HTML

(Hyper Text Markup Language)

What is HTML?

- HTML is a format that tells a computer how to display a web page.
- The documents themselves are plain text files (ASCII) with special "tags" or codes that a web browser knows how to interpret and display on your screen.

HTML Versions

- HTML 2.0
  - HTML 2.0 was developed by the Internet Engineering Task Force HTML Working Group in 1996.
- HTML 3.2
- HTML 4.0
- HTML 4.01
  - HTML 4.01 was released as a W3C Recommendation 24. December 1999.
  - HTML 4.01 is a minor update of corrections and bug fixes in HTML 4.0.
  - W3C will not continue to develop HTML. Future W3C work will be focusing on XHTML.

- XHTML 1.0
  - XHTML 1.0 reformulates HTML 4.01 in XML.
  - XHTML 1.0 was released as a W3C Recommendation 20. January 2000.
- HTML5
  - Web Hypertext Application Technology Working Group(WHATWG) and W3C came up with this in 2007
  - HTML 5 is a combination of HTML 4.01 and XHTML 1.0.
  - Many browsers are going to start supporting this in the future.
  - HTML 5 is backwards compatible.

What are HTML tags?

- When a web browser displays a page, it reads from a plain text file, and looks for special codes or "tags" that are marked by the < and > signs.
- The general format for a HTML tag is:

```
<tag_name>string of text</tag_name>
```

- As an example, the title for this section uses a header tag:

```
<h3>What are HTML tags?</h3>
```

- This tag tells a web browser to display the text What are HTML tags? in the style of header level 3
- HTML tags are case insensitive.
- Browser interprets the HTML and ignores the errors in the HTML code.

HCL

Types of tags

- Empty Tag
  - Tag that does not have an end tag
  - HTML 4.0 onwards, this tag when used must end with a/.
  - Example :
    - **
** older way
    - **
** HTML 4.0 onwards
- Container Tag
  - Tag that has start and the end tag
  - **<h3>What are HTML tags?</h3>**

Steps for creating HTML file

- Launch text editor program.
- Write the HTML code in the file
- Save the document as a file with any filename say “hello.html” or “hello.htm”

Structure Tags

- The tags that form the structure of HTML document
  - **<html>**
  - **<head>** → contains information about HTML page like the title, meta tags, scripts etc. The tags inside this will not be displayed in the page.
- **<body>** → This is the tag which holds all the controls and data that will be displayed in HTML page. The attribute that sets the color for the background, foreground, link are deprecated in HTML 4. **style** attribute **or** style sheet (CSS) can be used to set all of these.

```
<html>
  <head>   </head>
  <body>  Hello </body>
</html>
```

We will look at the style sheets later

Title and comment tags

- Every HTML document needs a title.
- This is given using **<title>** inside **the <head>**
- HTML comment is used to document HTML. These are not processed by the browser and hence they have no effect on the display.
- They are given between **<!--** and **-->**

```
<html>
<head>
<title>Hello world</title>
</head>
<!-- Hello world example -->
<body>
Hello World!!!!!!
</body>
</html>
```

This is an HTML comment

Textual and Formatting Tags

- Heading Tags:
  - **h1,h2,h3,h4,h5,h6** are header tag names
  - Example: **<h1>Main Heading</h1>**
  - h1 headings are bigger followed by h2 headings, h3 and so on
  - Sizes of this vary depending on browser.
- Paragraph Tags:
  - Paragraphs are defined with the ** ** tag.
  - ** How are you? **
  - **align** attribute can be used with this with values **left,right,center,justify**
- Adding a bit of emphasis using ****

```
This is a really <em>interesting</em> topic!
```

- Similarly to ****, **** can be used to define important text.
- Bold and italicized text: The **<b>** tag specifies bold text and **<i>** italicized

```
<p> His name is <i>Ram Bhandari</i></p>
```

Font and colors Tags

- The **<font>** tag is deprecated in HTML 4, and removed from HTML5.
- Style sheets (CSS) should be used to define the layout and display properties for many HTML elements.
- Older font tag

```
<font  face="Arial" size="4" color="blue"> Font face is Arial, size 4, blue color </font>
```


HCL

Links

- Adding links to other pages

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

- Adding linking parts of the same page : anchor
  - First parts of the page must be named using

```
<a name="some_name"> part 1</a>
```

  - Then using these names, links can be created beginning with a # followed by the name.

```
<HTML><BODY>
H1>Topics</H1>
<A href="#topic1">Introduction</A><BR>
<A href="#topic2">HTML</A><BR>
<A href="#topic2.1">CSS</A><BR>
```

Creating links

HCL

```
Some text<br> Some text<br> Some text<br> Some text<br>
Some text<br> Some text<br> Some text<br> Some text<br>
Some text<br> Some text<br> Some text<br> Some text<br>
Some text<br> Some text<br> Some text<br> Some text<br>
Some text<br> Some text<br> Some text<br> Some text<br>
Some text<br> Some text<br> Some text<br> Some text<br>
<H2>
<A name="topic1">Introduction</A></H2>
Introductionto web ...
<H2>
<A name="topic2">HTML</A></H2>
...Hyper text markup language...
<H3>
<A name="topic2.1">CSS</A></H3>
...Cascading Style Sheet ...
</BODY></HTML>
```

Naming parts of document

Adding images

- Images can be inserted using **<img>** and has two required attributes: **src** and **alt**.
- **src** specifies the URL of an image
- The **alt** attribute is used to give the short description, for those who can't see the image, can read in its absence .
- **<img>** is an empty tag.
- Examples:

```
<img src="pic.jpg" width="200" height="150" alt="My pic“ />
```

- Images are actually linked to HTML pages unlike Microsoft Word document where images can become part of the page.

HCL

Clickable regions within images

- Parts of the images can be made clickable by using **<map>.**
- The name attribute of the **<map>** element is linked with the **usemap** attribute of **<img>** and creates a relationship between the image and the map.

```
<HTML>
<BODY>
<p align="center">
<img src="scene.jpg" alt="scene" usemap="#scenemap"/>
<map name="scenemap">
<area shape="rect" coords=""6,116,97,184"
href="road.html" alt="road"/>
<area shape="circle" coords="210,10,10" href="home.html"
alt="home"/>
<area shape="poly" coords=""6,116,97,184"
href="ground.html" alt="sky"/>
</map> </p> </BODY></HTML>
```


HCL

Lists

- Creating unordered list.

```
<ul>
    <li> the first list item</li>
    <li> the second list item</li>
</ul>
```

- Creating ordered list where items are displayed numbered from 1.

```
<ol>
    <li> the first list item</li>
    <li> the second list item</li>
</ol>
```

  - `<ol>` can have **`type`** can be either **`1 or a or I or i.`** This specifies what ordering char to be used when list items are displayed. For example, if “I” is used, list item is displayed using Roman numbers.

Line breaks and horizontal line

- To insert a line breaks, **
** is used. **
** is an empty tag.

```
Why, when I needed you most, you have not been
there for me?" <br/>
The Lord replied, "The times when you have
seen only one set of footprints,<br/> is when I
carried you.
```

- To insert horizontal line, **<hr>** is used.

```
<p>Chapter 1</p>
<hr />
<p>Chapter 2</p>
```

HCL

Try out!

- How will you display

```
ArrayList<String> a = new ArrayList<String>();
```

in an HTML page?

```
<HTML>
<BODY>
<p>
ArrayList<String> a = new ArrayList<String>();
</p>
</BODY>
</HTML>
```

Will this work?

Special characters

HTML Entities	Special Characters
	Blank space
&copy;	©
&reg;	®
&#8482;	™
&lt;	<
&gt;	>

Note that HTML Entities are case sensitive.

Example:

```
ArrayList&lt;String&gt; a = new
ArrayList&lt;String&gt;();
```

Tables

- Tables are used for information as well as for layout.
- The **<table>** consists of one or more **<tr>, <th>, and <td>** elements.
- **<tr>** represents a row
- **<th>** represents heading cell/column
- **<td>** represents a cell/column

```
<table>
<tr><th>ID</th><th>Name</th></tr>
<tr><td>122</td><td>Meera</td></tr>
<tr><td>123</td><td>Mohan/td></tr>
<tr><td>124</td><td>Madhu</td></tr>
</table>
```

- **Cellpadding s**pecifies the space between the cell wall and the cell content,**cellspacing** Specifies the space between cells. Table can be given a border, width and height as well.

```
<table border="1" cellpadding="10">
<table border="1" cellpadding="10" cellspacing="10">
<table border="1" cellpadding="10" width="80%"
  height="500">
```

Forms

- Form tag is used to create controls on the page
- Form data will be received by a application on the server and processed. Hence the URL where the form data needs to be sent must be specified using **`action`** attribute. Apart from that **`method`** attribute is also very commonly used that specifies how form data is going to be submitted. (we will see this later).

```
<form action="go.do">
   <input type="text" name="txtuser">
   <input type="password" name="txtpass">
   <input type="submit" value="Login">
</form>
```

HCL

Form elements

- Some of the important <form> elements are
  - **`<input>`**
  - **`<select>, <option>`**
  - **`<textarea>`**
- A form can contain one or more of the following form elements.

<input>

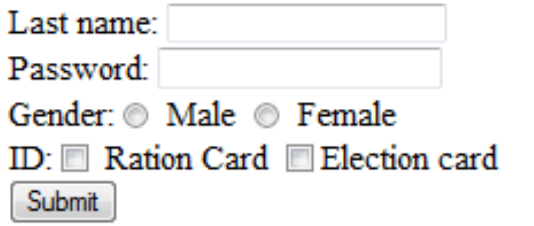
- Using the type attributes of this element many different controls can be created.
- align attribute can be used to specify where the control must be placed
- All these are empty tags.

- Values **type** can have
  - **button**
    checkbox
    file
    hidden
    image
    password
    radio
    reset
    submit
    text

- Other attributes
  - **checked** (Used with radio or checkbox type to specify that this is preselected)
  - **Maxlength** (maximum number of characters allowed)
  - **readonly**
  - **src** (used with image type that specifies URL of the image)
  - **name** (specifies name of the element)

Example

```
<HTML><BODY>
<form>
Last name: <input type="text" name="lastname"
/><br />
Password: <input type="password" name="pwd"
/><br />
Gender:<input type="radio" name="gender"
value="male" /> Male
 <input type="radio" name="gender"
value="female" /> Female<br />
ID:<input type="checkbox" name="rationid" />
Ration Card
<input type="checkbox" name="electionid"
/>Election card<br />
<input type="submit" value="Submit" />
</form>
</BODY></HTML>
```



<select> and <textarea>

- The **<select>** tag is used to create a drop-down list.
- Creating a Combo Box(only one selection at a time)

```
<select name="country">
  <option value="India"> India </option>
  <option value="China"> China </option>
</select>
```

- List Box(one/many selection allowed)

```
<select name="country" multiple>
  <option value="India"> India </option>
  <option value="China" selected> China </option>
</select>
```

- The **<textarea>** tag defines a multi-line text input control.

```
<textarea rows="4" cols="50">
   The woods are lovely, dark and deep.
But I have promises to keep,
And miles to go before I sleep,
And miles to go before I sleep </textarea>
```

Creating Frames

- Frames allows viewing multiple web pages in the same browser window.
- Frames can be created rows or columns
- The **<frameset>** contains <frame> elements which holds a separate HTML document.

Test1.html

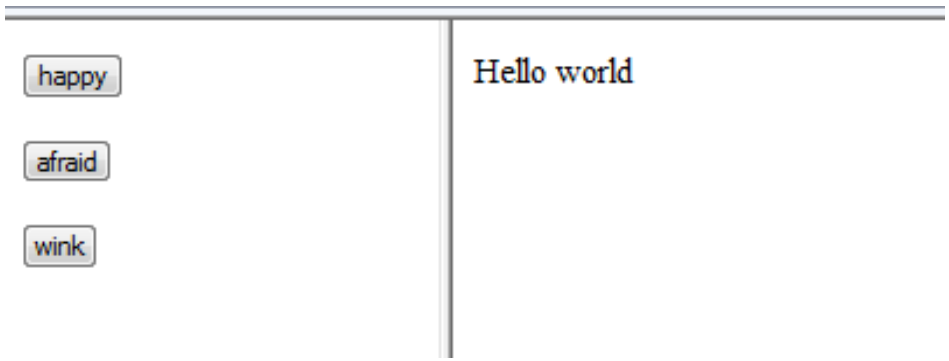
```
<html>
<head><title> frames</title>
</head>
<frameset cols="200,*">
<frame src="nav.html">
<frame src="body.html">
</frameset>
</html>
```

- nav.html

```
<html>
<body><form>
<input type=button value="happy">br><br>
<input type=button value="afraid"><br><br>
<input type=button value="wink">
</form> </body></html>
```

- body.html

```
<html>
<body>
Hello world
</body>
</html>
```



What is CSS?

- Cascading Style Sheets are used to display HTML elements
- Styles were added to HTML 4.0
- Style can be specified in two ways
  - Using **`style`** tag with html element
  - External Style Sheets are stored in CSS files

Steps for developing CSS

- Select desired element you want to style. The element that you select is called selector.
- Specify the property you want to style and state the value and append it with a semicolon. These are called declarations.
- Place all the styles in between braces.

```
/*This is a css comment*/
p
{
background-color : red ;
text-align:center;
}
```

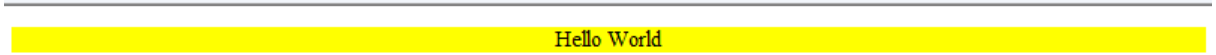
selector

declarations/rules

Example :Using CSS in the HTML

This example uses style attribute to set the style of <p> tag

```
<html>
<head>
<title>CSS Example</title>
<style type="text/css">
p{ background-color : yellow;
   text-align:center;
 }
</style>
</head>
<body>
 <p>Hello  World</p>
</body></html>
```



CSS Properties for text

- **`text-align`**
  - values: **`center`**, **`left`**, **`justify`**
- **`color`**
  - In general wherever we have color, it can have HEX value (#ff0000),an RGB value (rgb(255,0,0) or a color name (red)
- **`text-decoration`**
  - values: **`overline`**, **`line-through`**, **`underline`**, **`blink`**, **`none`**
- **`text-transform`**
  - **`p{text-transform:capitalize;}`**
  - **`uppercase`**, **`lowercase`**, **`capitalize`**
- **`text-indent`**

Example:

```
h1 {text-align:center; color:blue; text-
   decoration:overline; text-indent:50px;}
```


HCL

CSS Properties for font

- **font-family**
- **font-style**
  - Values: **normal, italic, oblique**
- **font-size**

Example:

- **p{font-family:"Times New Roman", verdana, geneva;**
  font-style:italic; font-size:12px;}

HCL

CSS Properties for background

- **background-color**
- **background-image**
  - **body {background-image:url('flower.gif');}**
- **background-repeat**
  - values: **no-repeat, repeat-x**
- **background-position**
  - **p{ background-position:right top;}**
- Short hand form of giving all background properties together

```
body {background:yellow url('flower.gif') no-repeat
left top;}
```


HCL

CSS Properties for tables

- **`border`** →values in the next slide
- **`width, height, padding`**
- **`border-collapse`**
  - sets whether the table borders are collapsed into a single border or separated
  - Values: **`collapse, none`**
- **`vertical-align`**
  - Values: **`top, bottom ,middle`**

- Example:

```
table
{
border: 1px solid black; width:100%; height:50%;
vertical-align:bottom; padding:5px;
}
```

CSS Properties for links

- Link can be in one of the four states
  - **`A:link`** - a normal, unvisited link
  - **`A:visited`** - a link the user has visited
  - **`A:hover`** - a link when the user mouse is over it
  - **`A:active`** - a link the moment it is clicked
- Style can be specified for each of these states
- Styles that can be specified are
  - All text styles and background styles can be used here.
  - More frequently used attribute are **`color, text-decoration, background-color`**
- Example: **`A:hover {text-decoration: underline; color: green;}`**

border values

Value	Description
none	Specifies no border
hidden	The same as "none", except in border conflict resolution for table elements
dotted	Specifies a dotted border
dashed	Specifies a dashed border
solid	Specifies a solid border
double	Specifies a double border
groove	Specifies a 3D grooved border. The effect depends on the border-color value
ridge	Specifies a 3D ridged border. The effect depends on the border-color value
inset	Specifies a 3D inset border. The effect depends on the border-color value
outset	Specifies a 3D outset border. The effect depends on the border-color value
inherit	Specifies that the border style should be inherited from the parent element

CSS Properties for lists

- For unordered lists
  - **`list-style-type`**
    - Values: **`circle, square`**
  - **`list-style-image`**
  - Example:

```
ul{ list-style-image: url('sqpurple.gif');}
```

- For ordered lists
  - **`list-style-type`**
  - Values: **`upper-roman, lower-alpha`**
  - Example:

```
ol{ list-style-type: lower-alpha;}
```

Combining styles

- When two elements have the same style rule then we combine the elements.

```
h1,h2 {
font-family:courier,verdana,geneva;
font-size : 170%;
color : grey;
}
```

- If we want to add any extra property only to element h1 we mention it separately along with the combined properties.

HCL

Linking CSS file externally

- Create a css file HTML in it, let us name it as “firstcss.css”.
- In this file **<style>** and **</style>** tags are eliminated as it does not contain any HTML.
- In the HTML file write the link to the file as

```
<link type="text/css" rel="stylesheet"
                href="firstcss.css">
```

- Omit the **<style>** and **</style>** tags from the HTML.

Creating different styles for same element

- Different styles can be created for same element.
- For this we need to create a class for the selector.
- **`p.mycolor{color : red ; }`**

  `mycolor` is class name.
- If we want to share the style rule for more than one elements then we could do this:

  `h1.mycolor, p.mycolor{color : red;}`
- Now if we want that all the elements that are in class mycolor to have a style than the better way is

```
. mycolor { color : red }
```

Example: using class

```
<html><head>
<title>CSS Example</title>
<style type="text/css">
      mycolor{
      color:red ;font-
family:courier,verdana,geneva;
      }
</style>
</head>
<body>
   <p class="mycolor">Hello  World</p>
   <h1 class="mycolor"> Heading</h1>
</body>
</html>
```

Using id of the element

- `id` attribute of an element can also be used to specify the style.
- This is done by using '#' symbol instead of dot (.)

```
<html><head>
<title>CSS Example</title>
<style type="text/css">
  p#p1{
  color:red ;font-family:courier,verdana,geneva;
  }
</style></head><body>
 <p id="p1">Hello  World</p>
</body></html>
```


HCL

Adding Styles to and descendants

- Let us assume that HTML has

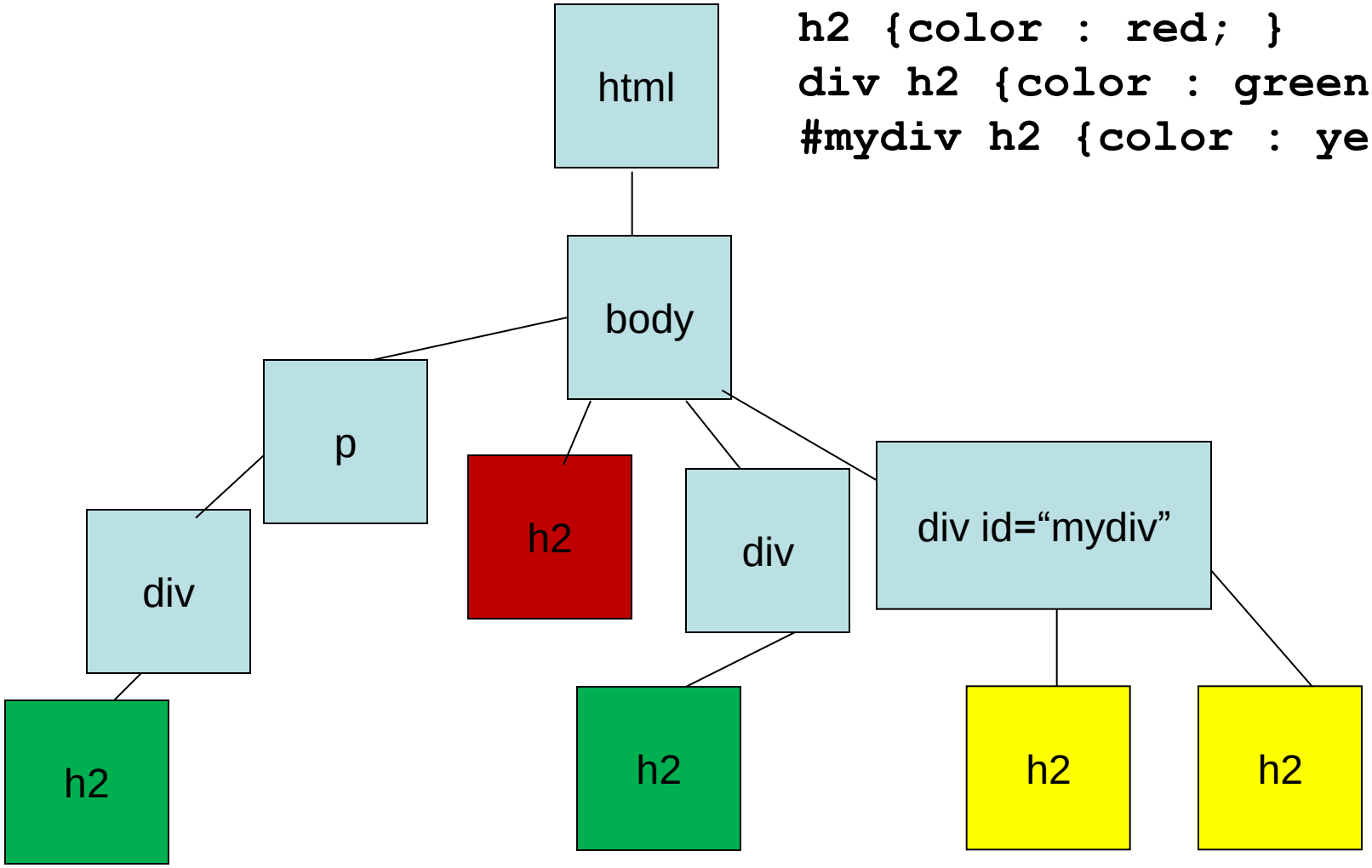
```
<div id="mydiv">
    <h2>From my div</h1>
     <p> Hello World</p>
 </div>
```

- Style for div is in the style sheet is

```
# mydiv {border-width : thin;
  border-style   : solid;
  border-color : #007e7e; }
```

- Now we need to add style for h2 which is inside the particular div.

Writing a descendant selector



#mydiv h2 {color : yellow;}

→select any <h2> that is descendant of an element with id "mydiv"