Madan Mohan Malaviya University of Technology, Gorakhpur

CES

Computer
Engineering Societ

CSS

Cascading Style Sheet

INTRODUCTION

CSS stands for Cascading Style Sheets.CSS describes how HTML elements are to be displayed.

CSS Syntax :

A CSS rule consists of a selector and a declaration block : The selector points to the HTML element to style. The declaration block (in curly braces) contains one or more declarations separated by semicolons. Each declaration includes a CSS property name and a value, separated by a colon.

Example :

```
<style>

p {

font-size:32px;

color: red;

text-align: center;}

</style>
```

EXTERNAL, INTERNAL AND INLINE CSS

There are three ways of inserting a style sheet:

- External CSS
- Internal CSS
- Inline CSS

External CSS

Each HTML page must include a reference to the external style sheet file inside the <link> element, inside the head section. If the external style sheet is "mystyle.css", it can be included as :

```
<link rel="stylesheet" href="mystyle.css">
```

Internal CSS

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.

Example of internal CSS :

```
<head>
<style>
h1 {
  color: maroon;
  margin-left: 40px;}
</style>
</head>
```

Discuss meaning of Cascading.
(Priority order)

Inline CSS

An inline style may be used to apply a unique style for a single element.
To use inline styles, add the style attribute to the relevant element. The style attribute can contain any CSS property.

```
<h1 style="color: blue; text-align: center; ">This is a heading</h1>
```

How we change property - property : value;

SELECTORS IN CSS

- CSS ELEMENT SELECTOR
  - Ex: p{
    color:blue;
    }
- CSS CLASS SELECTOR
  - Ex: .red-el{
    color: red;
    }
- CSS ID SELECTOR
  - Ex: #first-para{
    color : green;
    background-color:pink;
    }

Grouping selectors:

div, #first-para{
background-color:pink;
}

CSS STYLING

Font Styling

Choosing the right font has a huge impact on how the readers experience a website.

- font-family : In CSS, we use the font-family property to specify the font of a text.
- font-style : The font-style property is mostly used to specify italic text. This property has three values : normal, oblique and italic.
- font-weight : The font-weight property specifies the weight of a font : normal, bold etc.
- Font-size : The font-size property sets the size of the text.

Example :

```
.p{
        font-family: "Times New Roman", Times, serif;
        font-style: italic;
        font-weight: bold;
        font-size: 40px;
}
```

USE OF GOOGLE FONTS

Discuss color picker tool

Color

This is used to set the color of the text. You can use predefined color names, or RGB, HEX, HSL, RGBA, HSLA values.

Example : <h1 style="color: Tomato;">Hello World</h1>

Height and Width

The height and width properties are used to set the height and width of an element.

Example : div {
```
        height: 200px;
        width: 50%;}
```

Height and width are defined only for block and inline-block element

CSS borders

The CSS border properties allow you to specify the style, width, and color of an element's border

The border-style property specifies what kind of border to display.The following values are allowed:

->Dotted – Defines a dotted border ->dashed – Defines a dashed border

->solid – Defines a solid border ->double – Defines a double border

->groove – Defines a 3D grooved border. The effect depends on the border-color value

->ridge – Defines a 3D ridged border. The effect depends on the border-color value

->inset – Defines a 3D inset border. The effect depends on the border-color value

->outset – Defines a 3D outset border. The effect depends on the border-color value

->none – Defines no border ->hidden – Defines a hidden border

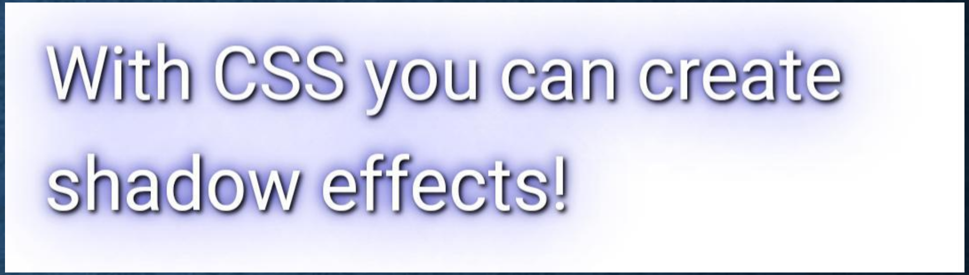
CSS Backgrounds

The CSS background properties are used to add background effects for elements.There are lots of properties to design the background.

CSS background properties are as follows:

- CSS Background- color Property
- CSS Background-image Property
- CSS Background-repeat Property
- CSS Background-attachment Property
- CSS Background-position Property
- CSS Background-origin Property
- CSS Background-clip Property

•


With CSS you can create shadow effects!

BOX SHADOW LINK

CSS Shadow Effects

With CSS you can add shadow to text and to elements.

```
box-shadow: 2px 2px 2px green;
Box-shadow: bottom | right| blur effect| color
```

CSS Text Shadow

The CSS text-shadow property applies shadow to text.

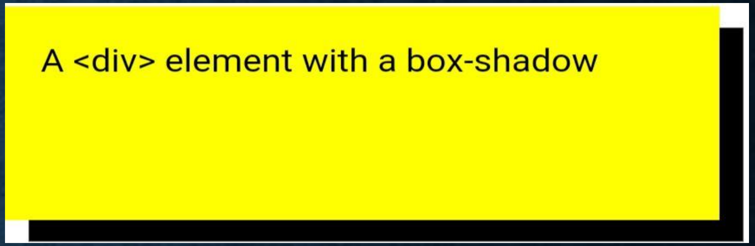
In its simplest use, you only specify the horizontal shadow (2px) and the vertical shadow (2px)

CSS box-shadow Property

The CSS box-shadow property is used to apply one or more shadows to an element.

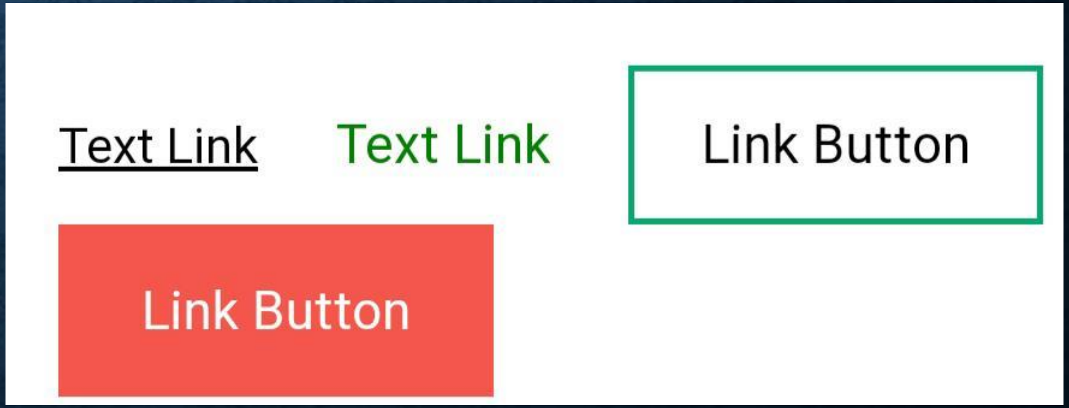
CSS Links

With CSS, links can be styled in many different ways. Links can be styled with any CSS property (e.g. color, font-family, background, etc.).In addition, links can be styled differently 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

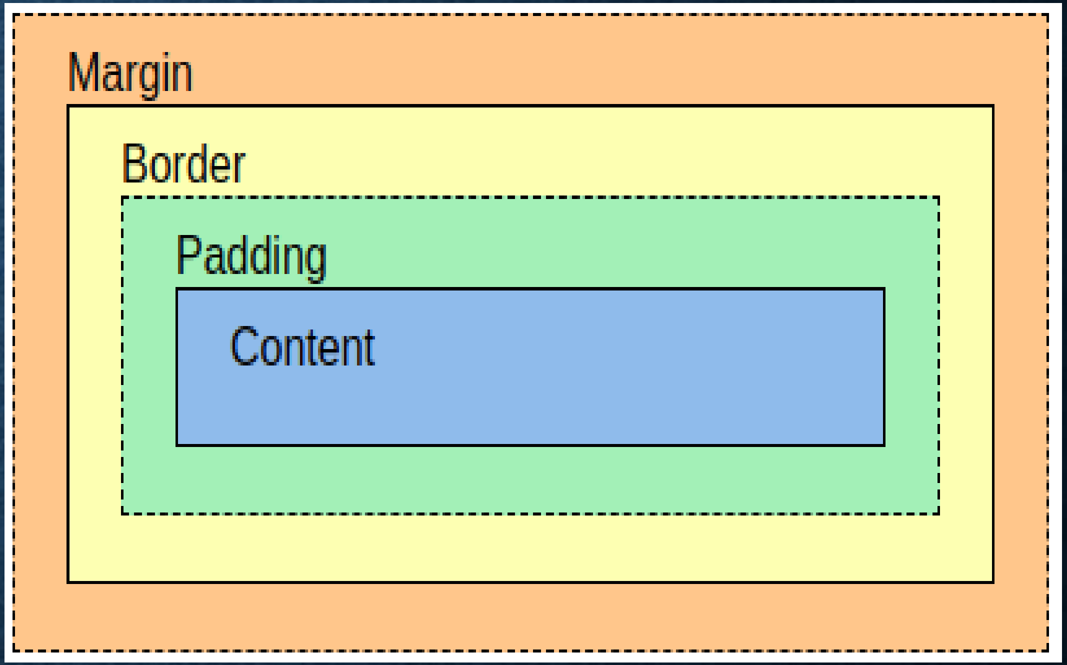


CSS BOX MODEL

The CSS box model is essentially a box that wraps around every HTML element. It consists of: margins, borders, padding, and the actual content.

Explanation of the different parts:

•Content - The content of the box, where text and images appear
•Padding - Clears an area around the content. The padding is transparent
•Border - A border that goes around the padding and content
•Margin - Clears an area outside the border. The margin is transparent



CSS Display

The display property specifies the display behavior (the type of rendering box) of an element.

Syntax :
display: value;
Where possible values are : inline, block, flex, grid, etc.

Position property

The position property specifies the type of positioning method used for an element(static, relative, absolute, fixed, or sticky) .

Static : Elements render in order, as they appear in the document flow
Absolute : positioned relative to its first positioned (not static) ancestor element
Relative : positioned relative to its normal position
Fixed : positioned relative to the browser window
Sticky : The element is positioned based on the user's scroll position

CSS Visibility

The visibility property specifies whether or not an element is visible.

Syntax :
Visibility : value;
Where possible values are : visible, hidden, collapse, initial or inherit.

Z-index

The z-index property specifies the stack order of an element. An element with greater stack order is always in front of an element with a lower stack order.

Note: z-index only works on positioned elements (position: absolute, position: relative, position: fixed, or position: sticky) and flex items (elements that are direct children of display: flex elements).

Note: If two positioned elements overlap without a z-index specified, the element positioned last in the HTML code will be shown on top.

Syntax :

z-index: `auto | number | initial | inherit;`

auto : Sets the stack order equal to its parents. This is default
Number : Sets the stack order of the element. Negative numbers are allowed
Initial : Sets this property to its default value. Read about initial
Inherit : Inherits this property from its parent element.

FLEXBOX

The **Flexbox Layout** (Flexible Box) module aims at providing a more efficient way to lay out, align and distribute space among items in a container, even when their size is unknown and/or dynamic (thus the word “flex”).

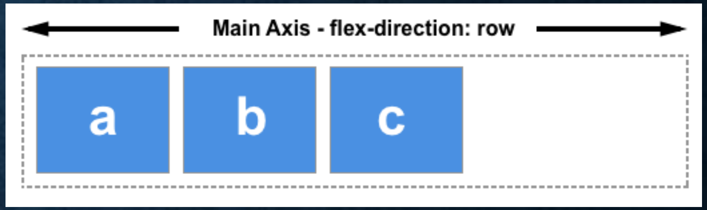
Flex BOX IMP LINK

The two axes of flexbox:

The main axis:

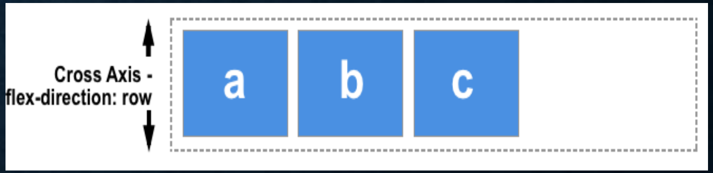
The main axis is defined by flex-direction, which has four possible values:

- row
- row-reverse
- column
- column-reverse



The cross axis:

The cross axis runs perpendicular to the main axis, therefore if your flex-direction (main axis) is set to row or row-reverse the cross axis runs down the columns.



The flex container:

An area of a document laid out using flexbox is called a **flex container**. To create a flex container, we set the value of the area's container’s display property to flex or inline-flex. As soon as we do this the direct children of that container become **flex items**.

- Items display in a row (the flex-direction property's default is row).
- The items start from the start edge of the main axis.
- The items do not stretch on the main dimension, but can shrink.
- The items will stretch to fill the size of the cross axis.
- The flex-basis property is set to auto.
- The flex-wrap property is set to nowrap.

Changing flex-direction:

Adding the flex-direction property to the flex container allows us to change the direction in which our flex items display. Setting flex-direction: row-reverse will keep the items displaying along the row, however the start and end lines are switched.
If we change flex-direction to column the main axis switches and our items now display in a column. Set column-reverse and the start and end lines are again switched.

Media Query:

In CSS, a media query is used to apply a set of styles based on the browser's characteristics including width, height, or screen resolution.

Media Query Syntax:

A media query consists of a media type and can contain one or more expressions, which resolve to either true or false.

```
@media not|only mediatype and (expressions)
{
  CSS-Code;
}
```

CSS3 Media Types:

Value	Description
all	Used for all media type devices
print	Used for printers
screen	Used for computer screens, tablets, smart-phones etc.
speech	Used for screenreaders that "reads" the page out loud

Media Queries Simple Examples:

- The following example changes the background-color to lightgreen if the viewport is 480 pixels wide or wider (if the viewport is less than 480 pixels, the background-color will be pink):

```
@media screen and (min-width: 480px) {
body {
 background-color: lightgreen;
 }
}
```

- The following example shows a menu that will float to the left of the page if the viewport is 480 pixels wide or wider (if the viewport is less than 480 pixels, the menu will be on top of the content):

```
@media screen and (min-width: 480px) {
#leftsidebar {width: 200px; float: left;}
#main {margin-left: 216px;}
}
```

Animation using keyframe:

The @keyframes rule specifies the animation code.
The animation is created by gradually changing from one set of CSS styles to another.

CSS Syntax:

@keyframes *animationname* {*keyframes-selector* {*css-styles;*}}

Property Values:

Value	**Description**
animationname	Required. Defines the name of the animation.
keyframes-selector	Required. Percentage of the animation duration.Legal values: 0-100% from (same as 0%) to (same as 100%) **Note:** You can have many keyframes-selectors in one animation.
css-styles	Required. One or more legal CSS style properties