Unit-1 :Introduction

By Prof. Vaibhavi Patel

Content

- HTML:

  Background of HTML,
- CSS:

  Background of CSS,
- XML:

  Background of XML.

HTML Image Maps

- With HTML image maps, you can create clickable areas on an image.
- The areas are defined with one or more <area> tags.

```
<img src="temp.jpg" usemap="#map1">
  <map name="map1">
    <area shape="rect" coords="107,55,178,91" href="https://en.wikipedia.org/wiki/Whiteboard">
    <area shape="rect" coords="4,144,280,177" href="https://en.wikipedia.org/wiki/Table">
    <area shape="circle" coords="144,41,5" href="https://en.wikipedia.org/wiki/Watch">
  </map>
```

HTML <picture> Element

- The HTML <picture> element allows you to display **different pictures** for different devices or screen sizes.
- The HTML <picture> element gives web developers **more flexibility** in specifying image resources.
- The <picture> element contains **one or more <source>** elements, each referring to different images through the srcset attribute. This way the browser can choose the image that **best fits** the current view and/or device.
- Each <source> element has a media attribute that defines when the image is the most suitable.

HTML <picture> Element

- There are two main purposes for the <picture> element:

1. **Bandwidth**

- If you have a small screen or device, it is not necessary to load a large image file. The browser will use the first <source> element with matching attribute values, and ignore any of the following elements.

2. **Format Support**

- Some browsers or devices may not support all image formats. By using the <picture> element, you can add images of all formats, and the browser will use the first format it recognizes, and ignore any of the following elements.

HTML <picture> Element

```
<picture>
  <source media="(min-width: 650px)" srcset=“temp.jpg">
  <source media="(min-width: 465px)" srcset=“temp1.jpg">
  <img src="img_girl.jpg">
</picture>
```

HTML Favicon

- A favicon is a small image displayed next to the page title in the browser tab.

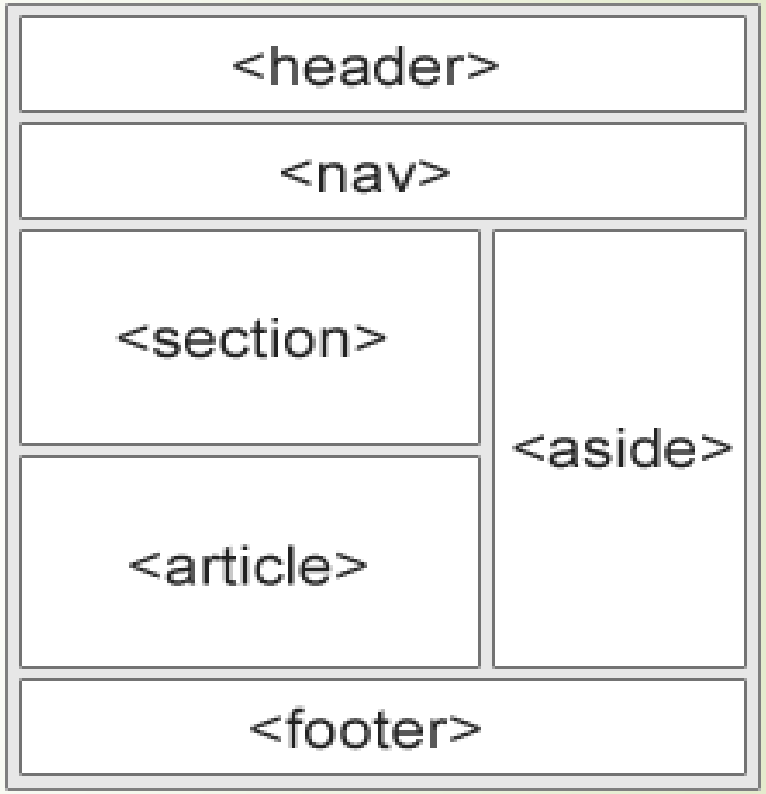
- To add a favicon to your website, either save your favicon image to the root directory of your webserver, or create a folder in the root directory called images, and save your favicon image in this folder. A common name for a favicon image is "favicon.ico".
- Next, add a <link> element to your file, after the <title> element, like this:

```
<head>
  <title>My Page Title</title>
  <link rel="icon" type="image/x-icon" href="/images/favicon.ico">
</head>
```

Semantic Elements in HTML

- Many web sites contain HTML code like: to indicate navigation, header, and footer.

- In HTML there are some semantic elements that can be used to define different parts of a web page:



Semantic Elements in HTML

- The **<section>** element defines a section in a document, is a grouping of content, typically with a heading.
- The **<article>** element specifies independent, self-contained content.

  An article should make sense on its own, and it should be possible to distribute it independently from the rest of the web site.
- The <header> element represents a container for introductory content or a set of navigational links.
- The <footer> element defines a footer for a document or section.
- The <nav> element defines a set of navigation links.
- The <aside> element defines some content aside from the content it is placed in (like a sidebar).

Using Emojis in HTML

- Emojis look like images, or icons, but they are not.

- They are letters (characters) from the UTF-8 (Unicode) character set.

- A is 65
- B is 66
- C is 67
- 😄 is 128516
- 😍 is 128525
- 💗 is 128151

CSS

CSS selector:

- Id selector
- Class selector
- Element selector
- Group selector
- Combinator selectors
- CSS Attribute Selectors

CSS combinator selector

A CSS selector can contain more than one simple selector. Between the simple selectors, we can include a combinator.

There are four different combinators in CSS:

- descendant selector (space)
- child selector (>)
- adjacent sibling selector (+)
- general sibling selector (~)

CSS combinator selector

- descendant selector (space):

  all elements that are descendants(inside) of a specified element.

- child selector (>):

  all elements that are the children of a specified element

- adjacent sibling selector (+):

  select an element that is directly after another specific element.

- general sibling selector (~)

  all elements that are next siblings of a specified element.

CSS Attribute Selectors

It is possible to style HTML elements that have specific attributes or attribute values.

- CSS [attribute] Selector

  The [attribute] selector is used to select elements with a specified attribute.

- CSS [attribute="value"] Selector

  The [attribute="value"] selector is used to select elements with a specified attribute and value.

CSS Layout

The position Property:

The position property specifies the type of positioning method used for an element.

- There are five different position values:

- static
- relative
- fixed
- absolute
- sticky

CSS Layout:

The position Property:

- Static: HTML elements are positioned static by default, always positioned according to the normal flow of the page.
- Relative: positioned relative to its normal position.
- Absolute: positioned relative to the nearest positioned ancestor
- Fixed: positioned relative to the viewport, which means it always stays in the same place even if the page is scrolled.
- Sticky: positioned based on the user's scroll position.

CSS Layout - The z-index Property

- When elements are positioned, they can overlap other elements.
- The z-index property specifies the stack order of an element (which element should be placed in front of, or behind, the others).
- An element can have a positive or negative stack order:

CSS Gradients

- CSS gradients let you display smooth transitions between two or more specified colors.
- CSS defines two types of gradients:
- Linear Gradients (goes down/up/left/right/diagonally)
- Radial Gradients (defined by their center)

CSS Linear Gradients

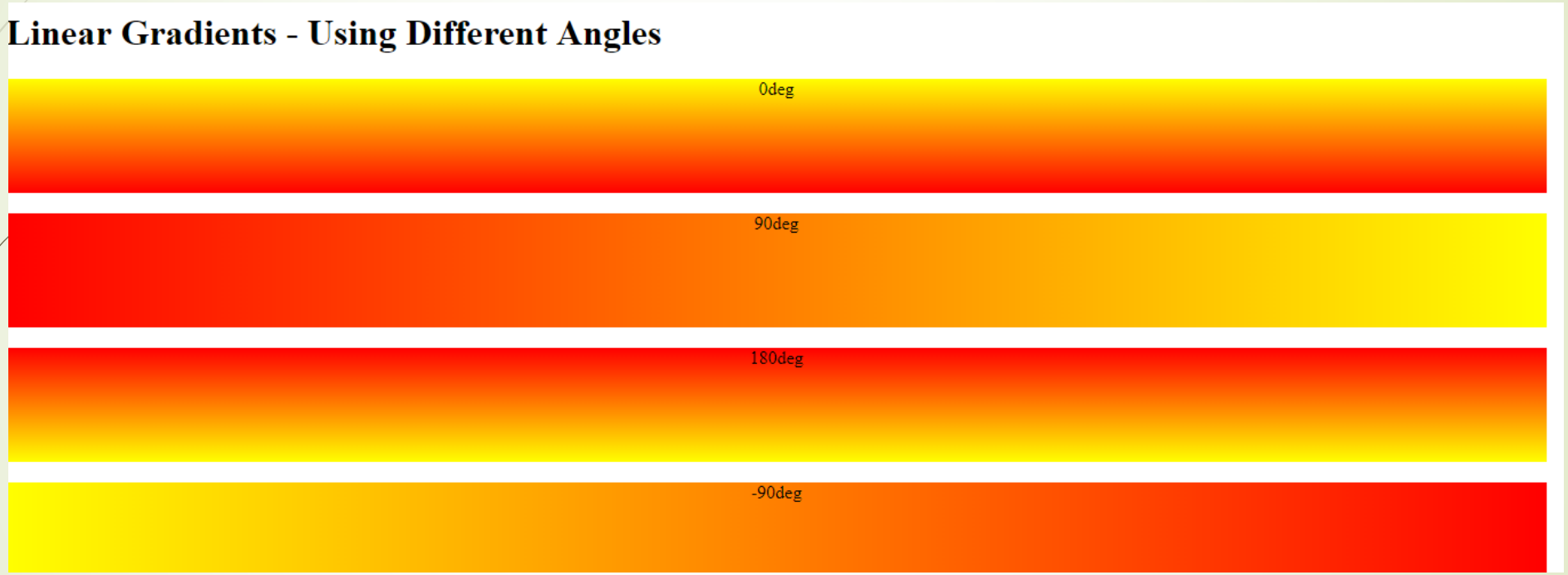
- To create a linear gradient you must define at least two color stops. Color stops are the colors you want to render smooth transitions among.
- You can also set a direction (or an angle) along with the gradient effect.
- The angle is specified as an angle between a horizontal line and the gradient line.

Syntax

- background: linear-gradient(direction, color-stop1, color-stop2, ...);

background: linear-gradient(to right, red , yellow);

CSS Linear Gradients



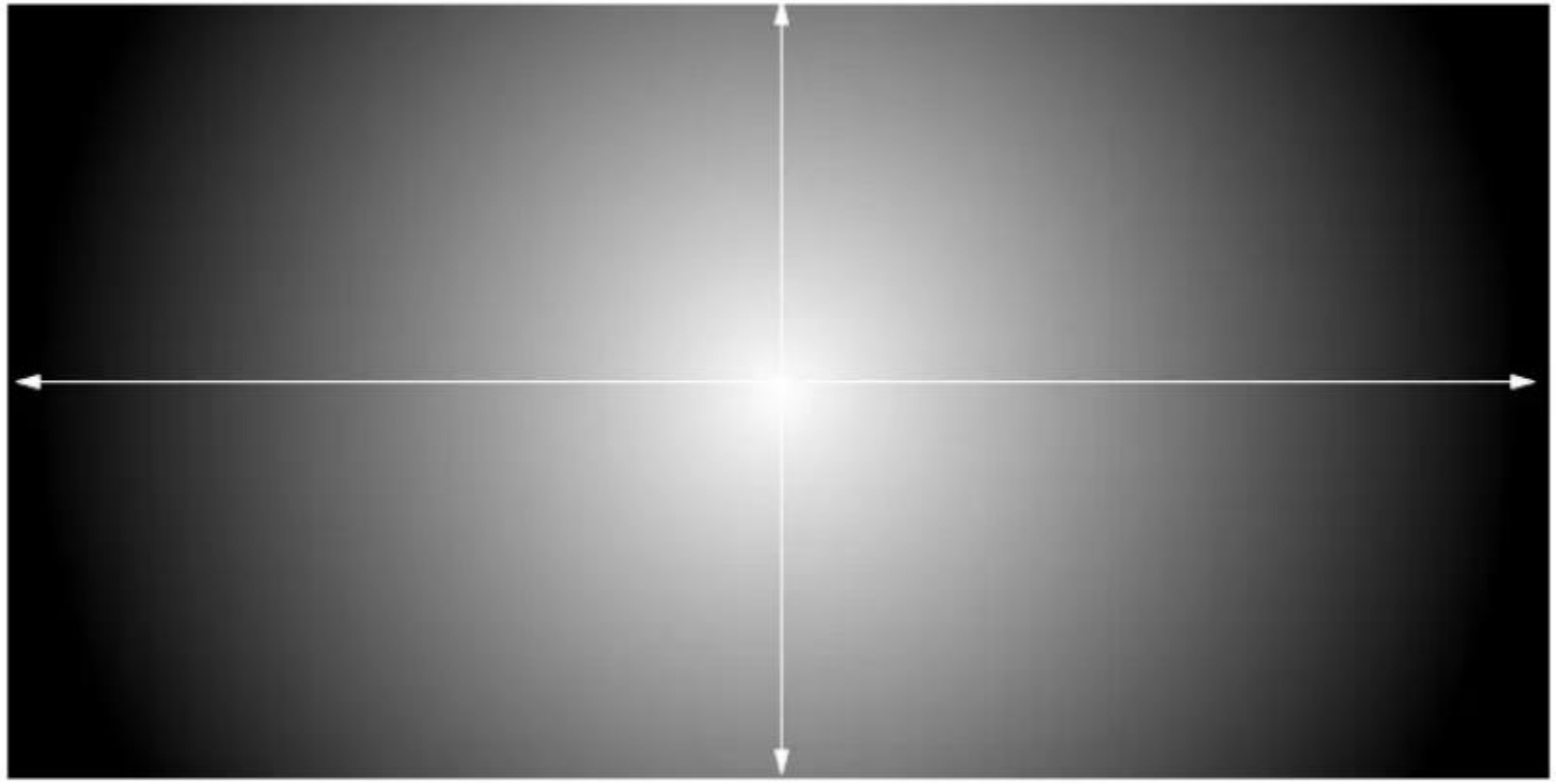
CSS Radial Gradients

- A radial gradient is defined by its center.
- To create a radial gradient you must also define at least two color stops.

- Syntax
- background: radial-gradient(shape size at position, start-color, ..., last-color);

background: radial-gradient(red, yellow, green);

The default is for the first color to start in the `center center` position of the element and then fade to the end color toward the edge of the element. The fade happens at an equal rate no matter which direction.



CSS Animations

- CSS allows animation of HTML elements without using JavaScript.

- An animation lets an element gradually change from one style to another.
- You can change as many CSS properties you want, as many times you want.
- To use CSS animation, you must first specify some keyframes for the animation.
- Keyframes hold what styles the element will have at certain times.

```
/* The animation code */
@keyframes example {
  from {background-color: red;}
  to {background-color: yellow;}
}

/* The element to apply the animation to */
div {
  width: 100px;
  height: 100px;
  background-color: red;
  animation-name: example;
  animation-duration: 4s;
}
```

```
/* The animation code */
@keyframes example {
  0%   {background-color: red;}
  25%  {background-color: yellow;}
  50%  {background-color: blue;}
  100% {background-color: green;}
}

/* The element to apply the animation to */
div {
  width: 100px;
  height: 100px;
  background-color: red;
  animation-name: example;
  animation-duration: 4s;
}
```

Delay an Animation

- The animation-delay property specifies a delay for the start of an animation.
- Negative values are also allowed. If using negative values, the animation will start as if it had already been playing for *N* seconds.

```
div {
  animation-name: example;
  animation-duration: 4s;
  animation-delay: 2s;
}
```

Set How Many Times an Animation Should Run

- The animation-iteration-count property specifies the number of times an animation should run.

Run Animation in Reverse Direction or Alternate Cycles

- The animation-direction property specifies whether an animation should be played forwards, backwards or in alternate cycles.

- The animation-direction property can have the following values:

- normal - The animation is played as normal (forwards). This is default
- reverse - The animation is played in reverse direction (backwards)
- alternate - The animation is played forwards first, then backwards
- alternate-reverse - The animation is played backwards first, then forwards

Specify the Speed Curve of the Animation

- The animation-timing-function property specifies the speed curve of the animation.
- The animation-timing-function property can have the following values:

- ease - Specifies an animation with a slow start, then fast, then end slowly (this is default)
- linear - Specifies an animation with the same speed from start to end
- ease-in - Specifies an animation with a slow start
- ease-out - Specifies an animation with a slow end
- ease-in-out - Specifies an animation with a slow start and end

animation-fill-mode

- CSS animations do not affect an element before the first keyframe is played or after the last keyframe is played. The animation-fill-mode property can override this behavior.

- The animation-fill-mode property can have the following values:
  - forwards - The element will retain the style values that is set by the last keyframe (depends on animation-direction and animation-iteration-count)
  - backwards - The element will get the style values that is set by the first keyframe (depends on animation-direction), and retain this during the animation-delay period

Grid Layout

- The CSS Grid Layout offers a grid-based layout system, with rows and columns, making it easier to design web pages.

- Properties:

  Display: grid; to make HTML element a grid container.

  grid-template-columns : defines the number of columns and width of each column

  grid-template-rows : defines the height of each row.

  column-gap : sets the gap between the columns.

  row-gap : sets the gap between the row.

  gap: shorthand property for the row-gap and the column-gap properties.

Grid Layout

- Properties:

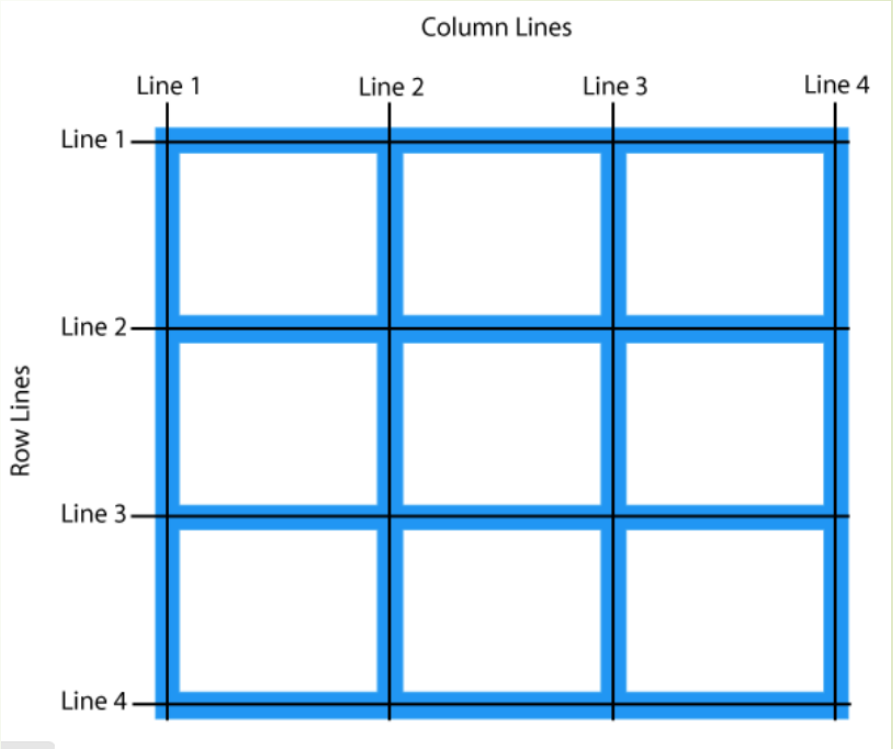
  grid-column-end: Specifies where to end the grid item column.

  grid-column-start: Specifies where to start the grid item column.

  grid-row-end: Specifies where to end the grid item row.

  grid-row-start : Specifies where to start the grid item row.

  grid-column and *grid-row* : A shorthand property



Grid and flexbox

- The basic difference between CSS Grid Layout and CSS Flexbox Layout is that
  - flexbox was designed for layout in one dimension - either a row or a column.
  - Grid was designed for two-dimensional layout - rows, and columns at the same time.

XML

e**x**tensible **M**arkup **L**anguage

-Prof Vaibhavi Patel

HTML

HTML is used to mark up text so it can be displayed to users

HTML describes both structure (e.g. <p>, <h2>,) and appearance (e.g.
, <font>, <i>)

HTML uses a fixed, unchangeable set of tags

XML

XML is used to mark up data so it can be processed by computers

XML describes only content, or "meaning"

In XML, you make up your own tags

Example well Formed XML document

```
<BOOKS>
<book id="123" loc="library">
   <author>Hull</author>
   <title>California</title>
   <year> 1995 </year>
</book>
<article id="555">
   <author>Su</author>
   <title> Purdue</title>
</article>
</BOOKS>
```

XML

- An XML document with correct syntax is called "**Well Formed**".
- An XML document validated against a DTD or Schema is both "Well Formed" and "Valid".
- A "**Valid" XML** document is "Well Formed", as well as it conforms to the rules of a DTD or Schema.

XSLT

XSLT

- XSLT stands for Extensible Stylesheet Language Transformations
- XSLT is used to transform XML documents into other kinds of documents--usually, but not necessarily, XHTML
- XSLT uses *two* input files:
  - The XML document containing the actual data
  - The XSL document containing both the "framework" in which to insert the data, *and* XSLT commands to do so

Very simple example

- File data.xml:

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="render.xsl"?>
<message>Howdy!</message>
```

- File render.xsl:

```
<?xml version="1.0"?>
<xsl:stylesheet version="1.0”
        xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <!-- one rule, to transform the input root (/) -->
  <xsl:template match="/">
    <html><body>
      <h1><xsl:value-of select="message"/></h1>
    </body></html>
  </xsl:template>
</xsl:stylesheet>
```

The .xsl file

- An XSLT document has the .xsl extension
- The XSLT document begins with:

```
<?xml version="1.0"?>

<xsl:stylesheet version="1.0"
    xmlns:xsl="http://www.w3.org/1999/
                XSL/Transform">
```

- Contains one or more templates, such as:

```
<xsl:template match="/"> ... </xsl:template>
```

- And ends with:

```
</xsl:stylesheet>
```

Finding the message text

- The template <xsl:template match="/"> says to select the entire file
  - You can think of this as selecting the *root node* of the XML tree
- Inside this template,
  - <xsl:value-of select="message"/> selects the message child

Putting it together

- The XSL was:

```
<xsl:template match="/">
    <html><body>
      <h1><xsl:value-of select="message"/></h1>
    </body></html>
  </xsl:template>
```

- The <xsl:template match="/"> chooses the root
- The <html><body> <h1> is written to the output file
- The contents of message is written to the output file
- The </h1> </body></html> is written to the output file

- The resultant file looks like:

```
<html><body>
    <h1>Howdy!</h1>
</body></html>
```

THANK YOU