

The background of the slide is a light beige color. In the top left corner, there is a corkboard with a few papers pinned to it. In the center, there is a large, stylized illustration of a laptop. The laptop screen displays a website layout with a blue header, a main content area with a colorful bar chart, and a footer with three grey rectangular boxes. To the left of the laptop, there is a red fire alarm pull station with two circular gauges. Above the laptop, there are several colorful circles connected by dashed lines, representing a network or data flow. The circles are labeled with various web technologies: 'www' (blue), 'HTML5' (red), 'js' (red), 'Cloud' (grey), 'XML' (orange), and 'PHP' (green).

**ECAP472**

# WEB TECHNOLOGIES

**Dr. Pritpal Singh**

Associate Professor

# Learning Outcomes



After this lecture, you will be able to

- understand CSS position Property.
- go over CSS fonts styles.

# CSS position Property

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

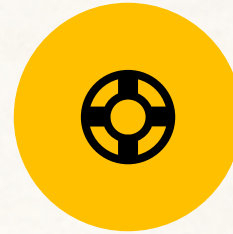
# CSS position-Syntax



STATIC



ABSOLUTE



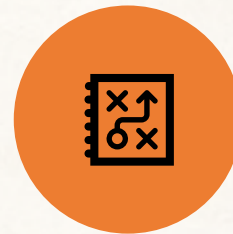
FIXED



RELATIVE



STICKY



INITIAL

# CSS position-Syntax

Value	Description
static	Default value. Elements render in order, as they appear in the document flow.
absolute	The element is positioned relative to its first positioned (not static) ancestor element
fixed	The element is positioned relative to the browser window
relative	The element is positioned relative to its normal position, so "left:20px" adds 20 pixels to the element's LEFT position
sticky	The element is positioned based on the user's scroll position

# Example

```
h2.pos_left
```

```
{
```

```
  position: relative;
```

```
  left: -20px;
```

```
}
```

```
h2.pos_right
```

```
{
```

```
  position: relative;
```

```
  left: 20px;
```

```
}
```

# Static

## CSS Property:

position:

- ☒ static
- ☐ absolute
- ☐ fixed
- ☐ relative
- ☐ initial

## Result:

myDIV

A red square representing a static positioned element. It is positioned in the top-left corner of the result area, indicating it follows the normal flow of the page.

A static positioned element is always positioned according to the normal flow of the page. HTML elements are positioned static by default. Static positioned elements are not affected by the top, bottom, left, right, and z-index properties.

# Absolute

## CSS Property:

## Result:

position:

myDIV

te


☐ relative

☐ initial

An absolutely positioned element is positioned relative to the first parent element that has a position other than static. If no such element is found, it will be positioned on a page relative to the 'top-left' corner of the browser window. The box's offsets further can be specified using one or more of the properties top, right, bottom, and left.



# Fixed

<div>myDIV</div> 	<b>CSS Property:</b>	<b>Result:</b>
	position:	
	<input type="radio"/> static	
	<input type="radio"/> absolute	
	<input checked="" type="radio"/> fixed	
	<input type="radio"/> relative	
	<input type="radio"/> initial	

- Fixed positioning is a subcategory of absolute positioning.
- The only difference is, a fixed positioned element is fixed with respect to the browser's [viewport](#) and does not move when scrolled.

# Relative

## CSS Property:

position:

- ☐ static
- ☐ absolute
- ☐ fixed
- ☒ relative
- ☐ initial

## Result:



myDIV

- A relative positioned element is positioned relative to its normal position.
- In the relative positioning scheme the element's box position is calculated according to the normal flow. Then the box is shifted from this normal position according to the properties — top or bottom and/or left or right.

# Initial

## CSS Property:

position:

- ☐ static
- ☐ absolute
- ☐ fixed
- ☐ relative
- ☒ initial

## Result:

myDIV



# CSS Fonts

- Font Selection is Important
- Choosing the right font has a huge impact on how the readers experience a website.
- The right font can create a strong identity for your brand.
- Using a font that is easy to read is important. The font adds value to your text. It is also important to choose the correct color and text size for the font.

# Generic Font Families

- In CSS there are five generic font families:
- Serif fonts have a small stroke at the edges of each letter. They create a sense of formality and elegance.
- Sans-serif fonts have clean lines (no small strokes attached). They create a modern and minimalistic look.

# Generic Font Families

- Monospace fonts - here all the letters have the same fixed width. They create a mechanical look.
- Cursive fonts imitate human handwriting.
- Fantasy fonts are decorative/playful fonts.

# Difference Between Serif and Sans-serif Fonts



Sans-serif



Serif



Serif  
(red serifs)

# The CSS font-family Property

- In CSS, we use the font-family property to specify the font of a text.



**Font-family Property**



# Example

```
.p1 {
```

```
    font-family: "Times New Roman", Times, serif;
```

```
}
```

```
.p2 {
```

```
    font-family: Arial, Helvetica, sans-serif;
```

```
}
```

```
.p3 {
```

```
    font-family: "Lucida Console", "Courier New", monospace;
```

```
}
```

# CSS Google Fonts

- Google Fonts
- If you do not want to use any of the standard fonts in HTML, you can use Google Fonts.
- Google Fonts are free to use and have more than 1000 fonts to choose from.

# How To Use Google Fonts

```
<head>
```

```
<link rel="stylesheet"
```

```
href="https://fonts.googleapis.com/css?family=Sofia">
```

```
<style>
```

```
body {
```

```
    font-family: "Sofia", sans-serif;
```

```
}
```

```
</style>
```

```
</head>
```

# Practical

---

That's all for now...