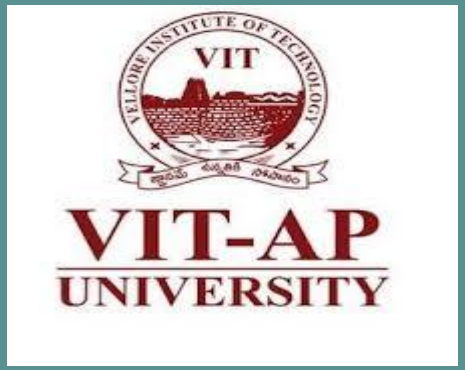
CSE 4004: Web Technologies
MODULE 1: HTML and CSS

Faculty: Prof. Daphna Chacko, Assistant Professor, SCOPE



HTML (HyperText Markup Language)

- HTML describes the structure of a web page
- Defined by Tim Berners Lee in 1990
- World Wide Web Consortium (W3C): Standardizes HTML and other web technologies.
- The latest version of HTML5 standard is widely used now.
- A web browser (Chrome, Edge, Firefox, Safari) is used to read HTML documents and display them correctly.
- **HTML uses tags** to define how web browser will format and display the content.
- A browser does not display the HTML tags, but uses them to determine how to display the document.

Structure of an html document

VIT-AP UNIVERSITY

An HTML document is said to contain following 2 types of information:

- **The markup information**: Instructions given to browser. It contains text enclosed within angle brackets (< and >) and are known as tags.
- **The character data**: Everything outside the markup tags. It is the information that is to be displayed by the browser.

- **Element of a html document** : A start tag, its corresponding end tag along with its contents.
- **Start tag**: *Element name* immediately follows **<** of the tag.
- **End tag**: *Element names* are preceded by **/**
- **Content of the element** : The portion between the tags.
- **Attributes:** used to provide additional information about the elements.

Outline of an html document

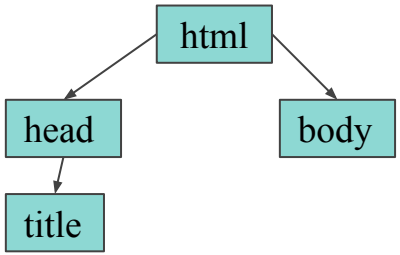
```
<!DOCTYPE html>
<html>
    <head>
        <title>
            My web page
        </title>
    </head>
    <body>
        Welcome to CSE 4004 Web Technologies!!
    </body>
</html>
```

Document instance

- **<!DOCTYPE html>**: To identify the version of html
- **<html>:** Root element of a HTML page. This should be the first tag in the document instance and appears only once in the document.
- **<head> :** This element is used to give instructions to the browser. Any text contained in the head element does not appear in the display area of the browser window.
- **<title>** : The contents are displayed as the title of the webpage as well as bookmark name
- **<body> :** Contains the information to be displayed in display area of the browser

Tree structure of an html document

- The markup tags give a tree structure to the html document
- Each start tag can be considered to be the starting of a nested level that is closed by its corresponding end tag



Where to write and execute html code

- Web pages can be created and modified by using professional HTML editors like VS Code, WebStorm.
- It is recommended to use notepad or gedit .
- Save the file as .htm or .html and set encoding as UTF-8.
- Open the saved html file in browser by typing the full path in the address bar.

Commonly used tags

- HTML tags are not case sensitive
- HTML **headings** are defined with the <h1> to <h6> tags where <h1> defines the most important heading and <h6> defines the least important heading
- Search engines use the headings to index the structure and content of your web pages. It is important to use headings to show the document structure.
- **Paragraphs** are defined with the <p> tag. A paragraph always starts on a new line, and some white space is added before and after a paragraph by browsers.
- The number of lines in a paragraph depends on the size of the browser window and the extra spaces/lines are removed by bowsers
- HTML **comments** are defined with <!-- -->
  - <!-- Hello, world! I am a comment -->

Empty html elements and pre formatted text

- HTML elements with no content are called empty elements.
-
 tag defines a line break. It is an empty element without a closing tag
- The <hr> tag is used to separate content (or define a change) in an HTML page,
- It is often displayed as a horizontal rule.
- The HTML <pre> element defines preformatted text.
- The text inside a <pre> element is displayed in a fixed-width font (usually Courier) preserving both spaces and line breaks
    - <pre>
      Hello All

      Welcome
      </pre>

Attributes of elements

- All html elements may have attributes that provide additional information about the elements.
- Attributes are specified in start tags.
- The **style** attribute is used to add styles to an element, such as color, font, size, and more.
- The **title** attribute defines some extra information about an element. The information is most often shown as a tooltip text when the mouse moves over the element.
- The value of the title attribute will be displayed as a tooltip when you move mouse over the element:
    - This is a paragraph.
- It is advisable to enclose the attribute value within quotes so that it is displayed correctly
- HTML **links** are defined with the <a> tag. The link's destination is specified in the **href** attribute.
    - Click here to know about VIT AP
- An element can have multiple attributes.
- HTML **images** defined with the <img> tag has source file (src), alternative text (alt), width, and height are provided as attributes.
    - <img src="../VIT.jpg" alt="VIT_AP" width="104" height="142">

Formatting tags

- <b> - Bold text
- - Important text (usually displayed in bold)
- <i> - Italic text
- - Emphasized text (displayed as italics. The screen reader reads it by giving verbal stress)
- <mark> - Marked text (to specify if the text is to be highlighted)
- <small> - Smaller text
- <del> - Deleted text (to indicate a deleted text. Usually browsers display the text as strikethrough)
- <ins> - Inserted text (to indicate an inserted text. Usually browsers display the text as underlined)
- <sub> - Subscript text
- <sup> - Superscript text

Quotation tags

- To specify short quotation <q> tag is used. Browsers insert quotation marks around the quotation specified.
  - <q>The more you know, the more you realize you know nothing</q>
- To define a section that is quoted from another source <blockquote> element is used. Browsers indent <blockquote> elements.
  - <blockquote cite="The C Programming Language, Kernighan and Ritchie">
    Since C is relatively small, it can be described in a small space, and learned quickly. A programmer can reasonably expect to know and understand and indeed regularly use the entire language.
    </blockquote>
- The cite attribute specifies the source of a quotation. Although it does not display any information, it can be used by search engines to get more information about the quotation.
- The <cite> tag is used to define the title of a creative work.The text in the <cite> element is displayed in *italic*.
  - <cite>The Imitation Game</cite> A movie based on the life of Alan Turing.

Abbreviation and address

- The <abbr> tag defines an abbreviation or an acronym, like "HTML", “Dr.”
  - <abbr title="Vellore Institute of Technology, Andhra Pradesh">VIT-AP</abbr>
- The HTML <address> tag defines the contact information for the author/owner of a document or an article.
- The address is displayed in italics and a line break is added before and after the <address> element.
  - <address>
    Nr Secretariat,
    Amaravati, 522237
    Andra Pradesh, India
    </address>

Links

- A link can be text, image, or any other HTML element.
- <a> tag is used to define a hyperlink
- By default, links will appear as follows in all browsers:
  - An unvisited link is underlined and blue
  - A visited link is underlined and purple
  - An active link is underlined and red
- The target attribute can have one of the following values:
  - _self - Default. Opens the document in the same window/tab as it was clicked
  - _blank - Opens the document in a new window or tab
  - _parent - Opens the document in the parent frame
  - _top - Opens the document in the full body of the window
- A link specified in in the href attribute can be absolute URL (a full web address) or relative (a link to a page within the same website)

Creating bookmarks and adding Images, emails as links

- To use an image as a link, put the <img> tag inside the <a> tag:
  -
    <img src="../VIT.jpg" alt="VIT_AP" width="104" height="120">

- Use **mailto**: inside the href attribute to create a link that opens the user's email program (to let them send a new email):
  - For general information
- Bookmarks are used to create links to sections in a long web page. We have to create a bookmark using *id* attribute and then create a link to it.
  - <h2 id="C4">Admissions</h2>

    Jump to Admissions
- The id attribute specifies a unique id for an HTML element.

Tables in html

- To arrange data into rows and columns
- The <table> tag defines an HTML table.
- Each table row is defined with a <tr> tag.
- Each table header is defined with a <th> tag. By default, the text in <th> elements are bold and centered.
- Each table data/cell is defined with a <td> tag. By default, the text in <td> elements are regular and left-aligned.
    - <table>

```
<tr>
  <th>Program</th>
  <th>School</th>
</tr>
<tr>
  <td>B.Tech CSE</td>
  <td>SCOPE</td>
</tr>
<tr>
  <td>B.Tech ECE</td>
  <td>SENSE</td>
</tr>
<tr>
  <td>B.Tech Mech</td>
  <td>SMSE</td>
</tr>
</table>
```

Table with caption

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

```
<caption>Programs in VIT-AP</caption>
  <tr>
    <th>Program</th>
    <th>School</th>
  </tr>
  <tr>
    <td>B.Tech CSE</td>
    <td>SCOPE</td>
  </tr>
  <tr>
    <td>B.Tech ECE</td>
    <td>SENSE</td>
  </tr>
  <tr>
    <td>B.Tech Mech</td>
    <td>SMSE</td>
  </tr>
</table>
```

Lists: Unordered and ordered

- HTML lists allow web developers to group a set of related items in lists.
- An unordered list starts with the <ul> tag. Each list item starts with the <li> tag.
- The list items will be marked with bullets by default:

```
<ul>
  <li>Chancellor</li>
  <li>Vice President</li>
  <li>Vice Chancellor</li>
  <li>Registrar</li>
</ul>
```

- An ordered list starts with the <ol> tag. Each list item starts with the <li> tag.
- The list items will be marked with numbers by default:

```
<ol>
  <li>Dean Academics</li>
  <li>COE</li>
  <li>Director Admissions</li>
</ol>
```

Description lists

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

```
<dl>
  <dt>Schools</dt>
  <dd>- Offer programmes  in various disciplines</dd>
  <dt>Programme</dt>
  <dd>-A course offered with a focus</dd>
</dl>
```

head element

- The <head> element is a container for metadata (data about data) and is placed between the <html> tag and the <body> tag.
- Metadata define the document title, character set, styles, scripts, and other meta information.
- The <style> element is used to define style information for a single HTML page:
  - <style>
    body {background-color: blue;}
    h1 {color: red;}
    p {color: blue;}
    </style>
- The <meta> element is typically used to specify the character set, page description, keywords, author of the document, and viewport settings.
  - <meta name="viewport" content="width=device-width, initial-scale=1.0">
- The metadata will not be displayed on the page, but are used by browsers (how to display content or reload page), by search engines (keywords), and other web services.

Forms in html

- An HTML form is used to collect user input and is created using <form> element.
- The user input is most often sent to a server for processing.
- The <form> element can contain various other elements and hence is known as a **container**.

Name	Value
Name	
Sex	Male Female
Eye color	green
Check all that apply	Over 6 feet tall Over 200 pounds
Describe your athletic ability:	
	Enter my information

Form attributes

- The *action* attribute defines the action to be performed when the form is submitted
  - <form action="/action_page.php">
- The *target* attribute specifies where to display the response that is received after submitting the form
- The *method* attribute specifies the HTTP method to be used when submitting the form data.
  - The form-data can be sent as URL variables (with method="get") or as HTTP post transaction (with method="post").
  - The default HTTP method when submitting form data is GET.
- The *autocomplete* attribute specifies whether a form should have autocomplete on or off.
  - When autocomplete is on, the browser displays values based on the previous user inputs.
- The *novalidate* attribute is a boolean attribute.
  - When present, it specifies that the form-data (input) should not be validated when submitted.

Form elements

- The <select> element defines a drop-down list
  - <select id="batch" name="Batch">
    <option value="Senior">Senior</option>
    <option value="Junior">Junior</option>
    <option value="Sophomore">Sophomore</option>
    <option value="Fresher">Fresher</option>
    </select>
  - By default, the first item in the drop-down list is selected. To define a pre-selected option, add the *selected* attribute to the option
  - The *multiple* attribute allows the user to select more than one value
  - *size* attribute to specify the number of visible values
- The <textarea> element defines a multi-line input field (a text area)
- The <button> element defines a clickable button
- The <fieldset> element is used to group related data in a form.
- The <legend> element defines a caption for the <fieldset> element.
- The <datalist> element specifies a list of pre-defined options for an <input> element which has a *list* attribute.
  - Users will see a drop-down list of the pre-defined options as they input data.

input and label elements

VIT-AP UNIVERSITY

- <input> element helps the user to enter input
- The default width of an input field is 20 characters.
- Each input field must have a *name* attribute so as to submit the value of the input field.
- The <label> element is useful for screen-reader users, because the screen-reader will read out loud the label when the user focus on the input element.
- The <label> element also help users who have difficulty clicking on very small regions (such as radio buttons or checkboxes) - because when the user clicks the text within the <label> element, it toggles the radio button/checkbox.
- The *for* attribute of the <label> tag should be equal to the *id* attribute of the <input> element to bind them together.

Types of input e*lement*

An <input> element can be displayed in many ways, depending on the *type* attribute

- <input type="text"> Displays a single-line text input field
- <input type="radio"> Displays a radio button (for selecting one of many choices)
- <input type="checkbox"> Displays a checkbox (for selecting zero or more of many choices)
- <input type="submit"> Displays a submit button (for submitting the form)
- <input type="button"> Displays a clickable button
- <input type="password"> defines a password field (The characters in a password field are masked)
- The <input type="email"> is used for input fields that should contain an e-mail address.
- <input type="file"> defines a file-select field and a "Browse" button for file uploads
- <input type="reset"> Displays a reset button(to reset all form values to their default values)
- <input type="time"> allows the user to select a time (Depending on browser support, a time picker can show up in the input field.)

Attributes of input element

- The input *value* attribute specifies an initial value for an input field
- The input *readonly* attribute specifies that an input field is read-only (The value of a read-only input field will be sent when submitting the form)
- The input *disabled* attribute specifies that an input field should be disabled (The value of a disabled input field will not be sent when submitting the form)
- The input *size* attribute specifies the visible width, in characters, of an input field (The size attribute works with the following input types: text, search, tel, url, email, and password.)
- The input *required* attribute specifies that an input field must be filled out before submitting the form
- The input *autofocus* attribute specifies that an input field should automatically get focus when the page loads.
- The input *autocomplete* attribute specifies whether a form or an input field should have autocomplete on or off.
- The input maxlength attribute specifies the maximum number of characters allowed in an input field.
- The input *list* attribute refers to a <datalist> element that contains pre-defined options for an <input> element.

Input form* attributes

- The input *form* attribute specifies the form the <input> element belongs to.
  - The value of this attribute must be equal to the id attribute of the <form> element it belongs to.
- The input *formaction* attribute specifies the URL of the file that will process the input when the form is submitted
  - This attribute works with submit and image input types.
- The input *formmethod* attribute defines the HTTP method for sending form-data to the action URL.
  - This attribute works with submit and image input types.
- The input *formtarget* attribute specifies a name or a keyword that indicates where to display the response that is received after submitting the form.
  - This attribute works with submit and image input types.
- The input *formnovalidate* attribute specifies that an <input> element should not be validated when submitted
  - This attribute works with submit input type

Block and inline elements

- A **block-level** element always starts on a new line and takes up the full width available.
- It also has a top and a bottom margin, whereas an inline element does not.
  - Be good; do good
  - The element is often used as a container for other HTML elements. When used together with CSS, it can be used to style blocks of content.
- An **inline** element does not start on a new line and only takes up as much width as necessary.
  - Be good; do good!!
  - The element is an inline container used to mark up a part of a text, or a part of a document. When used together with CSS, it can be used to style parts of the text.

Adding multimedia to HTML documents

- Multimedia on the web is sound, music, videos, movies, and animations.
- MP4, WebM, and Ogg video are supported video formats by the HTML standard.
- MP3, WAV, and Ogg audio are supported audio formats by the HTML standard.
- To show a video in HTML, use the <video> element
- The HTML <audio> element is used to play an audio file on a web page.
  - *controls* attribute adds video/audio controls, like play, pause, and volume.
  - *autoplay* attribute is used to start a video/audio automatically.
  - *muted* added after autoplay to let your video/audio start playing automatically but muted
  - The <source> element allows you to specify alternative audio/video files which the browser may choose from. The browser will use the first recognized format.
  - The text between the <audio> and </audio> tags will only be displayed in browsers that do not support the <audio> element. (same applies for <video> tags)

HTML iframes

- An HTML iframe is used to display a web page within a web page; i.e., to embed another document within the current HTML document.
    - <iframe width=100% src="Ex.html" title="CSE 4004:Web Technologies" name="iframe_a"> </iframe>
- The src attribute defines the URL of the page to embed
- Always include a title attribute (for screen readers)
- The height and width attributes specifies the size of the iframe
- Use border:none; to remove the border around the iframe

VIT-AP UNIVERSITY

CSS

Cascading Style Sheets (CSS)

- Used to format the layout of a webpage.
- A style applied to a parent element will also apply to all children elements within the parent unless specified.
- CSS allows us to control the color, font, the size of text, the spacing between elements, how elements are positioned and laid out, what background images or background colors are to be used, different displays for different devices and screen sizes, and so on.
- CSS can be added to HTML documents in 3 ways:
    - **Inline** - by using the style attribute inside HTML elements
    - **Internal** - by using a <style> element in the <head> section
    - **External** - by using a <link> element to link to an external CSS file
- Comments in CSS are inserted between /* and */

Setting rules in CSS

- A CSS rule consists of a selector and a declaration block.

- Multiple CSS declarations are separated with semicolons, and declaration blocks are surrounded by curly braces.

How to style a webpage?

- An **inline** CSS is used to apply a unique style to a single HTML element.
- An inline CSS uses the style attribute of an HTML element.
    - <h2 style="color:deeppink; text-align:center"> Let's learn to create webpages using HTML</h2>
- An **internal** style sheet may be used if one single HTML page has a unique style.
- The internal style is defined inside the <style> element, inside the head section.
    - <style>
      body{
      background-color:skyblue
      }
      q{
      color:tomato;
      }
      </style>

External CSS

- All relevant rules to change the look of an entire website website is written in a file.
- An external style sheet can be written in any text editor, and must be saved with a .css extension.
- The external .css file should not contain any HTML tags.
- Each HTML page must include a reference to the external style sheet file inside the <link> element, inside the head section.
  - <head>

    <link rel="stylesheet" href="style1.css">

    </head>
  - body {

    background-color: lightblue;

    }

Which style to choose?

- If some properties have been defined for the same selector (element) in different style sheets, the value from the last read style sheet will be used.
- When there is more than one style specified for an HTML element then cascading happens!!
- An inline style has the highest priority, and will override external and internal styles and browser defaults

Colors

- A color can be specified by using a predefined color name or RGB, HEX, HSL, RGBA, HSLA values
  - **RGB**: Red, Green, Blue with intensity varying from 0 to 255
  - **RGBA**: RGB with opacity specified using an alpha value varying from 0.0 (fully transparent) and 1.0 (not transparent at all). Eg, rgba(255, 99, 71, 0.5)
  - **HEX**: Hexadecimal integers are used to specify the colors. #RRGGBB
  - **HSL** (hue, saturation, and lightness): Hue is a degree on the color wheel from 0 to 360; Saturation and lightness are percentage values which specifies intensity and light resp.
  - **HSLA**: HSL with opacity specified using an alpha value
    - <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>

Color attributes

- **To set background color:** can be set for any element
  - `<h2 style="background-color:silver">`
- **To set color of text:** `<blockquote style="color:sienna">`
- **To set color of borders:** <iframe `style="border:2px solid Violet;">`

Background images

- An image can be used as the background of an element.
  - background-image: url("../WebTech/Gala-2018-Webpage-background-Top.png");
- By default, the *background-image* property repeats an image both horizontally and vertically.
  - Using background-repeat: repeat-y property the nature of repeat can be controlled
- The *background-position* property is used to specify the position of the background image.
- The *background-attachment* property specifies whether the background image should scroll or be fixed
- We can use shorthand property *background* to set the background properties in one declaration:
  - background: skyblue url("../WebTech/back.jpg") no-repeat fixed right top;
  - The order of the property values is:
    - background-color
    - background-image
    - background-repeat
    - background-attachment
    - background-position

Borders

- Allows you to specify the style, width, and color of an element's border.
- The *border-style* property specifies what kind of border to display.
  - dotted
  - dashed
  - solid
  - double
  - groove
  - ridge
  - inset
  - Outset
  - None
  - hidden
- The border-style property can have from one to four values (for the top border, right border, bottom border, and the left border).

More properties of borders

- The *border-width* property specifies the width of the four borders.
- The width can be set as a specific size (in px, pt, cm, em, etc) or by using one of the three pre-defined values: thin, medium, or thick
- The border-width property can have from one to four values (for the top border, right border, bottom border, and the left border)
    - border-width: 5px 20px; /* 5px top and bottom, 20px on the sides */
- The *border-color* property is used to set the color of the four borders. If border-color is not set, it inherits the color of the element.
- The border-color property can have from one to four values (for the top border, right border, bottom border, and the left border).
- The border-style, border-width and border-color properties can be set for individual sides.
    - If 3 values are specified then, it refers to top border, right and left borders and bottom border
    - If 2 values are specified then, it refers to top and bottom borders, right and left borders
    - If only one value is specified then all borders have the same value
- None of these border properties will have an effect unless the border-style property is set.

Individual borders, rounded borders and shorthand

- Individual border properties for just one side can be specified
  - p {
    border-left: 6px solid red;
    background-color: lightgrey;
    }
- *border-radius* property is used to add rounded borders to an element
  - p {
    border: 2px solid red;
    border-radius: 12px;
    }
- The *border* property is a shorthand property for the following individual border properties:
  - border-width
  - border-style (required)
  - border-color

Margins

- Margins are used to create space around elements, **outside of any defined borders**.
- CSS has properties for specifying the margin for each side of an element:
  a. margin-top
  b. margin-right
  c. margin-bottom
  d. margin-left
- All the margin properties can have the following values:
  a. *auto* - the browser calculates the margin and horizontally centers the element within its container.
  b. *length* - specifies a margin in px, pt, cm, etc.
  c. % - specifies a margin in % of the width of the containing element
  d. *inherit* - specifies that the margin should be inherited from the parent element
- The *margin* property is a shorthand property for the individual margin properties

Padding

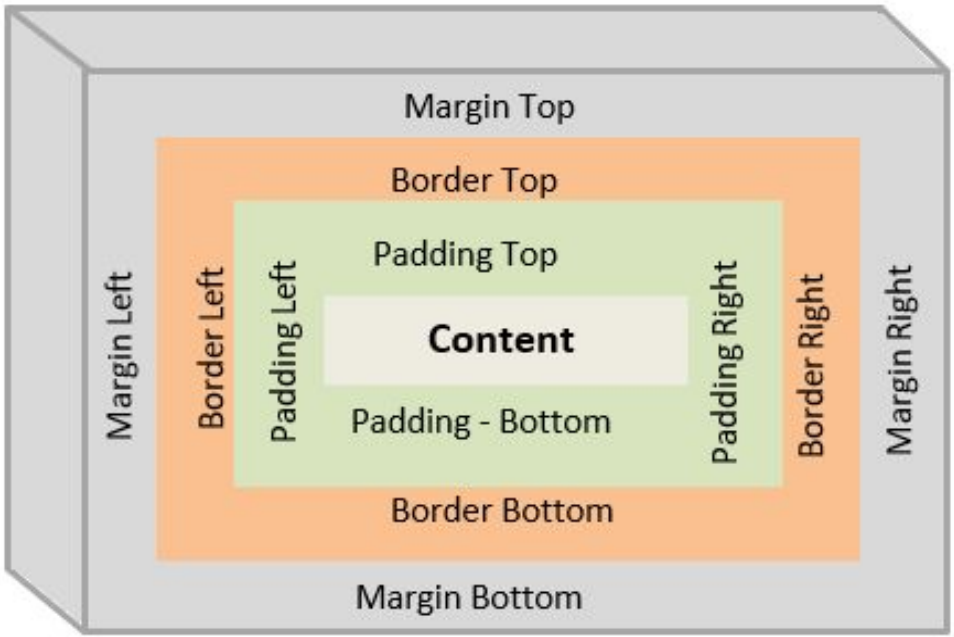
- Padding is used to create space around an element's content, **inside of any defined** borders.
- CSS has properties for specifying the padding for each side of an element:
  - padding-top
  - padding-right
  - padding-bottom
  - Padding-left
- All the padding properties can have the following values:
  - *length* - specifies a padding in px, pt, cm, etc.
  - *%* - specifies a padding in % of the width of the containing element
  - *inherit* - specifies that the padding should be inherited from the parent element
- To shorten the code, it is possible to specify all the above properties in the *padding* property.

Height and width properties

- Used to set the height and width of an element.
- The height and width properties do not include padding, borders, or margins. It sets the height/width of the area inside the padding, border, and margin of the element.
- The *height* and *width* properties may have the following values:
  - auto - This is default. The browser calculates the height and width
  - length - Defines the height/width in px, cm etc.
  - % - Defines the height/width in percent of the containing block
  - initial - Sets the height/width to its default value
  - inherit - The height/width will be inherited from its parent value
- The *max-width* property is used to set the maximum width of an element
- If an element has a specified width, the padding added to that element will be added to the total width of the element.
- *box-sizing* property helps the element to maintain its width. If you increase the padding, the available content space will decrease.

CSS box model

All HTML elements can be considered as boxes that wraps around every HTML element like margins, borders, padding, and the actual content.



Height and width calculation of elements

- When you set the width and height properties of an element with CSS, you just set the width and height of the content area.
- To calculate the full size of an element, you must also add padding, borders and margins.

```
div {
  width: 320px;
  padding: 10px;
  border: 5px solid gray;
  margin: 0;
}
```

Outline properties

- Unlike border, the outline is drawn outside the element's border, and may overlap other content.
- The outline is NOT a part of the element's dimensions; the element's total width and height is not affected by the width of the outline.
- The *outline-style* property specifies the style of the outline.
- The *outline-width* property specifies the width of the outline, and can have one of the following values:
    - thin (typically 1px)
    - medium (typically 3px)
    - thick (typically 5px)
    - A specific size (in px, pt, cm, em, etc)
- The *outline-color* property is used to set the color of the outline.
- The *outline* property is a shorthand property for setting the following individual outline properties:
    - outline-width
    - outline-style (required)
    - outline-color
- The *outline-offset* property adds space between an outline and the edge/border of an element. The space between an element and its outline is transparent.

Text property

- The *color* property is used to set the color of the text
- The *text-align* property is used to set the horizontal alignment of a text.
  - A text can be left or right aligned, centered, or justified.
  - The direction and unicode-bidi properties can be used to change the text direction of an element:
  - Left alignment is default if text direction is left-to-right, and right alignment is default if text direction is right-to-left
- The *vertical-align* property sets the vertical alignment of an element.
- The *text-decoration* property is used to set or remove decorations from text like underline, overline and line-through
- The *text-transform* property is used to turn everything into uppercase or lowercase letters, or capitalize the first letter of each word:
- The *text-indent* property is used to specify the indentation of the first line of a text
- The *letter-spacing* property is used to specify the space between the characters in a text
- The *line-height* property is used to specify the space between lines
- The *word-spacing* property is used to specify the space between the words in a text
- The *text-shadow* property allows to specify the horizontal shadow (2px) the vertical shadow (2px), a color and a blur effect.

Font property

- *font-family* property to specify the font of a text.
  - should hold several font names as a "fallback" system, to ensure maximum compatibility between browsers/operating systems
- **fallback** fonts are to be used so that if the first font does not work, the browser will try the next one, and the next one, and so on. Always end the list with a generic font family name.
  - `font-family: Lato, "Lucida Grande", Tahoma, Sans-Serif;`
- *font-style* property is mostly used to specify italic text.
- The *font-weight* property specifies the weight of a font
- The *font-size* property sets the size of the text
- **Web safe fonts** are fonts that are universally installed across all browsers and devices.

Icons

- Add the name of the specified icon class to any inline HTML element
- To use the Font Awesome icons: <script src="https://kit.fontawesome.com/*yourcode*.js" crossorigin="anonymous"></script>
- To use the Bootstrap glyphicons: <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">
- To use the Google icons: <link rel="stylesheet" href="https://fonts.googleapis.com/icon?family=Material+Icons">

Links

- Links can be styled with any CSS property and also depending on what state they are in.
  - The four links states are:
    - a:link - a normal, unvisited link
    - a:visited - a link the user has visited
    - a:hover - a link when the user mouses over it
    - a:active - a link the moment it is clicked
  - When setting the style for several link states, there are some order rules:
    - a:hover MUST come after a:link and a:visited
    - a:active MUST come after a:hover
- The text-decoration property is mostly used to remove underlines from links
- display links as boxes/buttons:

```
a:link, a:visited {
  background-color: #f44336;
  color: white;
  padding: 14px 25px;
  text-align: center;
  text-decoration: none;
  display: inline-block;
}
a:hover, a:active {
  background-color: red;
}
```

Lists

- *list-style-type* allows to change the list item markers:
  a. For ordered lists: upper-roman, lower-alpha
  b. For unordered lists: circle, square
- *list-style-image* property specifies an image as the list item marker

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

- *list-style-position* property specifies the position of the list-item markers: outside, inside
- *list-style* property is a shorthand property that contains values in the order: list-style-type, list-style-position, list-style-image
- Background colors, padding, margin etc can be added to lists and list items
- Anything added to the <ol> or <ul> tag, affects the entire list, while properties added to the <li> tag will affect the individual list item

Tables

- To specify table borders in CSS, use the *border* property.
  - table, th, td {
    border: 1px solid black;
    }
- The *border-collapse* property sets whether the table borders should be collapsed into a single border
- The width and height of a table are defined by the *width* and *height* properties:
  - For a table to span the entire screen (full-width), add *width*: 100%
- The *text-align* property sets the horizontal alignment (like left, right, or center) of the content in <th> or <td>.
- The *vertical-align* property sets the vertical alignment (like top, bottom, or middle) of the content in <th> or <td>.
- Add a container element (like) with overflow-x:auto around the <table> element to make it responsive:
- A responsive table will display a horizontal scroll bar if the screen is too small to display the full content
- Set the position of the table caption

```
caption {
  caption-side: bottom;
}
```

Table styles

- To control the space between the border and the content in a table, use the *padding* property on <td> and <th>
  - th, td {
    padding: 15px;
    text-align: left;
    }
- Add the *border-bottom* property to <th> and <td> for horizontal dividers
  - border-bottom: 1px solid #ddd;
- Use the :*hover* selector on <tr> to highlight table rows on mouse over:
  - tr:hover {background-color: #f5f5f5;}
- For zebra-striped tables, use the *nth-child*() selector and add a background-color to all even (or odd) table rows:
  - tr:nth-child(even) {background-color: #f2f2f2;}
- We can specify the background color and text color of table elements

Display properties

- The *display* property specifies if/how an element is displayed.
- The default display value for most elements is block or inline. Default values can often be overridden.
- Examples of **block-level** elements:
  -
  - <h1> - <h6>
  - <p>
  - <form>
  - <header>
  - <footer>
  - <section>
  -
- Examples of **inline** elements:
  -
  - <a>
  - <img>
- Hiding an element can be done by setting the *display* property to none. The element will be hidden, and the page will be displayed as if the element is not there.
- An element can be hidden using *visibility* property: visibility:hidden. The element will be hidden, but still affect the layout.

Positioning elements

- The *position* property specifies the type of positioning method used for an element (static, relative, fixed, absolute or sticky).
- Elements are then positioned using the top, bottom, left, and right properties. However, these properties will not work unless the position property is set first.
- An element with *position: static;* is not positioned in any special way; it is always positioned according to the normal flow of the page
- An element with *position: relative;* is positioned relative to its normal position
- An element with *position: fixed;* is positioned relative to the viewport, which means it always stays in the same place even if the page is scrolled.
- A fixed element does not leave a gap in the page where it would normally have been located.
- An element with *position: absolute;* is positioned relative to the nearest positioned (except static) ancestor
- If an absolute positioned element has no positioned ancestors, it uses the document body, and moves along with page scrolling.

Float and clear

- The *float* property is used for positioning and formatting content
  a. left - The element floats to the left of its container
  b. right - The element floats to the right of its container
  c. none - The element does not float (will be displayed just where it occurs in the text). This is default
  d. inherit - The element inherits the float value of its parent
- The *clear* property specifies what elements can float beside the cleared element and on which side
  a. none - Allows floating elements on both sides. This is default
  b. left - No floating elements allowed on the left side
  c. right- No floating elements allowed on the right side
  d. both - No floating elements allowed on either the left or the right side
  e. inherit - The element inherits the clear value of its parent
- The most common way to use the clear property is after you have used a float property on an element.

Navigation bars and Dropdown lists

- A navigation bar is a list of links and hence ordered/unordered list may be used by removing the list markers, padding, margins
- To build a vertical navigation bar, you can style the <a> elements inside the list
- One way to build a horizontal navigation bar is to specify the <li> elements as inline
- Another way of creating a horizontal navigation bar is to float the <li> elements, and specify a layout for the navigation links
- A hoverable dropdown list or dropdown menu can be built

CSS Flexbox

- The Flexible Box Layout Module, makes it easier to design flexible responsive layout structure without using float or positioning.
- A Flexible Layout must have a parent element with the *display* property set to *flex*.
- Direct child elements(s) of the flexible container automatically becomes flexible items.
- The flex container properties are:
  - *flex-direction*: Specifies the direction of the flexible items inside a flex container. row,row-reverse,column, column-reverse
  - *flex-wrap*: wrap, nowrap
  - *flex-flow* : shorthand property for the above two properties
  - *justify-content*: Horizontally aligns the flex items. center, flex-start, flex-end, space-around, space-between
  - *align-items*: Vertically aligns the flex items. center, flex-start, flex-end,stretch, baseline
  - *align-content*: aligns flex lines. enter, flex-start, flex-end,space-between, space-around, stretch

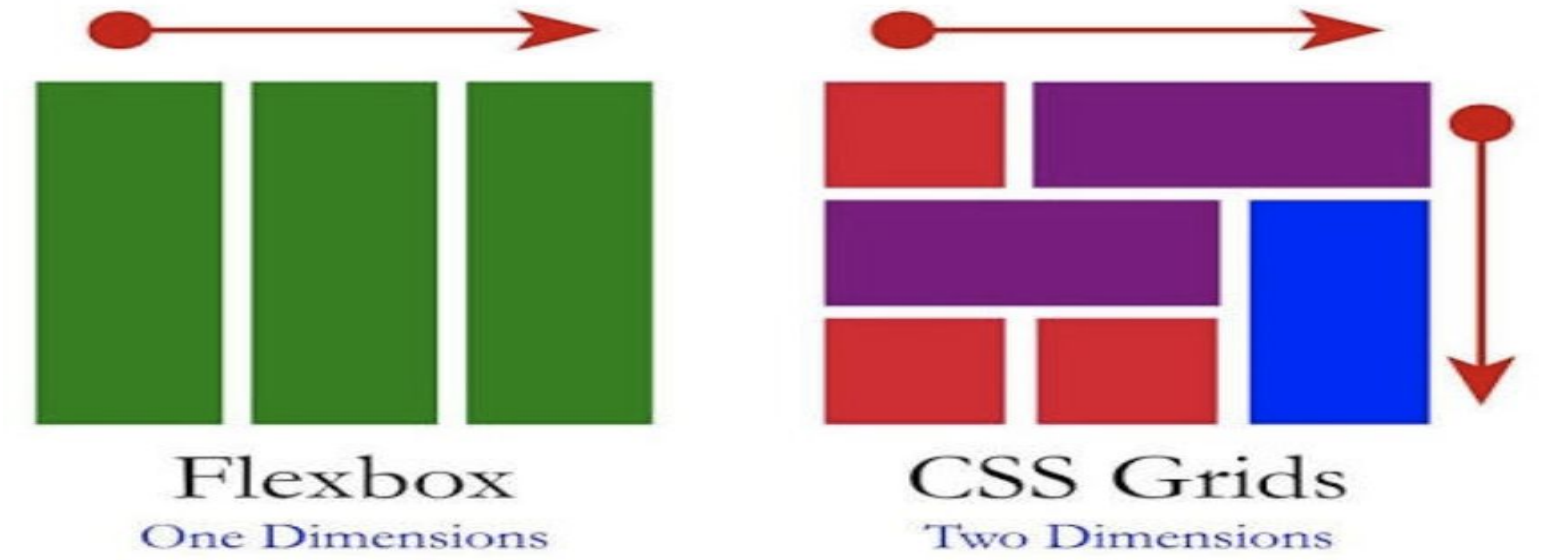
Flex items property

The flex item properties are:

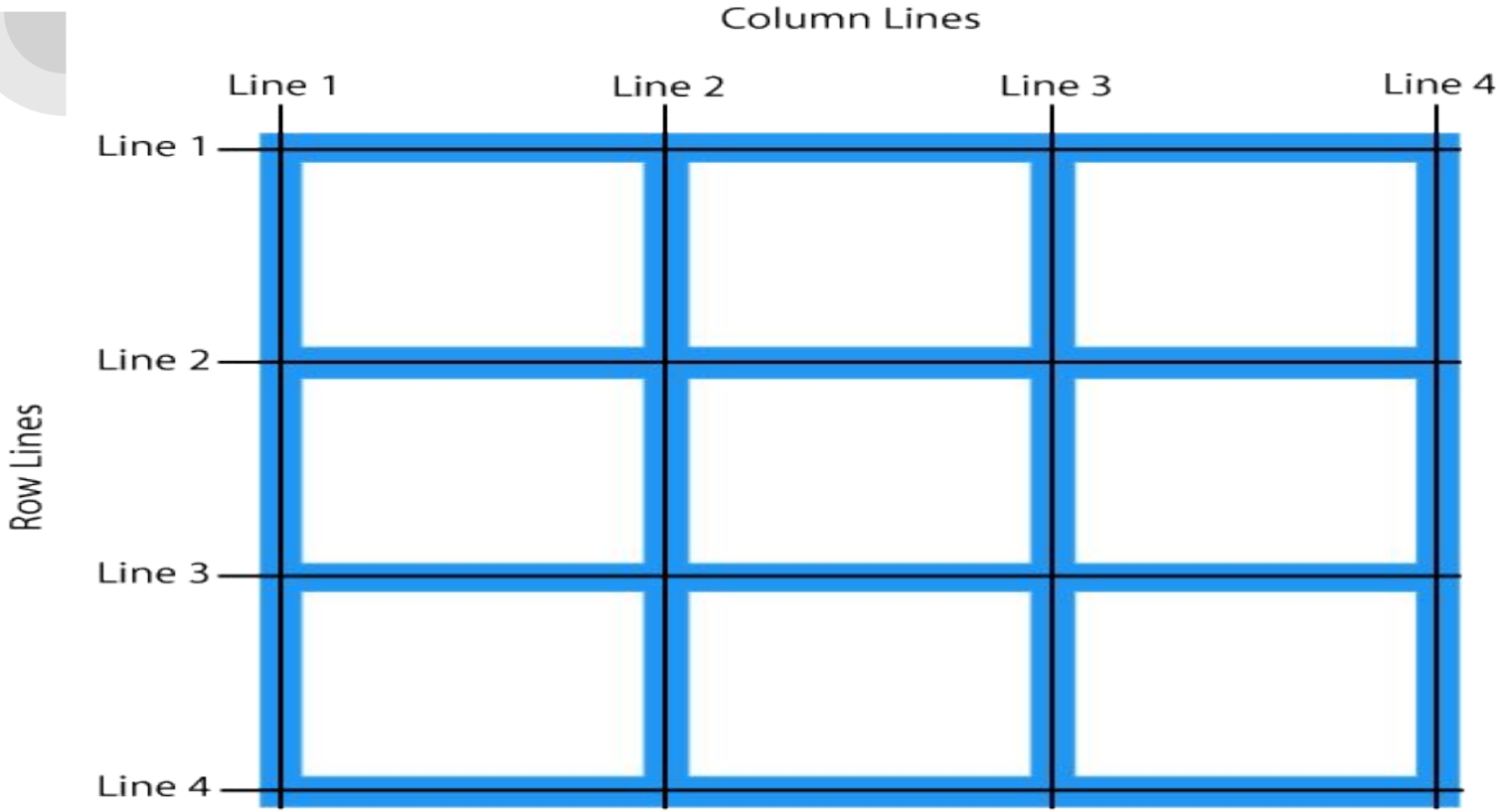
- *order*: specifies the order of the flex items
- *flex-grow*: specifies how much a flex item will grow relative to the rest of the flex items
- *flex-shrink*: specifies how much a flex item will shrink relative to the rest of the flex items.
- *flex-basis*: specifies the initial length of a flex item
- *flex*: shorthand property for the flex-grow, flex-shrink, and flex-basis properties
- *align-self*: specifies the alignment for the selected item inside the flexible container.

CSS Grid

- The CSS Grid Layout Module offers a grid-based layout system, with rows and columns, making it easier to design web pages without having to use floats and positioning.
- A grid layout consists of a parent element, with one or more child elements



Outline of a grid



Creating grids

- An HTML element becomes a grid container when its display property is set to grid or inline-grid.
- All direct children of the grid container automatically become *grid items*
- The vertical lines of grid items are called *columns*.
- The horizontal lines of grid items are called *rows*.
- The spaces between each column/row are called *gaps*.
- The the gap size can be adjusted using one of the following properties:
  - grid-column-gap
  - grid-row-gap
  - grid-gap
- The lines between columns are called *column lines* and the lines between rows are called *row lines*.
- We refer to line numbers when placing a grid item in a grid container:

Grid container

- Grid containers consist of grid items, placed inside columns and rows.
- The *grid-template-columns* property defines the number of columns in your grid layout, and it can define the width of each column.
- The value is a space-separated-list, where each value defines the width of the respective column or specify "auto" if all columns should have the same width.
- The *grid-template-rows* property defines the height of each row.
- The value is a space-separated-list, where each value defines the height of the respective row
- The *justify-content* property is used to align the whole grid inside the container.
- The *align-content* property is used to *vertically* align the whole grid inside the container

Grid items

- By default, a container has one grid item for each column, in each row, but you can style the grid items so that they will span multiple columns and/or rows.
- The *grid-column* property defines on which column(s) to place an item.
- The *grid-row* property defines on which row to place an item
- You define where the item will start, and where the item will end.
- To place an item, you can refer to *line numbers*, or use the keyword "span" to define how many columns/rows the item will span.
- The *grid-area* property can be used as a shorthand property for the grid-row-start, grid-column-start, grid-row-end and the grid-column-end properties.
- The grid-area property can also be used to assign names to grid items.
- Named grid items can be referred to by the *grid-template-areas* property of the grid container
    - Each row is defined by apostrophes (' ')
    - The columns in each row is defined inside the apostrophes, separated by a space.
    - A period sign represents a grid item with no name.
- The items do not have to be displayed in the same order as they are written in the HTML code

References

1. *"Web Technologies: A Computer Science Perspective"*, Jeffrey C Jackson
2. https://www.w3schools.com/html/
3. https://raw.githubusercontent.com/DaphnaChacko/Web-Technologies/main/Basics.html
4. https://raw.githubusercontent.com/DaphnaChacko/Web-Technologies/main/CSSAdv.html
5. https://raw.githubusercontent.com/DaphnaChacko/Web-Technologies/main/Flex.html

NB: Images are adapted from Internet