Program : **B.Tech**

Subject Name: **Internet and Web Technology**

Subject Code: **CS-504**

Semester: **5th**

Unit-2 Subject Notes

Subject code-CS 504 **Subject Name- Internet & Web Technology**

HTML

HTML (Hyper text Mark-up Language) is the most basic building block of the Web. It defines the meaning and structure of web content. Other technologies besides HTML are generally used to describe a web page's appearance/presentation.

HTML uses "mark-up" to annotate text, images, and other content for display in a Web browser. HTML mark-up includes special "elements" such as <head>, <title>, <body>, <header>, <footer>, <article>, <section>, <p>, <div>, <span>, <img>, <aside>, <audio>, <canvas>, <datalist>, <details>, <embed>, <nav>, <output>, <progress>, <video> and many others..

The below will help you learn more about HTML.

- HTML Introduction: If you're new to Web development, be sure to read our HTML Basics article to learn what HTML is and how to use it.
- HTML Tutorials: For articles about how to use HTML, as well as tutorials and complete examples, check out our HTML Learning Area.
- HTML Reference: In our extensive HTML reference section, you'll find the details about every element and attribute in HTML.

2.1 FORMATTING AND FONTS IN HTML

HTML Formatting Elements

In the previous chapter, you learned about the HTML style attribute. HTML also defines special elements for defining text with a special meaning. HTML uses elements like <b> and <i> for formatting output, like bold or *italic* text. Formatting elements were designed to display special types of text:

- <b> - Bold text
- <strong> - Important text
- <i> - Italic text
- <em> - Emphasized text
- <mark> - Marked text
- <small> - Small text
- <del> - Deleted text
- <ins> - Inserted text
- <sub> - Subscript text
- <sup> - Superscript text

HTML <sup> Element: **The HTML <sup> element defines superscripted text.**

Example: <p>This is <sup>superscripted</sup> text.</p>

HTML font Tag.

Example:

Specify the font size, font face and color of text:

```
<font size="3" color="red">This is some text!</font>
<font size="2" color="blue">This is some text!</font>
<font face="verdana" color="green">This is some text!</font>
```

2.2 COMMENT CODE

This element is used to add a comment to an HTML document. An HTML comment begins with <!— and the comment closes with —>. HTML comments are visible to anyone that views the page source code, but are not rendered when the HTML document is rendered by a browser

HTML comment:

- HTML comments are not displayed on the website. They are used to help you or other developers understand your code.
- Commenting serves various purposes, such as explaining a string of code or debugging a program.
- Multiline HTML comments usually explain large sections of code.
- Conditional HTML comments are interpreted only by the IE browser.
- Some browsers allow using the <comment> tag.
- A smart HTML comment is the one that adds value and meaning to the source code.
- **<!-- comment text -->**

HTML Comment Tags

This type of HTML comment is a regular single line comment. It is quite similar to the <q> element used for quoting. If you don't know much about it yet, you should visit our tutorial on HTML Quotation and Citation Elements.

You can use a single line comment to **deactivate a line of code** while debugging. Just insert HTML comment opening and closing tags around the part of the code you want to deactivate, and voila!

Multiline HTML Comments

HTML block comments or HTML multiline comments allow you to comment on a complex or a long piece of code. It works as a regular HTML comment tag, but as it can take several lines to explain a bigger part of the code, a single line comment won't be enough.

```
<!--
comment line1
comment line2
comment line3
-->
```

This comment tag works similarly as the <blockquote> element, which is used for quoting a bigger piece of text.

Multiline comment in HTML can also disable a block of code. All you have to do is include an opening and closing tags around the code you want to deactivate.

Example

```
<div>
<p>Block comment example</p>
<!--
A block comment helps when you need to comment out
large sections of code.
-->
</div>
```

2.3 COLOR

HTML colors are specified using predefined color names, or RGB, HEX, HSL, RGBA, HSLA values.

Color Values

In HTML, colors can also be specified using RGB values, HEX values, HSL values, RGBA values, and HSLA values:

Same as color name "Tomato"

Example

```
<h1 style="background-color:rgb(255, 99, 71);">...</h1>
<h1 style="background-color:#ff6347;">...</h1>
<h1 style="background-color:hsl(9, 100%, 64%);">...</h1>


<h1 style="background-color:rgba(255, 99, 71, 0.5);">...</h1>
<h1 style="background-color:hsla(9, 100%, 64%, 0.5);">...</h1>
```

RGB Value

In HTML, a color can be specified as an RGB value, using this formula: **rgb(*red, green*, *blue*).**
Each parameter (red, green, and blue) defines the intensity of the color between 0 and 255.
For example: rgb(255, 0, 0) is displayed as red, because red is set to its highest value (255) and the others are set to 0.
To display black, set all color parameters to 0, like this: rgb(0, 0, 0).
To display white, set all color parameters to 255, like this: rgb(255, 255, 255).
Experiment by mixing the RGB values below: rgb (255, 99, and 71)

HEX Value

In HTML, a color can be specified using a hexadecimal value in the form: ***#rrggbb***, where rr (red), gg (green) and bb (blue) are hexadecimal values between 00 and ff (same as decimal 0-255).

For example, #ff0000 is displayed as red, because red is set to its highest value (ff) and the others are set to the lowest value (00).

HSL Value

In HTML, a color can be specified using hue, saturation, and lightness (HSL) in the form:

Hsl (*hue*, *saturation*, *lightness*)

The degree on the color wheel from 0 to 360. 0 is red, 120 is green, and 240 is blue. Saturation is a percentage value, 0% means a shade of gray, and 100% is the full color. Lightness is also a percentage, 0% is black, 50% is neither light nor dark, and 100% is white

RGBA Value

RGBA color values are an extension of RGB color values with an alpha channel – which specifies the opacity for a color. An RGBA color value is specified with: rgba(*red, green*, *blue, alpha*)

HSLA Value

HSLA color values are an extension of HSL color values with an alpha channel - which specifies the opacity for a color. An HSLA color value is specified with: hsla(*hue, saturation*, *lightness, alpha*)

2.4 HYPERLINK

This page contains examples of HTML links - examples of hyperlink-specific code that **HTML**

Links

Hyperlinks are defined with the HTML <a> tag:
<a href="url">link text</a>

Local Links

The example above used an absolute URL (a full web address).
A local link (link to the same web site) is specified with a relative URL (without https://www....).<a href="html_images.asp">HTML Images</a>

Example:

```
<style>
a:link {
  color: green;
  background-color: transparent;
  text-decoration: none;
}

a:visited {
  color: pink;
  background-color: transparent;
  text-decoration: none;
}

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

a:active {
  color: yellow;
  background-color: transparent;
  text-decoration: underline;
}
</style>
```

2.6 LISTS

Unordered and Ordered List Example in HTML

- Ordered lists are numbered in some fashion, while unordered lists are bulleted.
- Definition lists consist of a term followed by its definition.
- Both ordered and unordered lists require start and end tags as well as the use of a special element to indicate where each list item begins (the <LI> tag

Unordered Lists

An unordered list is a list in which the order of the list items does not matter. Unordered lists should be used when rearranging the order of the list items would not create confusion or change the meaning of the information on the list.

The ul element opens and closes an unordered list. The items on the list are contained between list item, li, tags. A simple unordered list containing three items could be created with the following HTML.

Ordered Lists

Ordered lists are used for lists of items for which the order of the items does matter. The syntax for an ordered list is exactly the same as for an unordered list. However, to create an ordered list, the ol tag is used rather than the ul tag. By making this one change, we can convert the unordered list in our previous example into an ordered list.

Changing Numbering

There are times when you want to control the numbering of ordered lists. For example, your list may be broken up by a paragraph that appears mid-list to expand on a certain concept, or you may create a countdown list that begins at a high number and counts down. Lastly, maybe you'd rather use roman numerals. HTML and CSS make it easy to control the numbering of ordered lists.

Creating a Countdown List

To reverse the number of a list, simply add the reversed attributed to the opening ol tag.

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

Starting a List on a Specific Number

The start attribute is used to specify the number on which an ordered list starts. For example, imagine you have a list of 5 items, and after the second and fourth items you want to add a sentence or two with additional details. You could use the following HTML to do this without restarting the list numbering after each paragraph.

Notice that we used the start attribute on the ol tag to restart the numbering at “3” following the break in the list above. We'll use the same technique to properly number Step 5 below.

Changing the Numbering Style

You can use CSS to change the marker style of an ordered list. In addition to standard numbering (referred to as decimal in CSS), you can also use:

- upper-roman for uppercase roman numerals
- lower-roman for lowercase roman numerals
- decimal-leading-zero to add a “0” placeholder before single-digit list items

Description Lists

Description lists are created with the dl tag. Used far less frequently than their ordered and unordered peers, description lists are used to contain name-value groups. Each name-value group consists of one name, or term, placed between dt tags, followed by one or more values with each value, or description, placed between dd tags.

Using Lists for Menus

One of the most common uses of lists is to create website navigation menus. Unordered lists are usually the list-of-choice for this purpose. With just a few lines of CSS we can convert an unordered list into an attractive horizontal navigation menu.

Styling Lists

List typography is usually best styled to match the typography of the website's paragraph text. List-specific styling can be accomplished with CSS.

There are three list properties that can be styled with CSS:

- **list-style-type:** Defines the marker type that preceeds each list item. Common values include disc (the default unordered list style type), decimal (the default ordered list style type), circle, square, lower- or upper-roman, and lower- or upper-latin, although several additional styles may also be used.

- **list-style-position:** Determines whether the list item marker should be placed inside the content box, or outside of the content box in the item's left-hand padding area.
- **list-style-image:** An image can also be used as the item marker. This property is used to specify the image file to be used.

2.7 TABLE

An HTML table is defined with the <table> tag.
Each table row is defined with the <tr> tag. A table header is defined with the <th> tag. By default, table headings are bold and centered. A table data/cell is defined with the <td> tag.
Example:

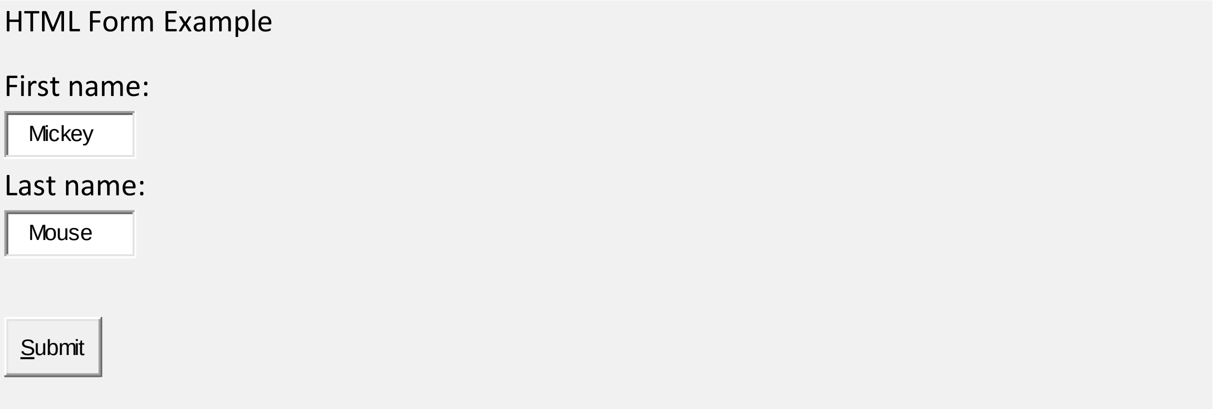
```
 <table style="width:100%">
  <tr>
    <th>Firstname</th>
    <th>Lastname</th>
    <th>Age</th>
  </tr>
  <tr>
    <td>Jill</td>
    <td>Smith</td>
    <td>50</td>
  </tr>
  <tr>
    <td>Eve</td>
    <td>Jackson</td>
    <td>94</td>
  </tr>
</table>
```

2.8 IMAGES

In HTML, images are defined with the <img> tag.
The <img> tag is empty; it contains attributes only, and does not have a closing tag.
The src attribute specifies the URL (web address) of the image:
<img src="*url*">

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 to store images in a sub-folder. You must then include the folder name in the src attribute:
Example: <img src="/images/html5.gif" alt="HTML5 Icon" style="width: 128px; height: 128px ;">

2.9 FORMS



HTML Form Example

First name:

Last name:

The <form> Element

The HTML <form> element defines a form that is used to collect user input:

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

An HTML form contains form elements.

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

The <input> Element

The <input> element is the most important form element.

The <input> element can be displayed in several ways, depending on the **type** attribute.

Examples:

Type	Description
<input type="text">	Defines a one-line text input field
<input type="radio">	Defines a radio button (for selecting one of many choices)
<input type="submit">	Defines a submit button (for submitting the form)

2.10 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>
```

The Most Important Differences from HTML:

Document Structure

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

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

XHTML Attributes

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

2.11 META TAGS

Meta tags are snippets of text that describe a page's content; the meta tags don't appear on the page itself, but only in the page's code. We all know tags from blog culture, and meta tags are more or less the same thing, little content descriptors that help tell search engines what a web page is about.

HTML **<Meta>** Tag

Example:

Describe metadata within an HTML document:

```
<head>
  <meta charset="UTF-8">
  <meta name="description" content="Free Web tutorials">
  <meta name="keywords" content="HTML,CSS,XML,JavaScript">
  <meta name="author" content="John Doe">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
</head>
```

2.12 CHARACTER ENTITIES

The following table lists the essential entities in HTML.

Character	Entity Name	Entity Number	Description
&	&amp;	&#38;	Ampersand
"	&quot;	&#34;	Double quote mark
<	&lt;	&#60;	Less than symbol
>	&gt;	&#62;	Greater than symbol
'	&apos;	&#39;	Apostrophe (**XHTML only**)

2.13 FRAMES AND FRAME SET

<**Frame**> Tag. Are used to divide the web browser window into multiple sections where each section can be loaded separately. A frameset tag is the collection of **frames** in the browser window **AMES.**

Example:

A simple three-framed page:

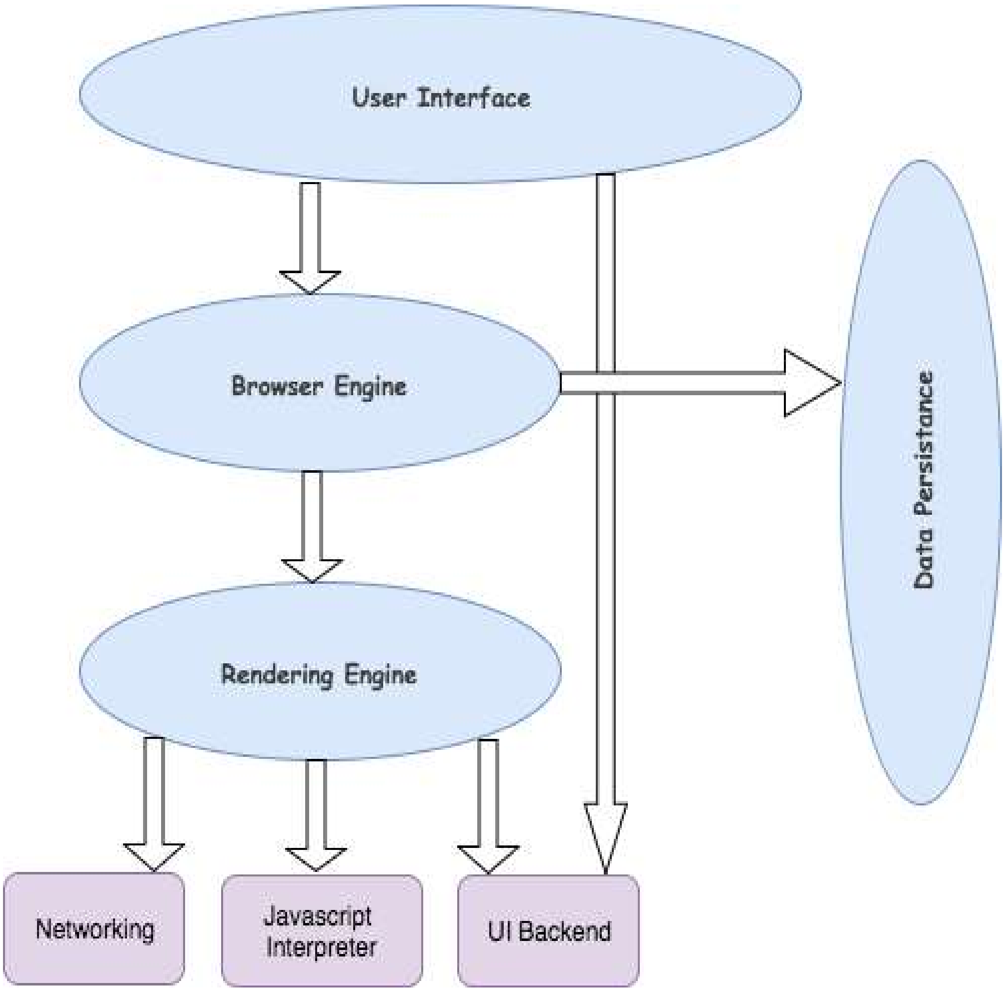
```
<frameset cols="25%,50%,25%">
  <frame src="frame_a.htm">
  <frame src="frame_b.htm">
  <frame src="frame_c.htm">
</frameset>
```

2.14.1 WEB BROWSER ARCHITECTURE

A browser is a software application used to locate, retrieve and display content on the World Wide Web, including Web pages, images, video and other files. As a client/server model, the browser is the client run on a computer that contacts the Web server and requests information. The Web server sends the information back to the Web browser which displays the results on the computer or other Internet-enabled device that supports a browser.

High-level architecture of browser

The below image shows the main components of a web browser:



2.1 Main components of the browser

The User Interface: The user interface is the space where User interacts with the browser. It includes the address bar, back and next buttons, home button, refresh and stop, bookmark option, etc. Every other part, except the window where requested web page is displayed, comes under it.

The Browser Engine: The browser engine works as a bridge between the User interface and the rendering engine. According to the inputs from various user interfaces, it queries and manipulates the rendering engine.

The Rendering Engine: The rendering engine, as the name suggests is responsible for rendering the requested web page on the browser screen. The rendering engine interprets the HTML, XML documents and images that are formatted using CSS and generates the layout that is displayed in the User Interface. However, using plugins or extensions, it can display other types data also. Different browsers user different rendering engines:

Networking: Component of the browser which retrieves the URLs using the common internet protocols of HTTP or FTP. The networking component handles all aspects of Internet communication and security. The network component may implement a cache of retrieved documents in order to reduce network traffic.

JavaScript Interpreter: It is the component of the browser which interprets and executes the javascript code embedded in a website. The interpreted results are sent to the rendering engine for display. If the script is external then first the resource is fetched from the network. Parser keeps on hold until the script is executed.

UI Backend: UI backend is used for drawing basic widgets like combo boxes and windows. This backend exposes a generic interface that is not platform specific. It underneath uses operating system user interface methods.

Data Persistence/Storage: This is a persistence layer. Browsers support storage mechanisms such as localStorage, IndexedDB, WebSQL and FileSystem. It is a small database created on the local drive of the computer where the browser is installed. It manages user data such as cache, cookies, bookmarks and preferences

2.14.2 WEB SITE STRUCTURE

Structuring your website is crucial for both its usability and findability. Many sites lack a sound structure to guide visitors to the information they're looking for. Having a clear site structure also leads to a better understanding of your site by Google, so it's incredibly important for your SEO. Let's take a closer look at how this works.

Ideal site structure

Let's start by looking at an ideal situation: if you're starting from scratch, how should you organize your site? We think a well-organized website looks like a pyramid with a number of levels:

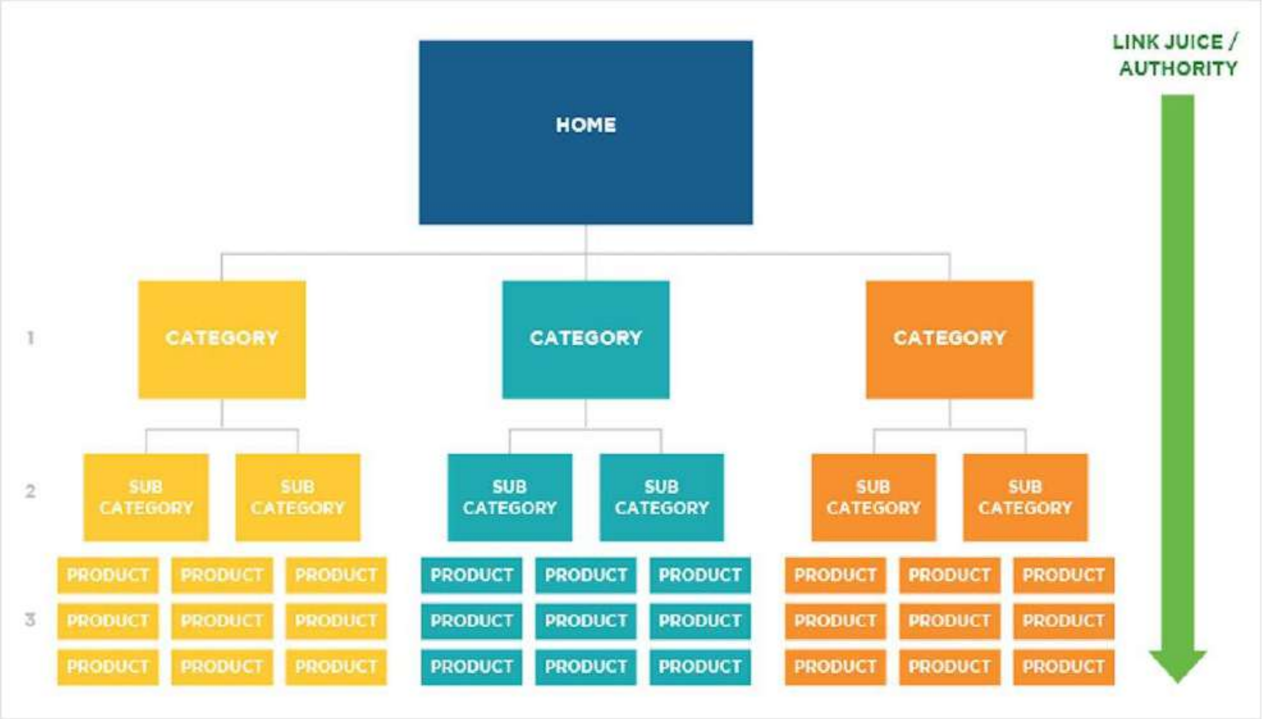
1. Homepage

2. Categories (or sections)

3. Subcategories (only for larger sites)

4. Individual pages and posts

The homepage should be all the way at the top. Then, you have some sections or category pages beneath it. You should be able to file all of your content under one of these categories. If your site is larger, you can divide these sections or categories into subcategories as well. Beneath your categories or subcategories are your individual pages and posts.



2.2 Website Structure

2.15 OVERVIEW AND FEATURE OF HTML5

HTML5 introduces a number of new elements and attributes that can help you in building modern websites. ... Forms 2.0 – Improvements to HTML web forms where new attributes have been introduced for <input> tag. Persistent Local Storage – To achieve without resorting to third-party plug-ins.

New Semantic/Structural Elements

HTML5 offers new elements for better document structure:

Tag	Description
<article>	Defines an article in a document
<aside>	Defines content aside from the page content
<bdi>	Isolates a part of text that might be formatted in a different direction from other text outside it

<details>	Defines additional details that the user can view or hide
<dialog>	Defines a dialog box or window
<figcaption>	Defines a caption for a <figure> element
<figure>	Defines self-contained content
<footer>	Defines a footer for a document or section
<header>	Defines a header for a document or section
<main>	Defines the main content of a document
<mark>	Defines marked/highlighted text
<meter>	Defines a scalar measurement within a known range (a gauge)
<nav>	Defines navigation links
<progress>	Represents the progress of a task
<rp>	Defines what to show in browsers that do not support ruby annotations
<rt>	Defines an explanation/pronunciation of characters (for East Asian typography)
<ruby>	Defines a ruby annotation (for East Asian typography)
<section>	Defines a section in a document

New Input Types

New Input Typ	**New Input Attributes**
• color • date • datetime • datetime-local • email • month • number • range • search • tel • time • ur	