">
        <li>Apples</li>
        <li>Bananas</li>
        <li>Lemons</li>
        <li>Oranges</li>
    </ul>
</body>
</html>

```

With a little extra style, you can make it look like a menu:

[Tables](#)

[Lists](#)

[Blocks](#)

[Classes](#)

New Style:

```

<!DOCTYPE html>
<html>
    <head>
        <style>
            ul#menu {
                padding: 0;
            }

            ul#menu li {
                display: inline;
            }

            ul#menu li a {
                background-color: black;
                color: white;
                padding: 10px 20px;
                text-decoration: none;
                border-radius: 4px 4px 0 0;
            }

            ul#menu li a:hover {
                background-color: orange;
            }
        </style>
    </head>
    <body>
        <h2>Horizontal List</h2>
        <ul id="menu">
            <li><a href="html_tables.html">Tables</a></li>
            <li><a href="html_lists.html">Lists</a></li>
        </ul>
    </body>
</html>

```

```

</li><a href="html_blocks.html">Blocks</a></li>
</li><a href="html_classes.html">Classes</a></li>
</ul>
</body>
</html>

```

Chapter Summary

- Use the HTML **** element to define an unordered list
- Use the HTML **style** attribute to define the bullet style
- Use the HTML **** element to define an ordered list
- Use the HTML **type** attribute to define the numbering type
- Use the HTML **** element to define a list item
- Use the HTML **<dl>** element to define a description list
- Use the HTML **<dt>** element to define the description term
- Use the HTML **<dd>** element to define the description data
- Lists can be nested inside lists
- List items can contain other HTML elements
- Use the CSS property **display:inline** to display a list horizontally

HTML List Tags

Tag	Description
	Defines an unordered list
	Defines an ordered list
	Defines a list item
<dl>	Defines a description list
<dt>	Defines the term in a description list
<dd>	Defines the description in a description list

CHAPTER SEVENTEEN

17.0 HTML BLOCK ELEMENTS

London

London is the capital city of England. It is the most populous city in the United Kingdom, with a metropolitan area of over 13 million inhabitants.

Standing on the River Thames, London has been a major settlement for two millennia, its history going back to its founding by the Romans, who named it Londinium.

Example

```
<!DOCTYPE html>
<html>
  <body>
    <div style="background-color:black; color:white; margin:20px;
padding:20px;">
      <h2>London</h2>
      <p>
        London is the capital city of England. It is the most populous city
        in the United Kingdom, with a metropolitan area of over 13
        million inhabitants.
      </p>
      <p>
        Standing on the River Thames, London has been a major
        settlement for two millennia, its history going back to its
        founding by the Romans, who named it Londinium.
      </p>
    </div>
  </body>
</html>
```

17.1 HTML Block Elements and Inline Elements

Most HTML elements are defined as **block level** elements or **inline** elements. Block level elements normally start (and end) with a new line, when displayed in a browser.

Examples: `<h1>, <p>, , <table>`

Inline elements are normally displayed without line breaks.

Examples: `, <td>, <a>, `

17.2 The HTML `<div>` Element

The HTML `<div>` element is a **block level element** that can be used as a container for other HTML elements. The `<div>` element has no special meaning. It has no required attributes, but **style** and **class** are common. Because it is a block level element, the browser will display line breaks before and after it. When used together with CSS, the `<div>` element can be used to style blocks of content.

17.3 The HTML `` Element

The HTML `` element is an **inline element** that can be used as a container for text. The `` element has no special meaning. It has no required attributes, but **style** and **class** are common. Unlike `<div>`, which is formatted with line breaks, the `` element does not have any automatic formatting. When used together with CSS, the `` element can be used to style parts of the text:

Example

```
<!DOCTYPE html>
<html>
  <body>
    <h1>My <span style="color:red">Important</span> Heading</h1>
  </body>
</html>
```

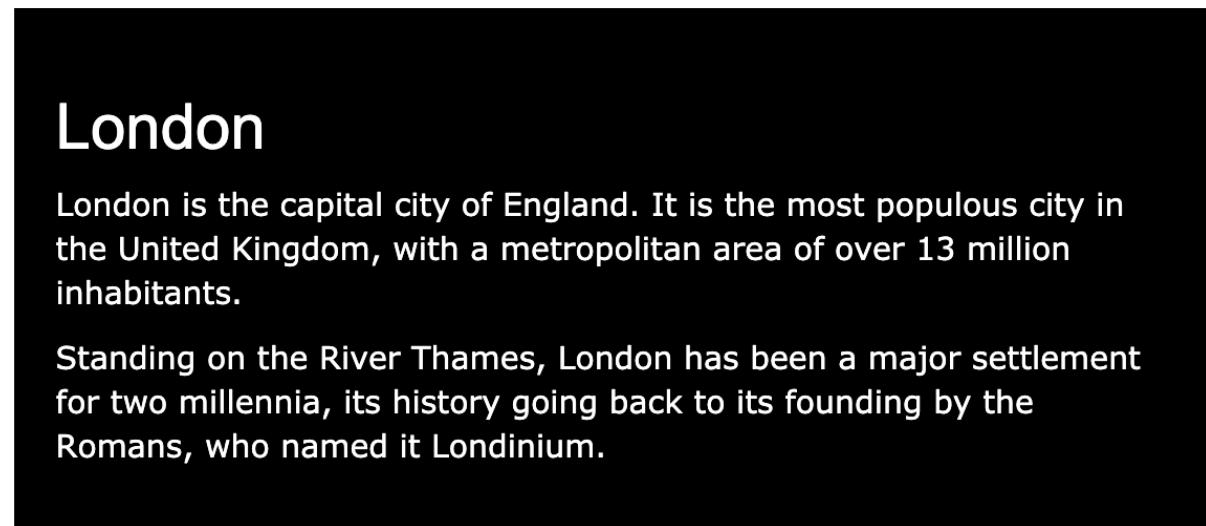
HTML Grouping Tags

Tag	Description
<div>	Defines a section in a document (block-level)
	Defines a section in a document (inline)

CHAPTER EIGHTEEN

HTML CLASSES

Classing HTML elements, makes it possible to define CSS styles for classes of elements. Equal styles for equal classes, or different styles for different classes.



Example

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      .cities {
        background-color:black;
        color:white;
        margin:20px;
        padding:20px;
      }
    </style>
  </head>
  <body>
    <div class="cities">
      <h2>London</h2>
      <p>
        London is the capital city of England. It is the most
        populous city in the United Kingdom, with a
        metropolitan area of over 13 million inhabitants.
      </p>
      <p>
```

Standing on the River Thames, London has been a major settlement for two millennia, its history going back to its founding by the Romans, who named it Londinium.

```
</p>
</div>
</body>
</html>
```

18.1 Classing Block Elements

The HTML `<div>` element is a **block level** element. It can be used as a container for other HTML elements.

Classing `<div>` elements, makes it possible to define equal styles for equal `<div>` elements:

London

London is the capital city of England. It is the most populous city in the United Kingdom, with a metropolitan area of over 13 million inhabitants.

Standing on the River Thames, London has been a major settlement for two millennia, its history going back to its founding by the Romans, who named it Londinium.

Paris

Paris is the capital and most populous city of France.

Situated on the Seine River, it is at the heart of the Île-de-France region, also known as the région parisienne.

Within its metropolitan area is one of the largest population centers in Europe, with over 12 million inhabitants.

Tokyo

Tokyo is the capital of Japan, the center of the Greater Tokyo Area, and the most populous metropolitan area in the world.

It is the seat of the Japanese government and the Imperial Palace, and the home of the Japanese Imperial Family.

The Tokyo prefecture is part of the world's most populous metropolitan area with 38 million people and the world's largest urban economy.

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      .cities {
        background-color:black;
        color:white;
        margin:20px;
        padding:20px;
      }
    </style>
  </head>
  <body>
    <div class="cities">
      <h2>London</h2>
      <p>
        London is the capital city of England. It is the most populous city in the United Kingdom, with a metropolitan area of over 13 million inhabitants.
      </p>
      <p>
        Standing on the River Thames, London has been a major settlement for two millennia, its history going back to its founding by the Romans, who named it Londinium.
      </p>
    </div>
    <div class="cities">
      <h2>Paris</h2>
      <p>Paris is the capital and most populous city of France.</p>
      <p>
        Situated on the Seine River, it is at the heart of the Île-de-France region, also known as the region Parisienne.
      </p>
    </div>
  </body>
</html>
```

```

<p>
    Within its metropolitan area is one of the largest
    population centers in Europe, with over 12 million
    inhabitants.
</p>
</div>
<div class="cities">
    <h2>Tokyo</h2>
    <p>
        Tokyo is the capital of Japan, the center of the Greater
        Tokyo Area, and the most populous metropolitan area in
        the world.
    </p>
    <p>
        It is the seat of the Japanese government and the Imperial
        Palace, and the home of the Japanese Imperial Family.
    </p>
    <p>
        The Tokyo prefecture is part of the world's most
        populous metropolitan area with 38 million people and
        the world's largest urban economy.
    </p>
</div>
</body>
</html>

```

18.2 Classing Inline Elements

The HTML `` element is an inline element that can be used as a container for text.

Classing `` elements makes it possible to design equal styles for equal `` elements.

Example

```

<!DOCTYPE html>
<html>
    <head>
        <style>
            span.red {
                color:red;
            }
        </style>
    </head>
    <body>
        <h1>My <span class="red">Important</span> Heading</h1>
    </body>
</html>

```

CHAPTER NINETEEN

19.0 HTML LAYOUT

Websites often display content in multiple columns (like a magazine or newspaper).

The screenshot shows a dark-themed website layout. At the top, a black header bar contains the title "City Gallery" in white. Below the header, a sidebar on the left lists three cities: "London", "Paris", and "Tokyo". The main content area to the right is titled "London". It contains two paragraphs of text: one describing London as the capital city of England with a population over 13 million, and another paragraph about its history as a major settlement since Roman times. At the bottom of the page, a black footer bar contains the copyright notice "Copyright © iFarouq Tech".

19.1 HTML Layout Using <div> Elements

The `<div>` element is often used as a layout tool, because it can easily be positioned with CSS.

This example uses 4 `<div>` elements to create a multiple column layout:

Example

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      #header {
        background-color:black;
        color:white;
        text-align:center;
        padding:5px;
      }
      #nav {
        line-height:30px;
        background-color:#eeeeee;
        height:300px;
        width:100px;
        float:left;
        padding:5px;
      }
      #content {
        margin-left:110px;
      }
    </style>
  </head>
  <body>
    <div id="header">Header</div>
    <div id="nav">Navigation</div>
    <div id="content">Content</div>
  </body>
</html>
```

```

        }
    #section {
        width:350px;
        float:left;
        padding:10px;
    }
    #footer {
        background-color:black;
        color:white;
        clear:both;
        text-align:center;
        padding:5px;
    }
</style>
</head>
<body>
    <div id="header">
        <h1>City Gallery</h1>
    </div>
    <div id="nav">
        London<br>
        Paris<br>
        Tokyo<br>
    </div>
    <div id="section">
        <h2>London</h2>
        <p>
            London is the capital city of England. It is the most
            populous city in the United Kingdom,
            with a metropolitan area of over 13 million inhabitants.
        </p>
        <p>
            Standing on the River Thames, London has been a major
            settlement for two millennia,
            its history going back to its founding by the Romans, who
            named it Londinium.
        </p>
    </div>
    <div id="footer">
        Copyright © iFarouq Tech
    </div>
</body>
</html>

```

The CSS:

```

<style>
#header {

```

```
background-color:black;  
color:white;  
text-align:center;  
padding:5px;  
}  
  
#nav {  
line-height:30px;  
background-color:#eeeeee;  
height:300px;  
width:100px;  
float:left;  
padding:5px;  
}  
  
#section {  
width:350px;  
float:left;  
padding:10px;  
}  
  
#footer {  
background-color:black;  
color:white;  
clear:both;  
text-align:center;  
padding:5px;
```

```
}
```

```
</style>
```

19.2 Website Layout Using HTML5

HTML5 offers new semantic elements that define different parts of a web page:



header	Defines a header for a document or a section
nav	Defines a container for navigation links
section	Defines a section in a document
article	Defines an independent self-contained article
aside	Defines content aside from the content (like a sidebar)
footer	Defines a footer for a document or a section
details	Defines additional details
summary	Defines a heading for the details element

This example uses `<header>`, `<nav>`, `<section>`, and `<footer>` to create a multiple column layout:

Example

```
<!DOCTYPE html>
<html>
<head>
```

```

<style>
    header {
        background-color:black;
        color:white;
        text-align:center;
        padding:5px;
    }
    nav {
        line-height:30px;
        background-color:#eeeeee;
        height:300px;
        width:100px;
        float:left;
        padding:5px;
    }
    section {
        width:350px;
        float:left;
        padding:10px;
    }
    footer {
        background-color:black;
        color:white;
        clear:both;
        text-align:center;
        padding:5px;
    }
</style>
</head>
<body>
    <header>
        <h1>City Gallery</h1>
    </header>
    <nav>
        London<br>
        Paris<br>
        Tokyo<br>
    </nav>
    <section>
        <h1>London</h1>
        <p>
            London is the capital city of England. It is the most populous city in the United Kingdom, with a metropolitan area of over 13 million inhabitants.
        </p>
        <p>
            Standing on the River Thames, London has been a major settlement for two millennia, its history going back to its founding by the Romans, who named it Londinium.
        </p>
    </section>
</body>

```

```

        </section>
        <footer>
            Copyright © iFarouq Tech
        </footer>
    </body>
</html>

```

The CSS

```

<style>
header {
    background-color:black;
    color:white;
    text-align:center;
    padding:5px;
}
nav {
    line-height:30px;
    background-color:#eeeeee;
    height:300px;
    width:100px;
    float:left;
    padding:5px;
}
section {
    width:350px;
    float:left;
    padding:10px;
}
footer {
    background-color:black;
    color:white;
    clear:both;
    text-align:center;
    padding:5px;
}

```

19.3 HTML Layout Using Tables

The `<table>` element was not designed to be a layout tool. The purpose of the `<table>` element is to display tabular data. Layout can be achieved using the `<table>` element, because table elements can be styled with CSS:

Example

```
<!DOCTYPE html>
```

```

<html>
  <head>
    <style>
      table.lamp {
        width:100%;
        border:1px solid #d4d4d4;
      }
      table.lamp th, td {
        padding:10px;
      }
      table.lamp th {
        width:40px;
      }
    </style>
  </head>
  <body>
    <table class="lamp">
      <tr>
        <th>
          
        </th>
        <td>
          The table element was not designed to be a
          layout tool.
        </td>
      </tr>
    </table>
  </body>
</html>

```

The CSS

```

<style>

table.lamp {
  width:100%;

  border:1px solid #d4d4d4;
}

table.lamp th, td {
  padding:10px;
}

table.lamp td {

```

```
width:40px;  
}  
</style>
```

CHAPTER TWENTY

20.0 HTML RESPONSIVE WEB DESIGN

20.1 What is Responsive Web Design?

- RWD stands for Responsive Web Design
- RWD can deliver web pages in variable sizes
- RWD is a must for tablets and mobile devices

20.2 Creating Your Own Responsive Design

One way to create a responsive design, is to create it yourself:

```
<!DOCTYPE html>
<html lang="en-US">
<head>
<style>
.city {
float: left;
margin: 5px;
padding: 15px;
width: 300px;
height: 300px;
border: 1px solid black;
}
</style>
</head>
<body>
<h1>iFarouq Tech Demo</h1>
<h2>Resize this responsive page!</h2>
<br>
<div class="city">
```

```

<h2>London</h2>
<p>London is the capital city of England.</p>
<p>It is the most populous city in the United Kingdom,
with a metropolitan area of over 13 million inhabitants.</p>
</div>
<div class="city">
<h2>Paris</h2>
<p>Paris is the capital and most populous city of France.</p>
</div>
<div class="city">
<h2>Tokyo</h2>
<p>Tokyo is the capital of Japan, the center of the Greater Tokyo Area,
and the most populous metropolitan area in the world.</p>
</div>
</body>
</html>

```

20.3 Using Bootstrap

Another way to create a responsive design, is to use an already existing CSS framework.

Bootstrap is the most popular HTML, CSS, and JS framework for developing responsive webs.

Bootstrap helps you to develop sites that look nice at any size; screen, laptop, tablet, or phone:

```

<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet" href="http://maxcdn.bootstrapcdn.com/bootstrap/3.2.0/css/bootstrap.min.css">
</head>

<body>
<div class="container">

```

```
<div class="jumbotron">  
  <h1>iFarouq Tech Demo</h1>  
  <p>Resize this responsive page!</p>  
</div>  
  
<div class="row">  
  <div class="col-md-4">  
    <h2>London</h2>  
    <p>London is the capital city of England.</p>  
    <p>It is the most populous city in the United Kingdom,  
      with a metropolitan area of over 13 million inhabitants.</p>  
  </div>  
  <div class="col-md-4">  
    <h2>Paris</h2>  
    <p>Paris is the capital and most populous city of France.</p>  
  </div>  
  <div class="col-md-4">  
    <h2>Tokyo</h2>  
    <p>Tokyo is the capital of Japan, the center of the Greater Tokyo Area,  
      and the most populous metropolitan area in the world.</p>  
  </div>  
</div>  
</body>  
</html>
```

CHAPTER TWENTY-ONE

21.0 HMTL iFRAME

21.1 Definition

An iframe is used to display a web page within a web page. The <iframe> tag specifies an inline frame. An inline frame is used to embed another document within the current HTML document.

21.2 Iframe Syntax

The syntax for adding an iframe is:

```
<iframe src="URL"></iframe>
```

The src attribute specifies the URL (web address) of the iframe page.

21.3 Iframe - Set Height and Width

Use the height and width attributes to specify the size. The attribute values are specified in pixels by default, but they can also be in percent (like "80%").

Example

```
<!DOCTYPE html>
<html>
  <body>
    <iframe src="demo_iframe.html" width="200"
            height="200"></iframe>
  </body>
</html>
```

21.4 Use iframe as a Target for a Link

An iframe can be used as the target frame for a link. The target attribute of the link must refer to the name attribute of the iframe:

Example

```

<!DOCTYPE html>
<html>
  <body>
    <iframe width="100%" height="300px" src="demo_iframe.html"
    name="iframe_a"></iframe>
    <p>
      <a href="../index.html" target="iframe_a">ifaroutech.com</a>
    </p>
    <p>
      When the target of a link matches the name of an iframe, the link
      will open in the iframe.
    </p>
  </body>
</html>

```

HTML iframe Tag

Tag	Description
<u><iframe></u>	Defines an inline frame

CHAPTER TWENTY-TWO

22.0 HTML COLOR NAMES

Colors are displayed combining RED, GREEN, and BLUE light.

22.1 140 Color Names are Supported by All Browsers

140 color names are defined in the HTML5 and CSS3 color specifications. 17 colors are from the HTML specification, 123 colors are from the CSS specification. The table below lists them all, along with their hexadecimal values.

The 17 colors from the HTML specification are: aqua, black, blue, fuchsia, gray, green, lime, maroon, navy, olive, orange, purple, red, silver, teal, white, and yellow.

22.2 Sorted by Color Name

Colors sorted by HEX values

Color Name	HEX	Color
AliceBlue	#FOFBFF	
AntiqueWhite	#FAEBD7	
Aqua	#00FFFF	
Aquamarine	#7FFFDD	
Azure	#F0FFFF	
Beige	#F5F5DC	
Bisque	#FFEAC4	
Black	#000000	
BlanchedAlmond	#FFEBCD	
Blue	#0000FF	
BlueViolet	#8A2BE2	
Brown	#A52A2A	
BurlyWood	#DEB887	
CadetBlue	#5F9EA0	
Chartreuse	#7FFF00	
Chocolate	#D2691E	
Coral	#FF7F50	
CornflowerBlue	#6495ED	
Cornsilk	#FFF8DC	
Crimson	#DC143C	
Cyan	#00FFFF	
DarkBlue	#00008B	
DarkCyan	#008B8B	
DarkGoldenRod	#B8860B	

GoldenRod	#DAA520		LightSlateGray	#778899	
Gray	#808080		LightSteelBlue	#B0C4DE	
Green	#008000		LightYellow	#FFFFE0	
GreenYellow	#ADFF2F		Lime	#00FF00	
HoneyDew	#FFF0FF		LimeGreen	#32CD32	
HotPink	#FF69B4		Linen	#FAF0E6	
IndianRed	#CD5C5C		Magenta	#FF00FF	
Indigo	#4B0082		Maroon	#800000	
Ivory	#FFFFF0		MediumAquaMarine	#66CDAA	
Khaki	#F0E68C		MediumBlue	#0000CD	
Lavender	#DDE6FD		MediumOrchid	#BA55D3	
LavenderBlush	#FFF0F5		MediumPurple	#9370DB	
LawnGreen	#7CFC00		MediumSeaGreen	#3CB371	
LemonChiffon	#FFFACD		MediumSlateBlue	#7B68EE	
LightBlue	#ADD8E6		MediumSpringGreen	#00FA9A	
LightCoral	#F08080		MediumTurquoise	#48D1CC	
LightCyan	#E0FFFF		MediumVioletRed	#C71585	
LightGoldenRodYellow	#FAFAD2		MidnightBlue	#191970	
LightGray	#D3D3D3		MintCream	#F5FFFA	
LightGreen	#90EE90		MistyRose	#FFE4E1	
LightPink	#FFB6C1		Moccasin	#FFE4B5	
LightSalmon	#FFA07A		NavajoWhite	#FFDEAD	
LightSeaGreen	#20B2AA		Navy	#000080	
LightSkyBlue	#87CEFA		OldLace	#FDF5E6	
LightSlateGray	#778899		Olive	#808000	

OliveDrab	#6B8BE2		SeaGreen	#2E8B57	
Orange	#FFA500		SeaShell	#FFF5EE	
OrangeRed	#FF4500		Sienna	#A0522D	
Orchid	#DA70D6		Silver	#C0C0C0	
PaleGoldenRed	#EEE8AA		SkyBlue	#87CEEB	
PaleGreen	#98FB98		SlateBlue	#6A5ACD	
PaleTurquoise	#AFEEEE		SlateGray	#708090	
PaleVioletRed	#DB7093		Snow	#FFFafa	
PapayaWhip	#FFFED5		SpringGreen	#00FF7F	
PeachPuff	#FFDAB9		SteelBlue	#4682B4	
Peru	#CD853F		Tan	#D2B48C	
Pink	#FFC0CB		Teal	#008080	
Plum	#DDA0DD		Thistle	#D8BFD8	
PowderBlue	#B0E0E6		Tomato	#FF6347	
Purple	#800080		Turquoise	#40E0D0	
RebeccaPurple	#663399		Violet	#EE82EE	
Red	#FF0000		Wheat	#FSDEB3	
RosyBrown	#BC8F8F		White	#FFFFFF	
RoyalBlue	#4169E1		WhiteSmoke	#F5F5F5	
SaddleBrown	#8B4513		Yellow	#FFFF00	
Salmon	#FA8072		YellowGreen	#9ACD32	
SandyBrown	#F4A460				
SeaGreen	#2E8B57				
SeaShell	#FFF5EE				
Sienna	#A0522D				

CHAPTER TWENTY-THREE

23.0 COLOR VALUES

23.1 HTML Color Values

Colors are displayed combining RED, GREEN, and BLUE light. Colors are defined using a hexadecimal (hex) notation for the Red, Green, and Blue values (RGB). The lowest value for each light source is 0 (hex 00). The highest value is 255 (hex FF). Hex values are written as # followed by either three or six hex characters.

Three-digit notations (#rgb) are automatically converted to six digits (#rrggb):

Color	Color 3 digit HEX	Color 6 digit HEX	Color RGB
Red	#F00	#FF0000	rgb(255,0,0)
Green	#0F0	#00FF00	rgb(0,255,0)
Blue	#00F	#0000FF	rgb(0,0,255)

```
<!DOCTYPE html>
<html>
  <body>
    <h2 style="background-color:#FFFF00">
      Color set by using HEX
    </h2>
    <h2 style="background-color:rgb(255,255,0)">
      Color set by using RGB
    </h2>
    <h2 style="background-color:yellow">
      Color set by using Name
    </h2>
  </body>
</html>
```

Shades of grey (from black to white) are defined using equal values for all the 3 light sources:

Color	Color 3 digit HEX	Color 6 digit HEX	Color RGB
Black	#000	#000000	rgb(0,0,0)
Grey	#888	#888888	rgb(136,136,136)
White	#FFF	#FFFFFF	rgb(255,255,255)

```
<!DOCTYPE html>
<html>
  <body>
    <h2 style="background-color:#000000;color:white">
      Color set by using #000000
    </h2>
    <h2 style="background-color:rgb(0,0,0);color:white">
      Color set by using rgb(0,0,0)
    </h2>
    <h2 style="background-color:black;color:white">
      Color set by using black
    </h2>
    <br>
    <h2 style="background-color:#808080;color:white">
      Color set by using #808080
    </h2>
    <h2 style="background-color:rgb(128,128,128);color:white">
      Color set by using rgb(128,128,128)
    </h2>
    <h2 style="background-color:gray;color:white">
      Color set by using gray
    </h2>
    <br>
    <h2 style="background-color:#D3D3D3;color:white">
      Color set by using #D3D3D3
    </h2>
    <h2 style="background-color:rgb(211,211,211);color:white">
      Color set by using rgb(211,211,211)
    </h2>
    <h2 style="background-color:lightgray;color:white">
      Color set by using lightgray
    </h2>
  </body>
</html>
```

23.2 Colors Sorted by HEX Value

Colors sorted by color name

Color Name	HEX	Color	Color Name	HEX	Color
Black	#000000		RoyalBlue	#4169E1	
Navy	#000080		SteelBlue	#4682B4	
DarkBlue	#00008B		DarkSlateBlue	#483D8B	
MediumBlue	#0000CD		MediumTurquoise	#483DCC	
Blue	#0000FF		Indigo	#480082	
DarkGreen	#006400		DarkOliveGreen	#556B2F	
Green	#008000		CadetBlue	#5F9EA0	
Teal	#008080		CornflowerBlue	#6495ED	
DarkCyan	#008B8B		RebeccaPurple	#663399	
DeepSkyBlue	#00BFFF		MediumAquaMarine	#66CDA8	
DarkTurquoise	#00CED1		DimGray	#696969	
MediumSpringGreen	#00FA9A		SlateBlue	#6A5ACD	
Lime	#00FF00		OliveDrab	#6B8BE2	
SpringGreen	#00FF7F		SlateGray	#708090	
Aqua	#00FFFF		LightSlateGray	#778899	
Cyan	#00FFFF		MediumSlateBlue	#7B68EE	
MidnightBlue	#191970		LawnGreen	#7CFC00	
DodgerBlue	#1E90FF		Chartreuse	#7FFF00	
LightSeaGreen	#20B2AA		Aquamarine	#7FFF40	
ForestGreen	#228B22		Maroon	#800000	
SeaGreen	#2E8B57		Purple	#800080	
DarkSlateGray	#2F4F4F		Olive	#808000	
LimeGreen	#32CD32		Gray	#808080	
MediumSeaGreen	#3CB371		SkyBlue	#87CEEB	
Turquoise	#40E0D0		LightSkyBlue	#87CEFA	
			BlueViolet	#8A2BE2	

DarkRed	#8B0000		Peru	#CD853F	
DarkMagenta	#8B008B		Chocolate	#D2691E	
SaddleBrown	#8B4513		Tan	#D2B48C	
DarkSeaGreen	#8FBBCF		LightGray	#D3D3D3	
LightGreen	#90EE90		Thistle	#D8bfd8	
MediumPurple	#9370DB		Orchid	#DA70D6	
DarkViolet	#9400D3		GoldenRod	#DAA520	
PaleGreen	#98FB98		PaleVioletRed	#DB7093	
DarkOrchid	#9932CC		Crimson	#DC143C	
YellowGreen	#9ACD32		Gainsboro	#D3D3D3	
Sienna	#A0522D		Plum	#DDA0DD	
Brown	#A52A2A		BurlyWood	#DEB887	
DarkGray	#A9A9A9		LightCyan	#E0FFFF	
LightBlue	#ADD8E6		Lavender	#E6E6FA	
GreenYellow	#ADFF2F		DarkSalmon	#F9967A	
PaleTurquoise	#AFEEEE		Violet	#EE82EE	
LightSteelBlue	#B0C4DE		PaleGoldenRod	#EEE8AA	
PowderBlue	#B0E0E6		LightCoral	#F08080	
FireBrick	#B22222		Khaki	#F0E68C	
DarkGoldenRod	#B8860B		AliceBlue	#F0F8FF	
MediumOrchid	#BA55D3		HoneyDew	#F0FFF0	
RosyBrown	#BC8F8F		Azure	#F0FFFF	
DarkKhaki	#BDB76B		SandyBrown	#F4A460	
Silver	#C0C0C0		Wheat	#F5DEB3	
MediumVioletRed	#C71585		Beige	#F5F5DC	
IndianRed	#CD5C5C		WhiteSmoke	#F5F5F5	

MintCream	#FFFFFA		
GhostWhite	#F8F8FF		
Salmon	#FA8072		
AntiqueWhite	#FAEBD7		
Linen	#FAF0E6		
LightGoldenRodYellow	#FAFAD2		
OldLace	#FDF5E6		
Red	#FF0000		
Fuchsia	#FF00FF		
Magenta	#FF00FF		
DeepPink	#FF1493		
OrangeRed	#FF4500		
Tomato	#FF6347		
HotPink	#FF69B4		
Coral	#FF7F50		
DarkOrange	#FF8C00		
LightSalmon	#FFA07A		
Orange	#FFA500		
LightPink	#FFB6C1		
Pink	#FFC0CB		
Gold	#FFD700		
PeachPuff	#FFDAB9		
NavajoWhite	#FFDEAD		
Moccasin	#FFE4B5		
Bisque	#FFE4C4		
MistyRose	#FFF4E1		
BlanchedAlmond	#FFFBCD		
PapayaWhip	#FFEED5		
LavenderBlush	#FFF0F5		
SeaShell	#FFFSEE		
Cornsilk	#FFF8DC		
LemonChiffon	#FFFACD		
FloralWhite	#FFFAFO		
Snow	#FFFafa		
Yellow	#FFFF00		
LightYellow	#FFFFE0		
Ivory	#FFFFFF		
White	#FFFFFF		

CHAPTER TWENTY-FOUR

24.0 HTML COLOR SHADES

Colors are displayed combining RED, GREEN, and BLUE light.

24.1 Shades of Gray

Gray colors are displayed using an equal amount of power to all of the light sources.

To make it easy for you to select a gray color we have compiled a table of gray shades for you:

Gray Shades	HEX	RGB
	#000000	rgb(0,0,0)
	#080808	rgb(8,8,8)
	#101010	rgb(16,16,16)
	#181818	rgb(24,24,24)
	#202020	rgb(32,32,32)
	#282828	rgb(40,40,40)
	#303030	rgb(48,48,48)
	#383838	rgb(56,56,56)
	#404040	rgb(64,64,64)
	#484848	rgb(72,72,72)
	#505050	rgb(80,80,80)
	#585858	rgb(88,88,88)
	#606060	rgb(96,96,96)
	#686868	rgb(104,104,104)
	#707070	rgb(112,112,112)
	#787878	rgb(120,120,120)
	#808080	rgb(128,128,128)
	#888888	rgb(136,136,136)

24.2 16 Million Different Colors

The combination of Red, Green and Blue values from 0 to 255 gives a total of more than 16 million different colors to play with ($256 \times 256 \times 256$). Most modern monitors are capable of

displaying at least 16384 different colors. If you look at the color table below, you will see the result of varying the red light from 0 to 255, while keeping the green and blue light at zero.

To see a full list of color mixes when the red light varies from 0 to 255, click on one of the hex or rgb values below.

Red Light	HEX	RGB
	#000000	rgb(0,0,0)
	#080000	rgb(8,0,0)
	#100000	rgb(16,0,0)
	#180000	rgb(24,0,0)
	#200000	rgb(32,0,0)
	#280000	rgb(40,0,0)
	#300000	rgb(48,0,0)
	#380000	rgb(56,0,0)
	#400000	rgb(64,0,0)
	#480000	rgb(72,0,0)
	#500000	rgb(80,0,0)
	#580000	rgb(88,0,0)
	#600000	rgb(96,0,0)
	#680000	rgb(104,0,0)
	#700000	rgb(112,0,0)
	#780000	rgb(120,0,0)
	#800000	rgb(128,0,0)
	#880000	rgb(136,0,0)

	<u>#800000</u>	<u>rgb(128,0,0)</u>
	<u>#880000</u>	<u>rgb(136,0,0)</u>
	<u>#900000</u>	<u>rgb(144,0,0)</u>
	<u>#980000</u>	<u>rgb(152,0,0)</u>
	<u>#A00000</u>	<u>rgb(160,0,0)</u>
	<u>#A80000</u>	<u>rgb(168,0,0)</u>
	<u>#B00000</u>	<u>rgb(176,0,0)</u>
	<u>#B80000</u>	<u>rgb(184,0,0)</u>
	<u>#C00000</u>	<u>rgb(192,0,0)</u>
	<u>#C80000</u>	<u>rgb(200,0,0)</u>
	<u>#D00000</u>	<u>rgb(208,0,0)</u>
	<u>#D80000</u>	<u>rgb(216,0,0)</u>
	<u>#E00000</u>	<u>rgb(224,0,0)</u>
	<u>#E80000</u>	<u>rgb(232,0,0)</u>
	<u>#F00000</u>	<u>rgb(240,0,0)</u>
	<u>#F80000</u>	<u>rgb(248,0,0)</u>
	<u>#FF0000</u>	<u>rgb(255,0,0)</u>

24.3 In the Stone Age

In the stone age, when computers only supported 256 different colors, a list of 216 "Web Safe Colors" was suggested as a Web standard, reserving 40 fixed system colors. These 216 cross-browser color palettes were created to ensure that all computers would display colors correctly:

000000	000033	000066	000099	0000CC	0000FF
003300	003333	003366	003399	0033CC	0033FF
006600	006633	006666	006699	0066CC	0066FF
009900	009933	009966	009999	0099CC	0099FF
00CC00	00CC33	00CC66	00CC99	00CCCC	00CCFF
00FF00	00FF33	00FF66	00FF99	00FFCC	00FFFF
330000	330033	330066	330099	3300CC	3300FF
333300	333333	333366	333399	3333CC	3333FF
336600	336633	336666	336699	3366CC	3366FF
339900	339933	339966	339999	3399CC	3399FF
33CC00	33CC33	33CC66	33CC99	33CCCC	33CCFF
33FF00	33FF33	33FF66	33FF99	33FFCC	33FFFF
660000	660033	660066	660099	6600CC	6600FF
663300	663333	663366	663399	6633CC	6633FF
666600	666633	666666	666699	6666CC	6666FF
669900	669933	669966	669999	6699CC	6699FF
66CC00	66CC33	66CC66	66CC99	66CCCC	66CCFF
66FF00	66FF33	66FF66	66FF99	66FFCC	66FFFF
990000	990033	990066	990099	9900CC	9900FF

993300	993333	993366	993399	9933CC	9933FF
996600	996633	996666	996699	9966CC	9966FF
999900	999933	999966	999999	9999CC	9999FF
99CC00	99CC33	99CC66	99CC99	99CCCC	99CCFF
99FF00	99FF33	99FF66	99FF99	99FFCC	99FFFF
CC0000	CC0033	CC0066	CC0099	CC00CC	CC00FF
CC3300	CC3333	CC3366	CC3399	CC33CC	CC33FF
CC6600	CC6633	CC6666	CC6699	CC66CC	CC66FF
CC9900	CC9933	CC9966	CC9999	CC99CC	CC99FF
CCCC00	CCCC33	CCCC66	CCCC99	CCCCCC	CCCCFF
CCFF00	CCFF33	CCFF66	CCFF99	CCFFCC	CCFFFF
FF0000	FF0033	FF0066	FF0099	FF00CC	FF00FF
FF3300	FF3333	FF3366	FF3399	FF33CC	FF33FF
FF6600	FF6633	FF6666	FF6699	FF66CC	FF66FF
FF9900	FF9933	FF9966	FF9999	FF99CC	FF99FF
FFCC00	FFCC33	FFCC66	FFCC99	FFCCCC	FFCCFF
FFFF00	FFFF33	FFFF66	FFFF99	FFFFCC	FFFFFF

CHAPTER TWENTY-FIVE

25.0 HTML SCRIPT

25.1 Script Definition

The `<script>` tag is used to define a client-side script, such as a JavaScript. The `<script>` element either contains scripting statements, or it points to an external script file through the `src` attribute. Common uses for JavaScript are image manipulation, form validation, and dynamic changes of content.

25.2 Noscript Definition

The `<noscript>` tag defines an alternate content for users that have disabled scripts in their browser or have a browser that doesn't support script. The `<noscript>` element can be used in both `<head>` and `<body>`. When used inside the `<head>` element: `<noscript>` must contain `<link>`, `<style>`, and `<meta>` elements. The content inside the `<noscript>` element will be displayed if scripts are not supported, or are disabled in the user's browser.

JavaScript make HTML pages more dynamic and interactive.

24.3 The HTML `<script>` Tag

The `<script>` tag is used to define a client-side script, such as a JavaScript. The `<script>` element either contains scripting statements or it points to an external script file through the `src` attribute.

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

The script below writes Hello JavaScript! into an HTML element with `id="demo"`:

Example

```
<!DOCTYPE html>
<html>
  <body>
    <p id="demo"></p>
    <script>
      document.getElementById("demo").innerHTML = "Hello
      JavaScript!";
    </script>
  </body>
</html>
```

24.4 The HTML <noscript> Tag

The <noscript> tag is used to provide an alternate content for users that have disabled scripts in their browser or have a browser that doesn't support client-side scripting. The <noscript> element can contain all the elements that you can find inside the <body> element of a normal HTML page.

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

Example

```
<!DOCTYPE html>
<html>
  <body>
    <p id="demo"></p>
    <script>
      document.getElementById("demo").innerHTML = "Hello
      JavaScript!";
    </script>
    <noscript>Sorry, your browser does not support JavaScript!</noscript>
    <p>
      A browser without support for JavaScript will show the text
      written inside the noscript element.
    </p>
  </body>
</html>
```

24.5 A Taste of JavaScript

Here are some examples of what JavaScript can do, **JavaScript can change HTML content:**

```
<!DOCTYPE html>

<html>

    <body>

        <h1>My First JavaScript</h1>

        <p>JavaScript can change the content of an HTML element:</p>

        <button type="button" onclick="myFunction()">Click Me!</button>

        <p id="demo">This is a demonstration.</p>

        <script>

            function myFunction() {

                document.getElementById("demo").innerHTML      =      "Hello
                JavaScript!";

            }

        </script>

    </body>

</html>
```

JavaScript can change HTML styles:

```
<!DOCTYPE html>

<html>

    <body>

        <h1>My First JavaScript</h1>

        <p id="demo">JavaScript can change the style of an HTML
        element.</p>

        <script>
```

```

        function myFunction() {
            document.getElementById("demo").style.fontSize = "25px";
        }
    </script>
<button type="button" onclick="myFunction()">Click Me!</button>
</body>
</html>
```

JavaScript can change HTML attributes:

```

<!DOCTYPE html>
<html>
<body>
<script>
function light(sw) {
    var pic;
    if (sw == 0) {
        pic = "pic_bulboff.gif"
    } else {
        pic = "pic_bulbon.gif"
    }
    document.getElementById('myImage').src = pic;
}
</script>

<p>
```

```
<button type="button" onclick="light(1)">Light On</button>  
<button type="button" onclick="light(0)">Light Off</button>  
</p>  
</body>  
</html>
```

HTML Script Tags

Tag	Description
<code><script></code>	Defines a client-side script
<code><noscript></code>	Defines an alternate content for users that do not support client-side scripts

CHAPTER TWENTY-FIVE

25.0 HTML HEAD

25.1 The HTML <head> Element

The <head> element is a container for meta data (data about data). HTML meta data is data about the HTML document. Metadata is not displayed. Meta data typically define document title, styles, links, scripts, and other meta information. The following tags describes meta data: <title>, <style>, <meta>, <link>, <script>, and <base>.

25.2 Omitting <html> and <body>

In the HTML5 standard, the <html> tag, the <body> tag, and the <head> tag can be omitted.

The following code will validate as HTML5:

Example

```
<!DOCTYPE html>

<title>Page Title</title>

<h1>This is a heading</h1>

<p>This is a paragraph.</p>
```

iFarouq Tech does not recommend omitting the <html> and <body> tags:

The <html> element is the document root. It is the recommended place for specifying the page language:

```
<!DOCTYPE html>

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

Declaring a language is important for accessibility applications (screen readers) and search engines.

Omitting `<html>` and `<body>` can crash badly written DOM and XML software.

Finally, omitting `<body>` can produce errors in older browsers (IE9).

25.3 Omitting `<head>`

In the HTML5 standard, the `<head>` tag can also be omitted. By default, browsers will add all elements before `<body>`, to a default `<head>` element.

You can reduce the complexity of HTML, by omitting the `<head>` tag:

Example

```
<!DOCTYPE html>

<html>

    <title>Page Title</title>

    <body>

        <h1>This is a heading</h1>

        <p>This is a paragraph.</p>

    </body>

</html>
```

Omitting tags is unfamiliar to web developers. It needs time to be established as a guideline.

25.4 The HTML `<title>` Element

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

A simplified HTML document:

Example

```
<!DOCTYPE html>

<html>

    <title>Page Title</title>

    <body>

        <p>The content of the body element is displayed in the browser window.</p>

        <p>The content of the title element is not displayed.</p>

    </body>

</html>
```

25.5 The HTML <style> Element

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

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

Example

```
<!DOCTYPE html>

<html>

    <title>Page Title</title>
```

```
<style>
    body{
        background-color:yellow;
    }
    p{
        color:blue;
    }
</style>

<body>
    <h1>This is a heading</h1>
    <p>This is a paragraph.</p>
</body>
</html>
```

25.6 The HTML <link> Element

The <link> element defines the page relationship to an external resource.

The <link> element is most often used to link to style sheets:

Example

```
<!DOCTYPE html>
<html>
    <title>Page Title</title>
    <link rel="stylesheet" href="mystyle.css">
    <body>
        <h1>This is a heading</h1>
        <p>This is a paragraph.</p>
```

```
</body>  
</html>
```

25.7 The HTML <meta> Element

The <meta> element is used to specify page description, keywords, author, and other metadata. Meta data is used by browsers (how to display content), by search engines (keywords), and other web services.

- Define keywords for search engines:

Example

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

- Define a description of your web page:

Example

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

- Define the character set used:

Example

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

- Define the author of a page:

Example

```
<meta name="author" content="iFarouq">
```

Example

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

```

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

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

<meta name="author" content="Hege Refsnes">

<meta charset="UTF-8">

<meta charset="UTF-8">

<body>

<p>All meta information goes before the body.</p>

</body>

</html>

```

Refresh document every 30 seconds:

Example

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

25.8 The HTML <script> Element

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

The script below writes Hello JavaScript! into an HTML element with `id="demo"`:

Example

```

<!DOCTYPE html>

<html>

<title>Page Title</title>

<script>

function myFunction() {

    document.getElementById("demo").innerHTML = "Hello
    JavaScript!";
}

```

```

        }

</script>

<body>

    <h1>My Web Page</h1>

    <p id="demo">A Paragraph</p>

    <button type="button" onclick="myFunction()">Try it</button>

</body>

</html>

```

To learn all about JavaScript, visit our [JavaScript Tutorial!](#)

25.9 The HTML <base> Element

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

Example

```

<!DOCTYPE html>

<html>

    <title>Page Title</title>

    <base target="_blank">

    <body>

        <p>

```

Since we have specified a base URL, the browser will look for
the image "html5.gif" at
["http://www.example.com/images/html5.gif"](http://www.example.com/images/html5.gif)

```
</p>
```

```
<p>
```

```

<a href="../index.html">iFarouq Tech</a>

</p>

<p>

The link above opens in a new window. This is because the base
target is set to "_blank".

</p>

</body>

</html>

```

HTML head Elements

Tag	Description
<u><head></u>	Defines information about the document
<u><title></u>	Defines the title of a document
<u><base></u>	Defines a default address or a default target for all links on a page
<u><link></u>	Defines the relationship between a document and an external resource
<u><meta></u>	Defines metadata about an HTML document
<u><script></u>	Defines a client-side script
<u><style></u>	Defines style information for a document

CHAPTER TWENTY-SIX

26.0 HTML ENTITIES

Reserved characters in HTML must be replaced with character entities. Characters, not present on your keyboard, can also be replaced by entities.

26.1 HTML Entities

Some characters are reserved in HTML. If you use the less than (<) or greater than (>) signs in your text, the browser might mix them with tags. Character entities are used to display reserved characters in HTML.

A character entity looks like this:

`&entity_name;`

OR

`&#entity_number;`

To display a less than sign we must write: `<` or `<`

The advantage of using an entity name, instead of a number, is that the name is easier to remember.

The disadvantage is that browsers may not support all entity names, but the support for numbers is good.

26.2 Non-Breaking Space

A common character entity used in HTML is the non-breaking space (). Remember that browsers will always truncate spaces in HTML pages. If you write 10 spaces in your text, the

browser will remove 9 of them. To add real spaces to your text, you can use the character entity.

Some Other Useful HTML Character Entities

Result	Description	Entity Name	Entity Number
	non-breaking space	 	
<	less than	<	<
>	greater than	>	>
&	ampersand	&	&
¢	cent	¢	¢
£	pound	£	£
¥	yen	¥	¥
€	euro	€	€
©	copyright	©	©
®	registered trademark	®	®

Entity names are case sensitive.

26.3 Combining Diacritical Marks

A diacritical mark is a "glyph" added to a letter. Some diacritical marks, like grave (`) and acute (') are called accents. Diacritical marks can appear both above and below a letter, inside a letter, and between two letters. Diacritical marks can be used in combination with alphanumeric characters, to produce a character that is not present in the character set (encoding) used in the page.

Mark	Character	Construct	Result
`	a	à	à
'	a	á	á
^	a	â	â
~	a	ã	ã
`	o	Ò	ò
'	o	Ó	ó
^	o	Ô	ô
~	o	Õ	õ

You will see more HTML symbols in the next chapter of this tutorial.

CHAPTER TWENTY-SEVEN

27.0 HTML SYMBOLS

27.1 HTML Symbol Entities

HTML entities were described in the previous chapter. Many mathematical, technical, and currency symbols, are not present on a normal keyboard. To add these symbols to an HTML page, you can use an HTML entity name. If no entity name exists, you can use an entity number; a decimal (or hexadecimal) reference.

If you use an HTML entity name or a hexadecimal number, the character will always display correctly. This is independent of what character set (encoding) your page uses!

Will display as:

```
<!DOCTYPE html>
<html>
  <body>
    <p>I will display &euro;</p>
    <p>I will display &#8364;</p>
    <p>I will display &#x20AC;</p>
  </body>
</html>
```

27.2 Some Mathematical Symbols Supported by HTML

Char	Number	Entity	Description
∀	∀	∀	FOR ALL
∂	∂	∂	PARTIAL DIFFERENTIAL
∃	∃	∃	THERE EXISTS
∅	∅	∅	EMPTY SETS
∇	∇	∇	NABLA
∈	∈	∈	ELEMENT OF
∉	∉	∉	NOT AN ELEMENT OF
∋	∋	∋	CONTAINS AS MEMBER
∏	∏	∏	N-ARY PRODUCT
Σ	∑	∑	N-ARY SUMMATION

UTF-8 Mathematical Operators

Range: Decimal 8704-8959. Hex 2200-22FF.

If you want any of these characters displayed in HTML, you can use the HTML entity found in the table below. If the character does not have an HTML entity, you can use the decimal (dec) or hexadecimal (hex) reference.

Example

Will display as:

```
<!DOCTYPE html>
<html>
  <body>
    <p>I will display &sum;</p>
    <p>I will display &#8721;</p>
    <p>I will display &#x2211;</p>
  </body>
```

</html>

Note: Some browsers may not support all HTML5 entities in the table below.

Currently, only IE 11 and Firefox 35 support all HTML5 entities.

Char	Dec	Hex	Entity	Name	Char	Dec	Hex	Entity	Name	Char	Dec	Hex	Entity	Name
v	8704	2200	M <small>athfrak{v}</small>	FOR ALL	v	8743	2227	&band;	LOGICAL AND	v	8782	2246	GEOMETRICALLY EQUIVALENT TO	
c	8705	2201	C <small>OMPLEMENT</small>		v	8744	2228	∨	LOGICAL OR	v	8783	2247	DIFFERENCE BETWEEN	
d	8706	2202	A <small>pert</small>	PARTIAL DIFFERENTIAL	v	8745	2229	&	INTERSECTION	v	8784	2250	APPROACHES THE LIMIT	
x	8707	2203	B <small>exist</small>	THERE EXISTS	v	8746	222A	∪	UNION	v	8785	2251	GEOMETRICALLY EQUAL TO	
#	8708	2204	B <small>exists</small>	THERE DOES NOT EXIST	f	8747	222B	∫	INTEGRAL	v	8786	2252	APPROXIMATELY EQUAL TO OR THE IMAGE OF	
o	8709	2205	B <small>empty</small>	EMPTY SET	f	8748	222C	&doubleint;	DOUBLE INTEGRAL	v	8787	2253	IMAGE OF OR APPROXIMATELY EQUAL TO	
Δ	8710	2206	I <small>ncrement</small>		f	8749	222D	&tripleint;	TRIPLE INTEGRAL	v	8788	2254	COLON EQUALS	
V	8711	2207	B <small>bbra</small>	NABLA	f	8750	222E	&contourint;	CONTOUR INTEGRAL	v	8789	2255	EQUALS COLON	
c	8712	2208	B <small>bigrm</small>	ELEMENT OF	f	8751	222F	&surfaceint;	SURFACE INTEGRAL	v	8790	2256	RING IN EQUAL TO	
#	8713	2209	B <small>bnotin</small>	NOT AN ELEMENT OF	f	8752	2230	&volumeint;	VOLUME INTEGRAL	v	8791	2257	RING EQUAL TO	
x	8714	220A	B <small>smallrm</small>	SMALL ELEMENT OF	f	8753	2231	&clockwiseint;	CLOCKWISE INTEGRAL	v	8792	2258	COINCIDES WITH	
#	8715	220B	B <small>smallit</small>	CONTAINS AS MEMBER	f	8754	2232	&anticlockwiseint;	COUNTERCLOCKWISE INTEGRAL	v	8793	2259	ESTIMATES	
#	8716	220C	B <small>smallit</small>	DOES NOT CONTAIN AS MEMBER	f	8755	2233	∴	THE THEREFORE	v	8794	225A	EQUANGULAR TO	
-	8717	220D	B <small>smallit</small>	SMALL CONTAINS AS MEMBER	f	8756	2234	∵	BECAUSE	v	8795	225B	STAR EQUALS	
▪	8718	220E	E <small>ND</small> OF <small>PROOF</small>		f	8757	2235	&endofproof;	END OF PROOF	v	8796	225C	DELTA EQUAL TO	
Π	8719	220F	N <small>ARY</small> PRODUCT		f	8758	2236	∶	RATIO	v	8797	225D	EQUAL TO BY DEFINITION	
Π	8720	2210	N <small>ARY</small> CPRODUCT		f	8759	2237	&proportion;	PROPORTION	v	8798	225E	MEASURED BY	
Σ	8721	2211	B <small>sum</small>	N <small>ARY</small> SUMMATION	f	8760	2238	∸	DOT MINUS	v	8799	225F	QUESTIONED EQUAL TO	
-	8722	2212	B <small>minus</small>	MINUS SIGN	f	8761	2239	!	EXCESS	v	8800	2260	NOT EQUAL TO	
⊖	8723	2213	B <small>minusplus</small>	MINUS-OR-PLUS SIGN	f	8762	223A	&geometricproportion;	GEOMETRIC PROPORTION	v	8801	2261	BEQUIV;	
÷	8724	2214	D <small>O</small> T P <small>LUS</small>	DOT PLUS	f	8763	223B	&homothetic;	HOMOTHETIC	v	8802	2262	NOT IDENTICAL TO	
/	8725	2215	D <small>IVISION</small> S <small>LASH</small>		f	8764	223C	˜	TITLE OPERATOR	v	8803	2263	STRICTLY EQUIVALENT TO	
¬	8726	2216	S <small>ET</small> M <small>INUS</small>	SET MINUS	f	8765	223D	&reversedtilde;	REVERSED TILDE	v	8804	2264	LESS-THAN OR EQUAL TO	
+*	8727	2217	B <small>lowast</small>	ASTERISK OPERATOR	f	8766	223E	&invertedlazy;	INVERTED LAZY'S	v	8805	2265	GREATER-THAN OR EQUAL TO	
-	8728	2218	K <small>ING</small> O <small>PERATOR</small>		f	8767	223F	&sine;	SINE WAVE	v	8806	2266	LESS-THAN OVER EQUAL TO	
·	8729	2219	B <small>ullet</small> O <small>PERATOR</small>		f	8768	2240	≀	WREATH PRODUCT	v	8807	2267	GREATER-THAN OVER EQUAL TO	
✓	8730	221A	B <small>radic</small>	SQUARE ROOT	f	8769	2241	¬tilde;	NOT TILDE	v	8808	2268	LESS-THAN BUT NOT EQUAL TO	
√	8731	221B	C <small>URE</small> R <small>OOT</small>	CUBE ROOT	f	8770	2242	&minustilde;	MINUS TILDE	v	8809	2269	GREATER-THAN BUT NOT EQUAL TO	
∜	8732	221C	F <small>OURTH</small> R <small>OOT</small>	FOURTH ROOT	f	8771	2243	≈	ASYMPTOTICALLY EQUAL TO	v	8810	226A	MUCH LESS-THAN	
-	8733	221D	B <small>propop</small>	PROPORTIONAL TO	f	8772	2244	¬asymp;	NOT ASYMPTOTICALLY EQUAL TO	v	8811	226B	MUCH GREATER-THAN	
∞	8734	221E	B <small>infinf</small>	INFINITY	f	8773	2245	≅	APPROXIMATELY EQUAL TO	v	8812	226C	BETWEEN	
∠	8735	221F	R <small>IGHT</small> A <small>NGLE</small>	RIGHT ANGLE	f	8774	2246	≈	APPROXIMATELY BUT NOT ACTUALLY EQUAL TO	v	8813	226D	NOT EQUIVALENT TO	
≈	8736	2220	B <small>ang</small>	ANGLE	f	8775	2247	&neq;	NEITHER APPROXIMATELY NOR ACTUALLY EQUAL TO	v	8814	226E	NOT LESS-THAN	
≈	8737	2221	M <small>EASURED</small> A <small>NGLE</small>	MEASURED ANGLE	f	8776	2248	&almosteq;	ALMOST EQUAL TO	v	8815	226F	NOT GREATER-THAN	
≈	8738	2222	S <small>PHERICAL</small> A <small>NGLE</small>	SPHERICAL ANGLE	f	8777	2249	¬almosteq;	NOT ALMOST EQUAL TO	v	8816	2270	NEITHER LESS-THAN NOR EQUAL TO	
÷	8739	2223	D <small>IVIDES</small>	DIVIDES	f	8778	224A	&almosteq; OR EQUAL TO	NEITHER GREATER-THAN NOR EQUAL TO	v	8817	2271	NEITHER GREATER-THAN NOR EQUAL TO	
÷	8740	2224	D <small>OES</small> N <small>O</small> D <small>IVIDE</small>	DOES NOT DIVIDE	f	8779	224B	&tripletilde;	TRIPLE TILDE	v	8818	2272	LESS-THAN OR EQUIVALENT TO	
÷	8741	2225	P <small>ARALLEL</small> T <small>O</small>	PARALLEL TO	f	8780	224C	&all;	ALL EQUAL TO	v	8819	2273	GREATER-THAN OR EQUIVALENT TO	
÷	8742	2226	N <small>O</small> P <small>ARALLEL</small> T <small>O</small>	NOT PARALLEL TO	f	8781	224D	≡	EQUIVALENT TO	v	8820	2274	NEITHER LESS-THAN NOR EQUIVALENT TO	
÷	8743	2227	N <small>O</small> P <small>ARALLEL</small> T <small>O</small>	NOT PARALLEL TO	f	8782	224E	&geometricallyequiv;	GEOMETRICALLY EQUIVALENT TO	v	8821	2275	NEITHER GREATER-THAN NOR EQUIVALENT TO	
÷	8744	2228	N <small>O</small> P <small>ARALLEL</small> T <small>O</small>	NOT PARALLEL TO	f	8783	224F	&diff;	DIFFERENCE BETWEEN	v	8822	2276	LESS-THAN OR GREATER-THAN	

≤	8822	2276	<	LESS-THAN OR GREATER-THAN	⊕	8862	229E	+	SQUARED PLUS	⊗	8900	22C4	&dotdot;	DIAMOND OPERATOR
≥	8823	2277	>	GREATER-THAN OR LESS-THAN	⊖	8863	229F	−	SQUARED MINUS	⊘	8901	22C5	&dotdot;	DOT OPERATOR
≠	8824	2278	&neq;	NEITHER LESS-THAN NOR GREATER-THAN	⊗	8864	22A0	×	SQUARED TIMES	⊘	8902	22C6	☆	STAR OPERATOR
≺	8825	2279	≺	PRECEDES	⋈	8865	22A1	&dotdot;	SQUARED DOT OPERATOR	⋈	8903	22C7	÷	DIVISION TIMES
≻	8826	227A	≻	SUCCEEDS	⋈	8866	22A2	&righttack;	RIGHT TACK	⋈	8904	22C8	⋈	BOWTIE
≈	8827	227B	≈	APPROXIMATELY	⋈	8867	22A3	&lefttack;	LEFT TACK	⋈	8905	22C9	&left;	LEFT NORMAL FACTOR SEMIDIRECT PRODUCT
≈	8828	227C	≈	PRECESSES OR EQUAL TO	⋈	8868	22A4	&downtack;	DOWN TACK	⋈	8906	22CA	&right;	RIGHT NORMAL FACTOR SEMIDIRECT PRODUCT
≈	8829	227D	≈	SUCCEEDS OR EQUAL TO	⋈	8869	22A5	↑	UP TACK	⋈	8907	22CB	&leftdot;	LEFT SEMIDIRECT PRODUCT
≈	8830	227E	≈	PRECESSES OR EQUIVALENT TO	⋈	8870	22A6	&assert;	ASSERTION	⋈	8908	22CC	&rightdot;	RIGHT SEMIDIRECT PRODUCT
≈	8831	227F	≈	SUCCEEDS OR EQUIVALENT TO	⋈	8871	22A7	⊧	MODELS	⋈	8909	22CD	&reversedtilde;	REVERSED TILDE EQUALS
≈	8832	2280	≈	DOES NOT PRECIDE	⋈	8872	22A8	&true;	TRUE	⋈	8910	22CE	⋎	CURLY LOGICAL OR
≈	8833	2281	≈	DOES NOT SUCCEED	⋈	8873	22A9	&forces;	FORCES	⋈	8911	22CF	⋎	CURLY LOGICAL AND
≈	8834	2282	⊂	SUBSET OF	⋈	8874	22AA	&triplebar;	TRIPLE VERTICAL BAR RIGHT TURNSTILE	⋈	8912	22D0	&doublesubset;	DOUBLE SUBSET
≈	8835	2283	⊃	SUPERSET OF	⋈	8875	22AB	&doublebar;	DOUBLE VERTICAL BAR DOUBLE RIGHT TURNSTILE	⋈	8913	22D1	&doublecup;	DOUBLE SUPERSET
≈	8836	2284	⫕	NOT A SUBSET OF	⋈	8876	22AC	&doesnotprove;	DOES NOT PROVE	⋈	8914	22D2	&doubleintersection;	DOUBLE INTERSECTION
≈	8837	2285	⫕	NOT A SUPERSET OF	⋈	8877	22AD	¬true;	NOT TRUE	⋈	8915	22D3	&doubleunion;	DOUBLE UNION
≈	8838	2286	⫕	SUBSET OF OR EQUAL TO	⋈	8878	22AE	&doesnotforce;	DOES NOT FORCE	⋈	8916	22D4	⋔	PITCHFORK
≈	8839	2287	⫕	SUPERSET OF OR EQUAL TO	⋈	8879	22AF	&negateddoubleverticalbar;	NEGATED DOUBLE VERTICAL BAR DOUBLE RIGHT TURNSTILE	⋈	8917	22D5	&equalandparallel;	EQUAL AND PARALLEL TO
≈	8840	2288	⫕	NEITHER A SUBSET OF NOR EQUAL TO	⋈	8880	22B0	&precedesunder;	PRECESSES UNDER RELATION	⋈	8918	22D6	&lessthanwithdot;	LESS-THAN WITH DOT
≈	8841	2289	⫕	NEITHER A SUPERSET OF NOR EQUAL TO	⋈	8881	22B1	&succesunder;	SUCCEEDS UNDER RELATION	⋈	8919	22D7	&greaterthanwithdot;	GREATER-THAN WITH DOT
≈	8842	228A	⫕	SUBSET OF WITH NOT EQUAL TO	⋈	8882	22B2	&normalsubgroup;	NORMAL SUBGROUP OF	⋈	8920	22D8	&verymuchless;	VERY MUCH LESS-THAN
≈	8843	228B	⫕	SUPERSET OF WITH NOT EQUAL TO	⋈	8883	22B3	&containsas;	CONTAINS AS NORMAL SUBGROUP	⋈	8921	22D9	&verymuchgreater;	VERY MUCH GREATER-THAN
≈	8844	228C	⫕	MULTISET	⋈	8884	22B4	&normalsubgroupor;	NORMAL SUBGROUP OR EQUAL TO	⋈	8922	22DA	&lessthanorequal;	LESS-THAN EQUAL TO OR GREATER-THAN
≈	8845	228D	⫕	MULTISET MULTIPLICATION	⋈	8885	22B5	&containsasnormalsubgroup;	CONTAINS AS NORMAL SUBGROUP OR EQUAL TO	⋈	8923	22DB	&greaterthanorequal;	GREATER-THAN EQUAL TO OR LESS-THAN
≈	8846	228E	⫕	MULTISET UNION	⋈	8886	22B6	&originalof;	ORIGINAL OF	⋈	8924	22DC	&equaltoorequal;	EQUAL TO OR LESS-THAN
≈	8847	228F	⫕	SQUARE IMAGE OF	⋈	8887	22B7	&imageof;	IMAGE OF	⋈	8925	22DD	&equaltoorequal;	EQUAL TO OR GREATER-THAN
≈	8848	228G	⫕	SQUARE ORIGINAL OF	⋈	8888	22B8	⊸	MULTIMAP	⋈	8926	22DE	&equaltoorecedes;	EQUAL TO OR PRECEDES
≈	8849	228H	⫕	L <small>OGIC</small> A <small>LGIC</small>	⋈	8889	22B9	&hermitianconjugate;	HERMITIAN CONJUGATE MATRIX	⋈	8927	22DF	&equaltooreucceeds;	EQUAL TO OR SUCCEEDS
≈	8850	228I	⫕	SQUARE ORIGINAL OF OR EQUAL TO	⋈	8890	22BA	&intercalate;	INTERCALATE	⋈	8928	22E0	&doesnotprecede;	DOES NOT PRECIDE OR EQUAL
≈	8851	228J	⫕	SQUARE CAP	⋈	8891	22B8	&xor;	XOR	⋈	8929	22E1	&doesnotsucceedorequal;	DOES NOT SUCCEED OR EQUAL
≈	8852	228K	⫕	SQUARE CUP	⋈	8892	22BC	&nand;	NAND	⋈	8930	22E2	¬squareimage;	NOT SQUARE IMAGE OF OR EQUAL TO
≈	8853	228L	⫕	CIRCLED PLUS	⋈	8893	22BD	&nor;	NOR	⋈	8931	22E3	¬squareoriginal;	NOT SQUARE ORIGINAL OF OR EQUAL TO
≈	8854	2296	⫕	CIRCLED MINUS	⋈	8894	22BE	&rightanglewitharc;	RIGHT ANGLE WITH ARC	⋈	8932	22E4	&squareimage;	SQUARE IMAGE OF OR NOT EQUAL TO
≈	8855	2297	⫕	CIRCLED TIMES	⋈	8895	22BF	&righttriangle;	RIGHT TRIANGLE	⋈	8933	22E5	&squareoriginal;	SQUARE ORIGINAL OF OR NOT EQUAL TO
≈	8856	2298	⫕	CIRCLED DIVISION SLASH	⋈	8896	22C0	&narylogicaland;	N-ARY LOGICAL AND	⋈	8934	22E6	&less-thanbutnotequiv;	LESS-THAN BUT NOT EQUIVALENT TO
≈	8857	2299	⫕	CIRCLED DOT OPERATOR	⋈	8897	22C1	&narylogicalor;	N-ARY LOGICAL OR	⋈	8935	22E7	&greater-thanbutnotequiv;	GREATER-THAN BUT NOT EQUIVALENT TO
≈	8858	229A	⫕	CIRCLED KONG OPERATOR	⋈	8898	22C2	&naryintersection;	N-ARY INTERSECTION	⋈	8936	22E8	&precedesbutnotequiv;	PRECESDES BUT NOT EQUIVALENT TO
≈	8859	229B	⫕	CIRCLED ASTERISK OPERATOR	⋈	8899	22C3	&naryunion;	N-ARY UNION	⋈	8			

≠	8932	22E4	SQUARE IMAGE OF OR NOT EQUAL TO
≣	8933	22E5	SQUARE ORIGINAL OF OR NOT EQUAL TO
≢	8934	22E6	LESS-THAN BUT NOT EQUIVALENT TO
≣	8935	22E7	GREATER-THAN BUT NOT EQUIVALENT TO
≤	8936	22E8	PRECEDES BUT NOT EQUIVALENT TO
≥	8937	22E9	SUCCEEDS BUT NOT EQUIVALENT TO
≮	8938	22EA	NOT NORMAL SUBGROUP OF
≯	8939	22EB	DOES NOT CONTAIN AS NORMAL SUBGROUP
≷	8940	22EC	NOT NORMAL SUBGROUP OF OR EQUAL TO
≸	8941	22ED	DOES NOT CONTAIN AS NORMAL SUBGROUP OR EQUAL
⋮	8942	22EE	VERTICAL ELLIPSIS
⋮⋮	8943	22EF	MIDLINIE HORIZONTAL ELLIPSIS
⋮⋮⋮	8944	22F0	UP RIGHT DIAGONAL ELLIPSIS
⋮⋮⋮⋮	8945	22F1	DOWN RIGHT DIAGONAL ELLIPSIS
⋮⋮⋮⋮⋮	8946	22F2	ELEMENT OF WITH LONG HORIZONTAL STROKE
⋮⋮⋮⋮⋮⋮	8947	22F3	ELEMENT OF WITH VERTICAL BAR AT END OF HORIZONTAL STROKE
⋮⋮⋮⋮⋮⋮⋮	8948	22F4	SMALL ELEMENT OF WITH VERTICAL BAR AT END OF HORIZONTAL STROKE
⋮⋮⋮⋮⋮⋮⋮⋮	8949	22F5	ELEMENT OF WITH DOT ABOVE
⋮⋮⋮⋮⋮⋮⋮⋮⋮	8950	22F6	ELEMENT OF WITH OVERBAR
⋮⋮⋮⋮⋮⋮⋮⋮⋮⋮	8951	22F7	SMALL ELEMENT OF WITH OVERBAR
⋮⋮⋮⋮⋮⋮⋮⋮⋮⋮⋮	8952	22F8	ELEMENT OF WITH UNDERBAR
⋮⋮⋮⋮⋮⋮⋮⋮⋮⋮⋮⋮	8953	22F9	ELEMENT OF WITH TWO HORIZONTAL STROKES
⋮⋮⋮⋮⋮⋮⋮⋮⋮⋮⋮⋮⋮	8954	22FA	CONTAINS WITH LONG HORIZONTAL STROKE
⋮⋮⋮⋮⋮⋮⋮⋮⋮⋮⋮⋮⋮⋮	8955	22FB	CONTAINS WITH VERTICAL BAR AT END OF HORIZONTAL STROKE
⋮⋮⋮⋮⋮⋮⋮⋮⋮⋮⋮⋮⋮⋮⋮	8956	22FC	SMALL CONTAINS WITH VERTICAL BAR AT END OF HORIZONTAL STROKE
⋮⋮⋮⋮⋮⋮⋮⋮⋮⋮⋮⋮⋮⋮⋮⋮	8957	22FD	CONTAINS WITH OVERBAR
⋮⋮⋮⋮⋮⋮⋮⋮⋮⋮⋮⋮⋮⋮⋮⋮⋮	8958	22FE	SMALL CONTAINS WITH OVERBAR
⋮⋮⋮⋮⋮⋮⋮⋮⋮⋮⋮⋮⋮⋮⋮⋮⋮⋮	8959	22FF	Z NOTATION BAG MEMBERSHIP

27.3 Some Greek Letters Supported by HTML

UTF-8 Greek and Coptic

Range: Decimal 880-1023. Hex 0370-03FF.

If you want any of these characters displayed in HTML, you can use the HTML entity found in the table below.

If the character does not have an HTML entity, you can use the decimal (dec) or hexadecimal (hex) reference.

Example

```
<p>I will display &Delta;<p>
<p>I will display &#916;<p>
<p>I will display &#x0394;<p>
```

Will display as:

```
<!DOCTYPE html>
<html>
  <body>
    <p>I will display &Delta;<p>
    <p>I will display &#916;<p>
    <p>I will display &#x0394;<p>
  </body>
</html>
```

Char	Dec	Hex	Entity	Name
Ϛ	880	0370		GREEK CAPITAL LETTER HETA
Ϛ	881	0371		GREEK SMALL LETTER HETA
Ϛ	882	0372		GREEK CAPITAL LETTER ARCHAIC SAMPI
Ϛ	883	0373		GREEK SMALL LETTER ARCHAIC SAMPI
Ϛ	884	0374		GREEK NUMERAL SIGN
Ϛ	885	0375		GREEK LOWER NUMERAL SIGN
Ϛ	886	0376		GREEK CAPITAL LETTER PAMPHYLIAN DIGAMMA
Ϛ	887	0377		GREEK SMALL LETTER PAMPHYLIAN DIGAMMA
Ϛ	890	037A		GREEK PROSEGGRAMMENI
Ϛ	891	037B		GREEK SMALL REVERSED LUNATE SIGMA SYMBOL
Ϛ	892	037C		GREEK SMALL DOTTED LUNATE SIGMA SYMBOL
Ϛ	893	037D		GREEK SMALL REVERSED DOTTED LUNATE SIGMA SYMBOL
Ϛ	894	037E		GREEK QUESTION MARK
Ϛ	900	0384		GREEK TONOS
Ϛ	901	0385		GREEK DIALYTIKA TONOS
Ϛ	902	0386		GREEK CAPITAL LETTER ALPHA WITH TONOS
Ϛ	903	0387		GREEK AND TELEIA
Ϛ	904	0388		GREEK CAPITAL LETTER EPSILON WITH TONOS
Ϛ	905	0389		GREEK CAPITAL LETTER ETA WITH TONOS
Ϛ	906	038A		GREEK CAPITAL LETTER IOTA WITH TONOS
Ϛ	908	038C		GREEK CAPITAL LETTER OMICRON WITH TONOS
Ϛ	910	038E		GREEK CAPITAL LETTER UPSILON WITH TONOS
Ϛ	911	038F		GREEK CAPITAL LETTER OMEGA WITH TONOS
Ϛ	912	0390		GREEK SMALL LETTER IOTA WITH DIALYTIKA AND TONOS
Ϛ	913	0391	ϐαϳϟ;	GREEK CAPITAL LETTER ALPHA
Ϛ	914	0392	ϐδϳϟ;	GREEK CAPITAL LETTER BETA
Ϛ	915	0393	ϐγϳϟ;	GREEK CAPITAL LETTER GAMMA
Ϛ	916	0394	ϐδϳϟ;	GREEK CAPITAL LETTER DELTA
Ϛ	917	0395	ϐϳϳϟ;	GREEK CAPITAL LETTER EPSILON
Ϛ	918	0396	ϐζϳϟ;	GREEK CAPITAL LETTER ZETA
Ϛ	919	0397	ϐϳϳϟ;	GREEK CAPITAL LETTER ETA
Ϛ	920	0398	ϐ϶ϳϟ;	GREEK CAPITAL LETTER THETA
Ϛ	921	0399	ϐ϶ϳϟ;	GREEK CAPITAL LETTER IOTA
Ϛ	922	039A	ϐϗϳϟ;	GREEK CAPITAL LETTER KAPPA
Ϛ	923	039B	ϐϘϳϟ;	GREEK CAPITAL LETTER LAMDA
Ϛ	924	039C	ϐϙϳϟ;	GREEK CAPITAL LETTER MU
Ϛ	925	039D	ϐϘϳϟ;	GREEK CAPITAL LETTER NU
Ϛ	926	039E	ϐϘϳϟ;	GREEK CAPITAL LETTER XI
Ϛ	927	039F	ϐϘϘϳϟ;	GREEK CAPITAL LETTER OMICRON
Ϛ	928	03A0	ϐϘϳϟ;	GREEK CAPITAL LETTER PI
Ϛ	929	03A1	ϐϳϳ;	GREEK CAPITAL LETTER RHO
Ϛ	931	03A3	ϐϳϳ;	GREEK CAPITAL LETTER SIGMA
Ϛ	932	03A4	ϐͲϳ;	GREEK CAPITAL LETTER TAU
Ϛ	933	03A5	ϐϳϳ;	GREEK CAPITAL LETTER UPSILON
Ϛ	934	03A6	ϐϳϳ;	GREEK CAPITAL LETTER PHI
Ϛ	935	03A7	ϐϘϳ;	GREEK CAPITAL LETTER CHI
Ϛ	936	03A8	ϐϘϳ;	GREEK CAPITAL LETTER PSI
Ϛ	937	03A9	ϐϘϘϳ;	GREEK CAPITAL LETTER OMEGA
Ϛ	938	03AA		GREEK CAPITAL LETTER IOTA WITH DIALYTIKA
Ϛ	939	03AB		GREEK CAPITAL LETTER UPSILON WITH DIALYTIKA
Ϛ	940	03AC		GREEK SMALL LETTER ALPHI WITH TONOS
Ϛ	941	03AD		GREEK SMALL LETTER EPSILON WITH TONOS
Ϛ	942	03AE		GREEK SMALL LETTER ETA WITH TONOS
Ϛ	943	03AF		GREEK SMALL LETTER IOTA WITH TONOS
Ϛ	944	03B0		GREEK SMALL LETTER UPSILON WITH DIALYTIKA AND TONOS
Ϛ	945	03B1	ϐϳϳ;	GREEK SMALL LETTER ALPHA
Ϛ	946	03B2	ϐϳϳ;	GREEK SMALL LETTER BETA
Ϛ	947	03B3	ϐϳϳ;	GREEK SMALL LETTER GAMMA
Ϛ	948	03B4	ϐϳϳ;	GREEK SMALL LETTER DELTA
Ϛ	949	03B5	ϐϳϳ;	GREEK SMALL LETTER EPSILON
Ϛ	950	03B6	ϐϳϳ;	GREEK SMALL LETTER ZETA
Ϛ	951	03B7	ϐϳϳ;	GREEK SMALL LETTER ETA
Ϛ	952	03B8	ϐϳϳ;	GREEK SMALL LETTER THETA
Ϛ	953	03B9	ϐϳϳ;	GREEK SMALL LETTER IOTA
Ϛ	954	03BA	ϐϳϳ;	GREEK SMALL LETTER KAPPA
Ϛ	955	03BB	ϐϳϳ;	GREEK SMALL LETTER LAMDA
Ϛ	956	03BC	ϐϳϳ;	GREEK SMALL LETTER MU
Ϛ	957	03BD	ϐϳϳ;	GREEK SMALL LETTER NU
Ϛ	958	03BE	ϐϳϳ;	GREEK SMALL LETTER XI
Ϛ	959	03BF	ϐϳϳ;	GREEK SMALL LETTER OMICRON
Ϛ	960	03C0	ϐϳϳ;	GREEK SMALL LETTER PI
Ϛ	961	03C1	ϐϳϳ;	GREEK SMALL LETTER RHO
Ϛ	962	03C2	ϐϳϳ;	GREEK SMALL LETTER FINAL SIGMA
Ϛ	963	03C3	ϐϳϳ;	GREEK SMALL LETTER SIGMA
Ϛ	964	03C4	ϐϳϳ;	GREEK SMALL LETTER TAU
Ϛ	965	03C5	ϐϳϳ;	GREEK SMALL LETTER UPSILON
Ϛ	966	03C6	ϐϳϳ;	GREEK SMALL LETTER PHI
Ϛ	967	03C7	ϐϳϳ;	GREEK SMALL LETTER CHI
Ϛ	968	03C8	ϐϳϳ;	GREEK SMALL LETTER PSI
Ϛ	969	03C9	ϐϳϳ;	GREEK SMALL LETTER OMEGA
Ϛ	970	03CA		GREEK SMALL LETTER IOTA WITH DIALYTIKA
Ϛ	971	03CB		GREEK SMALL LETTER UPSILON WITH DIALYTIKA
Ϛ	972	03CC		GREEK SMALL LETTER OMICRON WITH TONOS
Ϛ	973	03CD		GREEK SMALL LETTER UPSILON WITH TONOS
Ϛ	974	03CE		GREEK SMALL LETTER OMEGA WITH TONOS
Ϛ	975	03CF		GREEK CAPITAL KAI SYMBOL
Ϛ	976	03D0		GREEK BETH SYMBOL
Ϛ	977	03D1	ϐ&thetaisym;	GREEK THETA SYMBOL
Ϛ	978	03D2	ϐυ	GREEK UPSILON WITH HOOK SYMBOL
Ϛ	979	03D3		GREEK UPSILON WITH ACUTE AND HOOK SYMBOL
Ϛ	980	03D4		GREEK UPSILON WITH DIAERESIS AND HOOK SYMBOL
Ϛ	981	03D5	ϐ&straightpsilon;	GREEK PHI SYMBOL
Ϛ	982	03D6	ϐ`	GREEK PI SYMBOL
Ϛ	983	03D7		GREEK KAI SYMBOL
Ϛ	984	03D8		GREEK LETTER ARCHAIC KOPPA
Ϛ	985	03D9		GREEK SMALL LETTER ARCHAIC KOPPA
Ϛ	986	03DA		GREEK LETTER STIGMA
Ϛ	987	03DB		GREEK SMALL LETTER STIGMA
Ϛ	988	03DC	ϐϜ	GREEK LETTER DIGAMMA
Ϛ	989	03DD	ϐϝ	GREEK SMALL LETTER DIGAMMA
Ϛ	990	03DE		GREEK LETTER KOPPA
Ϛ	991	03DF		GREEK SMALL LETTER KOPPA
Ϛ	992	03E0		GREEK LETTER SAMPI
Ϛ	993	03E1		GREEK SMALL LETTER SAMPI
Ϛ	994	03E2		Coptic Capital Letter Shei
Ϛ	995	03E3		Coptic Small Letter Shei
Ϛ	996	03E4		Coptic Capital Letter Fei
Ϛ	997	03E5		Coptic Small Letter Fei
Ϛ	998	03E6		Coptic Capital Letter Khei
Ϛ	999	03E7		Coptic Small Letter Khei
Ϛ	1000	03E8		Coptic Capital Letter Shma
Ϛ	1001	03E9		Coptic Small Letter Hor
Ϛ	1002	03EA		Coptic Capital Letter Gangia
Ϛ	1003	03EB		Coptic Small Letter Gangia
Ϛ	1004	03EC		Coptic Capital Letter Shema
Ϛ	1005	03ED		Coptic Small Letter Shma
Ϛ	1006	03EE		Coptic Capital Letter Dei
Ϛ	1007	03EF		Coptic Small Letter Dei
Ϛ	1008	03F0	ϐϰ	Greek Kappa Symbol
Ϛ	1009	03F1	ϐϱ	Greek Rho Symbol
Ϛ	1010	03F2		Greek Lunate Sigma Symbol

c	1010	03F2	GREEK LUNATE SIGMA SYMBOL
j	1011	03F3	GREEK LETTER YOT
Ϛ	1012	03F4	GREEK CAPITAL THETA SYMBOL
Ϛ	1013	03F5	ϵ
Ϛ	1014	03F6	϶
Ϛ	1015	03F7	GREEK CAPITAL LETTER SHO
Ϛ	1016	03F8	GREEK SMALL LETTER SHO
C	1017	03F9	GREEK CAPITAL LUNATE SIGMA SYMBOL
M	1018	03FA	GREEK CAPITAL LETTER SAN
M	1019	03FB	GREEK SMALL LETTER SAN
Ϛ	1020	03FC	GREEK RHO WITH STROKE SYMBOL
Ϛ	1021	03FD	GREEK CAPITAL REVERSED LUNATE SIGMA SYMBOL
Ϛ	1022	03FE	GREEK CAPITAL DOTTED LUNATE SIGMA SYMBOL
Ϛ	1023	03FF	GREEK CAPITAL REVERSED DOTTED LUNATE SIGMA SYMBOL

27.4 Some Other Entities Supported by HTML

Range: Decimal 8352-8399. Hex 20A0-20CF.

If you want any of these characters displayed in HTML, you can use the HTML entity found in the table below. If the character does not have an HTML entity, you can use the decimal (dec) or hexadecimal (hex) reference.

Example

```
<p>I will display &euro;</p>
<p>I will display &#8364;</p>
<p>I will display &#x20AC;</p>

<!DOCTYPE html>
<html>
  <body>
    <p>I will display &euro;</p>
    <p>I will display &#8364;</p>
    <p>I will display &#x20AC;</p>
  </body>
</html>
```

Note: Some browsers may not support all HTML5 entities in the table below.

Currently, only IE 11 and Firefox 35 support all HTML5 entities.

Char	Dec	Hex	Entity	Name
₵	8352	20A0		EURO-CURRENCY SIGN
₵	8353	20A1		COLON SIGN
₵	8354	20A2		CRUZEIRO SIGN
₣	8355	20A3		FRENCH FRANC SIGN
₺	8356	20A4		LIRA SIGN
ℳ	8357	20A5		MILL SIGN
₦	8358	20A6		NAIRA SIGN
₱	8359	20A7		PESETA SIGN
₹	8360	20A8		RUPEE SIGN
₩	8361	20A9		WON SIGN
₪	8362	20AA		NEW SHEQEL SIGN
đ	8363	20AB		DONG SIGN
€	8364	20AC	€	EURO SIGN
₭	8365	20AD		KIP SIGN
₮	8366	20AE		TUGRIK SIGN
₯	8367	20AF		DRACHMA SIGN
₢	8368	20B0		GERMAN PENNY SYMBOL
₱	8369	20B1		PESO SIGN
₲	8370	20B2		GUARANI SIGN
₼	8371	20B3		AUSTRAL SIGN
₴	8372	20B4		HRYVNIA SIGN
₵	8373	20B5		CEDI SIGN
₮	8374	20B6		LIVRE TOURNOIS SIGN
₮	8375	20B7		SPESMOLO SIGN
₮	8376	20B8		TENGE SIGN
₹	8377	20B9		INDIAN RUPEE SIGN

27.4.1 UTF-8 Arrows

Range: Decimal 8592-8703. Hex 2190-21FF.

If you want any of these characters displayed in HTML, you can use the HTML entity found in the table below.

If the character does not have an HTML entity, you can use the decimal (dec) or hexadecimal (hex) reference.

Example

```
<p>I will display &larr;<p>  
<p>I will display &#8592;<p>  
<p>I will display &#x2190;<p>
```

Will display as:

```
<!DOCTYPE html>  
<html>  
  <body>  
    <p>I will display &larr;<p>  
    <p>I will display &#8592;<p>  
    <p>I will display &#x2190;<p>  
  </body>  
</html>
```

Char	Dec	Hex	Entity	Name			
←	8592	2190	←	LEFTWARDS ARROW	↳	8623	21AF
↑	8593	2191	↑	UPWARDS ARROW	↴	8624	21B0
→	8594	2192	→	RIGHTWARDS ARROW	↶	8625	21B1
↓	8595	2193	↓	DOWNWARDS ARROW	↷	8626	21B2
↔	8596	2194	↔	LEFT RIGHT ARROW	↶	8627	21B3
↕	8597	2195		UP DOWN ARROW	↷	8628	21B4
↖	8598	2196		NORTH WEST ARROW	↶	8629	21B5
↗	8599	2197		NORTH EAST ARROW	↷	8630	21B6
↘	8600	2198		SOUTH EAST ARROW	↶	8631	21B7
↙	8601	2199		SOUTH WEST ARROW	↷	8632	21B8
↶	8602	219A		LEFTWARDS ARROW WITH STROKE	↶	8633	21B9
↷	8603	219B		RIGHTWARDS ARROW WITH STROKE	↶	8634	21BA
↶	8604	219C		LEFTWARDS WAVE ARROW	↷	8635	21BB
↷	8605	219D		RIGHTWARDS WAVE ARROW	↶	8636	21BC
↶	8606	219E		LEFTWARDS TWO HEADED ARROW	↷	8637	21BD
↑↑	8607	219F		UPWARDS TWO HEADED ARROW	↑↑	8638	21BE
⤠	8608	21A0		RIGHTWARDS TWO HEADED ARROW	⤠	8639	21BF
⤡	8609	21A1		DOWNWARDS TWO HEADED ARROW	⤡	8640	21C0
⤢	8610	21A2		LEFTWARDS ARROW WITH TAIL	⤢	8641	21C1
⤣	8611	21A3		RIGHTWARDS ARROW WITH TAIL	⤢	8642	21C2
⤤	8612	21A4		LEFTWARDS ARROW FROM BAR	⤢	8643	21C3
⤥	8613	21A5		UPWARDS ARROW FROM BAR	⤤	8644	21C4
⤦	8614	21A6		RIGHTWARDS ARROW FROM BAR	⤥	8645	21C5
⤧	8615	21A7		DOWNWARDS ARROW FROM BAR	⤦	8646	21C6
⤨	8616	21A8		UP DOWN ARROW WITH BASE	⤧	8647	21C7
⤩	8617	21A9		LEFTWARDS ARROW WITH HOOK	⤨	8648	21C8
⤪	8618	21AA		RIGHTWARDS ARROW WITH HOOK	⤩	8649	21C9
⤫	8619	21AB		LEFTWARDS ARROW WITH LOOP	⤪	8650	21CA
⤬	8620	21AC		RIGHTWARDS ARROW WITH LOOP	⤫	8651	21CB
					⤬	8652	21CC

⤱	8653	21CD		LEFTWARDS DOUBLE ARROW WITH STROKE	⤱	8683	21EB
⤲	8654	21CE		LEFT RIGHT DOUBLE ARROW WITH STROKE	⤲	8684	21EC
⤳	8655	21CF		RIGHTWARDS DOUBLE ARROW WITH STROKE	⤳	8685	21ED
⤴	8656	21D0	&IArr;	LEFTWARDS DOUBLE ARROW	⤴	8686	21EE
⤵	8657	21D1	⇑	UPWARDS DOUBLE ARROW	⤵	8687	21EF
⤶	8658	21D2	⇒	RIGHTWARDS DOUBLE ARROW	⤶	8688	21F0
⤷	8659	21D3	⇓	DOWNWARDS DOUBLE ARROW	⤷	8689	21F1
⤸	8660	21D4	⇔	LEFT RIGHT DOUBLE ARROW	⤸	8690	21F2
⤹	8661	21D5		UP DOWN DOUBLE ARROW	⤹	8691	21F3
⤺	8662	21D6		NORTH WEST DOUBLE ARROW	⤺	8692	21F4
⤻	8663	21D7		NORTH EAST DOUBLE ARROW	⤻	8693	21F5
⤼	8664	21D8		SOUTH EAST DOUBLE ARROW	⤼	8694	21F6
⤽	8665	21D9		SOUTH WEST DOUBLE ARROW	⤽	8695	21F7
⤾	8666	21DA		LEFTWARDS TRIPLE ARROW	⤾	8696	21F8
⤿	8667	21DB		RIGHTWARDS TRIPLE ARROW	⤿	8697	21F9
⤿	8668	21DC		LEFTWARDS SQUIGGLE ARROW	⤿	8698	21FA
⤿	8669	21DD		RIGHTWARDS SQUIGGLE ARROW	⤿	8699	21FB
⤿	8670	21DE		UPWARDS ARROW WITH DOUBLE STROKE	⤿	8700	21FC
⤿	8671	21DF		DOWNWARDS ARROW WITH DOUBLE STROKE	⤿	8701	21FD
⤿	8672	21E0		LEFTWARDS DASHED ARROW	⤿	8702	21FE
⤿	8673	21E1		UPWARDS DASHED ARROW	⤿	8703	21FF
⤿	8674	21E2		RIGHTWARDS DASHED ARROW			
⤿	8675	21E3		DOWNWARDS DASHED ARROW			
⤿	8676	21E4		LEFTWARDS ARROW TO BAR			
⤿	8677	21E5		RIGHTWARDS ARROW TO BAR			
⤿	8678	21E6		LEFTWARDS WHITE ARROW			
⤿	8679	21E7		UPWARDS WHITE ARROW			
⤿	8680	21E8		RIGHTWARDS WHITE ARROW			
⤿	8681	21E9		DOWNWARDS WHITE ARROW			
⤿	8682	21EA		UPWARDS WHITE ARROW FROM BAR			

27.4.2 UTF-8 Miscellaneous Symbols

Range: Decimal 9728-9983. Hex 2600-26FF.

If you want any of these characters displayed in HTML, you can use the HTML entity found in the table below. If the character does not have an HTML entity, you can use the decimal (dec) or hexadecimal (hex) reference.

Example

```
<p>I will display &spades;<p>
<p>I will display &#9824;<p>
<p>I will display &#x2660;<p>
```

Will display as:

```
<!DOCTYPE html>
<html>
  <body>
    <p>I will display &spades;<p>
    <p>I will display &#9824;<p>
    <p>I will display &#x2660;<p>
  </body>
</html>
```

Note: Some browsers may not support all HTML5 entities in the table below. Currently, only IE 11 and Firefox 35 support all HTML5 entities.

Char	Dec	Hex	Entity	Name	9757	261D	WHITE UP POINTING INDEX	•	9787	263B	BLACK SMILING FACE	
★	9728	2600		BLACK SUN WITH RAYS	◐	9758	261E	WHITE RIGHT POINTING INDEX	*	9788	263C	WHITE SUN WITH RAYS
●	9729	2601		CLOUD	◑	9759	261F	WHITE DOWN POINTING INDEX	›	9789	263D	FIRST QUARTER MOON
▲	9730	2602		UMBRELLA	■	9760	2620	SKULL AND CROSSBONES	‹	9790	263E	LAST QUARTER MOON
▷	9731	2603		SNOWMAN	▬	9761	2621	CAUTION SIGN	♀	9791	263F	MERCURY
／	9732	2604		COMET	▬	9762	2622	RADIOACTIVE SIGN	♂	9792	2640	FEMALE SIGN
★	9733	2605		BLACK STAR	▬	9763	2623	BIOHAZARD SIGN	δ	9793	2641	EARTH
☆	9734	2606		WHITE STAR	▬	9764	2624	CADUCEUS	♂	9794	2642	MALE SIGN
◀	9735	2607		LIGHTNING	▬	9765	2625	ANIKH	▬	9795	2643	JUPITER
☈	9736	2608		THUNDERSTORM	▬	9766	2626	ORTHODOX CROSS	▬	9796	2644	SATURN
○	9737	2609		SUN	▬	9767	2627	CHI RHO	♀	9797	2645	URANUS
□	9738	260A		ASCENDING NODE	▬	9768	2628	CROSS OF LORRAINE	▬	9798	2646	NEPTUNE
□	9739	260B		DESCENDING NODE	▬	9769	2629	CROSS OF JERUSALEM	▬	9799	2647	PLUTO
▫	9740	260C		CONJUNCTION	▬	9770	262A	STAR AND CRESCENT	▬	9800	2648	ARIES
❖	9741	260D		OPPOSITION	▬	9771	262B	FARSI SYMBOL	▬	9801	2649	TAURUS
☎	9742	260E		BLACK TELEPHONE	▬	9772	262C	ADI SHAKTI	▬	9802	264A	GEMINI
☏	9743	260F		WHITE TELEPHONE	▬	9773	262D	HAMMER AND SICKLE	▬	9803	264B	CANCER
□	9744	2610		BALLOT BOX	▬	9774	262E	PEACE SYMBOL	▬	9804	264C	LEO
☒	9745	2611		BALLOT BOX WITH CHECK	▬	9775	262F	YIN YANG	▬	9805	264D	VIRGO
☰	9746	2612		BALLOT BOX WITH X	▬	9776	2630	TRIGRAM FOR HEAVEN	▬	9806	264E	LIBRA
X	9747	2613		SALTIRE	▬	9777	2631	TRIGRAM FOR LAKE	▬	9807	264F	SCORPIUS
☂	9748	2614		UMBRELLA WITH RAIN DROPS	▬	9778	2632	TRIGRAM FOR FIRE	▬	9808	2650	SAGITTARIUS
❀	9749	2615		HOT BEVERAGE	▬	9779	2633	TRIGRAM FOR THUNDER	▬	9809	2651	CAPRICORN
□	9750	2616		WHITE SHOGI PIECE	▬	9780	2634	TRIGRAM FOR WIND	▬	9810	2652	AQUARIUS
☗	9751	2617		BLACK SHOGI PIECE	▬	9781	2635	TRIGRAM FOR WATER	▬	9811	2653	PISCES
▲	9752	2618		SHAMROCK	▬	9782	2636	TRIGRAM FOR MOUNTAIN	▬	9812	2654	WHITE CHESS KING
▬	9753	2619		REVERSED ROTATED FLORAL HEART BULLET	▬	9783	2637	TRIGRAM FOR EARTH	▬	9813	2655	WHITE CHESS QUEEN
▼	9754	261A		BLACK LEFT POINTING INDEX	▬	9784	2638	WHEEL OF DHARMA	▬	9814	2656	WHITE CHESS ROOK
▼	9755	261B		BLACK RIGHT POINTING INDEX	▬	9785	2639	WHITE FROWNING FACE	▬	9815	2657	WHITE CHESS BISHOP
▬	9756	261C		WHITE LEFT POINTING INDEX	▬	9786	263A	WHITE SMILING FACE	▬	9816	2658	WHITE CHESS KNIGHT

♟	9817	2659		WHITE CHESS PAWN	▲	9847	2677	RECYCLING SYMBOL FOR TYPE-5 PLASTICS
♚	9818	265A		BLACK CHESS KING	▲	9848	2678	RECYCLING SYMBOL FOR TYPE-6 PLASTICS
♛	9819	265B		BLACK CHESS QUEEN	▲	9849	2679	RECYCLING SYMBOL FOR TYPE-7 PLASTICS
♜	9820	265C		BLACK CHESS ROOK	△	9850	267A	RECYCLING SYMBOL FOR GENERIC MATERIALS
♝	9821	265D		BLACK CHESS BISHOP	♻	9851	267B	BLACK UNIVERSAL RECYCLING SYMBOL
♞	9822	265E		BLACK CHESS KNIGHT	♾	9852	267C	RECYCLED PAPER SYMBOL
♙	9823	265F		BLACK CHESS PAWN	♾	9853	267D	PARTIALLY-RECYCLED PAPER SYMBOL
♠	9824	2660	♠	BLACK SPADE SUIT	♾	9854	267E	PERMANENT PAPER SIGN
♥	9825	2661		WHITE HEART SUIT	♿	9855	267F	WHEELCHAIR SYMBOL
♦	9826	2662		WHITE DIAMOND SUIT	▣	9856	2680	DIE FACE-1
♣	9827	2663	♣	BLACK CLUB SUIT	▣	9857	2681	DIE FACE-2
♤	9828	2664		WHITE SPADE SUIT	▣	9858	2682	DIE FACE-3
♡	9829	2665	♥	BLACK HEART SUIT	▣	9859	2683	DIE FACE-4
♦	9830	2666	♦	BLACK DIAMOND SUIT	▣	9860	2684	DIE FACE-5
♧	9831	2667		WHITE CLUB SUIT	▣	9861	2685	DIE FACE-6
♨	9832	2668		HOT SPRINGS	○	9862	2686	WHITE CIRCLE WITH DOT RIGHT
♩	9833	2669		QUARTER NOTE	○	9863	2687	WHITE CIRCLE WITH TWO DOTS
♪	9834	266A		EIGHTH NOTE	●	9864	2688	BLACK CIRCLE WITH WHITE DOT RIGHT
♫	9835	266B		BEAMED EIGHTH NOTES	●	9865	2689	BLACK CIRCLE WITH TWO WHITE DOTS
♬	9836	266C		BEAMED SIXTEENTH NOTES	-	9866	268A	MONOGRAM FOR YANG
♪	9837	266D		MUSIC FLAT SIGN	--	9867	268B	MONOGRAM FOR YIN
♮	9838	266E		MUSIC NATURAL SIGN	=	9868	268C	DIGRAM FOR GREATER YANG
#	9839	266F		MUSIC SHARP SIGN	=	9869	268D	DIGRAM FOR LESSER YIN
+	9840	2670		WEST SYRIAC CROSS	=	9870	268E	DIGRAM FOR LESSER YANG
+	9841	2671		EAST SYRIAC CROSS	=	9871	268F	DIGRAM FOR GREATER YIN
♾	9842	2672		UNIVERSAL RECYCLING SYMBOL	▶	9872	2690	WHITE FLAG
♻	9843	2673		RECYCLING SYMBOL FOR TYPE-1 PLASTICS	▶	9873	2691	BLACK FLAG
♻	9844	2674		RECYCLING SYMBOL FOR TYPE-2 PLASTICS	✖	9874	2692	HAMMER AND PICK
♻	9845	2675		RECYCLING SYMBOL FOR TYPE-3 PLASTICS	⚓	9875	2693	ANCHOR
♻	9846	2676		RECYCLING SYMBOL FOR TYPE-4 PLASTICS	x	9876	2694	CROSSED SWORDS

⌚	9877	2695	STAFF OF AESCULAPIUS		⌚	9907	26B3	CERES
⚖	9878	2696	SCALES		⚖	9908	26B4	PALLAS
⚗	9879	2697	ALEMBIC		⚗	9909	26B5	JUNO
✿	9880	2698	FLOWER		✿	9910	26B6	VESTA
⚙	9881	2699	GEAR		⚙	9911	26B7	CHIRON
☿	9882	269A	STAFF OF HERMES		☿	9912	26B8	BLACK MOON LILITH
✳	9883	269B	ATOM SYMBOL		✳	9913	26B9	SEXTILE
✳	9884	269C	FLEUR-DE-LIS		✳	9914	26BA	SEMISEXITLE
◻	9885	269D	OUTLINED WHITE STAR		◻	9915	26BB	QUINCUNX
◻	9886	269E	THREE LINES CONVERGING RIGHT		◻	9916	26BC	SESQUIQUADRATRE
◻	9887	269F	THREE LINES CONVERGING LEFT		⚽	9917	26BD	SOCCER BALL
⚠	9888	26A0	WARNING SIGN		⚾	9918	26BE	BASEBALL
⚡	9889	26A1	HIGH VOLTAGE SIGN		◻	9919	26BF	SQUARED KEY
⊗	9890	26A2	DOUBLED FEMALE SIGN		◻	9920	26C0	WHITE DRAUGHTS MAN
♂	9891	26A3	DOUBLED MALE SIGN		◻	9921	26C1	WHITE DRAUGHTS KING
⚥	9892	26A4	INTERLOCKED FEMALE AND MALE SIGN		◻	9922	26C2	BLACK DRAUGHTS MAN
⚥	9893	26A5	MALE AND FEMALE SIGN		◻	9923	26C3	BLACK DRAUGHTS KING
♂	9894	26A6	MALE WITH STROKE SIGN		⛄	9924	26C4	SNOWMAN WITHOUT SNOW
Ѷ	9895	26A7	MALE WITH STROKE AND MALE AND FEMALE SIGN		┉	9925	26C5	SUN BEHIND CLOUD
܂	9896	26A8	VERTICAL MALE WITH STROKE SIGN		┉	9926	26C6	RAIN
܂	9897	26A9	HORIZONTAL MALE WITH STROKE SIGN		┉	9927	26C7	BLACK SNOWMAN
┉	9898	26AA	MEDIUM WHITE CIRCLE		┉	9928	26C8	THUNDER CLOUD AND RAIN
●	9899	26AB	MEDIUM BLACK CIRCLE		┉	9929	26C9	TURNED WHITE SHOGI PIECE
○	9900	26AC	MEDIUM SMALL WHITE CIRCLE		┉	9930	26CA	TURNED BLACK SHOGI PIECE
♾	9901	26AD	MARRIAGE SYMBOL		┉	9931	26CB	WHITE DIAMOND IN SQUARE
♾	9902	26AE	DIVORCE SYMBOL		┉	9932	26CC	CROSSING LANES
♾	9903	26AF	UNMARRIED PARTNERSHIP SYMBOL		┉	9933	26CD	DISABLED CAR
┉	9904	26B0	COFFIN		┉	9934	26CE	OPHIUCHUS
┉	9905	26B1	FUNERAL URN		┉	9935	26CF	PICK
┉	9906	26B2	NEUTER		┉	9936	26D0	CAR SLIDING

⛑	9937	26D1	HELMET WITH WHITE CROSS		┉	9967	26EF	MAP SYMBOL FOR LIGHTHOUSE
┉	9938	26D2	CIRCLED CROSSING LANES		┉	9968	26F0	MOUNTAIN
┉	9939	26D3	CHAINS		┉	9969	26F1	UMBRELLA ON GROUND
┉	9940	26D4	NO ENTRY		┉	9970	26F2	FOUNTAIN
┉	9941	26D5	ALTERNATE ONE-WAY LEFT WAY TRAFFIC		┉	9971	26F3	FLAG IN HOLE
┉	9942	26D6	BLACK TWO-WAY LEFT WAY TRAFFIC		┉	9972	26F4	FERRY
┉	9943	26D7	WHITE TWO-WAY LEFT WAY TRAFFIC		┉	9973	26F5	SAILBOAT
┉	9944	26D8	BLACK LEFT LANE MERGE		┉	9974	26F6	SQUARE FOUR CORNERS
┉	9945	26D9	WHITE LEFT LANE MERGE		┉	9975	26F7	SKIER
┉	9946	26DA	DRIVE SLOW SIGN		┉	9976	26F8	ICE SKATE
┉	9947	26D8	HEAVY WHITE DOWN-POINTING TRIANGLE		┉	9977	26F9	PERSON WITH BALL
┉	9948	26DC	LEFT CLOSED ENTRY		┉	9978	26FA	TENT
┉	9949	26DD	SQUARED SALTIRE		┉	9979	26FB	JAPANESE BANK SYMBOL
┉	9950	26DE	FALLING DIAGONAL IN WHITE CIRCLE IN BLACK SQUARE		┉	9980	26FC	HEADSTONE GRAVEYARD SYMBOL
┉	9951	26DF	BLACK TRUCK		┉	9981	26FD	FUEL PUMP
┉	9952	26E0	RESTRICTED LEFT ENTRY-1		┉	9982	26FE	CUP ON BLACK SQUARE
┉	9953	26E1	RESTRICTED LEFT ENTRY-2		┉	9983	26FF	WHITE FLAG WITH HORIZONTAL MIDDLE BLACK STRIPE
┉	9954	26E2	ASTRONOMICAL SYMBOL FOR URANUS					
┉	9955	26E3	HEAVY CIRCLE WITH STROKE AND TWO DOTS ABOVE					
┉	9956	26E4	PENTAGRAM					
┉	9957	26E5	RIGHT-HANDED INTERLACED PENTAGRAM					
┉	9958	26E6	LEFT-HANDED INTERLACED PENTAGRAM					
┉	9959	26E7	INVERTED PENTAGRAM					
┉	9960	26E8	BLACK CROSS ON SHIELD					
┉	9961	26E9	SHINTO SHRINE					
┉	9962	26EA	CHURCH					
┉	9963	26EB	CASTLE					
┉	9964	26EC	HISTORIC SITE					
┉	9965	26ED	GEAR WITHOUT HUB					
┉	9966	26EE	GEAR WITH HANDLES					

CHAPTER TWENTY-EIGHT

28.0 HTML ENCODING (CHARACTER SET)

To display an HTML page correctly, a web browser must know the character set (character encoding) to use.

28.1 What is Character Encoding?

ASCII was the first **character encoding standard** (also called character set). It defines 127 different alphanumeric characters that could be used on the internet. ASCII supported numbers (0-9), English letters (A-Z), and some special characters like ! \$ + - () @ < > .

ANSI (Windows-1252) was the original Windows character set. It supported 256 different character codes.

ISO-8859-1 was the default character set for HTML 4. It also supported 256 different character codes. Because ANSI and ISO was limited, the default character encoding was changed to UTF-8 in HTML5. UTF-8 (Unicode) covers almost all of the characters and symbols in the world.

All HTML 4 processors also support UTF-8.

28.2 The HTML charset Attribute

To display an HTML page correctly, a web browser must know the character set used in the page.

This is specified in the <meta> tag:

For HTML4:

```
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
```

For HTML5:

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

If a browser detects ISO-8859-1 in a web page, it defaults to ANSI, because ANSI is identical to ISO-8859-1 except that ANSI has 32 extra characters.

28.3 Differences Between Character Sets

The following table displays the differences between the character sets described above:

Numb	ASCII	ANSI	8859	UTF-8	Description
32					space
33	!	!	!	!	exclamation mark
34	"	"	"	"	quotation mark
35	#	#	#	#	number sign
36	\$	\$	\$	\$	dollar sign
37	%	%	%	%	percent sign
38	&	&	&	&	ampersand
39	'	'	'	'	apostrophe
40	((((left parenthesis
41))))	right parenthesis
42	*	*	*	*	asterisk
43	+	+	+	+	plus sign
44	,	,	,	,	comma
45	-	-	-	-	hyphen-minus
46	full stop
47	/	/	/	/	solidus
48	0	0	0	0	digit zero
49	1	1	1	1	digit one
50	2	2	2	2	digit two
51	3	3	3	3	digit three
52	4	4	4	4	digit four
53	5	5	5	5	digit five
54	6	6	6	6	digit six
55	7	7	7	7	digit seven
56	8	8	8	8	digit eight
57	9	9	9	9	digit nine
58	:	:	:	:	colon
59	;	;	;	;	semicolon
60	<	<	<	<	less-than sign
61	=	=	=	=	equals sign
62	>	>	>	>	greater-than sign
63	?	?	?	?	question mark
64	@	@	@	@	commercial at
65	A	A	A	A	Latin capital letter A
66	B	B	B	B	Latin capital letter B
67	C	C	C	C	Latin capital letter C
68	D	D	D	D	Latin capital letter D
69	E	E	E	E	Latin capital letter E
70	F	F	F	F	Latin capital letter F
71	G	G	G	G	Latin capital letter G
72	H	H	H	H	Latin capital letter H
73	I	I	I	I	Latin capital letter I
74	J	J	J	J	Latin capital letter J
75	K	K	K	K	Latin capital letter K
76	L	L	L	L	Latin capital letter L
77	M	M	M	M	Latin capital letter M
78	N	N	N	N	Latin capital letter N
79	O	O	O	O	Latin capital letter O
80	P	P	P	P	Latin capital letter P
81	Q	Q	Q	Q	Latin capital letter Q
82	R	R	R	R	Latin capital letter R
83	S	S	S	S	Latin capital letter S
84	T	T	T	T	Latin capital letter T
85	U	U	U	U	Latin capital letter U
86	V	V	V	V	Latin capital letter V
87	W	W	W	W	Latin capital letter W
88	X	X	X	X	Latin capital letter X
89	Y	Y	Y	Y	Latin capital letter Y
90	Z	Z	Z	Z	Latin capital letter Z
91	[[[[left square bracket
92	\	\	\	\	reverse solidus
93]]]]	right square bracket
94	^	^	^	^	circumflex accent
95	–	–	–	–	low line
96	‘	‘	‘	‘	grave accent
97	à	à	à	à	Latin small letter a
98	ò	ò	ò	ò	Latin small letter b
99	ç	ç	ç	ç	Latin small letter c
100	đ	đ	đ	đ	Latin small letter d
101	é	é	é	é	Latin small letter e
102	ƒ	ƒ	ƒ	ƒ	Latin small letter f
103	g	g	g	g	Latin small letter g
104	h	h	h	h	Latin small letter h
105	í	í	í	í	Latin small letter i
106	ј	ј	ј	ј	Latin small letter j
107	ќ	ќ	ќ	ќ	Latin small letter k
108	љ	љ	љ	љ	Latin small letter l

109	m	m	m	m	Latin small letter m		135	‡	double dagger
110	n	n	n	n	Latin small letter n		136	^	modifier letter circumflex accent
111	o	o	o	o	Latin small letter o		137	%o	per mille sign
112	p	p	p	p	Latin small letter p		138	Š	Latin capital letter S with caron
113	q	q	q	q	Latin small letter q		139	„	single left-pointing angle quotation mark
114	r	r	r	r	Latin small letter r		140	Œ	Latin capital ligature OE
115	s	s	s	s	Latin small letter s		141		NOT USED
116	t	t	t	t	Latin small letter t		142	Ž	Latin capital letter Z with caron
117	u	u	u	u	Latin small letter u		143		NOT USED
118	v	v	v	v	Latin small letter v		144		NOT USED
119	w	w	w	w	Latin small letter w		145	‘	left single quotation mark
120	x	x	x	x	Latin small letter x		146	’	right single quotation mark
121	y	y	y	y	Latin small letter y		147	“	left double quotation mark
122	z	z	z	z	Latin small letter z		148	”	right double quotation mark
123	{	{	{	{	left curly bracket		149	•	bullet
124					vertical line		150	–	en dash
125	}	}	}	}	right curly bracket		151	—	em dash
126	~	~	~	~	tilde		152	˜	small tilde
127	DEL						153	™	trade mark sign
128	€				euro sign		154	š	Latin small letter s with caron
129					NOT USED		155	>	single right-pointing angle quotation mark
130	,				single low-9 quotation mark		156	œ	Latin small ligature oe
131	f				Latin small letter f with hook		157		NOT USED
132	„				double low-9 quotation mark		158	ž	Latin small letter z with caron
133	…				horizontal ellipsis		159	Ÿ	Latin capital letter Y with diaeresis
134	†				dagger		160		no-break space

161	¡	¡	¡	inverted exclamation mark		187	»	»	»	right-pointing double angle quotation mark
162	¢	¢	¢	cent sign		188	¼	¼	¼	vulgar fraction one quarter
163	£	£	£	pound sign		189	½	½	½	vulgar fraction one half
164	¤	¤	¤	currency sign		190	¾	¾	¾	vulgar fraction three quarters
165	¥	¥	¥	yen sign		191	¿	¿	¿	inverted question mark
166	¦	¦	¦	broken bar		192	À	À	À	Latin capital letter A with grave
167	§	§	§	section sign		193	Á	Á	Á	Latin capital letter A with acute
168	˝	˝	˝	diaeresis		194	Â	Â	Â	Latin capital letter A with circumflex
169	©	©	©	copyright sign		195	Ã	Ã	Ã	Latin capital letter A with tilde
170	ª	ª	ª	feminine ordinal indicator		196	Ä	Ä	Ä	Latin capital letter A with diaeresis
171	«	«	«	left-pointing double angle quotation mark		197	Å	Å	Å	Latin capital letter A with ring above
172	¬	¬	¬	not sign		198	Æ	Æ	Æ	Latin capital letter AE
173				soft hyphen		199	Ҫ	Ҫ	Ҫ	Latin capital letter C with cedilla
174	®	®	®	registered sign		200	È	È	È	Latin capital letter E with grave
175	—	—	—	macron		201	É	É	É	Latin capital letter E with acute
176	°	°	°	degree sign		202	Ê	Ê	Ê	Latin capital letter E with circumflex
177	±	±	±	plus-minus sign		203	Ë	Ë	Ë	Latin capital letter E with diaeresis
178	²	²	²	superscript two		204	Ì	Ì	Ì	Latin capital letter I with grave
179	³	³	³	superscript three		205	Í	Í	Í	Latin capital letter I with acute
180	ˊ	ˊ	ˊ	acute accent		206	Î	Î	Î	Latin capital letter I with circumflex
181	µ	µ	µ	micro sign		207	Ï	Ï	Ï	Latin capital letter I with diaeresis
182	¶	¶	¶	pilcrow sign		208	Ð	Ð	Ð	Latin capital letter Eth
183	·	·	·	middle dot		209	Ñ	Ñ	Ñ	Latin capital letter N with tilde
184	¸	¸	¸	cedilla		210	Ò	Ò	Ò	Latin capital letter O with grave
185	¹	¹	¹	superscript one		211	Ó	Ó	Ó	Latin capital letter O with acute
186	º	º	º	masculine ordinal indicator		212	Ô	Ô	Ô	Latin capital letter O with circumflex

213	Ö	Ö	Ö	Latin capital letter O with tilde	230	æ	æ	æ	Latin small letter ae
214	Ö	Ö	Ö	Latin capital letter O with diaeresis	231	ç	ç	ç	Latin small letter c with cedilla
215	×	×	×	multiplication sign	232	è	è	è	Latin small letter e with grave
216	Ø	Ø	Ø	Latin capital letter O with stroke	233	é	é	é	Latin small letter e with acute
217	Ù	Ù	Ù	Latin capital letter U with grave	234	ê	ê	ê	Latin small letter e with circumflex
218	Ú	Ú	Ú	Latin capital letter U with acute	235	ë	ë	ë	Latin small letter e with diaeresis
219	Û	Û	Û	Latin capital letter U with circumflex	236	ì	ì	ì	Latin small letter i with grave
220	Ü	Ü	Ü	Latin capital letter U with diaeresis	237	í	í	í	Latin small letter i with acute
221	Ý	Ý	Ý	Latin capital letter Y with acute	238	î	î	î	Latin small letter i with circumflex
222	Þ	Þ	Þ	Latin capital letter Thorn	239	ï	ï	ï	Latin small letter i with diaeresis
223	ß	ß	ß	Latin small letter sharp s	240	ð	ð	ð	Latin small letter eth
224	à	à	à	Latin small letter a with grave	241	ñ	ñ	ñ	Latin small letter n with tilde
225	á	á	á	Latin small letter a with acute	242	ò	ò	ò	Latin small letter o with grave
226	â	â	â	Latin small letter a with circumflex	243	ó	ó	ó	Latin small letter o with acute
227	ã	ã	ã	Latin small letter a with tilde	244	ô	ô	ô	Latin small letter o with circumflex
228	ä	ä	ä	Latin small letter a with diaeresis	245	ö	ö	ö	Latin small letter o with tilde
229	å	å	å	Latin small letter a with ring above	246	ö	ö	ö	Latin small letter o with diaeresis
230	æ	æ	æ	Latin small letter ae	247	÷	÷	÷	division sign
231	ç	ç	ç	Latin small letter c with cedilla	248	ø	ø	ø	Latin small letter o with stroke
232	è	è	è	Latin small letter e with grave	249	ù	ù	ù	Latin small letter u with grave
233	é	é	é	Latin small letter e with acute	250	ú	ú	ú	Latin small letter u with acute
234	ê	ê	ê	Latin small letter e with circumflex	251	û	û	û	Latin small letter u with circumflex
235	ë	ë	ë	Latin small letter e with diaeresis	252	ü	ü	ü	Latin small letter u with diaeresis
236	ì	ì	ì	Latin small letter i with grave	253	ý	ý	ý	Latin small letter y with acute
237	í	í	í	Latin small letter i with acute	254	þ	þ	þ	Latin small letter thorn
238	î	î	î	Latin small letter i with circumflex	255	ÿ	ÿ	ÿ	Latin small letter y with diaeresis

28.4 The ASCII Character Set

ASCII uses the values from 0 to 31 (and 127) for control characters. ASCII uses the values from 32 to 126 for letters, digits, and symbols. ASCII does not use the values from 128 to 255.

28.5 The ANSI Character Set (Windows-1252)

ANSI is identical to ASCII for the values from 0 to 127. ANSI has a proprietary set of characters for the values from 128 to 159. ANSI is identical to UTF-8 for the values from 160 to 255.

28.6 The ISO-8859-1 Character Set

8859-1 is identical to ASCII for the values from 0 to 127. 8859-1 does not use the values from 128 to 159. 8859-1 is identical to UTF-8 for the values from 160 to 255.

28.7 The UTF-8 Character Set

UTF-8 is identical to ASCII for the values from 0 to 127. UTF-8 does not use the values from 128 to 159. UTF-8 is identical to both ANSI and 8859-1 for the values from 160 to 255. UTF-8 continues from the value 256 with more than 10.000 different characters.

CHAPTER TWENTY-NINE

29.0 HTML UNIFOARM RE COURSE LOCATOR

A URL is another word for a web address. A URL can be composed of words (example.com), or an Internet Protocol (IP) address (192.68.20.50). Most people enter the name when surfing, because names are easier to remember than numbers.

29.1 URL - Uniform Resource Locator

Web browsers request pages from web servers by using a URL. When you click on a link in an HTML page, an underlying <a> tag points to an address on the web. A Uniform Resource Locator (URL) is used to address a document (or other data) on the web.

Example

scheme://host.domain:port/path/filename

Explanation:

- **scheme** - defines the **type** of Internet service (most common is **http**)
- **host** - defines the **domain host** (default host for http is **www**)
- **domain** - defines the Internet **domain name** (example.com)
- **port** - defines the **port number** at the host (default for http is **80**)
- **path** - defines a **path** at the server (If omitted: the root directory of the site)
- **filename** - defines the name of a document or resource
-

29.2 Common URL Schemes

The table below lists some common schemes:

Scheme	Short for	Used for
http	HyperText Transfer Protocol	Common web pages. Not encrypted
https	Secure HyperText Transfer Protocol	Secure web pages. Encrypted
ftp	File Transfer Protocol	Downloading or uploading files
file		A file on your computer

29.3 URL Encoding

URLs can only be sent over the Internet using the ASCII character-set. Since URLs often contain characters outside the ASCII set, the URL has to be converted into ASCII. URL encoding converts characters into a format that can be transmitted over the Internet. URL encoding replaces non ASCII characters with a "%" followed by hexadecimal digits. URLs cannot contain spaces. URL encoding normally replaces a space with a plus (+) sign, or %20.

Try It Yourself

If you click "Submit", the browser will URL encode the input before it is sent to the server. A page at the server will display the received input. Try some other input and click Submit again.

29.4 ASCII Encoding Examples

Your browser will encode input, according to the character-set used in your page.

The default character-set in HTML5 is UTF-8.

Character	From Windows-1252	From UTF-8
€	%80	%E2%82%AC
£	%A3	%C2%A3
©	%A9	%C2%A9
®	%AE	%C2%AE
À	%C0	%C3%80
Á	%C1	%C3%81
Â	%C2	%C3%82
Ã	%C3	%C3%83
Ä	%C4	%C3%84
Å	%C5	%C3%85

CHAPTER THIRTY

30.0 HTML AND XHTML

XHTML is HTML written as XML.

30.1 What Is XHTML?

- XHTML stands for EXtensible HyperText Markup Language
- XHTML is almost identical to HTML
- XHTML is stricter than HTML
- XHTML is HTML defined as an XML application
- XHTML is supported by all major browsers

30.2 Why XHTML?

Many pages on the internet contain "bad" HTML.

This HTML code works fine in most browsers (even if it does not follow the HTML rules):

```
<html>  
<head>  
<title>This is bad HTML</title>  
<body>  
<h1>Bad HTML  
<p>This is a paragraph  
</body>
```

Today's market consists of different browser technologies. Some browsers run on computers, and some browsers run on mobile phones or other small devices. Smaller devices

often lack the resources or power to interpret "bad" markup. XML is a markup language where documents must be marked up correctly (be "well-formed"). If you want to study XML, please read our XML tutorial. By combining the strengths of HTML and XML, XHTML was developed. XHTML is HTML redesigned as XML.

30.3 The Most Important Differences from HTML:

30.3.1 Document Structure

- XHTML DOCTYPE is **mandatory**
- The xmlns attribute in <html> is **mandatory**
- <html>, <head>, <title>, and <body> are **mandatory**

30.3.2 XHTML Elements

- XHTML elements must be **properly nested**
- XHTML elements must always be **closed**
- XHTML elements must be in **lowercase**
- XHTML documents must have **one root element**

30.3.3 XHTML Attributes

- Attribute names must be in **lower case**
- Attribute values must be **quoted**
- Attribute minimization is **forbidden**

30.4 <!DOCTYPE> Is Mandatory

An XHTML document must have an XHTML DOCTYPE declaration. A complete list of all the XHTML Docypes is found in our HTML Tags Reference. The <html>, <head>, <title>,

and <body> elements must also be present, and the xmlns attribute in <html> must specify the xml namespace for the document.

This example shows an XHTML document with a minimum of required tags:

```
<!DOCTYPE html PUBLIC "-//ifarouqtech//DTD XHTML 1.0 Transitional//EN"  
"http://www.example.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">  
  
<html xmlns="http://www.example.org/1999/xhtml">  
  
<head>  
  
    <title>Title of document</title>  
  
</head>  
  
<body>  
  
    some content  
  
</body>  
  
</html>
```

30.5 XHTML Elements Must Be Properly Nested

In HTML, some elements can be improperly nested within each other, like this:

```
<b><i>This text is bold and italic</b></i>
```

In XHTML, all elements must be properly nested within each other, like this:

```
<b><i>This text is bold and italic</i></b>
```

30.6 XHTML Elements Must Always Be Closed

This is wrong:

```
<p>This is a paragraph  
<p>This is another paragraph
```

This is correct:

```
<p>This is a paragraph</p>  
<p>This is another paragraph</p>
```

30.7 Empty Elements Must Also Be Closed

This is wrong:

```
A break: <br>  
A horizontal rule: <hr>  
An image: 
```

This is correct:

```
A break: <br />  
A horizontal rule: <hr />  
An image: 
```

30.8 XHTML Elements Must Be In Lower Case

This is wrong:

```
<BODY>  
<P>This is a paragraph</P>  
</BODY>
```

This is correct:

```
<body>  
<p>This is a paragraph</p>  
</body>
```

30.9 XHTML Attribute Names Must Be In Lower Case

This is wrong:

```
<table WIDTH="100%">
```

This is correct:

```
<table width="100%">
```

30.10 Attribute Values Must Be Quoted

This is wrong:

```
<table width=100%>
```

This is correct:

```
<table width="100%">
```

30.11 Attribute Minimization Is Forbidden

Wrong:

```
<input type="checkbox" name="vehicle" value="car" checked />
```

Correct:

```
<input type="checkbox" name="vehicle" value="car" checked="checked" />
```

Wrong:

```
<input type="text" name="lastname" disabled />
```

Correct:

```
<input type="text" name="lastname" disabled="disabled" />
```

30.12 How to Convert from HTML to XHTML

1. Add an XHTML <!DOCTYPE> to the first line of every page

2. Add an xmlns attribute to the html element of every page
3. Change all element names to lowercase
4. Close all empty elements
5. Change all attribute names to lowercase
6. Quote all attribute values

CHAPTER THIRTY-ONE

31.0 HTML FORMS

31.1 The <form> Element

HTML forms are used to collect user input. The **<form>** element defines an HTML form:

Example

```
<form>
  .
  form elements
  .
</form>
```

HTML forms contain **form elements**.

Form elements are different types of input elements, checkboxes, radio buttons, submit buttons, and more.

31.2 The <input> Element

The **<input>** element is the most important **form element**. The **<input>** element has many variations, depending on the **type** attribute.

Here are the types used in this chapter:

Type	Description
text	Defines normal text input
radio	Defines radio button input (for selecting one of many choices)
submit	Defines a submit button (for submitting the form)

You will learn a lot more about input types later in this tutorial.

31.3 Text Input

`<input type="text">` defines a one-line input field for **text input**:

Example

```
<!DOCTYPE html>
<html>
  <body>
    <form>
      First name:<br>
      <input type="text" name="firstname">
      <br>
      Last name:<br>
      <input type="text" name="lastname">
    </form>
    <p>Note that the form itself is not visible.</p>
    <p>Also note that the default width of a text field is 20 characters.</p>
  </body>
</html>
```

Note: The form itself is not visible. Also note that the default width of a text field is 20 characters.

31.4 Radio Button Input

`<input type="radio">` defines a **radio button**.

Radio buttons let a user select ONE of a limited number of choices:

Example

```
<!DOCTYPE html>
<html>
<body>
<form>
<input type="radio" name="sex" value="male" checked>Male
<br>
<input type="radio" name="sex" value="female">Female
</form>
</body>
</html>
```

31.5 The Submit Button

`<input type="submit">` defines a button for **submitting** a form to a **form-handler**. The form-handler is typically a server page with a script for processing input data.

The form-handler is specified in the form's **action** attribute:

Example

```
<!DOCTYPE html>
<html>
<body>
<form action="action_page.php">
First name:<br>
<input type="text" name="firstname" value="Mickey">
<br>
```

```

Last name:<br>

<input type="text" name="lastname" value="Mouse">

<br><br>

<input type="submit" value="Submit">

</form>

<p>If you click "Submit", the form-data will be sent to a page called "action_page.php".</p>

</body>

</html>

```

31.6 The Action Attribute

The **action attribute** defines the action to be performed when the form is submitted. The common way to submit a form to a server, is by using a submit button. Normally, the form is submitted to a web page on a web server.

In the example above, a server-side script is specified to handle the submitted form:

```
<form action="action_page.php">
```

If the action attribute is omitted, the action is set to the current page.

31.7 The Method Attribute

The **method attribute** specifies the HTTP method (**GET** or **POST**) to be used when submitting the forms:

```
<form action="action_page.php" method="GET">
```

or:

```
<form action="action_page.php" method="POST">
```

31.7.1 When to Use GET?

You can use GET (the default method):

If the form submission is passive (like a search engine query), and without sensitive information.

When you use GET, the form data will be visible in the page address:

`action_page.php?firstname=John&lastname=Mark`

GET is best suited to short amounts of data. Size limitations are set in your browser.

31.7.2 When to Use POST?

You should use POST:

If the form is updating data, or includes sensitive information (password).

POST offers better security because the submitted data is not visible in the page address.

31.8 The Name Attribute

To be submitted correctly, each input field must have a name attribute.

This example will only submit the "Last name" input field:

Example

```
<!DOCTYPE html>
<html>
<body>
<form action="action_page.php">
First name:<br>
<input type="text" value="John">
```

```

<br>

Last name:<br>

<input type="text" name="lastname" value="Mark">

<br><br>

<input type="submit" value="Submit">

</form>

<p>If you click "Submit", the form-data will be sent to a page called "action_page.php".</p>

<p>The first name will not be submitted, because the input element does not have a name attribute.</p>

</body>

</html>

```

31.9 Grouping Form Data with <fieldset>

The **<fieldset>** element groups related data in a form. The **<legend>** element defines a caption for the **<fieldset>** element.

Example

```

<!DOCTYPE html>

<html>

<body>

<form action="action_page.php">

<fieldset>

<legend>Personal information:</legend>

First name:<br>

<input type="text" name="firstname" value="John">

<br>

```

```

Last name:<br>

<input type="text" name="lastname" value="Mark">

<br><br>

<input type="submit" value="Submit">

</fieldset>

</form>

</body>

</html>

```

31.10 HTML Form Attributes

An HTML <form> element, with all possible attributes set, will look like this:

Example

```

<form action="action_page.php" method="GET" target="_blank" accept-
charset="UTF-8"
      enctype="application/x-www-form-urlencoded" autocomplete="off" novalidate>
    .
    form elements
    .
  </form>

```

Here is the list of <form> attributes:

Attribute	Description
accept-charset	Specifies the charset used in the submitted form (default: the page charset).
action	Specifies an address (url) where to submit the form (default: the submitting page).
autocomplete	Specifies if the browser should autocomplete the form (default: on).
enctype	Specifies the encoding of the submitted data (default: is url-encoded).
method	Specifies the HTTP method used when submitting the form (default: GET).
name	Specifies a name used to identify the form (for DOM usage: document.forms.name).
novalidate	Specifies that the browser should not validate the form.
target	Specifies the target of the address in the action attribute (default: _self).

Send e-mail from a form

How to send e-mail from a form.

```
<!DOCTYPE html>

<html>

<body>

<h2>Send e-mail to someone@example.com:</h2>

<form action="MAILTO:someone@example.com" method="post"
enctype="text/plain">

    Name:<br>

    <input type="text" name="name" value="your name"><br>

    E-mail:<br>

    <input type="text" name="mail" value="your email"><br>

    Comment:<br>

    <input type="text" name="comment" value="your comment"
size="50"><br><br>

    <input type="submit" value="Send">

    <input type="reset" value="Reset">

</form>

</body>

</html>
```

CHAPTER THIRTY-TWO

32.0 HTML FORM ELEMENTS

This chapter describes all HTML form elements.

32.1 The `<input>` Element

The most important form element is the `<input>` element. The `<input>` element can vary in many ways, depending on the `type` attribute.

All HTML input types are covered in the next chapter.

32.2 The `<select>` Element (Drop-Down List)

The `<select>` element defines a **drop-down** list:

Example

```
<!DOCTYPE html>

<html>

<body>

<form action="action_page.php">

    <select name="fruits">

        <option value="Apple">Apple</option>

        <option value="Pineapple">Pineapple</option>

        <option value="Orange">Orange</option>

        <option value="Water Melon">Water Melon</option>

        <option value="Mango">Mango</option>

    </select>

    <br><br>
```

```
<input type="submit">  
</form>  
</body>  
</html>
```

The **<option>** elements defines the options to select. The list will normally show the first item as selected.

You can add a selected attribute to define a predefined option.

Example

```
<!DOCTYPE html>  
<html>  
<body>  
  <p>You can preselect an option with the selected attribute.</p>  
  <form action="action_page.php">  
    <select name="fruits">  
      <option value="Apple">Apple</option>  
      <option value="Pineapple">Pineapple</option>  
      <option value="Orange">Orange</option>  
      <option value="Water Melon" selected>Water  
        Melon</option>  
      <option value="Mango">Mango</option>  
    </select>  
    <br><br>  
    <input type="submit">  
</form>  
</body>
```

```
</html>
```

32.3 The <textarea> Element

The <textarea> element defines a multi-line input field (**a text area**):

Example

```
<!DOCTYPE html>

<html>

  <body>

    <form action="action_page.php">

      <textarea name="message" rows="10" cols="30">

        The cat was playing in the garden.

      </textarea>

      <br><br>

      <input type="submit">

    </form>

  </body>

</html>
```

32.4 The <button> Element

The <button> element defines a clickable **button**:

Example

```
<!DOCTYPE html>

<html>

  <body>
```

```

<button type="button" onclick="alert('Hello World!')">Click
Me!
</button>

</body>

</html>

```

32.5 HTML5 Form Elements

HTML5 added the following form elements:

- <**datalist**>
- <**keygen**>
- <**output**>

By default, browsers do not display unknown elements. New elements will not destroy your page.

32.5.1 HTML5 <**datalist**> Element

The <**datalist**> element specifies a list of pre-defined options for an <**input**> element. Users will see a drop-down list of pre-defined options as they input data. The **list** attribute of the <**input**> element, must refer to the **id** attribute of the <**datalist**> element.

Example

An <**input**> element with pre-defined values in a <**datalist**>:

```

<!DOCTYPE html>

<html>

<body>

<form action="action_page.php">

<input list="clubs" name="club">

```

```

<datalist id="clubs">

    <option value="Man United">
    <option value="Barcelona">
    <option value="Real Madrid">
    <option value="Chelsea">
    <option value="Bayern Munich">

</datalist>

<input type="submit">

</form>

<p><b>Note:</b> The datalist tag is not supported in Safari or IE9 (and earlier).</p>

</body>

</html>

```

32.5.2 HTML5 <keygen> Element

The purpose of the <keygen> element is to provide a secure way to authenticate users. The <keygen> element specifies a key-pair generator field in a form. When the form is submitted, two keys are generated, one private and one public.

The private key is stored locally, and the public key is sent to the server. The public key could be used to generate a client certificate to authenticate the user in the future.

Example

A form with a keygen field:

```

<!DOCTYPE html>

<html>

<body>

```

```

<form action="action_page.php">

    Username:
    <br>
    <input type="text" name="user">
    <br><br>

    Encryption:
    <br>
    <keygen name="security">
    <br><br>
    <input type="submit">

</form>

</body>

</html>

```

32.5.3 HTML5 <output> Element

The <output> element represents the result of a calculation (like one performed by a script).

Example

Perform a calculation and show the result in an <output> element:

```

<!DOCTYPE html>

<html>

    <body>

        <form action="action_page.php"
            oninput="x.value=parseInt(a.value)+parseInt(b.value)">

            0
            <input type="range" id="a" name="a" value="50">

```

```
100 +  
<input type="number" id="b" name="b" value="50">  
=  
<output name="x" for="a b"></output>  
<br><br>  
<input type="submit">  
</form>  
</body>  
</html>
```

HTML Form Elements



= new in HTML5.

Tag	Description
<u><form></u>	Defines an HTML form for user input
<u><input></u>	Defines an input control
<u><textarea></u>	Defines a multiline input control (text area)
<u><label></u>	Defines a label for an <input> element
<u><fieldset></u>	Groups related elements in a form
<u><legend></u>	Defines a caption for a <fieldset> element
<u><select></u>	Defines a drop-down list
<u><optgroup></u>	Defines a group of related options in a drop-down list
<u><option></u>	Defines an option in a drop-down list
<u><button></u>	Defines a clickable button
<u><datalist></u>	Specifies a list of pre-defined options for input controls
<u><keygen></u>	Defines a key-pair generator field (for forms)
<u><output></u>	Defines the result of a calculation

CHAPTER THIRTY-THREE

33.0 HTML INPUT TYPES

This chapter describes the input types of the <input> element.

33.1 Input Type: text

```
<input type="text"> defines a one-line input field for text input:  
  
<!DOCTYPE html>  
  
<html>  
  
    <body>  
  
        <form action="action_page.php">  
  
            First name:<br>  
  
            <input type="text" name="firstname">  
  
            <br>  
  
            Last name:<br>  
  
            <input type="text" name="lastname">  
  
            <br><br>  
  
            <input type="submit">  
  
        </form>  
  
        <p>Note that the form itself is not visible.</p>  
  
        <p>Also note that the default width of a text field is 20 characters.</p>  
  
    </body>  
  
</html>
```

33.2 Input Type: password

`<input type="password">` defines a password field:

```
<!DOCTYPE html>

<html>

  <body>

    <form action="#">

      User name:<br>

      <input type="text" name="userid">

      <br>

      User password:<br>

      <input type="password" name="psw">

    </form>

    <p>The characters in a password field are masked (shown as asterisks or circles).</p>

  </body>

</html>
```

The characters in a password field are masked (shown as asterisks or circles).

33.3 Input Type: submit

`<input type="submit">` defines a button for submitting form input to a form-handler.

The form-handler is typically a server page with a script for processing input data.

The form-handler is specified in the form's action attribute:

Example

```
<!DOCTYPE html>

<html>

<body>

<form action="action_page.php">

    First name:<br>

    <input type="text" name="firstname" value="Mickey">

    <br>

    Last name:<br>

    <input type="text" name="lastname" value="Mouse">

    <br><br>

    <input type="submit" value="Submit">

</form>

<p>If you click "Submit", the form-data will be sent to a page called "action_page.php".</p>

</body>

</html>
```

If you omit the submit button's value attribute, the button will get a default text:

33.4 Input Type: radio

`<input type="radio">` defines a radio button.

Radio buttons let a user select ONLY ONE of a limited number of choices:

```
<!DOCTYPE html>

<html>
```

```

<body>

<form action = “action_page.php”>

    <input type="radio" name="sex" value="male" checked>Male

    <br>

    c

    <br><br>

    <input type="submit">

</form>

</body>

</html>

```

33.5 Input Type: checkbox

<input type="checkbox"> defines a checkbox.

Checkboxes let a user select ZERO or MORE options of a limited number of choices.

Example

```

<!DOCTYPE html>

<html>

<body>

<form action="action_page.php">

    <input type="checkbox" name="vehicle" value="Bike">I have
    a bike

    <br>

    b

    <br><br>

    <input type="submit">

```

```
</form>

</body>

</html>
```

33.6 Input Type: button

<input type="button"> defines a button:

```
<!DOCTYPE html>

<html>

<body>

    <input type="button" value="Submit" />

</body>

</html>
```

33.7 HTML5 Input Types

HTML5 added several new input types:

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

- url
- week

Input types, not supported by old web browsers, will behave as input type text.

33.7.1 Input Type: number

The `<input type="number">` is used for input fields that should contain a numeric value. You can set restrictions on the numbers. Depending on browser support, the restrictions can apply to the input field.

Example

```
<!DOCTYPE html>
<html>
  <body>
    <p>
      Depending on browser support:<br>
      Numeric restrictions will apply in the input field.
    </p>
    <form action="action_page.php">
      Quantity (between 1 and 5):
      <input type="number" name="quantity" min="1" max="5">
      <input type="submit">
    </form>
    <p><b>Note:</b> type="number" is not supported in IE9 and earlier.</p>
  </body>
```

```
</html>
```

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

You will learn more about input restrictions in the next chapter.

Example

```
<!DOCTYPE html>

<html>

  <body>

    <p>
      Depending on browser support:<br>
      Fixed steps will apply in the input field.
    </p>
```

```

<form action="action_page.php">

    Quantity:

    <input type="number" name="points"
        min="0" max="100" step="10" value="30">

    <input type="submit">

</form>

<p>

    <b>Note:</b><input type="number" is not supported in IE9 and earlier.

</p>

</body>

</html>

```

33.7.2 Input Type: date

The `<input type="date">` is used for input fields that should contain a date. Depending on browser support, a date picker can show up in the input field.

Example

```

<!DOCTYPE html>

<html>

    <body>

        <p>

            Depending on browser support:<br>

            A date picker can pop-up when you enter the input field.

        </p>

        <form action="action_page.php">

            Birthday:

```

```

<input type="date" name="bday">

<input type="submit">

</form>

<p><b>Note:</b> type="date" is not supported in Internet Explorer.</p>

</body>

</html>

```

You can add restrictions to the input:

Example

```

<!DOCTYPE html>

<html>

<body>

<form action="action_page.php">

    Enter a date before 1980-01-01:<br>

    c<br><br>

    Enter a date after 2000-01-01:<br>

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

    <input type="submit">

</form>

<p><strong>Note:</strong>

    type="date" is not supported in Internet Explorer.</p>

</body>

</html>

```

33.7.3 Input Type: color

The `<input type="color">` is used for input fields that should contain a color. Depending on browser support, a color picker can show up in the input field.

Example

```
<!DOCTYPE html>

<html>

    <body>

        <p>
            Depending on browser support:<br>
            A color picker can pop-up when you enter the input field.
        </p>

        <form action="action_page.php">
            Select your favorite color:
            <input type="color" name="favcolor" value="#000000">
            <input type="submit">
        </form>

        <p><b>Note:</b> type="color" is not supported in Internet Explorer.</p>
    </body>

</html>
```

33.7.4 Input Type: range

The `<input type="range">` is used for input fields that should contain a value within a range. Depending on browser support, the input field can be displayed as a slider control.

Example

```
<!DOCTYPE html>

<html>

<body>

<p>

    Depending on browser support:<br>

    The input type "range" can be displayed as a slider control.

</p>

<form action="action_page.php" method="get">

    Points:

    <input type="range" name="points" min="0" max="10">

    <input type="submit">

</form>

<p>

    <b>Note:</b>

    type="range" is not supported in Internet Explorer 9 and earlier versions.

</p>

</body>

</html>
```

You can use the following attributes to specify restrictions: min, max, step, value.

33.7.5 Input Type: month

The `<input type="month">` allows the user to select a month and year. Depending on browser support, a date picker can show up in the input field.

Example

```
<!DOCTYPE html>

<html>

<body>

<p>

    Depending on browser support:<br>

    A date picker can pop-up when you enter the input field.

</p>

<form action="action_page.php">

    Birthday (month and year):

    <input type="month" name="bdaysmonth">

    <input type="submit">

</form>

<p><b>Note:</b>

    type="month" is not supported in Internet Explorer.

</p>

</body>

</html>
```

33.7.6 Input Type: week

The `<input type="week">` allows the user to select a week and year. Depending on browser support, a date picker can show up in the input field.

Example

```
<!DOCTYPE html>

<html>

<body>

<p>

    Depending on browser support:<br>

    A date picker can pop-up when you enter the input field.

</p>

<form action="action_page.php">

    Select a week:

    <input type="week" name="year_week">

    <input type="submit">

</form>

<p><b>Note:</b>

    type="week" is not supported in Internet Explorer.

</p>

</body>

</html>
```

13.7.7 Input Type: time

The `<input type="time">` allows the user to select a time (no time zone). Depending on browser support, a time picker can show up in the input field.

Example

```
<!DOCTYPE html>

<html>

<body>

<p>
    Depending on browser support:<br>
    A time picker might pop-up when you enter the input field.
</p>

<form action="action_page.php">
    Select a time:
    <input type="time" name="usr_time">
    <input type="submit">
</form>

<p><b>Note:</b>
    type="time" is not supported in Firefox or Internet Explorer.
</p>

</body>

</html>
```

33.7.8 Input Type: datetime

The `<input type="datetime">` allows the user to select a date and time (with time zone).

Depending on browser support, a date picker can show up in the input field.

Example

```
<!DOCTYPE html>

<html>

<body>

<p>
    Depending on browser support:<br>
    A date picker can pop-up when you enter the input field.
</p>

<form action="action_page.php">
    Birthday (date and time):
    <input type="datetime" name="bdytime">
    <input type="submit">
</form>

<p><b>Note:</b> type="datetime" is not supported in Chrome, Firefox, Safari, or Internet Explorer.</p>

</body>

</html>
```

33.7.9 Input Type: datetime-local

The `<input type="datetime-local">` allows the user to select a date and time (no time zone). Depending on browser support, a date picker can show up in the input field.

Example

```
<!DOCTYPE html>
<html>
<body>
<p>
    Depending on browser support:<br>
    A date picker can pop-up when you enter the input field.
</p>
<form action="action_page.php">
    Birthday (date and time):
    <input type="datetime-local" name="bdaytime">
    <input type="submit" value="Send">
</form>
<p><b>Note:</b>
    type="datetime-local" is not supported in Firefox and Internet
    Explorer.
</p>
</body>
</html>
```

33.7.10 Input Type: email

The `<input type="email">` is used for input fields that should contain an e-mail address.

Depending on browser support, the e-mail address can be automatically validated when submitted.

Some smartphones recognize the email type, and adds ".com" to the keyboard to match email input.

Example

```
<!DOCTYPE html>

<html>

<body>

<form action="action_page.php">

    E-mail:

    <input type="email" name="email">

    <input type="submit">

</form>

<p>

<b>Note:</b><input type="email" name="email"> is not supported in IE9 and earlier.</p>

</body>

</html>
```

33.7.11 Input Type: search

The `<input type="search">` is used for search fields (a search field behaves like a regular text field).

Example

```
<!DOCTYPE html>

<html>

<body>

<form action="action_page.php">
```

```

Search Google:

<input type="search" name="googlesearch">

<input type="submit">

</form>

</body>

</html>

```

33.7.12 Input Type: tel

The `<input type="tel">` is used for input fields that should contain a telephone number. The tel type is currently supported only in Safari 8.

Example

```

<!DOCTYPE html>
<html>
  <body>
    <form action="action_page.php">
      Telephone:
      <input type="tel" name="usrtel">
      <input type="submit">
    </form>
    <p><b>Note:</b> type="tel" is supported only in Safari 8.</p>
  </body>
</html>

```

33.7.13 Input Type: url

The `<input type="url">` is used for input fields that should contain a URL address. Depending on browser support, the url field can be automatically validated when submitted. Some smartphones recognize the url type, and adds ".com" to the keyboard to match url input.

Example

```
<!DOCTYPE html>
<html>
  <body>
    <form action="action_page.php">
      Add your homepage:
      <input type="url" name="homepage">
      <input type="submit">
    </form>
    <p><b>Note:</b>
      The type="url" is not supported in IE9 and earlier versions.
    </p>
  </body>
</html>
```

CHAPTER THIRTY-FOUR

34.0 HTML INPUT ATTRIBUTES

34.1 The value Attribute

The **value** attribute specifies the initial value for an input field:

Example

```
<!DOCTYPE html>
<html>
  <body>
    <form action="#">
      First name:<br>
      <input type="text" name="firstname" value="John">
      <br>
      Last name:<br>
      <input type="text" name="lastname">
    </form>
  </body>
</html>
```

34.2 The readonly Attribute

The **readonly** attribute specifies that the input field is read only (cannot be changed):

Example

```
<!DOCTYPE html>
<html>
  <body>
    <form action="#">
      First name:<br>
      <input type="text" name="firstname" value ="John" readonly>
      <br>
      Last name:<br>
```

```

<input type="text" name="lastname">
</form>
</body>
</html>

```

The **readonly** attribute does not need a value. It is the same as writing `readonly="readonly"`.

34.3 The disabled Attribute

The **disabled** attribute specifies that the input field is disabled. A disabled element is un-useable and un-clickable. Disabled elements will not be submitted.

Example

```

<!DOCTYPE html>
<html>
<body>
<form action="#">
    First name:<br>
    <input type="text" name="firstname" value ="John" disabled>
    <br>
    Last name:<br>
    <input type="text" name="lastname">
</form>
</body>
</html>

```

The **disabled** attribute does not need a value. It is the same as writing `disabled="disabled"`.

34.4 The size Attribute

The **size** attribute specifies the size (in characters) for the input field:

Example

```
<!DOCTYPE html>
<html>
  <body>
    <form action="#">
      First name:<br>
      <input type="text" name="firstname" value="John" size="40">
      <br>
      Last name:<br>
      <input type="text" name="lastname">
    </form>
  </body>
</html>
```

34.5 The maxlength Attribute

The **maxlength** attribute specifies the maximum allowed length for the input field:

Example

```
<!DOCTYPE html>
<html>
  <body>
    <form action="#">
      First name:<br>
      <input type="text" name="firstname" maxlength="10">
      <br>
      Last name:<br>
      <input type="text" name="lastname">
    </form>
  </body>
</html>
```

With a maxlen attribute, the input control will not accept more than the allowed number of characters.

The attribute does not provide any feedback. If you want to alert the user, you must write JavaScript code.

Input restrictions are not foolproof. JavaScript provides many ways to add illegal input.

To safely restrict input, restrictions must be checked by the receiver (the server) as well.

34.6 HTML5 Attributes

HTML5 added the following attributes for <input>:

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

- step

34.6.1 The autocomplete Attribute

The autocomplete attribute specifies whether a form or input field should have autocomplete on or off.

When autocomplete is on, the browser automatically complete values based on values that the user has entered before.

Tip: It is possible to have autocomplete "on" for the form, and "off" for specific input fields, or vice versa.

The autocomplete attribute works with <form> and the following <input> types: text, search, url, tel, email, password, datepickers, range, and color.

Example

An HTML form with autocomplete on (and off for one input field):

```
<!DOCTYPE html>
<html>
<body>
<form action="action_page.php" autocomplete="on">
    First name:<input type="text" name="fname"><br>
    Last name: <input type="text" name="lname"><br>
    E-mail:      <input      type="email"      name="email"
               autocomplete="off"><br>
    <input type="submit">
</form>
```

```
<p>Fill in and submit the form, then reload the page to see how  
autocomplete works.</p>
```

```
<p>Notice that autocomplete is "on" for the form, but "off" for the e-  
mail field.</p>
```

```
</body>
```

```
</html>
```

Tip: In some browsers you may need to activate the autocomplete function for this to work.

34.6.2 The novalidate Attribute

The novalidate attribute is a <form> attribute. When present, novalidate specifies that form data should not be validated when submitted.

Example

Indicates that the form is not to be validated on submit:

```
<!DOCTYPE html>  
  
<html>  
  
<body>  
  
<form action="action_page.php" novalidate>  
  
    E-mail: <input type="email" name="user_email">  
  
    <input type="submit">  
  
</form>  
  
<p><strong>Note:</strong> The novalidate attribute of the form tag is  
not supported in Internet Explorer 9 and earlier versions, or in  
Safari.</p>  
  
</body>  
  
</html>
```

34.6.3 The autofocus Attribute

The autofocus attribute is a Boolean attribute. When present, it specifies that an <input> element should automatically get focus when the page loads.

Example

Let the "First name" input field automatically get focus when the page loads:

```
<!DOCTYPE html>

<html>

  <body>

    <form action="action_page.php">

      First name:<input type="text" name="fname" autofocus><br>

      Last name: <input type="text" name="lname"><br>

      <input type="submit">

    </form>

    <p><strong>Note:</strong> The autofocus attribute of the input tag is
    not supported in Internet Explorer 9 and earlier versions.</p>

  </body>

</html>
```

34.6.4 The form Attribute

The form attribute specifies one or more forms an <input> element belongs to.

Tip: To refer to more than one form, use a space-separated list of form ids.

Example

An input field located outside the HTML form (but still a part of the form):

```

<!DOCTYPE html>

<html>

<body>

<form action="action_page.php" id="form1">

    First name: <input type="text" name="fname"><br>

    <input type="submit" value="Submit">

</form>

<p>The "Last name" field below is outside the form element, but still
part of the form.</p>

    Last name: <input type="text" name="lname" form="form1">

</body>

</html>

```

34.6.5 The formaction Attribute

The formaction attribute specifies the URL of a file that will process the input control when the form is submitted. The formaction attribute overrides the action attribute of the <form> element.

The formaction attribute is used with type="submit" and type="image".

Example

An HTML form with two submit buttons, with different actions:

```

<!DOCTYPE html>

<html>

<body>

<form action="action_page.php">

```

```

First name: <input type="text" name="fname"><br>

Last name: <input type="text" name="lname"><br>

<input type="submit" value="Submit"><br>

<input type="submit" formaction="demo_admin.asp"
      value="Submit as admin">

</form>

<p><strong>Note:</strong> The formaction attribute of the input tag is
not supported in Internet Explorer 9 and earlier versions.</p>

</body>

</html>

```

34.6.6 The formenctype Attribute

The formenctype attribute specifies how the form-data should be encoded when submitting it to the server (only for forms with method="post"). The formenctype attribute overrides the enctype attribute of the <form> element.

The formenctype attribute is used with type="submit" and type="image".

Example

Send form-data that is default encoded (the first submit button), and encoded as "multipart/form-data" (the second submit button):

```

<!DOCTYPE html>

<html>

<body>

<form action="" method="post">

First name: <input type="text" name="fname"><br>

<input type="submit" value="Submit">

```

```

<input type="submit" formenctype="multipart/form-data"
value="Submit as Multipart/form-data">

</form>

<p><strong>Note:</strong> The formenctype attribute of the input tag
is not supported in Internet Explorer 9 and earlier versions.</p>

</body>

</html>

```

34.6.7 The formmethod Attribute

The formmethod attribute defines the HTTP method for sending form-data to the action URL. The formmethod attribute overrides the method attribute of the <form> element.

The formmethod attribute can be used with type="submit" and type="image".

Example

The second submit button overrides the HTTP method of the form:

```

<!DOCTYPE html>

<html>

<body>

<form action="action_page.php" method="get">

First name: <input type="text" name="fname"><br>

Last name: <input type="text" name="lname"><br>

<input type="submit" value="Submit">

<input type="submit" formmethod="post"
formaction="demo_post.asp" value="Submit using POST">

</form>

```

<p>Note: The formmethod attribute of the input tag
is not supported in Internet Explorer 9 and earlier versions.</p>

```
</body>  
</html>
```

34.6.8 The formnovalidate Attribute

The novalidate attribute is a boolean attribute. When present, it specifies that the <input> element should not be validated when submitted. The formnovalidate attribute overrides the novalidate attribute of the <form> element.

The formnovalidate attribute can be used with type="submit".

Example

A form with two submit buttons (with and without validation):

```
<!DOCTYPE html>  
<html>  
  <body>  
    <form action="action_page.php">  
      E-mail: <input type="email" name="userid"><br>  
      <input type="submit" value="Submit"><br>  
      <input type="submit" formnovalidate value="Submit without validation">  
    </form>  
  
    <p><strong>Note:</strong> The formnovalidate attribute of the input tag is not supported in Internet Explorer 9 and earlier versions, or in Safari.</p>  
  </body>  
</html>
```

34.6.9 The formtarget Attribute

The formtarget attribute specifies a name or a keyword that indicates where to display the response that is received after submitting the form. The formtarget attribute overrides the target attribute of the <form> element.

The formtarget attribute can be used with type="submit" and type="image".

Example

A form with two submit buttons, with different target windows:

```
<!DOCTYPE html>

<html>

<body>

<form action="action_page.php">

    First name: <input type="text" name="fname"><br>

    Last name: <input type="text" name="lname"><br>

    <input type="submit" value="Submit as normal">

    <input type="submit" formtarget="_blank" value="Submit to a
    new window/tab">

</form>

<p><strong>Note:</strong> The formtarget attribute of the input tag is
not supported in Internet Explorer 9 and earlier versions.</p>

</body>

</html>
```

34.6.10 The height and width Attributes

The height and width attributes specify the height and width of an <input> element.

The height and width attributes are only used with <input type="image">.

Always specify the size of images. If the browser does not know the size, the page will flicker while images load.

Example

Define an image as the submit button, with height and width attributes:

```
<!DOCTYPE html>
<html>
    <body>
        <form action="action_page.php">
            First name: <input type="text" name="fname"><br>
            Last name: <input type="text" name="lname"><br>
            <input type="image" src="img_submit.gif" alt="Submit"
width="48" height="48">
        </form>
        <p><b>Note:</b> The input type="image" sends the X and Y
coordinates of the click that activated the image button.</p>
    </body>
</html>
```

34.6.11 The list Attribute

The list attribute refers to a <datalist> element that contains pre-defined options for an <input> element.

Example

An <input> element with pre-defined values in a <datalist>:

```
<!DOCTYPE html>

<html>

<body>

<form action="action_page.php" method="get">

    <input list="browsers" name="browser">

    <datalist id="browsers">

        <option value="Internet Explorer">

        <option value="Firefox">

        <option value="Chrome">

        <option value="Opera">

        <option value="Safari">

    </datalist>

    <input type="submit">

</form>

<p><b>Note:</b> The datalist tag is not supported in Internet Explorer 9 and earlier versions, or in Safari.</p>

</body>

</html>
```

34.6.12 The min and max Attributes

The min and max attributes specify the minimum and maximum value for an <input> element.

The min and max attributes work with the following input types: number, range, date, datetime, datetime-local, month, time and week.

Example

<input> elements with min and max values:

```
<!DOCTYPE html>
<html>
  <body>
    <form action="action_page.php">
      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>
      Quantity (between 1 and 5):
      <input type="number" name="quantity" min="1" max="5"><br>
      <input type="submit">
    </form>
    <p><strong>Note:</strong> The max and min attributes of the input tag is not supported in Internet Explorer 9 and earlier versions, or in Firefox.</p>
    <p><strong>Note:</strong> The max and min attributes will not work for dates and time in Internet Explorer 10, since IE 10 does not support these input types.</p>
  </body>
</html>
```

34.6.13 The multiple Attribute

The multiple attribute is a boolean attribute. When present, it specifies that the user is allowed to enter more than one value in the <input> element.

The multiple attribute works with the following input types: email, and file.

Example

A file upload field that accepts multiple values:

```
<!DOCTYPE html>

<html>

    <body>

        <form action="action_page.php">

            Select images: <input type="file" name="img" multiple>

            <input type="submit">

        </form>

        <p>Try selecting more than one file when browsing for files.</p>

        <p><strong>Note:</strong> The multiple attribute of the input tag is not supported in Internet Explorer 9 and earlier versions.</p>

    </body>

</html>
```

34.6.14 The pattern Attribute

The pattern attribute specifies a regular expression that the <input> element's value is checked against. The pattern attribute works with the following input types: text, search, url, tel, email, and password.

Tip: Use the global title attribute to describe the pattern to help the user.

Tip: Learn more about regular expressions in our JavaScript tutorial.

Example

An input field that can contain only three letters (no numbers or special characters):

```
<!DOCTYPE html>

<html>

    <body>

        <form action="action_page.php">

            Country code:

            <input type="text" name="country_code" pattern="[A-Za-z]{3}" title="Three letter country code">

            <input type="submit">

        </form>

        <p><strong>Note:</strong> The pattern attribute of the input tag is not supported in Internet Explorer 9 and earlier versions, or in Safari.</p>

    </body>

</html>
```

34.6.15 The placeholder Attribute

The placeholder attribute specifies a hint that describes the expected value of an input field (a sample value or a short description of the format). The hint is displayed in the input field before the user enters a value.

The placeholder attribute works with the following input types: text, search, url, tel, email, and password.

Example

An input field with a placeholder text:

```
<!DOCTYPE html>
<html>
<body>
<form action="action_page.php">
    <input type="text" name="fname" placeholder="First name"><br>
    <input type="text" name="lname" placeholder="Last name"><br>
    <input type="submit" value="Submit">
</form>
<p><strong>Note:</strong> The placeholder attribute of the input tag is not supported in Internet Explorer 9 and earlier versions.</p>
</body>
</html>
```

34.6.16 The required Attribute

The required attribute is a boolean attribute.

When present, it specifies that an input field must be filled out before submitting the form. The required attribute works with the following input types: text, search, url, tel, email, password, date pickers, number, checkbox, radio, and file.

Example

A required input field:

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

```

<body>

    <form action="html/action_page.php">

        Username: <input type="text" name="username" required>

        <input type="submit">

    </form>

    <p><strong>Note:</strong> The required attribute of the input tag is not
    supported in Internet Explorer 9 and earlier versions, or in Safari.</p>

</body>

</html>

```

34.6.17 The step Attribute

The step attribute specifies the legal number intervals for an `<input>` element.

Example: if `step="3"`, legal numbers could be -3, 0, 3, 6, etc.

Tip: The step attribute can be used together with the max and min attributes to create a range of legal values.

The step attribute works with the following input types: number, range, date, datetime, datetime-local, month, time and week.

Example

An input field with a specified legal number intervals:

```

<!DOCTYPE html>

<html>

    <body>

        <form action="action_page.php">

            <input type="number" name="points" step="3">

```

```
<input type="submit">  
</form>  


<p><strong>Note:</strong> The step attribute of the input tag is not supported in Internet Explorer 9 and earlier versions, or in Firefox.</p>



</body>



</html>


```

CHAPTER THIRTY-FIVE

35.0 HTML MULTIMEDIA

Multimedia on the web, is sound, music, videos, movies, and animations.

35.1 What is Multimedia?

Multimedia comes in many different formats. It can be almost anything you can hear or see.

Examples: Pictures, music, sound, videos, records, films, animations, and more. Web pages often contains multimedia elements of different types and formats. In this chapter you will learn about the different multimedia formats.

35.2 Browser Support

The first web browsers had support for text only, limited to a single font in a single color. Later came browsers with support for colors and fonts, and even support for pictures! The support for sounds, animations, and videos is handled differently by various browsers. Different types and formats are supported, and some formats requires extra helper programs (plug-ins) to work. Hopefully this will become history. HTML5 multimedia promises an easier future for multimedia.

35.3 Multimedia Formats

Multimedia elements (like sounds or videos) are stored in media files. The most common way to discover the type of a file, is to look at the file extension. When a browser sees the file extension .htm or .html, it will treat the file as an HTML file. The .xml extension indicates an XML file, and the .css extension indicates a style sheet file. Pictures are recognized by extensions like .gif, .png and .jpg. Multimedia files also have their own formats and different extensions like: .swf, .wav, .mp3, .mp4, .mpg, .wmv, and .avi.

Common Video Formats

MP4 is the new and upcoming format for internet video.



MP4 is recommended by YouTube.

MP4 is supported by Flash Players

MP4 is supported by HTML5.

Format	File	Description
MPEG	.mpg .mpeg	MPEG. Developed by the Moving Pictures Expert Group. The first popular video format on the web. Used to be supported by all browsers, but it is not supported in HTML5 (See MP4).
AVI	.avi	AVI (Audio Video Interleave). Developed by Microsoft. Commonly used in video cameras and TV hardware. Plays well on Windows computers, but not in web browsers.
WMV	.wmv	WMV (Windows Media Video). Developed by Microsoft. Commonly used in video cameras and TV hardware. Plays well on Windows computers, but not in web browsers.
QuickTime	.mov	QuickTime. Developed by Apple. Commonly used in video cameras and TV hardware. Plays well on Apple computers, but not in web browsers. (See MP4)
RealVideo	.rm .ram	RealVideo. Developed by Real Media to allow video streaming with low bandwidths. It is still used for online video and Internet TV, but does not play in web browsers.
Flash	.swf .flv	Flash. Developed by Macromedia. Often requires an extra component (plug-in) to play in web browsers.
Ogg	.ogg	Theora Ogg. Developed by the Xiph.Org Foundation. Supported by HTML5.
WebM	.webm	WebM. Developed by the web giants, Mozilla, Opera, Adobe, and Google. Supported by HTML5.
MPEG-4 or MP4	.mp4	MP4. Developed by the Moving Pictures Expert Group. Based on QuickTime. Commonly used in newer video cameras and TV hardware. Supported by all HTML5 browsers. Recommended by YouTube.

NOTE! Only MP4, WebM, and Ogg video is supported by the newest HTML5 standard.

35.4 Sound Formats

MP3 is the newest format for compressed recorded music. The term MP3 has become synonymous with digital music. If your website is about recorded music, MP3 is the choice.

Format	File	Description
MIDI	.mid .midi	MIDI (Musical Instrument Digital Interface). Main format for all electronic music devices like synthesizers and PC sound cards. MIDI files do not contain sound, but digital notes that can be played by electronics. Plays well on all computers and music hardware, but not in web browsers.
RealAudio	.rm .ram	RealAudio. Developed by Real Media to allow streaming of audio with low bandwidths. Does not play in web browsers.
WMA	.wma	WMA (Windows Media Audio). Developed by Microsoft. Commonly used in music players. Plays well on Windows computers, but not in web browsers.
AAC	.aac	AAC (Advanced Audio Coding). Developed by Apple as the default format for iTunes. Plays well on Apple computers, but not in web browsers.
WAV	.wav	WAV. Developed by IBM and Microsoft. Plays well on Windows, Macintosh, and Linux operating systems. Supported by HTML5.
Ogg	.ogg	Ogg. Developed by the Xiph.Org Foundation. Supported by HTML5.
MP3	.mp3	MP3 files are actually the sound part of MPEG files. MP3 is the most popular format for music players. Combines good compression (small files) with high quality. Supported by all browsers.
MP4	.mp4	MP4 is a video format, but can also be used for audio. MP4 video is the upcoming video format on the internet. This leads to automatic support for MP4 audio by all browsers.

CHAPTER THIRTY-SIX

36.0 HTML VIDEO

```
<!DOCTYPE html>

<html>

    <body>

        <video width="400" controls>

            <source src="mov_bbb.mp4" type="video/mp4">

            <source src="mov_bbb.ogg" type="video/ogg">

        Your browser does not support HTML5 video.

        </video>

        <p>

            Video courtesy of

            <a href="http://www.bigbuckbunny.org/" target="_blank">Big Buck
            Bunny</a>.

        </p>

    </body>

</html>
```

36.1 Playing Videos in HTML

Before HTML5, there was no standard for showing videos on a web page. Before HTML5, videos could only be played with a plug-in (like flash). The HTML5 `<video>` element specifies a standard way to embed a video in a web page.

Browser Support:



Internet Explorer 9+, Firefox, Opera, Chrome, and Safari support the <video> element.

Note: Internet Explorer 8 and earlier versions, do not support the <video> element.

36.2 The HTML <video> Element

To show a video in HTML, use the <video> element:

Example:

```
<!DOCTYPE html>

<html>

    <body>

        <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>

    </body>

</html>
```

How it Works

The **controls** attribute adds video controls, like play, pause, and volume. It is a good idea to always include **width** and **height** attributes. If height and width are not set, the browser does not know the size of the video. The effect will be that the page will change (or flicker) while the video loads. Text between the <video> and </video> tags will only display in browsers that do not support the <video> element. Multiple <source> elements can link to different video files. The browser will use the first recognized format.

36.4 HTML <video> Autoplay

To start a video automatically use the **autoplay** attribute:

Example:

```
<!DOCTYPE html>

<html>

<body>

<video width="320" height="240" autoplay>

    <source src="movie.mp4" type="video/mp4">

    <source src="movie.ogg" type="video/ogg">

        Your browser does not support the video tag.

</video>

</body>

</html>
```

HTML Video - Browser Support

Currently, there are 3 supported video formats for the <video> element: MP4, WebM, and Ogg:

Browser	MP4	WebM	Ogg
Internet Explorer	YES	NO	NO
Chrome	YES	YES	YES
Firefox	YES	YES	YES
Safari	YES	NO	NO
Opera	YES (from Opera 25)	YES	YES

HTML Video - Media Types

File Format	Media Type
MP4	video/mp4
WebM	video/webm
Ogg	video/ogg

36.5 HTML Video - Methods, Properties, and Events

HTML5 defines DOM methods, properties, and events for the <video> element. This allows you to load, play, and pause videos, as well as setting duration and volume. There are also DOM events that can notify you when a video begins to play, is paused, etc.

Example: Using JavaScript

```
<!DOCTYPE html>

<html>

<body>

<div style="text-align:center">

<button onclick="playPause()">Play/Pause</button>

<button onclick="makeBig()">Big</button>

<button onclick="makeSmall()">Small</button>

<button onclick="makeNormal()">Normal</button>

<br>

<video id="video1" width="420">

<source src="mov_bbb.mp4" type="video/mp4">

<source src="mov_bbb.ogg" type="video/ogg">

Your browser does not support HTML5 video.


```

```
</video>

</div>

<script>

var myVideo = document.getElementById("video1");

function playPause() {

    if (myVideo.paused)

        myVideo.play();

    else

        myVideo.pause();

}

function makeBig() {

    myVideo.width = 560;

}

function makeSmall() {

    myVideo.width = 320;

}

function makeNormal() {

    myVideo.width = 420;

}

</script>

<p>Video courtesy of <a href="http://www.bigbuckbunny.org/" target="_blank">Big Buck Bunny</a>.</p>

</body>

</html>
```

HTML5 Video Tags

Tag	Description
<code><video></code>	Defines a video or movie
<code><source></code>	Defines multiple media resources for media elements, such as <code><video></code> and <code><audio></code>
<code><track></code>	Defines text tracks in media players

CHAPTER THIRTY-SEVEN

37.0 HTML5 AUDIO

HTML5 provides a standard for playing audio files.

37.1 Audio on the Web

Before HTML5, there was no standard for playing audio files on a web page. Before HTML5, audio files could only be played with a plug-in (like flash). The HTML5 `<audio>` element specifies a standard way to embed audio in a web page.

Browser Support



Internet Explorer 9+, Firefox, Opera, Chrome, and Safari support the `<audio>` element.

Note: Internet Explorer 8 and earlier versions, do not support the `<audio>` element.

37.2 The HTML `<audio>` Element

To play an audio file in HTML, use the `<audio>` element:

Example

```
<!DOCTYPE html>

<html>

  <body>

    <audio controls>

      <source src="horse.ogg" type="audio/ogg">
```

```

<source src="horse.mp3" type="audio/mpeg">

Your browser does not support the audio element.

</audio>

</body>

</html>

```

37.3 HTML Audio - How It Works

The **controls** attribute adds audio controls, like play, pause, and volume. Text between the <audio> and </audio> tags will display in browsers that do not support the <audio> element. Multiple <source> elements can link to different audio files. The browser will use the first recognized format.

HTML Audio - Browser Support

Currently, there are 3 supported file formats for the <audio> element: MP3, Wav, and Ogg:

Browser	MP3	Wav	Ogg
Internet Explorer	YES	NO	NO
Chrome	YES	YES	YES
Firefox	YES	YES	YES
Safari	YES	YES	NO
Opera	YES	YES	YES

HTML Audio - Media Types

File Format	Media Type
MP3	audio/mpeg
Ogg	audio/ogg
Wav	audio/wav

37.4 HTML Audio - Methods, Properties, and Events

HTML5 defines DOM methods, properties, and events for the <audio> element. This allows you to load, play, and pause audios, as well as setting duration and volume.

There are also DOM events that can notify you when an audio begins to play, is paused, etc.

For a full DOM reference, go to our [HTML5 Audio/Video DOM Reference](#).

HTML5 Audio Tags

Tag	Description
<audio>	Defines sound content
<source>	Defines multiple media resources for media elements, such as <video> and <audio>

CHAPTER THIRTY-EIGHT

38.0 HTML PLUG-INS

The purpose of a plug-in, is to extend the functionality of the HTML browser.

38.1 HTML Helpers (Plug-ins)

Helper applications are computer programs that extend the standard functionality of a web browser. Helper applications are also called plug-ins. Examples of well-known plug-ins are Java applets.

Plug-ins can be added to web pages with the `<object>` tag or the `<embed>` tag.

Plug-ins can be used for many purposes: display maps, scan for viruses, verify your bank id, etc.

NOTE! To display video and audio: Use the `<video>` and `<audio>` tags.

38.2 The `<object>` Element

The `<object>` element is supported by all browsers. The `<object>` element defines an embedded object within an HTML document. It is used to embed plug-ins (like Java applets, PDF readers, Flash Players) in web pages.

Example:

```
<!DOCTYPE html>

<html>

    <body>

        <object width="400" height="50" data="bookmark.swf"></object>

    </body>
```

```
</html>
```

The <object> element can also be used to include HTML in HTML:

Example:

```
<!DOCTYPE html>

<html>

    <body>

        <object width="100%" height="500px" data="snippet.html"></object>

    </body>

</html>
```

Or images if you like:

Example:

```
<!DOCTYPE html>

<html>

    <body>

        <object data="audi.jpg"></object>

    </body>

</html>
```

38.3 The <embed> Element

The <embed> element is supported in all major browsers. The <embed> element also defines an embedded object within an HTML document. Web browsers have supported the <embed> element for a long time. However, it has not been a part of the HTML specification before HTML5. The element will validate in an HTML5 page, but not in an HTML 4 page.

Example:

```
<!DOCTYPE html>

<html>

    <body>

        <embed width="400" height="50" src="bookmark.swf">

    </body>

</html>
```

NOTE! Note that the `<embed>` element does not have a closing tag. It can not contain alternative text.

The `<embed>` element can also be used to include HTML in HTML:

Example:

```
<!DOCTYPE html>

<html>

    <body>

        <embed width="100%" height="500px" src="snippet.html">

    </body>

</html>
```

Or images if you like:

Example:

```
<!DOCTYPE html>

<html>

    <body>

        <embed src="audi.jpg">
```

</body>

</html>

CHAPTER THIRTY-NINE

39.0 HTML YOUTUBE VIDEOS

The easiest way to play videos in HTML, is to use YouTube.

39.1 Struggling with Video Formats?

Different versions of different browsers support different video formats. Earlier in this tutorial, you have seen that you might have to convert your videos to different video formats, to make sure they play in all browsers. Converting videos to different format can be difficult and time consuming. An easier solution might be to let YouTube play the videos in your web pages.

39.2 Playing a YouTube Video in HTML

To play your video on a web page, you can upload it to YouTube, and then insert the proper HTML code in your web page to display it:

Example - Using iFrame (this is the recommended method)

```
<!DOCTYPE html>

<html>

    <body>

        <iframe width="420" height="345"
src="http://www.youtube.com/embed/XGSy3_Czz8k">

    </iframe>

    </body>

</html>
```

Example - Using <object>

```
<!DOCTYPE html>

<html>

<body>

<object width="420" height="315"
       data="http://www.youtube.com/v/XGSy3_Czz8k">

</object>

</body>

</html>
```

Example - Using <embed>

```
<!DOCTYPE html>

<html>

<body>

<embed width="420" height="315"
       src="http://www.youtube.com/v/XGSy3_Czz8k">

</body>

</html>
```

YouTube will display the code to use (like:XGSy3_Czz8k), when you click "Share" under the video.

Goodbye

Well, I suppose this brings us to a close. Hopefully, the book lived up to what you expected, and, just maybe, you see testing in your future. You know what? It better dang well be in your future. I worked hard on this book!

The truth, though, is that this is an incredibly complex topic that consists of multiple methodologies, terminologies, frameworks, helpers, and more. No, it's not a skill that you pick up in a few hours. I've been at this for a long time, but still find myself learning new tricks and finding pitfalls every day. I hope the same will be true for you!

Stay in Touch

Let's be friends! I'm incredibly active on [Whatsapp](#), [Twitter](#), [Instagram](#), and [Facebook](#), of course, spend much of my time making.

Also

You can learn this entire Document by following the tutorials in my [YouTube](#) Channel. If this document is really helpful to you, please try as much as possible to spread this document across social media and around your friends and relatives.

Lastly, Please show some love by Subscribing to my [YouTube](#) Channel if you haven't yet, like, comment, and share the Videos. Thank you so much for you time and maximum cooperation. Much Love, iFarouq Tech.

Goodbye...