

Course Code: CSE3150

Course Title: Front End Full Stack Development

Lab sheet 1 - Module 1

Problem Statement:

Mr. Kiran Student of Presidency University wants to create an attractive web page for Presidency University using HTML and CSS with following specifications.

- Split the web page into two parts using frames, top section 20% and bottom section 80%.
- Place your college logo at the top of the page with menu's (About, Academic, Admission, Facilities, etc). Create hyperlink for all menus.
- Create a web page about the college linked at the bottom of the frame.
- Create a web page about the Academic linked at the bottom of the frame.
- Create a web page about the admission, list out the departments, click any department name that connects to the registration page and apply the position concept in CSS.
- Create a web page about the Facilities, list out the college facilities and display the images using floating concepts in CSS.
- Create a web page about the login page linked at the bottom of the frame.

Solution

Home.html

```
frameset rows="25%, *">
  <frame src="top.html" name="top">
  <frame src="bottom.html" noresize name="bottom" >
</frameset>
</frameset>
```

Top.html

```
<html>
  <head>
    <link rel="stylesheet" href="mystyle.css">
    <style>
      *  {
```

```
margin:0;
padding:0;
box-sizing: border-box;
}
body{
    display: flex;
    justify-content: center;
    align-items: center;
    height: 100vh;
    background-color: #151925;
    font-family: sans-serif;
    font-weight: bold;
}
a
{
    position: relative;
    display: inline-block;
    padding: 25px 15px;
    margin: 20px 10px;
    color: #03e9f4;
    text-decoration: none;
    text-transform: uppercase;
    transition: 0.9s;
    /* letter-spacing: 2px; */
    overflow: hidden;
    margin-right: 20px;
}
a:hover
{
    background: #03e9f4;
    color: #050801;
    box-shadow: 0 0 5px #03e9f4, 0 0 25px #03e9f4, 0 0 50px #03e9f4, 0 0 200px #03e9f4;
    -webkit-box-reflect: below 1px linear-gradient(transparent, #0005);
}
a:nth-child(1)
{
    filter: hue-rotate(270deg);
}
a:nth-child(2)
{
    filter: hue-rotate(110deg);
}
```

```
a:nth-child(3)
{
    filter:hue-rotate(200deg);
}
a:nth-child(4)
{
    filter:hue-rotate(270deg);
}
a:nth-child(5)
{
    filter:hue-rotate(110deg);
}
a:nth-child(6)
{
    filter:hue-rotate(186deg);
}
a span
{
    position: absolute;
    display:block;
}
a span:nth-child(1)
{
    top:0;
    left:0;
    width:100%;
    height:2px;
    background: linear-gradient(90deg, transparent,#03e9f4);
    animation: animate1 1s linear infinite;
}
@keyframes animate1
{
    0%{left:-100%;}
    50%,100%{left:100%;}
}
a span:nth-child(2)
{
    top:-100%;
    right:0;
    width:2px;
    height:100%;
    background: linear-gradient(180deg, transparent,#03e9f4);
    animation: animate2 1s linear infinite;
    animation-delay: 0.25s;
```

```

}

@keyframes animate2
{
  0%{top:-100%}
  50%, 100%{top:100%;}
}

a span:nth-child(3)
{
  bottom:0;
  right:0;
  width:100%;
  height:2px;
  background: linear-gradient(270deg, transparent,#03e9f4);
  animation: animate3 1s linear infinite;
  animation-delay: 0.5s;
}

@keyframes animate3
{
  0%{right:-100%}
  50%, 100%{right:100%;}
}

a span:nth-child(4)
{
  bottom:-100%;
  left:0;
  width:2px;
  height:100%;
  background: linear-gradient(360deg, transparent,#03e9f4);
  animation: animate4 1s linear infinite;
  animation-delay: 0.75s;
  /* border-radius:20px; */
}

@keyframes animate4
{
  0%{bottom:-100%}
  50%, 100%{bottom:100%;}
}

</style>
</head>
<body>

  <imgsrc="pulogo.png" alt="PU Logo" style="background-color:white; padding:10px; border-radius: 10px; margin-top: 5px; margin-bottom: 5px; height:100px; border:3px solid lightgrey">

```

```
<a href="about.html" target="bottom">
    Home<span></span>
    <span></span>
    <span></span>
    <span></span>
</a>
<a href="academics.html" target="bottom">
    <span></span>
    <span></span>
    <span></span>
    <span></span>
    Academic
</a>
<a href="admission.html" target="bottom">
    <span></span>
    <span></span>
    <span></span>
    <span></span>
    Admissions
</a>
<a href="facilities.html" target="bottom">
    <span></span>
    <span></span>
    <span></span>
    <span></span>
    Facilities
</a>
<a href="login.html" target="bottom">
    <span></span>
    <span></span>
    <span></span>
    <span></span>
    Login
</a>
<a href="Register.html" target="bottom">
    <span></span>
    <span></span>
    <span></span>
    <span></span>
    Register
</a>
</body>
</html>
```

About.html

```
<html>
  <head>
    <style>
      body
      {
        background-color: rgb(193, 193, 189);
      }
    </style>
  </head>
<body>
  <h1> Over View</h1>
  <p><b>About the University</b></p>
  <imgsrc="pu.jpg">
</body>
</html>
```

Academics.html

```
<html>
<body>

  <h1 style="text-align:center"> WELCOME<br> TO <br>PRESIDENCY UNIVERSITY
</h1>

  </body>
</html>
```

Admission.html

```
<html>
<head>
  <link rel="stylesheet" href="mystyle.css">
  <style>
    body
    {
      background-color: rgba(189, 189, 188, 0.79);
    }
    .static
    {
      position:sticky;
      top:0;
      right:0;
    }
  </style>
</head>
<body>
```

```

display: inline;
float: right;
font-size: 25px;
border-radius: 15px;
padding: 10px;
background-color: aliceblue;
box-shadow: 0 0 25px black;
}
li
{
list-style: none;
margin: 8px 5px;
}
a
{
text-decoration: none;
color: rgb(50, 53, 51);
}
li:hover
{
border: 2px solid black;
display: inline-block;
padding: 8px;
color: azure;
border-radius: 20px;
background-color: rgb(37, 38, 43);
}
h1
{
/* border: 2px solid black; */
display: inline-block;
padding: 5px;
color: darkslateblue;
text-shadow: 2px 1px 2px rgb(120, 127, 131);
}

</style>
</head>
<body>
<div class="static">
<a href="register.html"> For Admission Click Here</a></div>
<h1>Admissions Open </h1>
<ul>
<a href="register.html"><li>School of Engineering</li></a>
<a href="register.html"><li>School of Law</li></a>
<a href="register.html"><li>School of Management</li></a>
<a href="register.html"><li>School of Commerce</li></a>
<a href="register.html"><li>School of Design</li></a>
</ul>
</body>

```

```
</html>
```

Facilities.html

```
<html>
<head>
<title>Facilities</title>
<style>
body
{
    background-color: rgb(211, 215, 211);
}
img
{
    float:right;
    height:200px;
    width:350px;
    border-radius: 20px;
    margin:110px 19px;
}
p
{
    margin:29px 23px;
    font-size: 24px;
    width:750px;
    text-align: justify;
}
h1
{
    font-family:'Courier New', Courier, monospace;
    font-size:30px;
    margin:25px 203px;
    border:2px solid black;
    display: inline-block;
    padding:5px;
    border-radius: 15px;
    background-color: lightgray;
}
#a2
{
    float: left;
    margin:160px 10px;
}
#P2
{
    /* border:2px solid black; */
    margin:10px 370px;
    width:650px;
}
```

```
#h2
{
    margin:10px 10px;
}
</style>
</head>
<body>

    <imgsrc="Library.jpg" alt="Library" class="image" id="a1">
    <h1>Library</h1>
    <p>
        The Presidency University reference and lending library is well-stocked with a vast collection of books, reports, journals and periodicals. The library also provides access to the best of business and management-related digital resources through its subscription to various databases consisting of scholarly and industry-relevant content. Students utilize these services for their classroom-related work and for further research into their areas of interest.
    </p>

    <imgsrc="classroom.jpg" alt="Library" class="image" id="a2">
    <h1 #id="H2">Classrooms</h1>
    <p id="P2">
        The Presidency University reference and lending library is well-stocked with a vast collection of books, reports, journals and periodicals. The library also provides access to the best of business and management-related digital resources through its subscription to various databases consisting of scholarly and industry-relevant content. Students utilize these services for their classroom-related work and for further research into their areas of interest.
    </p>
    <imgsrc="Library.jpg" alt="Library" class="image" id="a1">
    <h1>Library</h1>
    <p>
        The Presidency University reference and lending library is well-stocked with a vast collection of books, reports, journals and periodicals. The library also provides access to the best of business and management-related digital resources through its subscription to various databases consisting of scholarly and industry-relevant content. Students utilize these services for their classroom-related work and for further research into their areas of interest.
    </p>

    <imgsrc="classroom.jpg" alt="Library" class="image" id="a2">
    <h1 #id="H2">Classrooms</h1>
    <p id="P2">
        The Presidency University reference and lending library is well-stocked with a vast collection of books, reports, journals and periodicals. The library also provides access to the best of business and management-related digital resources through its subscription to various databases consisting of scholarly and industry-relevant content. Students utilize these services for their classroom-related work and for further research into their areas of interest.
    </p>
</body>
</html>
```

Login.html

```
<html>
<head>

    <title>Login_Page</title>
</head>
<body>
    <h1 >
        Login Page - // code here
    </h1>
</body>
</html>
```

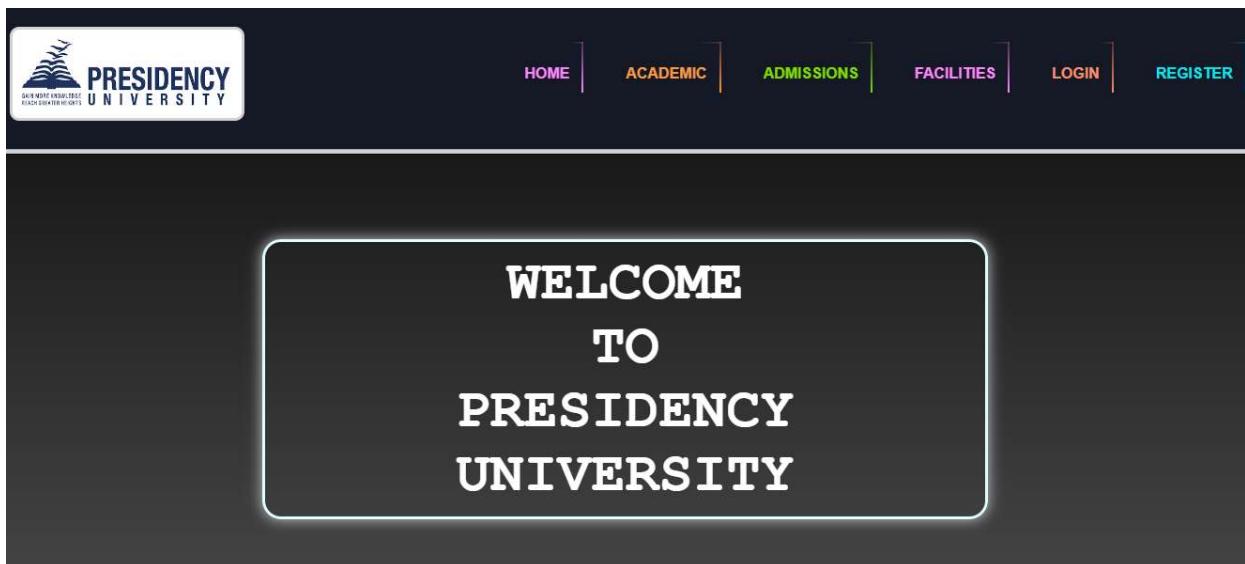
Registration .html

```
<html>
<head>

    <title>Login_Page</title>
</head>
<body>
    <h1 >
        RegistrationPage - // code here
    </h1>
</body>
</html>
```

Output:

Home page:



Admission page

Admissions Open

For Admission Click Here

School of Engineering
School of Law
School of Management
School of Commerce
School of Design
School of Engineering

Facilities Page

library

The Presidency University reference and lending library is well-stocked with a vast collection of books, reports, journals and periodicals. The library also provides access to the best of business and management-related digital resources through its subscription to various databases consisting of scholarly and industry-relevant content. Students utilize these services for their classroom-related work and for further research into their areas of interest.

Login page

HOME ACADEMIC ADMISSIONS FACILITIES LOGIN REGISTER

Log in

Name:

Enter Name:

Password:

Password

Submit Register

Registration page



The logo of Presidency University features a stylized book icon above the word "PRESIDENCY" and "UNIVERSITY" stacked vertically.

HOME ACADEMIC ADMISSIONS FACILITIES LOGIN REGISTER

Student Registration Form

First Name:

Last Name:

Email:

Mobile:

Gender:

Male Female

Date Of Birth:

Address:

Course Code: CSE3150

Course Title: Front End Full Stack Development

Lab sheet 2 - Module 1-HTML Canvas

The purpose for designing a website is to entertain a user of that website. Designer designs the web site using various elements that catches the eyes of the viewers. It is not an easy task to design the website beautifully. It takes a lot of time and effort to design beautiful website. Many elements are considered while designing a website. Among those elements, canvas is also one of them. Therefore, the topic of discussion here is Awesome HTML5 Canvas Examples with Source Code.

Canvas is a graphical element that we can use in our website. The use of canvas is generally done to add up the graphics element in the website. The graphical element means, that can be some kind of effect or animations that we can use in games, movies or anywhere.

HTML5 element <canvas> gives you an easy and powerful way to draw graphics using JavaScript. It can be used to draw graphs, make photo compositions or do simple (and not so simple) animations.

Here is a simple <canvas> element which has only two specific attributes width and height plus all the core HTML5 attributes like id, name and class, etc.

Problem Statement:

The problem statement: To create a canvas drawing application that allows users to draw on the canvas by clicking and dragging the mouse. To achieve this, use HTML5 code that includes a canvas element with an event attribute that listens for mousedown, mousemove, and mouseup events. These events shoud trigger JavaScript functions that draw lines on the canvas based on the user's mouse movements. The canvas element can be styled using CSS to have a black border. Use article, section, attributes to enhance the web page.

Solution

```
<!DOCTYPE html>
<html>
<head>
<title>Canvas Example</title>
<style>
```

```
canvas {
    border: 1px solid black;
}
</style>
</head>
<body>
<header>
<h1>Canvas Example</h1>
<p>Draw on the canvas by clicking and dragging the mouse</p>
</header>
<article>
<h2>Canvas</h2>
<canvas id="myCanvas" width="400" height="400"
onmousedown="startDrawing(event)" onmousemove="drawLine(event)"
onmouseup="stopDrawing(event)"></canvas>
</article>
<script>
var canvas = document.getElementById("myCanvas");
var ctx = canvas.getContext("2d");
var isDrawing = false;
function startDrawing(event) {
    isDrawing = true;
    var x = event.clientX - canvas.offsetLeft;
    var y = event.clientY - canvas.offsetTop;
    ctx.beginPath();
    ctx.moveTo(x, y);
}
function drawLine(event) {
    if (isDrawing) {
        var x = event.clientX - canvas.offsetLeft;
        var y = event.clientY - canvas.offsetTop;
        ctx.lineTo(x, y);
        ctx.stroke();
    }
}
```

```
}

function stopDrawing(event) {
    isDrawing = false;
}

</script>
</body>
</html>
```

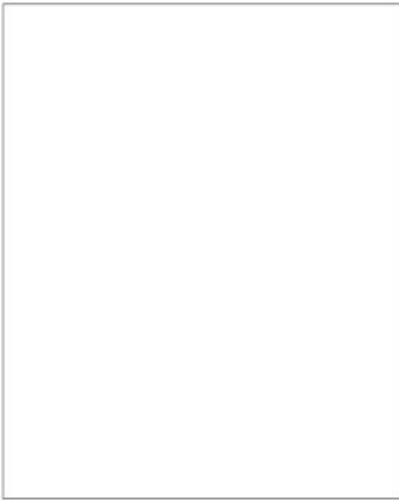
Output:



Canvas Example

Draw on the canvas by clicking and dragging the mouse

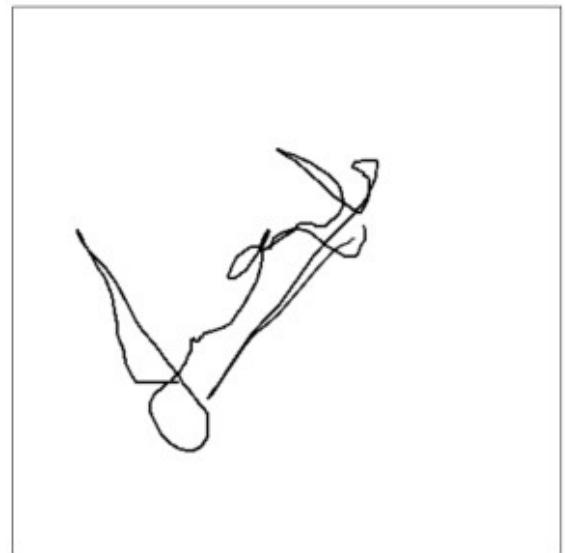
Canvas



Canvas Example

Draw on the canvas by clicking and dragging the mouse

Canvas



Course Code: CSE3150

Course Title: Front End Full Stack Development

Lab sheet 3 - Module 1

Problem Statement:

A company has recently launched a new website and wants to gather information from visitors using a sign-up form. They want to make sure the form is user-friendly, collects the necessary information, and is visually appealing. They also want to ensure the form is accessible on different devices and screens.

To achieve these goals, the company has decided to use Web form 2.0 controls. The form includes inputs for the user's name, date of birth, age, email, and website. The form is designed to be easy to use, with inline validation and conditional logic to show or hide fields based on user input. The company hopes that by using these controls, they will increase the number of sign-ups and improve the overall user experience on their website.

Solution

```
<form>
  <label for="name">Name:</label>
  <input type="text" id="name" name="name" required>

  <label for="dob">Date of Birth:</label>
  <input type="date" id="dob" name="dob" required>

  <label for="age">Age:</label>
  <input type="range" id="age" name="age" min="18" max="100">

  <label for="email">Email:</label>
  <input type="email" id="email" name="email" required>

  <label for="website">Website:</label>
```

```
<input type="url" id="website" name="website">  
  
<button type="submit">Sign Up</button>  
</form>
```

Output:

Name:

Date of Birth: mm/dd/yyyy

Age:

Email:

Website:

Course Code: CSE3150

Course Title: Front End Full Stack Development

Lab sheet 4 - Module 1

Problem Statement: You have been tasked with designing a webpage for a new restaurant that will be opening soon. The webpage should showcase the restaurant's ambiance and cuisine using attractive visual elements.

Note: To create an appealing webpage for the restaurant, you can use various CSS3 features such as colors, text styles, transforms, and gradients. Here's a sample webpage design using these features:

Solution:

Code:index.htm

```
<!DOCTYPE html>
<html>
<head>
  <link rel="stylesheet" href="style.css">
  <title>My Restaurant</title>
</head>
<body>
  <header>
    <h1>Welcome to My Restaurant</h1>
    <nav>
      <a href="#h1">Home</a>
      <a href="#m1">Menu</a>
      <a href="#c1">Contact Us</a>
    </nav>
  </header>
```

```
<main>

<div id="grad2">

    <h2>About Us</h2></div>

    <p>My Restaurant is a new restaurant opening soon in your city. Our mission is to provide our customers with delicious food and excellent service in a comfortable and friendly environment.</p>

</div>

</main>

<h2 id="m1">MENU</h2>

<section>

    <div class="menu-item">

        <h3>pizza</h3>

        <p>If you're a vegetarian, you'll love Domino's Veg Extravaganza, Indi Tandoori Paneer, and Mexican Green Wave. For non-vegetarians, Non-Veg Supreme, Chicken Fiesta, and Indi Chicken Tikka top the list. Though these are the most loved Domino's pizzas along with the cheesy Margherita pizza, you should check out the Domino's menu to see what you like the most.</p>

    </div>

    <div class="menu-item">

        <h3>Burger</h3>

        <p>Nine Different Types of Burgers to Make</p>

    </div>

</section>
```

Turkey burger. ...

Portobello mushroom burger. ...

Veggie burger. ...

Wild salmon burger. ...

Bean burger. ...

Cheeseburger</p>

</div>

<div class="menu-item">

<h3>pasta</h3>

<p> here are some popular pasta dishes you might consider:

Spaghetti carbonara

Linguine with clam sauce

Fettuccine Alfredo

Penne alla vodka

Lasagna

Bucatini all'Amatriciana

Pappardelle with mushroom ragù

Rigatoni alla Norma

Spaghetti aglio e olio (garlic and oil)</p>

</div>

</section>

```
<section>
<div id="grad1">

<h1 id="c1"> contact us</h1>

</div>
<div id="rot">
<h1> feedback</h1></div>
</section>

<footer>
    <p>&copy; 2023 My Restaurant</p>
</footer>

</body>
</html>
```

Output : style.css

```
/* General styles */
body {
    font-family: Arial, sans-serif;
    margin: 0;
    padding: 0;
}
#rot{
    transform: rotate(20deg);
```

```
}

#grad1 {
    height: 200px;
    background-color: red; /* For browsers that do not support gradients */
    background-image: linear-gradient(blue, yellow);
    background-image: conic-gradient(red, yellow, orange);
}

#grad2 {
    height: 200px;
    background-color: red; /* For browsers that do not support gradients */
    background-image: linear-gradient(black, red);
}

/* Header styles */
header {
    background-color: #333;
    color: white;
    padding: 20px;
}

nav a {
    color: white;
    text-decoration: none;
    margin-right: 20px;
}

nav a:hover {
    color: #ffcc00;
}
```

```
/* Main section styles */

main {
    background-color: #eee;
    padding: 50px;
    text-align: center;
}

main h2 {
    color: #333;
    font-size: 36px;
    margin-bottom: 20px;
}

main p {
    color: #666;
    font-size: 18px;
    line-height: 1.5;
    margin-bottom: 50px;
}

main img {
    max-width: 100%;
    height: auto;
    margin-top: 20px;
    box-shadow: 0px 0px 10px #333;
}

/* Menu section styles */

section {
```

```
display: flex;  
flex-wrap: wrap;  
justify-content: space-around;  
background-color: #fff;  
padding: 50px;  
}  
  
.menu-item {  
flex-basis: 30%;  
margin-bottom: 50px;  
box-shadow: 0px 0px 10px #999;  
padding: 20px;  
text-align: center;  
}  
  
.menu-item img {  
max-width: 100%;  
height: auto;  
margin-bottom: 20px;  
}  
  
.menu-item h3 {  
color: #333;  
font-size: 24px;  
margin-bottom: 10px;  
}  
  
.menu-item p {  
color: #666;  
font-size: 16px;  
line-height: 1.5;
```

```
}
```

```
/* Footer styles */
footer {
    background-color: #333;
    color: white;
    padding: 20px;
    text-align: center;
}
footer p {
    font-size: 14px;
    margin: 0;
}
```

```
/* Gradient styles */
header {
    background: linear-gradient(to bottom, #333 0%, #666 100%);
}
nav a:hover {
    background: linear-gradient(to bottom, #ffcc00 0%, #ff9900 100%);
}
section {
    background: radial-gradient(circle, #fff 0%, #eee 100%);
}
.menu-item:hover {
    background: linear-gradient(to bottom, #eee 0%, #fff 100%);
    transform: translateY(-5px);
```

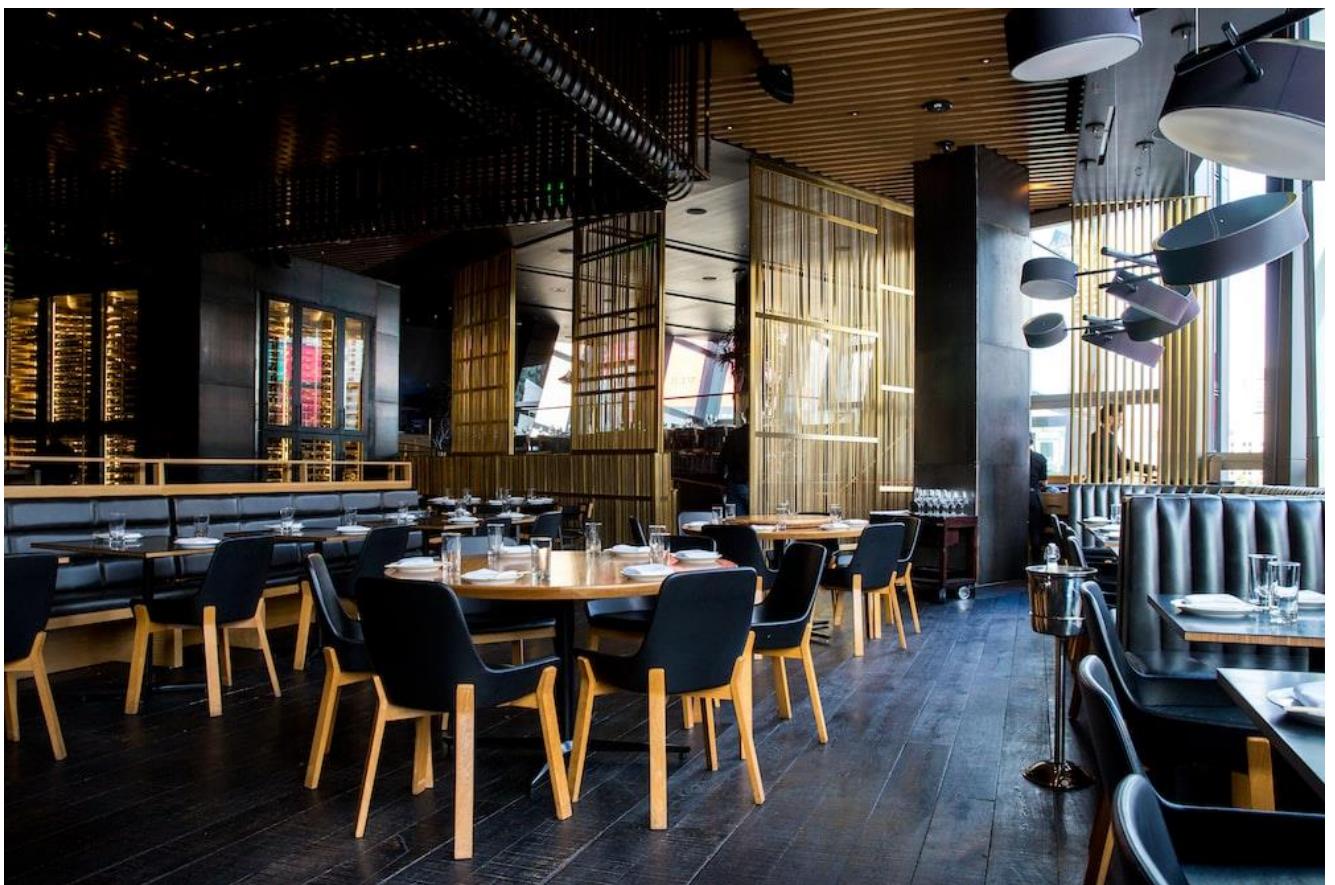
```
}
```

```
footer {
```

```
background: linear-gradient(to bottom, #666 0%, #333 100%);
```

```
}
```

Download :Restaurant image



Output:

A screenshot of a website for "My Restaurant". The header includes the text "Welcome to My Restaurant" and links for "Home", "Menu", and "Contact Us". Below the header, there is a large redacted area where the restaurant's logo or main image would normally appear. To the right of this redacted area, the text "About Us" is visible. At the bottom of the page, there is a small image of the restaurant's interior, which appears to be the same scene shown in the previous photograph.

```
<!DOCTYPE html>
<html>
<head>
    <link rel="stylesheet" href="style.css">
        <title>My Restaurant</title>
</head>
<body>
    <header>
        <h1>Welcome to My Restaurant</h1>
        <nav>
            <a href="#">Home</a>
            <a href="#">Menu</a>
            <a href="#">Contact Us</a>
        </nav>
    </header>

    <main>
        <h2>About Us</h2>
        <p>My Restaurant is a new restaurant opening soon in your city. Our mission is to provide our customers with delicious food and excellent service in a comfortable and friendly environment.</p>
        
    </main>

    <section>
        <div class="menu-item">
            
    <h3>Menu Item 1</h3>
    <p>Lorem ipsum dolor sit amet,
consectetur adipiscing elit. Mauris vel felis sit amet lectus gravida
efficitur non vel nibh.</p>
    </div>
    <div class="menu-item">
        
    <h3>Menu Item 2</h3>
    <p>Phasellus ac vestibulum mauris.
Fusce tincidunt nibh eu libero luctus dictum. Sed quis ultrices
lacus.</p>
    </div>
    <div class="menu-item">
        
    <h3>Menu Item 3</h3>
    <p>Nulla facilisi. Praesent eget
lobortis ex. Fusce dictum libero sit amet mauris auctor, at bibendum
odio placerat.</p>
    </div>
</section>

<footer>
    <p>&copy; 2023 My Restaurant</p>
</footer>

</body>
```

</html>

```
/* General styles */
body {
    font-family: Arial, sans-serif;
    margin: 0;
    padding: 0;
}

/* Header styles */
header {
    background-color: #333;
    color: white;
    padding: 20px;
}
nav a {
    color: white;
    text-decoration: none;
    margin-right: 20px;
}
nav a:hover {
    color: #ffcc00;
}

/* Main section styles */
main {
    background-color: #eee;
    padding: 50px;
    text-align: center;
}
main h2 {
```

```
        color: #333;
        font-size: 36px;
        margin-bottom: 20px;
    }
main p {
    color: #666;
    font-size: 18px;
    line-height: 1.5;
    margin-bottom: 50px;
}
main img {
    max-width: 100%;
    height: auto;
    margin-top: 20px;
    box-shadow: 0px 0px 10px #333;
}

/* Menu section styles */
section {
    display: flex;
    flex-wrap: wrap;
    justify-content: space-around;
    background-color: #fff;
    padding: 50px;
}
.menu-item {
    flex-basis: 30%;
    margin-bottom: 50px;
    box-shadow: 0px 0px 10px #999;
```

```
        padding: 20px;
        text-align: center;
    }

.menu-item img {
    max-width: 100%;
    height: auto;
    margin-bottom: 20px;
}

.menu-item h3 {
    color: #333;
    font-size: 24px;
    margin-bottom: 10px;
}

.menu-item p {
    color: #666;
    font-size: 16px;
    line-height: 1.5;
}

/* Footer styles */
footer {
    background-color: #333;
    color: white;
    padding: 5px;
    text-align: center;
}

footer p {
    font-size: 14px;
    margin: 0;
}
```

```
}

/* Gradient styles */
header {
    background: linear-gradient(to bottom, #333 0%, #666
100%);
}
nav a:hover {
    background: linear-gradient(to bottom, #ffcc00 0%,
#ff9900 100%);
}
section {
    background: radial-gradient(circle, #fff 0%, #eee 100%);
}
.menu-item:hover {
    background: linear-gradient(to bottom, #eee 0%, #fff
100%);
    transform: translateY(-5px);
}
footer {
    background: linear-gradient(to bottom, #666 0%, #333
100%);
}
```

```
<!DOCTYPE html>
<html>
<body>

<script>

document.write("Hello World");

</script>

</body>
</html>
```

```
<!DOCTYPE html>
<html>
<body>

<script>

let x, y, z;
x = 12;
y = 123;
z = x + y;
document.write("The value of z is " + z + ".");

</script>

</body>
</html>
```

```
<!DOCTYPE html>
<html>
<body>

<script>

let num1 = parseInt(prompt('Enter the first number '));
let num2 = parseInt(prompt('Enter the second number '));

let sum = num1 + num2;

document.write(`The sum of ${num1} and ${num2} is ${sum}`);

</script>

</body>
</html>
```

```
<!DOCTYPE html>
<html>
<body>

<script>
let name = 'Welcome';
let name1 = "Presidency";
let result = `The names are ${name} and ${name1}`;
document.write(result);
</script>

</body>
</html>
```

```
<!DOCTYPE html>
<html>
<body>

<script>
let number = prompt("Enter a number: ");

if(number % 2 == 0)
{
    document.write("The number is even.");
}

else {
    document.write("The number is odd.");
}
</script>

</body>
</html>
```

Course Code: CSE3150

Course Title: Front End Full Stack Development

Lab sheet 6 Javascript - Module 2

Problem Statement:

Ravi is trying to write a program in JavaScript to calculate the Fibonacci series of a given number and the square of a given number series as the assignment is given by class teacher, help Ravi to develop and demonstrate a HTML file that includes JavaScript script for the following problems:

a) Input: A number n obtained using prompt

Output: The first n Fibonacci numbers

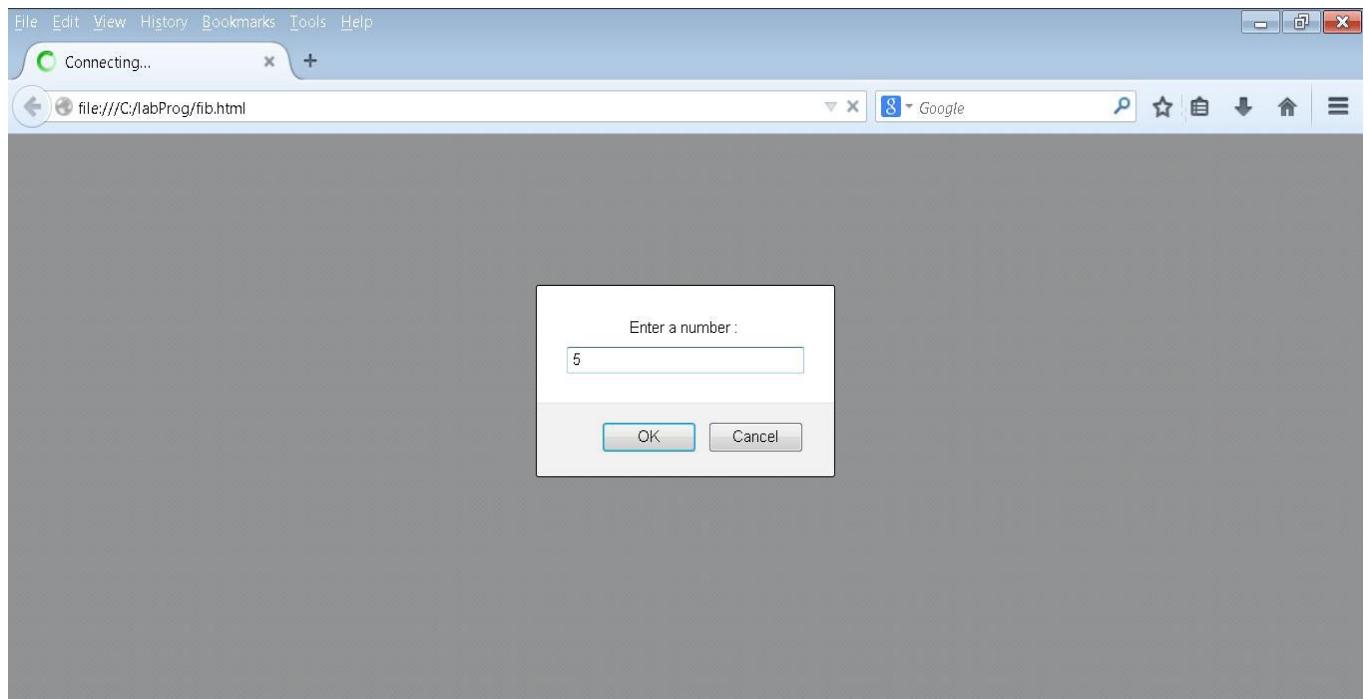
b) Input: A number n obtained using prompt

Output: A table of numbers from 1 to n and their squares using **alert**

a)

```
<html>
<head>
<title>Fibonacci Series</title>
</head>
<body>
<script type="text/javascript">
var fib1=0,fib2=1,fib=0;
var num=prompt("Enter a number : \n", "");
if(num != null && num > 0 )
{
document.write("<h1>The first "+num+" numbers in the fibonacci series </h1>");
if(num==1)
document.write("<h2> "+ fib1 + "</h2>");
else
{
```

```
document.write("<h2>" + fib1 + "</h2>");  
document.write("<h2>" + fib2 + "</h2>");  
}  
for(i=3;i<=num; i++)  
{  
fib= fib1 + fib2;  
document.write("<h2> " + fib + "</h2>");  
fib1=fib2;  
fib2=fib;  
}  
}  
else  
alert("Invalid Input");  
</script>  
</body>  
</html>
```





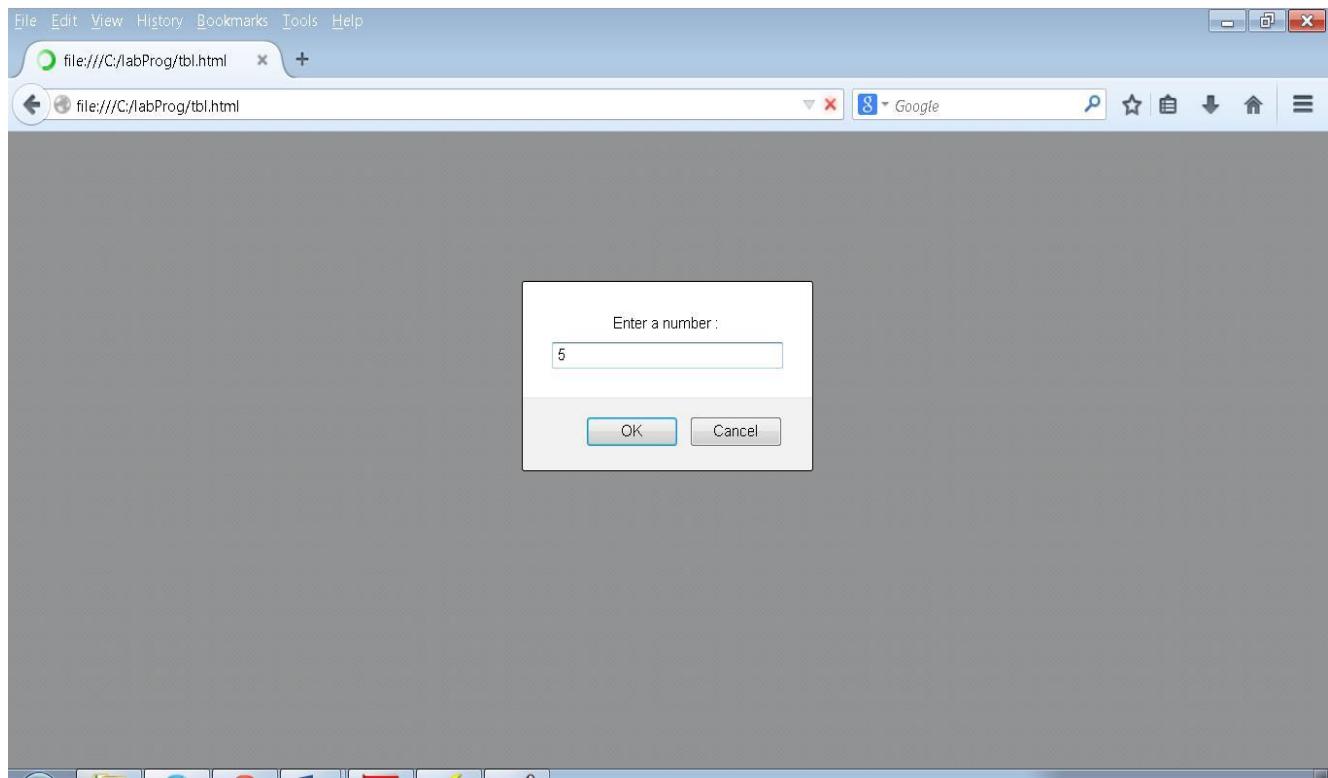
The first 5 numbers in the fibonacci series

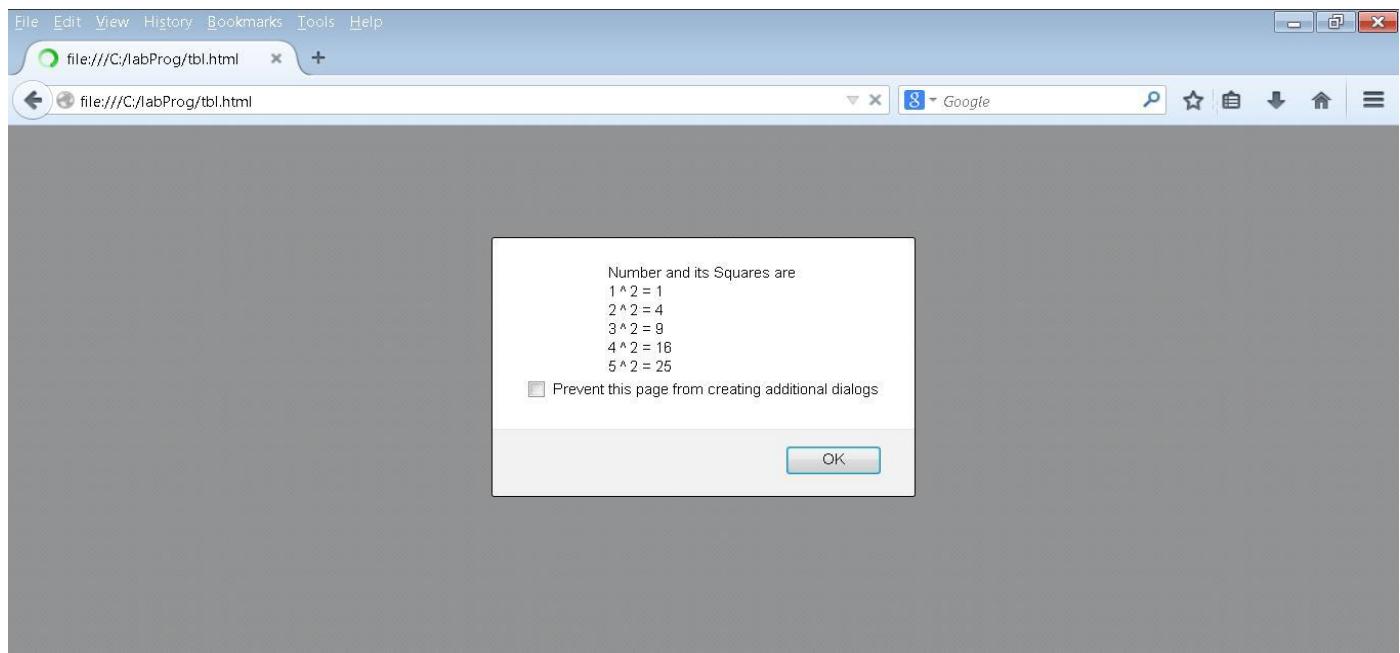
0
1
1
2
3

b) <!DOCTYPE html>

```
<html>
<head>
<title>Number and its squares</title>
</head>
<body>
<script type="text/javascript">
var num = prompt("Enter a number : \n", "");
var msgstr;
if(num > 0 && num !=null){
msgstr="Number and its Squares are \n";
for(i=1;i <= num; i++)
{
msgstr = msgstr + i + " ^ 2 = " + i*i + "\n";
}
alert(msgstr);
}
```

```
else
alert("Invalid Input");
</script>
</body>
</html>
```





OVER
40
YEARS
OF ACADEMIC
WISDOM



PRESIDENCY UNIVERSITY

CSE3150 – Front-end Full Stack Development



**Department of Computer Science Engineering
School of Engineering**

Module I - Syllabus

[Lecture-5Hrs,Knowledge]

Introduction to Agile Methodology; Scrum Fundamentals; Scrum Roles, Artifacts and Rituals;

DevOps – Architecture, Lifecycle, Workflow & Principles;

DevOps Tools Overview – Jenkins, Docker, Kubernetes.

Review of GIT source control.

HTML5 – Syntax, Attributes, Events, Web Forms 2.0, Web Storage, Canvas, Web Sockets;

CSS3 – Colors, Gradients, Text, Transform



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



Full stack front end developer

An individual who operates on the front end and back end of a web - based application is known as a full stack developer.

back end is in charge of the site's architecture and logic in the background,

the front end is in charge of the website's aesthetic appearance and feel.



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



Front end Development (Client side)

- Everything on a web page from the logo, to the search bar, buttons, overall layout and how the user interacts with the page was created by a front end developer. Front end developers are in charge of the look and feel of the website.
- Front end developers also have to make sure the website looks good on all devices (phones, tablets, and computer screens). This is called Responsive Web Design.



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



Back end Development (Server side)

- Back end development refers to the part of the application that the user does not see. Back end developers focus on the logic of the site, creating servers, and working with databases and API's (Application Programming Interfaces).
- For example, the content and layout for a form would be created on the client side, but when the users submits their information it gets processed on the server side (back end).



Full Stack Development	Front End Developer	Back End Developer
Able to create both sides of a website	Is incharge of the client-side	Is incharge of server side
A frontend technology, along with at least one backend language	Knowledge of HTML/CSS, React, Vue.js, and Angular	Ruby, PHP, Node.js, Java, Python, etc.
Complete lifecycle of a web application	Improves the program's usability and aesthetic	Database management



- HTML 1.0 was released in 1993 with the intention of sharing information that can be readable and accessible via web browsers. But not many of the developers were involved in creating websites. So the language was also not growing.
- Then comes HTML 2.0, published in 1995, which contains all the features of HTML 1.0 along with a few additional features, which remained the standard markup language for designing and creating websites until January 1997 and refined various core features of HTML.



- Then comes HTML 3.0, where Dave Raggett introduced a fresh paper or draft on HTML. It included improved new features of HTML, giving more powerful characteristics for webmasters in designing web pages. But these powerful features of the new HTML slowed down the browser in applying further improvements.
- Then comes HTML 4.01, which is widely used and was a successful version of HTML before HTML 5.0, which is currently released and used worldwide. HTML 5 can be said for an extended version of HTML 4.01, which was published in the year 2012.



New Features

- HTML5 introduces a number of new elements and attributes that can help you in building modern websites. Here is a set of some of the most prominent features introduced in HTML5.



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



CSE-367 Data Visualization

- HTML5 comes with a lot of flexibility and it supports the following features
 -
- Uppercase tag names.
- Quotes are optional for attributes.
- Attribute values are optional.
- Closing empty elements are optional.



Type of content	HTML 1.2	HTML 4.01	HTML5	Purpose
Heading	Yes	Yes	Yes	Organize page content by adding headings and subheadings to the top of each section of the page
Paragraph	Yes	Yes	Yes	Identify paragraphs of text
Address	Yes	Yes	Yes	Identify a block of text that contains contact information
Anchor	Yes	Yes	Yes	Link to other web content
List	Yes	Yes	Yes	Organize items into a list
Image	Yes	Yes	Yes	Embed a photograph or drawing into a web page
Table	No	Yes	Yes	Organize data into rows and columns
Style	No	Yes	Yes	Add CSS to control how objects on a web page are presented
Script	No	Yes	Yes	Add Javascript to make pages respond to user behaviors (more interactive)
Audio	No	No	Yes	Add audio to a web page with a single tag
Video	No	No	Yes	Add video to a web page with a single tag
Canvas	No	No	Yes	Add an invisible drawing pad to a web page, on which you can add drawings (animations, games, and other interactive features) using Javascript



- **New Semantic Elements** – These are like <header>, <footer>, and <section>.
- **Forms 2.0** – Improvements to HTML web forms where new attributes have been introduced for <input> tag.
- **Persistent Local Storage** – To achieve without resorting to third-party plugins.
- **WebSocket** – A next-generation bidirectional communication technology for web applications.
- **Server-Sent Events** – HTML5 introduces events which flow from web server to the web browsers and they are called Server-Sent Events (SSE).
- **Canvas** – This supports a two-dimensional drawing surface that you can program with JavaScript.
- **Audio & Video** – You can embed audio or video on your webpages without resorting to third-party plugins.
- **Geolocation** – Now visitors can choose to share their physical location with your web application.
- **Microdata** – This lets you create your own vocabularies beyond HTML5 and extend your web pages with custom semantics.
- **Drag and drop** – Drag and drop the items from one location to another location on the same webpage.



What is CSS3?

- CSS3, also known as Cascading Style Sheets Level 3, is a more advanced version of CSS and the successor of CSS2.
- CSS3 is used for the same thing as CSS, namely to style web pages and make them more attractive and user-friendly.
- In addition, CSS3 incorporates more up-to-date features designed to increase efficiency and make it more convenient for developers to use.



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



What is CSS3?

CSS Color Names

In CSS, a color can be specified by using a predefined color name:



CSS Background Color

```
<h1 style="background-color:DodgerBlue;">Hello World</h1>
```

CSS Text Color

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



**PRESIDENCY
UNIVERSITY**



Private University Estd. in Karnataka State by Act No. 41 of 2013

What is CSS3?

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



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



HTML5 - Web Storage

- HTML5 introduces two mechanisms, similar to HTTP session cookies, for storing structured data on the client side and to overcome following drawbacks.
- Cookies are included with every HTTP request, thereby slowing down your web application by transmitting the same data.
- Cookies are included with every HTTP request, thereby sending data unencrypted over the internet.
- Cookies are limited to about 4 KB of data. Not enough to store required data.
- The two storages are **session storage** and **local storage** and they would be used to handle different situations.



HTML5 - Web Storage

- Session Storage
- The *Session Storage* is designed for scenarios where the user is carrying out a single transaction, but could be carrying out multiple transactions in different windows at the same time.
- Local Storage
- The *Local Storage* is designed for storage that spans multiple windows, and lasts beyond the current session. In particular, Web applications may wish to store megabytes of user data, such as entire user-authored documents or a user's mailbox, on the client side for performance reasons.



HTML5 - Web Storage

- Example
- *For example, if a user buying plane tickets in two different windows, using the same site. If the site used cookies to keep track of which ticket the user was buying, then as the user clicked from page to page in both windows, the ticket currently being purchased would "leak" from one window to the other, potentially causing the user to buy two tickets for the same flight without really noticing.*
- HTML5 introduces the `sessionStorage` attribute which would be used by the sites to add data to the session storage, and it will be accessible to any page from the same site opened in that window, i.e., **session** and as soon as you close the window, the session would be lost.



HTML5 - Web Forms 2.0

- Web Forms 2.0 is an extension to the forms features found in HTML4. Form elements and attributes in HTML5 provide a greater degree of semantic mark-up than HTML4 and free us from a great deal of tedious scripting and styling that was required in HTML4.



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



CSE-367 Data Visualization

HTML5 - Events

- When users visit your website, they perform various activities such as clicking on text and images and links, hover over defined elements, etc. These are examples of what JavaScript calls **events**.

onkeydown	script	Triggers when a key is pressed
onkeypress	script	Triggers when a key is pressed and released
onkeyup	script	Triggers when a key is released
onload	script	Triggers when the document loads
onloadeddata	script	Triggers when media data is loaded
onloadedmetadata	script	Triggers when the duration and other media data of a media element is loaded



HTML5 - Attributes

- elements may contain attributes that are used to set various properties of an element.
- Some attributes are defined globally and can be used on any element, while others are defined for specific elements only.

Attribute	Options	Function
accesskey	User Defined	Specifies a keyboard shortcut to access an element.
align	right, left, center	Horizontally aligns tags
background	URL	Places an background image behind an element
bgcolor	numeric, hexidecimal, RGB values	Places a background color behind an element
class	User Defined	Classifies an element for use with Cascading Style Sheets.
contenteditable	true, false	Specifies if the user can edit the element's content or not.



HTML5 - Syntax

- The HTML 5 language has a "custom" HTML syntax that is compatible with HTML 4 and XHTML1 documents published on the Web, but is not compatible with the more esoteric SGML features of HTML 4.
- HTML 5 does not have the same syntax rules as XHTML where we needed lower case tag names, quoting our attributes, an attribute had to have a value and to close all empty elements.
- HTML5 comes with a lot of flexibility and it supports the following features –
 - Uppercase tag names.
 - Quotes are optional for attributes.
 - Attribute values are optional.
 - Closing empty elements are optional.



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



HTML5 - Syntax

The DOCTYPE

DOCTYPES in older versions of HTML were longer because the HTML language was SGML based and therefore required a reference to a DTD.

HTML 5 authors would use simple syntax to specify DOCTYPE as follows –

```
<!DOCTYPE html>
```

The above syntax is case-insensitive.

Character Encoding

HTML 5 authors can use simple syntax to specify Character Encoding as follows –

```
<meta charset = "UTF-8">
```

The above syntax is case-insensitive.

The <script> tag

It's common practice to add a type attribute with a value of "text/javascript" to script elements as follows –

```
<script type = "text/javascript" src = "scriptfile.js"></script>
```

HTML 5 removes extra information required and you can use simply following syntax –

```
<script src = "scriptfile.js"></script>
```



HTML5 - WebSockets

- WebSockets is a next-generation bidirectional communication technology for web applications which operates over a single socket and is exposed via a JavaScript interface in HTML 5 compliant browsers.
- Once you get a Web Socket connection with the web server, you can send data from browser to server by calling a **send()** method, and receive data from server to browser by an **onmessage** event handler.



HTML5 - WebSockets

WebSocket Attributes

Following are the attribute of WebSocket object. Assuming we created Socket object as mentioned above –

Sr.No.	Attribute & Description
1	Socket.readyState The readonly attribute readyState represents the state of the connection. It can have the following values – <ul style="list-style-type: none">■ A value of 0 indicates that the connection has not yet been established.■ A value of 1 indicates that the connection is established and communication is possible.■ A value of 2 indicates that the connection is going through the closing handshake.■ A value of 3 indicates that the connection has been closed or could not be opened.
2	Socket.bufferedAmount The readonly attribute bufferedAmount represents the number of bytes of UTF-8 text that have been queued using send() method.



HTML5 - WebSockets

WebSocket Events

Following are the events associated with WebSocket object. Assuming we created Socket object as mentioned above –

Event	Event Handler	Description
open	Socket.onopen	This event occurs when socket connection is established.
message	Socket.onmessage	This event occurs when client receives data from server.
error	Socket.onerror	This event occurs when there is any error in communication.
close	Socket.onclose	This event occurs when connection is closed.



HTML5 - WebSockets

WebSocket Methods

Following are the methods associated with WebSocket object. Assuming we created Socket object as mentioned above –

Sr.No.	Method & Description
1	Socket.send() The send(data) method transmits data using the connection.
2	Socket.close() The close() method would be used to terminate any existing connection.



HTML5 - Web Canvas

Canvas

HTML5 element `<canvas>` gives you an easy and powerful way to draw graphics using JavaScript. It can be used to draw graphs, make photo compositions or do simple (and not so simple) animations.



The HTML `<canvas>` element is used to draw graphics on a web page.

The graphic to the left is created with `<canvas>`. It shows four elements: a red rectangle, a gradient rectangle, a multicolor rectangle, and a multicolor text.



HTML5 - Web Canvas

What is HTML Canvas?

- ✓ The HTML <canvas> element is used to draw graphics, on the fly, via JavaScript.
- ✓ The <canvas> element is only a container for graphics. You must use JavaScript to actually draw the graphics.
- ✓ Canvas has several methods for drawing paths, boxes, circles, text, and adding images.



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



CSE-367 Data Visualization

Canvas Examples

A canvas is a rectangular area on an HTML page. By default, a canvas has no border and no content.

The markup looks like this:

```
<canvas id="myCanvas" width="200" height="100"></canvas>
```

Note: Always specify an `id` attribute (to be referred to in a script), and a `width` and `height` attribute to define the size of the canvas. To add a border, use the `style` attribute.

Here is an example of a basic, empty canvas:



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013

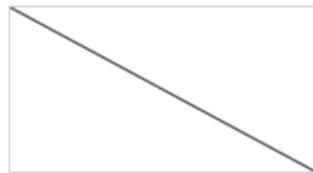


CSE-367 Data Visualization

Add a JavaScript

- After creating the rectangular canvas area, you must add a JavaScript to do the drawing.
- Here are some examples:

Draw a Line



Example

```
<script>
var c = document.getElementById("myCanvas");
var ctx = c.getContext("2d");
ctx.moveTo(0, 0);
ctx.lineTo(200, 100);
ctx.stroke();
</script>
```



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



Introduction to Agile Methodology:

- What is Agile?
- Agile is an iterative approach to project management and software development that helps teams deliver value to their customers faster and with fewer headaches. Instead of betting everything on a "big bang" launch, an agile team delivers work in small, but consumable, increments. Requirements, plans, and results are evaluated continuously so teams have a natural mechanism for responding to change quickly.



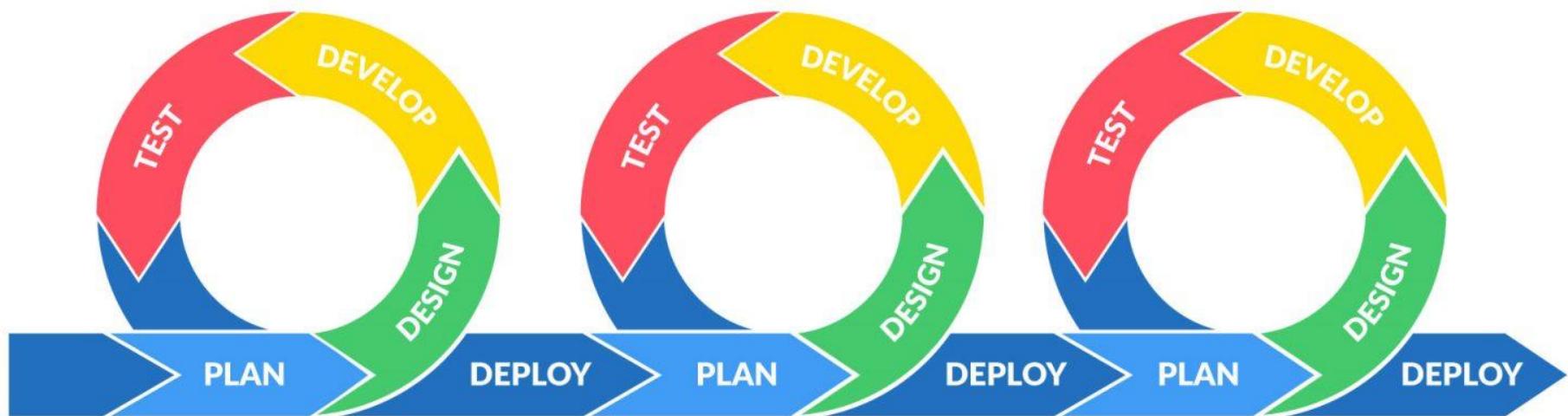
**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013

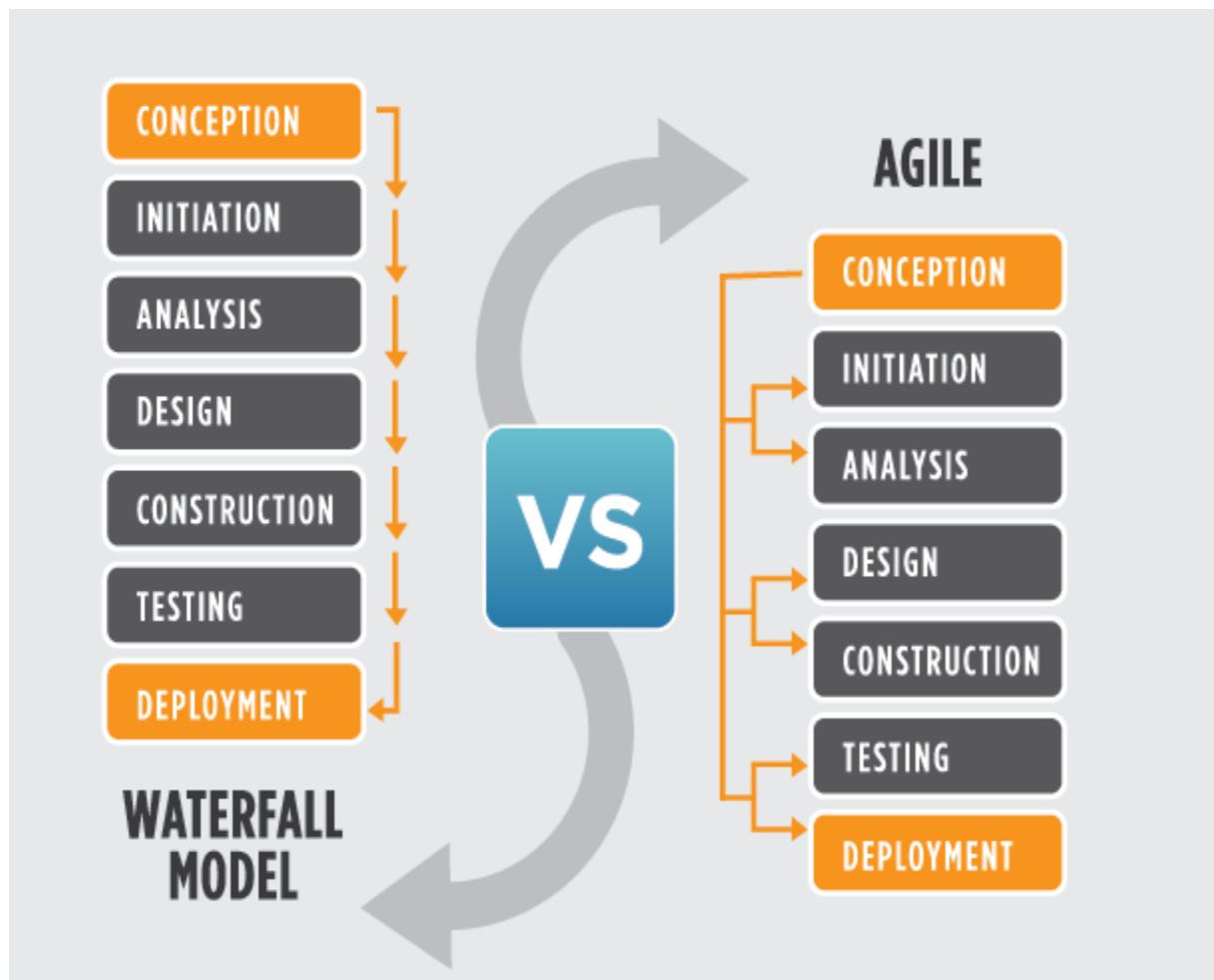


Introduction to Agile Methodology:

- Agile is a project management approach that emphasizes collaboration, flexibility, and customer satisfaction. It values delivering a working product incrementally and embracing change.



Introduction to Agile Methodology:



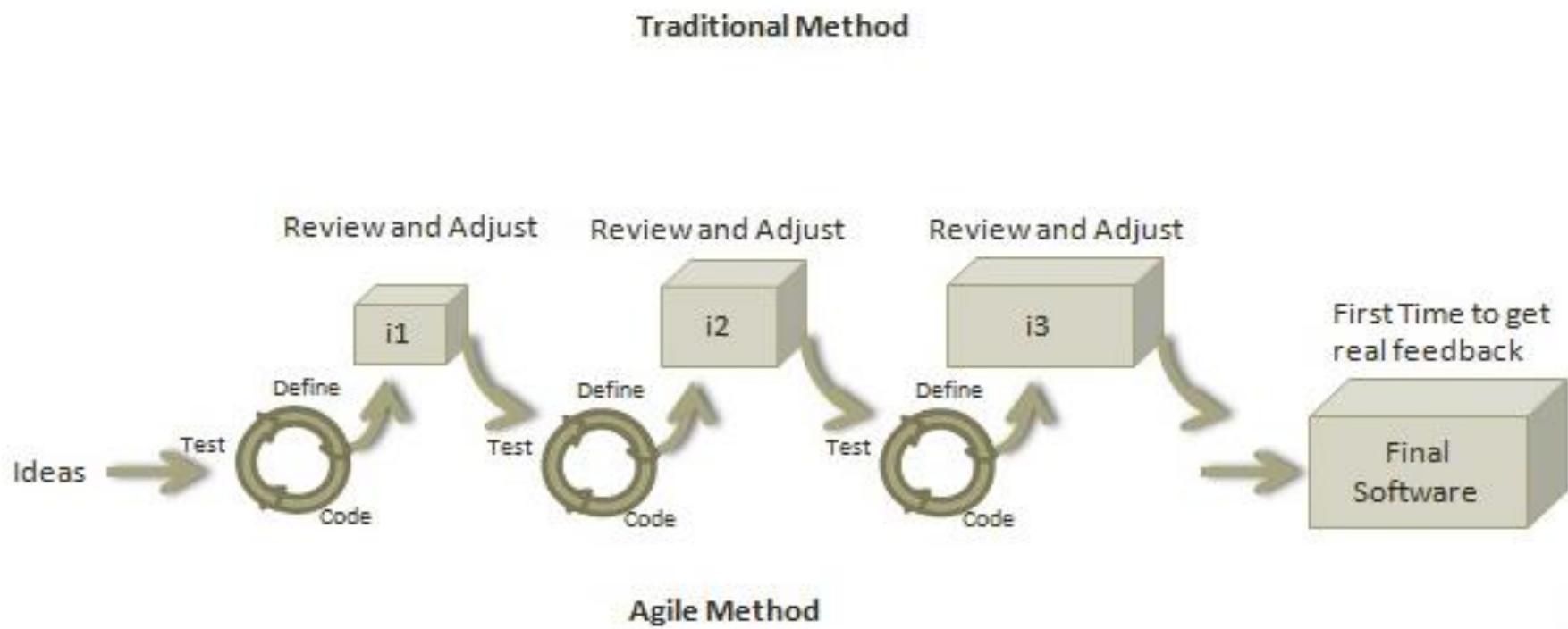
**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



- Agile is a software development methodology to build a software incrementally using short iterations of 1 to 4 weeks so that the development process is aligned with the changing business needs. Instead of a single-pass development of 6 to 18 months where all the requirements and risks are predicted upfront, Agile adopts a process of frequent feedback where a workable product is delivered after 1 to 4 week iteration.





**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



How to Apply Agile to your Everyday Life



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



How to Apply Agile to your Everyday Life

- List out all your goals and tasks, long and short-term. ...
- Divide and prioritize, focusing on the next 15 days.
...
- Do your sprint. ...
- Come back to your list and review your work every 15 days. ...
- Make changes if you need to adapt. ...
- Focus on specific tasks. ...
- You keep your tasks under control.



Manifesto for Agile Software Development

We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

- **Individuals and interactions** over processes and tools
- **Working software** over comprehensive documentation
- **Customer collaboration** over contract negotiation
 Responding to change over following a plan



Principles behind the Agile Manifesto

We follow these principles:

- Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
- Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
- Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
- Business people and developers must work together daily throughout the project.



- Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
- The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
- Working software is the primary measure of progress.
- Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.



Pros of Agile Methodology

- The customer has frequent and early opportunities to see the work being delivered, and to make decisions and changes throughout the development project.
- The customer gains a strong sense of ownership by working extensively and directly with the project team throughout the project.
- If time to market for a specific application is a greater concern than releasing a full feature set at initial launch, Agile can more quickly produce a basic version of working software which can be built upon in successive iterations.
- Development is often more user-focused, likely a result of more and frequent direction from the customer.



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



Cons of Agile Methodology

- Implementation challenges
- Lack of structure
- Difficulty with longer-term planning
- Requires a dedicated team
- Can lead to scope creep



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



What is meant by scrum?

- Scrum is **a management framework that teams use to self-organize and work towards a common goal**. It describes a set of meetings, tools, and roles for efficient project delivery. Much like a sports team practicing for a big match, Scrum practices allow teams to self-manage, learn from experience, and adapt to change.
- **Agile is a development methodology based on iterative and incremental approach. Scrum is an implementation of agile methodology in which incremental changes are delivered timely**



What is meant by scrum?

- A team's success with scrum depends on five values: **commitment, courage, focus, openness, and respect.**
- Why is Agile called scrum?
- It is actually **inspired by a scrum in the sport of rugby**. In rugby, the team comes together in what they call a scrum to work together to move the ball forward. In this context, Scrum is where the team comes together to move the product forward.



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



Who uses Scrum Why?

- Most companies are faced with hardships or difficult tasks at some point in time. By distributing the work among different members in a collaborative group setting, the task tends to become much simpler. Examples of other industries using a scrum approach include **education, military, marketing, and automotive**.



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



- The Scrum framework suggests that each Sprint starts with a brief planning meeting, and ends with a review meeting of the work done at the end of the Sprint. Choose the days and times for these two meetings.. For example: **Planning on Mondays, from 8 am to 9 am, and Review on Fridays, from 4 pm to 5 pm**



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



Physical Scrum board

Goal	TO DO	In Progress	Completed	Defined
Own new job	Get a new job			Proposed changes in job
Get Married		Proposed wedding date		
Find Church	Find church Get confirmed			
Invest	Investing in mutual funds & stocks Invest in real estate			
Fix up house	Get done - Get kitchen - Get bathroom - Get roof - Get exterior	Get done	Get done	
Travel	Get done - Get travel - Get tickets - Get visa			



Digital Scrum board

To Do

- Take EmergenC
- prepare tips

Doing

- + Add a card

Done

- Foundational**
Pick a Venue
- Stationary**
print a schedule for jay
- Stationary**
Do I need programs?
- Ceremony**
Get speech and schedule from 목사님
🕒 3 MH
- Stationary**
Print Favor sign
🕒 1
- Reception**
Write toast
🕒 1
- Honeymoon**
Exchange money
- Reception**
Get Toast from Dane's dad and translate it on paper for Brittany
🕒 1
- Honeymoon**
pay all bills before honeymoon
- Honeymoon**
Create ~~new and unique~~ Itinerary

To Buy

- Decoration**
Flower petals for aisle
+ Add another card

Bought

- Ceremony**
Wedding Rings
- Bridal Party**
Bridesmaid gifts
- Decoration**
5 more air plant containers
- Decoration**
Air plants
- Decoration**
Table Numbers
- Decoration**
smaller vases for bm bouquets on card table
- Ceremony**
Bubbles
- Reception**
Cake topper
- Bridal Party**
pastors gift
- Bridal Party**
parents gifts
- Decoration**
Picture Frames
- Ceremony**
Dane Tux
- gift for pastor
+ Add another card



PRESIDENCY
UNIVERSITY

Private University Estd. in Karnataka State by Act No. 41 of 2013



- *The more tasks you complete, the better you feel, the easier it becomes, and the habit forms.*
- By applying basic Scrum principles to your everyday life you will start being more productive and habits will form.



Scrum FUNDAMENTALS

1. Empirical Process Control: Scrum is based on the principle of empirical process control, which means that it relies on experience and experimentation to continuously improve the process.
2. Cross-functional Teams: Scrum teams are cross-functional, meaning they consist of individuals with a range of skills and expertise.
3. Sprint: A sprint is a time-boxed period, usually one to four weeks, during which the team works on completing a set of tasks.
4. Incremental Delivery: Scrum allows for the delivery of a usable product increment at the end of each sprint.
5. Transparency: Scrum emphasizes transparency in all aspects of the process, from the work being done by the team to the progress being made.



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



Scrum Methodology

Scrum is simple light weighted agile project management methodology that enables product teams to build products incrementally in an iterative fashion through effective team collaboration.

Scrum forms the base for many of the other frameworks and hence it is important for an agile practitioner to understand this methodology.

The Scrum Methodology is defined by:

- Scrum Roles
- Scrum Events
- Scrum Artifacts

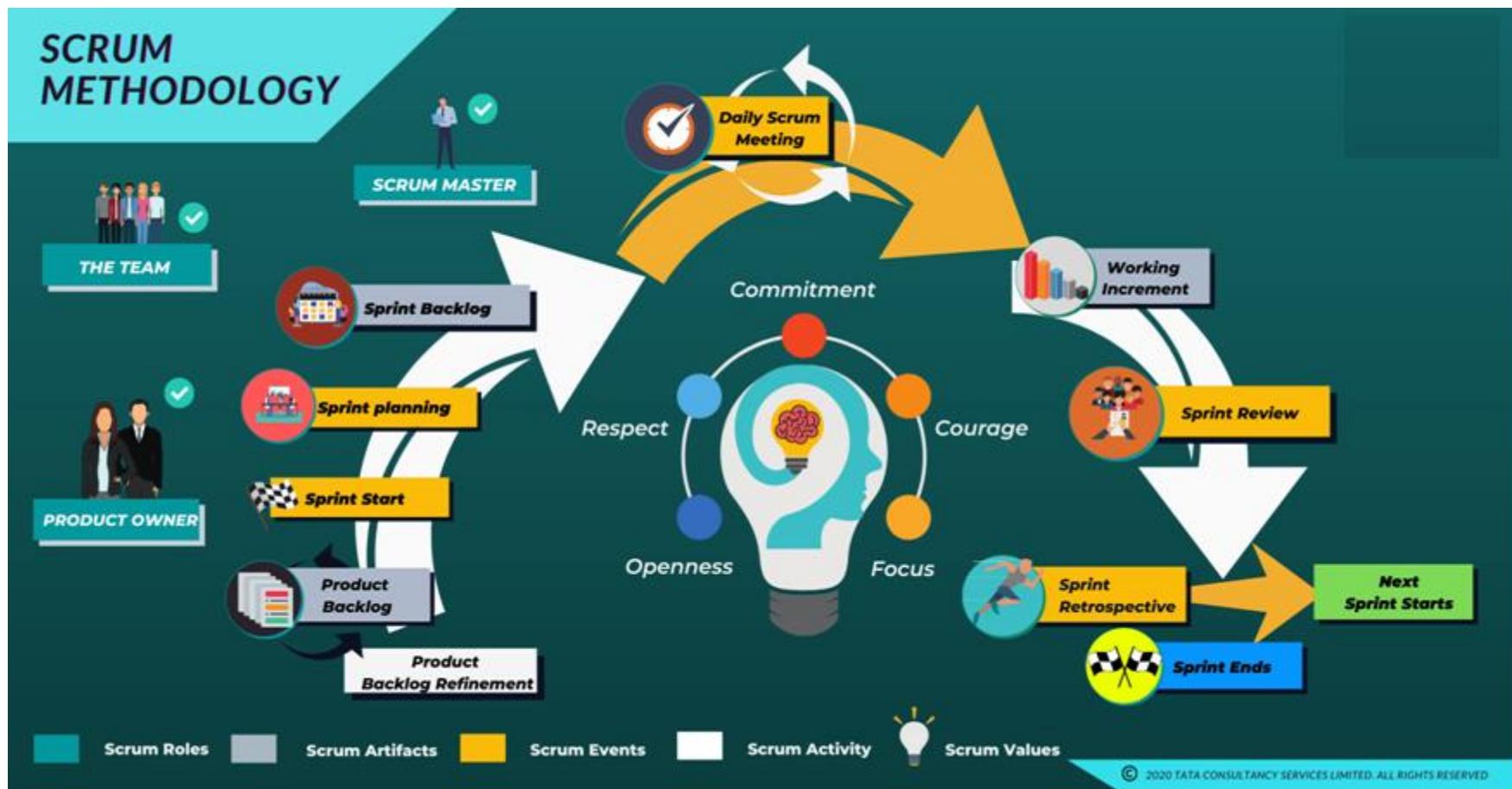


**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



Scrum Methodology



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



What are Scrum Roles?

The scrum team is made up of three roles:

1. A Product Owner
2. A Scrum Master
3. The Development Team.



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



What are Scrum Roles?

1. Product Owner:

A Product Owner in a scrum team decides what needs to be built.

Unlike traditional delivery, this person is a part of the team that delivers the product.

Following are the key responsibilities of the Product Owner:

- Creates the vision
- Represents business, and is responsible for the ROI
- Cascades the vision to the teams
- Owns the backlog of features
- Prioritizes features by market value
- Is empowered to take decisions
- Negotiates with the team and business to deliver the right product at the right time



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



What are Scrum Roles?

2. Scrum Master:

The Scrum Master's major responsibility is to ensure that scrum is understood and practiced by every team member in the true spirit.

Following are the key responsibilities of the Scrum Master :

- Is a servant leader
- Helps remove obstacles/impediments
- Facilitates collaboration
- Teaches scrum to the team.
- Protects the teams from external disruptions such as changes to stories in the current sprint.
- Is a change agent in growing the organization to deliver early and often, and removing waste.



What are Scrum Roles?

3. The Development team:

The Development team in scrum is the team that has all the skills necessary to execute the backlog items.

The following are the special characteristics of 'The Development team':

- Self-organizing
- Empowered
- Cross-functional
- Small-sized
- Co-located
- Committed
- Dedicated



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



What are Scrum Artifacts?

Scrum is an Agile framework for managing software development projects. It involves a set of practices and principles designed to help teams deliver high-quality products in a fast and efficient manner.

The following are the three main artifacts in Scrum:

1. Product Backlog
2. Sprint Backlog
3. Increment



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



Scrum Artifacts

1. Product backlog

A product backlog is a dynamic list of functionalities the product might include, such that it provides value to users.

These are a few unique characteristics of a product backlog:

- It is dynamic in nature as it evolves based on changing market needs
- Lists all the features and capabilities that will be taken up in iteration and delivered as a product increment
- It is refined on a continuous basis. The Product Owner and Development team collaborate and update the details, estimate, and prioritize based on business value and size



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



Scrum Artifacts

2. Sprint backlog:

Sprint backlog is a subset of the entire product backlog that the scrum team plans to implement in one iteration or sprint.

Sprint backlog has:

- Subset of product backlog items that the teams commit to implement in one sprint
- Items broken into smaller pieces of work as tasks
- A focus on 'HOW' the team does the work and delivers the value in one sprint
- A story or task board that is used by the teams to view backlog and what the individuals sign up for work after backlog prioritization
- Provision for the Development teams to track the sprint progress and check their alignment with sprint goals



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



Scrum Artifacts

3. Increment:

An increment is the work delivered at the end of every sprint. Typically, after every iteration there will be a Product Increment (PI) that delivers value and the final product will be a working software.

This increment is a sum of all the capabilities that were delivered in the previous sprints as a part of the PI. At the end of every sprint, the Product Owner decides whether to release the working product increment or wait until the next release.



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



Scrum

Scrum events and rituals

Events: Events are time-boxed activities with specific agendas and outcomes that must be achieved within the time frame. There are four main events in Scrum: Sprint, Sprint Planning, Daily Scrum, and Sprint Review.

Rituals: Rituals are repeated activities that are performed regularly to reinforce the principles and values of Scrum. Rituals are less formal and less time-boxed than events and do not have specific agendas or outcomes. Examples of Scrum rituals include the product backlog refinement, sprint retrospectives, and product demos.



SRUM Rituals

Scrum rituals are regular activities that help reinforce the principles and values of Scrum and improve the development process.

Some of the main Scrum rituals:

Sprint Planning: The team meets to plan the upcoming sprint, review the product backlog, and determine which items can be completed during the sprint. This is a collaborative effort between the development team and the product owner.

Daily Scrum: Also known as the "stand-up," this is a daily meeting where the development team members share updates on their progress and discuss any obstacles they may be facing. The purpose of the Daily Scrum is to increase transparency and accountability within the team.



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



SRUM Rituals

Sprint Review: The team meets to review the work completed during the sprint and demonstrate the product increments to stakeholders. This is an opportunity for the team to receive feedback and make improvements for the next sprint.

Sprint Retrospective: The team meets to reflect on the past sprint and identify ways to improve the development process for the next sprint. The Sprint Retrospective is an opportunity for the team to continuously improve and adapt to changing circumstances.

Product Backlog Refinement: The product backlog is regularly reviewed and updated to ensure that it accurately reflects the priorities and goals of the project. This helps to ensure that the team is focused on delivering the most valuable features and functionality to the customer.



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



What is DevOps?

The DevOps is a combination of two words, one is software Development, and second is Operations. This allows a single team to handle the entire application lifecycle, from development to **testing**, **deployment**, and **operations**. DevOps helps you to reduce the disconnection between software developers, quality assurance (QA) engineers and system administrators.

Various DevOps tools such as Git, Ansible, Docker, Puppet, Jenkins, Chef, Nagios, and Kubernetes.

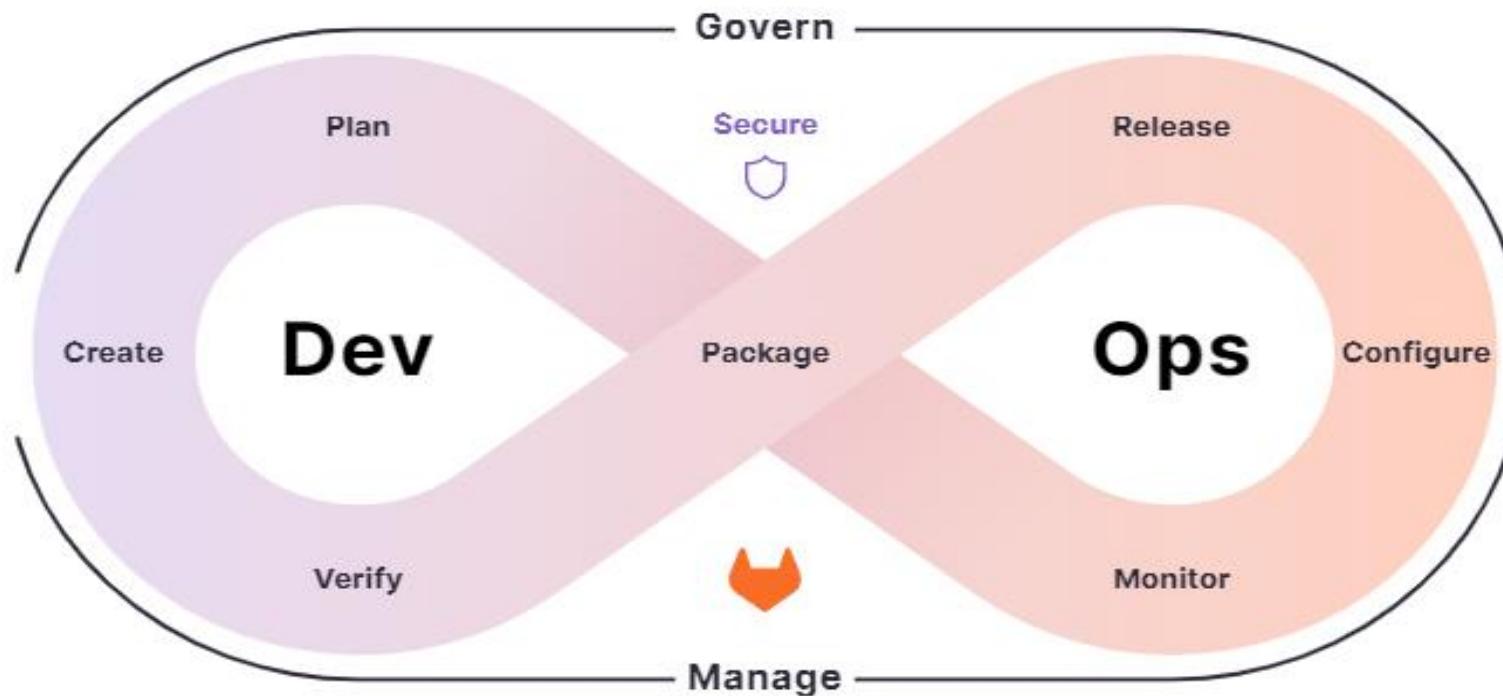
DevOps practices enable software development (dev) and operations (ops) teams to accelerate delivery through automation, collaboration, fast feedback, and iterative improvement.



- **DevOps** is a combination of software development (dev) and operations (ops).
- It is defined as a software engineering methodology which aims to integrate the work of development teams and operations teams by facilitating a culture of collaboration and shared responsibility.



DevOps combines development and operations to increase the efficiency, speed, and security of software development and delivery compared to traditional processes. A more nimble software development lifecycle results in a competitive advantage for businesses and their customers.



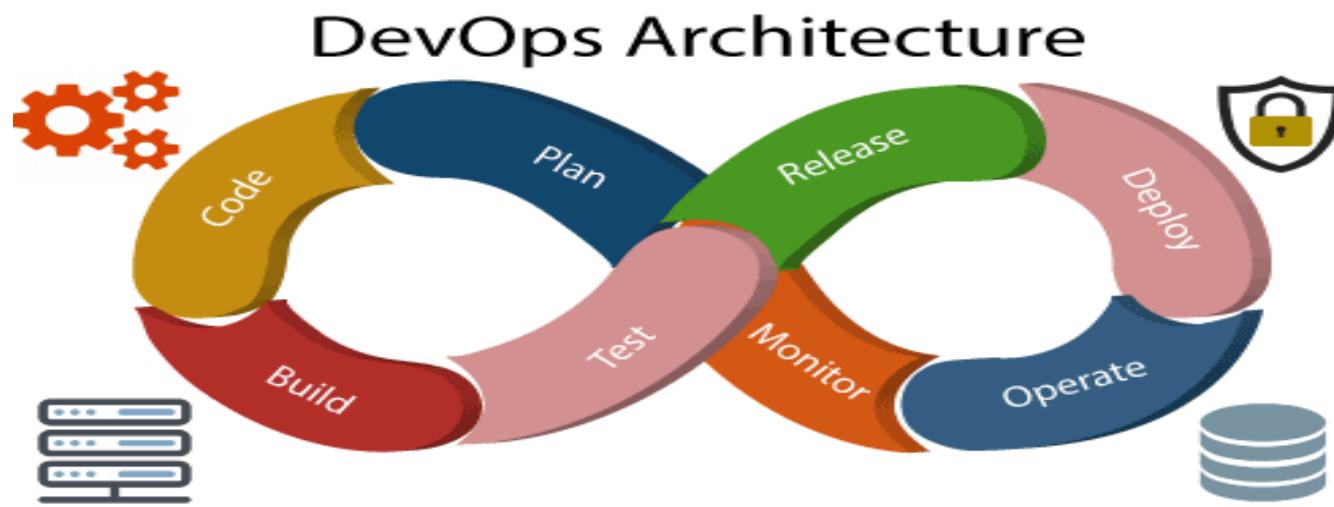
Benefits of DevOps

- **Speed.** DevOps practices let you move at the velocity you need to innovate faster, adapt to changing markets better, and become more efficient at driving business results.
- **Rapid delivery.** When you increase the pace of releases, you can improve your product faster and build competitive advantage.
- **Reliability.** DevOps practices like continuous integration and continuous delivery can ensure the quality of application updates and infrastructure changes so you can reliably deliver at a more rapid pace while maintaining an optimum experience for end users.
- **Improved collaboration.** Under a DevOps model, developers and operations teams collaborate closely, share responsibilities, and combine their workflows. This reduces inefficiencies and saves time.
- **Security.** You can adopt a DevOps model without sacrificing security by using automated, integrated security testing tools.



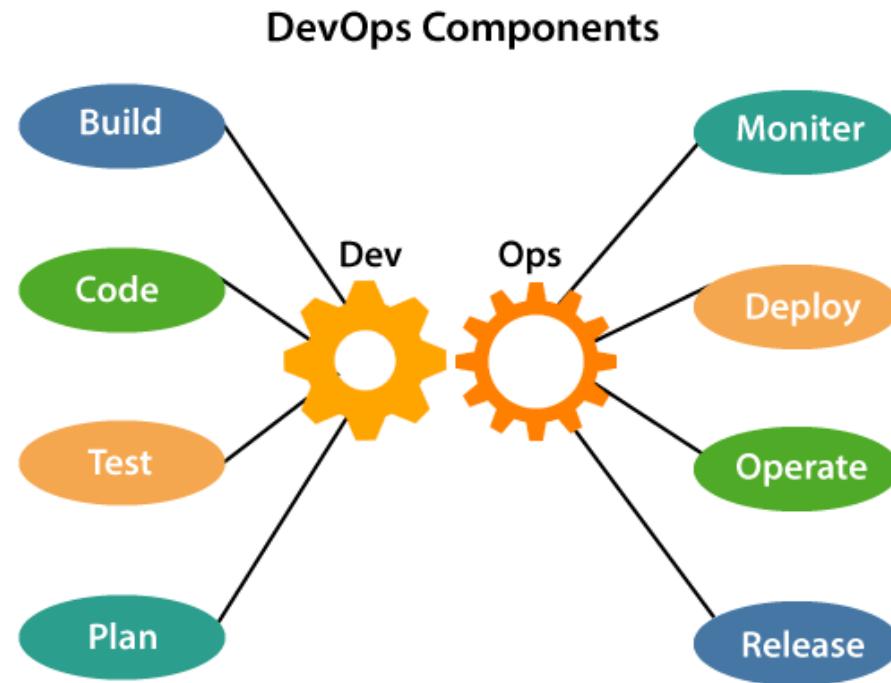
DevOps Architecture

The operation consists of the administrative processes, services, and support for the software. When both the development and operations are combined with collaborating, then the DevOps architecture is the solution to fix the gap between deployment and operation terms; therefore, delivery can be faster.



DevOps Architecture

The various components that are used in the DevOps architecture:



DevOps Architecture

The various components that are used in the DevOps architecture:

- Build: Cloud and shared resources are used to control resource usage.
- Code: Good practices like Git make code use, tracking, and reusability easier.
- Test: Automated testing saves time and prepares the app for production.
- Plan: Agile methodology helps plan development and improve productivity.
- Monitor: Continuous monitoring reduces risk of failure and tracks app health.
- Deploy: Automated deployment captures insights and optimizes performance.
- Operate: DevOps collaborates throughout the service lifecycle.
- Release: Deployment to production is done manually to minimize impact.



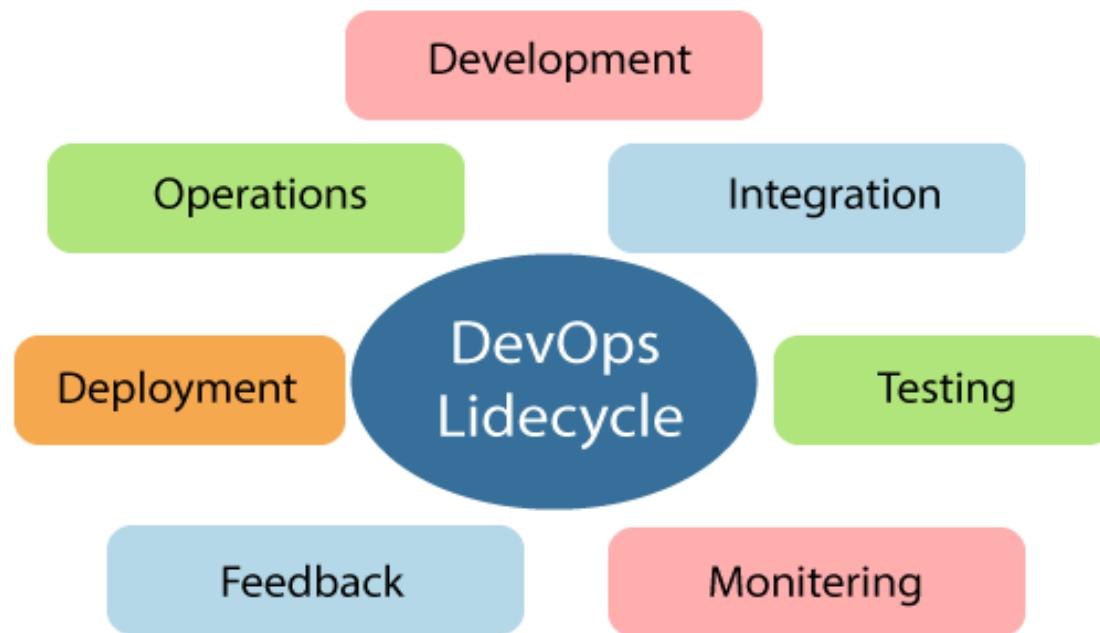
**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



DevOps Lifecycle

DevOps defines an agile relationship between operations and Development. It is a process that is practiced by the development team and operational engineers together from beginning to the final stage of the product



DevOps Lifecycle

- The DevOps lifecycle consists of seven phases: Continuous Development, Continuous Integration, Continuous Testing, Continuous Monitoring, Continuous Feedback, Continuous Deployment, and Continuous Operations. Each phase plays a crucial role in the software development process and contributes to the overall efficiency and effectiveness of the DevOps approach.
- Continuous Development involves planning and coding the software, where the vision of the project is decided and the developers begin coding the application.
- Continuous Integration is the heart of the DevOps lifecycle and involves committing changes to the source code frequently and building the code, including unit testing, integration testing, code review, and packaging.
- Continuous Testing involves constantly testing the developed software for bugs using automation testing tools such as TestNG, JUnit, and Selenium.



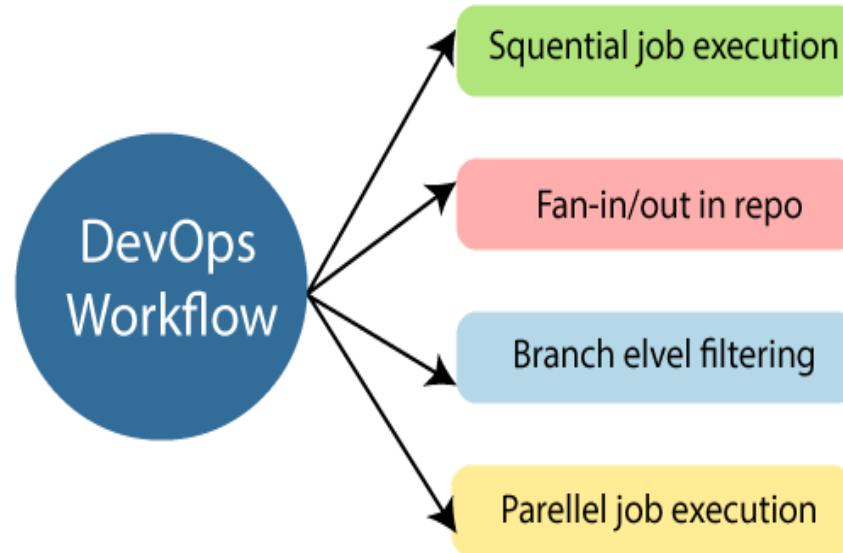
DevOps Lifecycle contd..

- Continuous Monitoring involves monitoring the operational factors of the DevOps process and recording important information about the use of the software.
- Continuous Feedback involves analyzing the results from the operations of the software to continuously improve the application development. Continuous Deployment involves deploying the code to the production servers and ensuring that the code is correctly used on all servers. Configuration management tools play a crucial role in executing tasks frequently and quickly in this phase.
- Continuous Operations are based on continuity with complete automation of the release process and allow organizations to accelerate the overall time to market. With DevOps, the software product becomes more efficient and increases the overall count of interested customers.



DevOps Workflow

- DevOps workflow provides a visual overview of the sequence in which input is provided. Also, it tells about which one action is performed, and output is generated for an operations process
- DevOps workflow allows the ability to separate and arrange the jobs which are top requested by the users. Also, it gives the ability to mirror their ideal process in the configuration jobs.



DevOps Principles

The main principles of DevOps are Continuous delivery, automation, and fast reaction to the feedback.

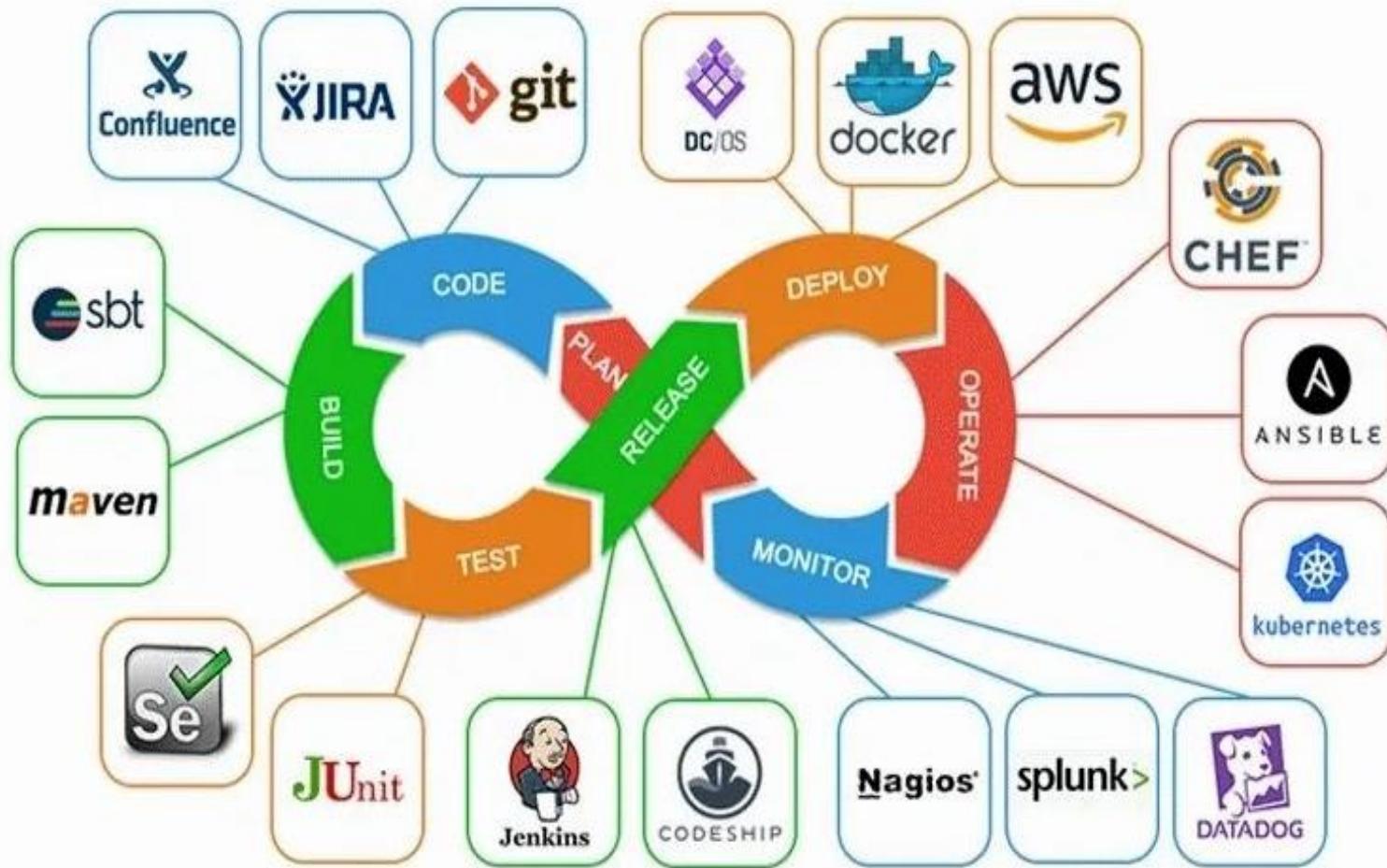
- **End to End Responsibility**
- **Continuous Improvement**
- **Automate Everything**
- **Custom Centric Action**
- **Monitor and test everything**
- **Work as one team**

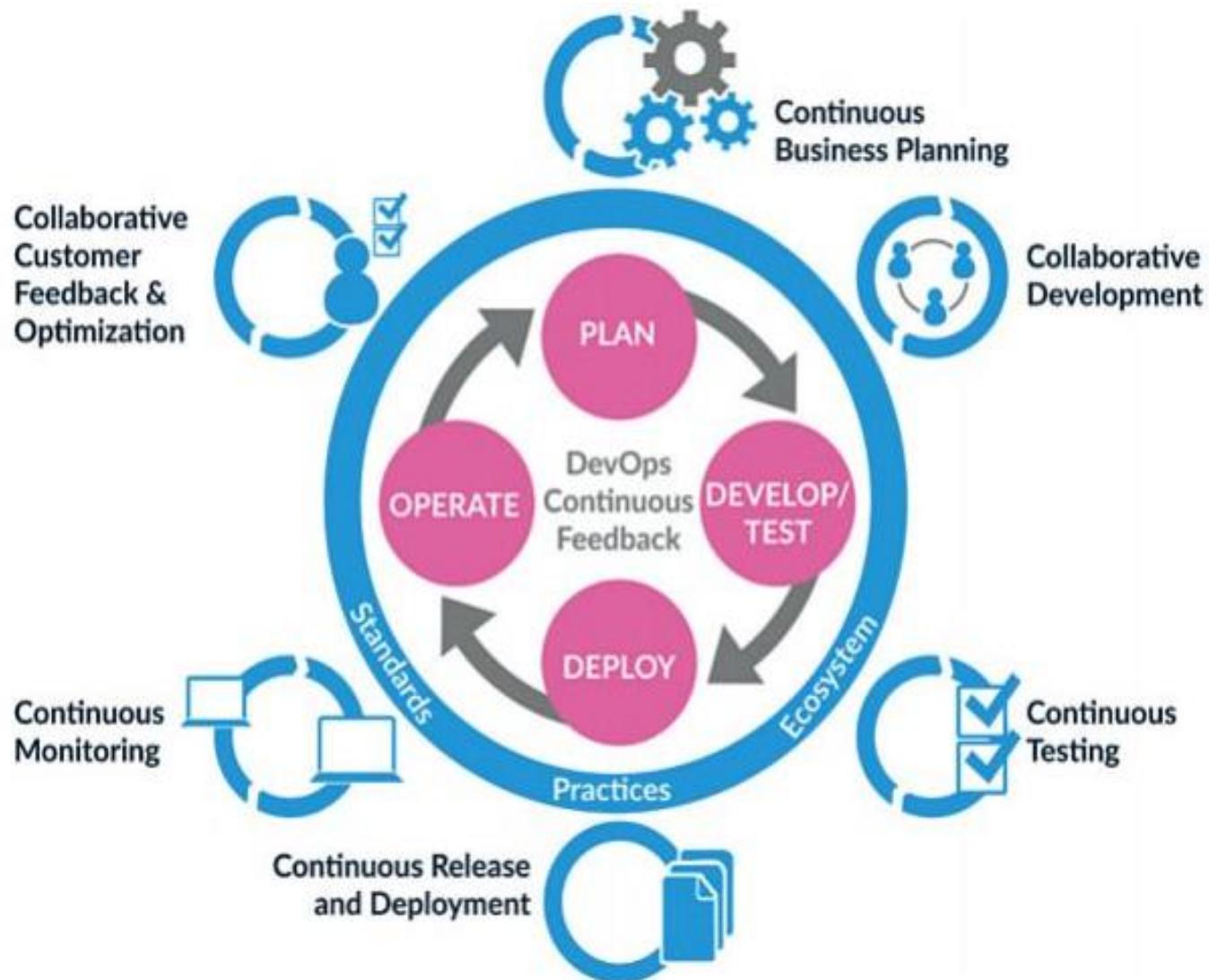


**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013







DevOps Tools

Docker

- Docker is a high-end DevOps tool that allows building, ship, and run distributed applications on multiple systems. It also helps to assemble the apps quickly from the components, and it is typically suitable for container management.

Features

- It configures the system more comfortable and faster.
- It increases productivity.
- It provides containers that are used to run the application in an isolated environment.
- It routes the incoming request for published ports on available nodes to an active container. This feature enables the connection even if there is no task running on the node.
- It allows saving secrets into the swarm itself.



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



DevOps Tools

Jenkins

Jenkins is a DevOps tool for monitoring the execution of repeated tasks. Jenkins is a software that allows continuous integration. Jenkins will be installed on a server where the central build will take place. It helps to integrate project changes more efficiently by finding the issues quickly

Features

- Jenkins increases the scale of automation.
- It can easily set up and configure via a web interface.
- It can distribute the tasks across multiple machines, thereby increasing concurrency.
- It supports continuous integration and continuous delivery.
- It offers 400 plugins to support the building and testing any project virtually.
- It requires little maintenance and has a built-in GUI tool for easy updates.



DevOps Tools

Kubernetes

Kubernetes is a powerful and feature-rich platform for managing containerized applications

Features

- Container orchestration: Kubernetes provides a platform for deploying, scaling, and managing containers, making it easier to manage and scale applications.
- Self-healing: Kubernetes has built-in mechanisms for automatically replacing failed containers, ensuring that applications remain available and running even in the face of failures.
- Automatic scaling: Kubernetes can automatically scale the number of containers running based on demand, providing a way to ensure that applications can handle increased traffic and load.
- Load balancing: Kubernetes provides built-in load balancing capabilities, making it easy to distribute incoming traffic across multiple containers.



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



DevOps Tools

Kubernetes contd..

Kubernetes is a powerful and feature-rich platform for managing containerized applications

Features

- Configuration management: Kubernetes provides a way to manage the configuration of containers, making it easier to deploy and manage applications at scale.
- Secret and configuration management: Kubernetes provides a secure way to manage secrets and configuration data, helping to ensure that sensitive information is kept secure.
- Resource management: Kubernetes provides a way to manage the resources required by containers, including CPU, memory, and storage. This helps to ensure that applications get the resources they need to run effectively.
- Rollouts and rollbacks: Kubernetes provides a way to easily perform rolling updates and rollbacks of applications, making it easier to manage the deployment of updates.
- Batch processing: Kubernetes provides a platform for running batch processing jobs, making it a great choice for organizations that need to process large amounts of data.
- Service discovery: Kubernetes provides a way to discover and access services running within the cluster, making it easier to build distributed applications.



What Is Source Control?

Source control, also known as version control, is a system for managing changes to a codebase or other collection of files over time. It provides a way to track the history of changes made to the code, making it easier to manage and collaborate on software development projects.

Source Control vs. Version Control

These two terms are used interchangeably. However, source control is specific to source code. Version control also covers large binary files and digital assets.



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



Source Control Features

Features

- Versioning
- Branching and Merging
- Collaboration
- Conflict resolution
- Continuous integration and delivery
- Auditing
- Security



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



What Is Source Control Management?

- Source control management (SCM) refers to tools that help you keep track of your code with a complete history of changes.

Source control tools include:

- Git -Git is a popular open-source version control system that allows multiple users to collaborate and manage changes to software projects efficiently. It enables teams to track changes and work together in real-time, making it a crucial tool for DevOps.
- Helix Core
- Subversion
- ClearCase
- Team Foundation Server
- Mercurial



Review of GIT source control.

- Speed: Git is designed to be fast, even when working with large projects or repositories. It uses a unique data structure called a "packfile" to efficiently store and retrieve version control data.
- Easy Collaboration: Git allows multiple developers to work on the same project simultaneously.
- Security: Git provides robust security features, including cryptographic hashing and digital signatures to secure version control data.
- Large Community: Git has a large and active community of developers who contribute to the project and provide support.
- Git is a widely used open-source distributed version control system that helps track changes made to software source code. It was created by Linus Torvalds in 2005 and has since become one of the most popular version control systems in use today.
- Git offers a number of benefits over other version control systems, including:
 1. Distributed Architecture.
 2. Branching and Merging.



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013





THANK YOU



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013

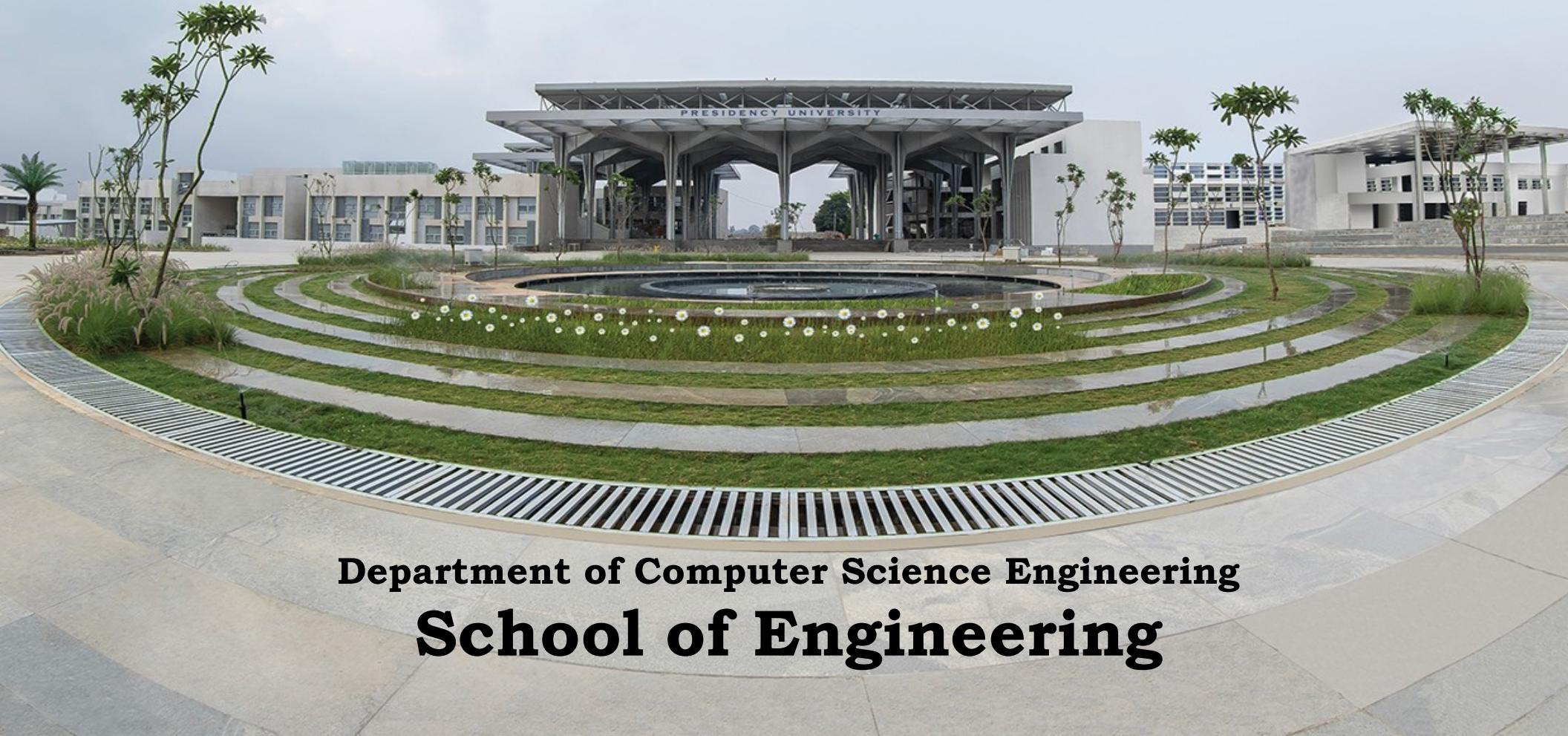


CSE-367 Data Visualization

OVER
40
YEARS
OF ACADEMIC
WISDOM



PRESIDENCY UNIVERSITY



**Department of Computer Science Engineering
School of Engineering**

Advanced CSS



```
body {  
    font: x-sma;  
    background:  
    color: black;  
    margin: 0;  
    padding: 0;
```

Cascading Style Sheet(CSS)

- Cascading Style Sheet(CSS) is used to set the style in web pages that contain HTML elements.
- It sets the background color, font-size, font-family, color, ... etc property of elements on a web page

There are three types of CSS which are given below:

- **Inline CSS** - by using the style attribute inside HTML elements
- **Internal or Embedded CSS** -by using a `<style>` element in the `<head>` section
- **External CSS** - by using a `<link>` element to link to an external CSS

Cascading Style Sheet(CSS)

Inline CSS :

Inline CSS contains the CSS property in the body section attached with element is known as inline CSS.

This kind of style is specified within an HTML tag using the style attribute.

`<h1 style="color: green; text-decoration: underline;">Hello world!</h1>`

`<p style="font-size: 25px; font-family: 'Trebuchet MS';">I Love CSS</p>`

Inline styles are generally the safest way to ensure rendering compatibility across various email clients, programs and devices, but can be time-consuming to write and a bit challenging to manage.

Cascading Style Sheet(CSS)

Inline CSS :

```
<!DOCTYPE html>
<html>
    <head>        <title>Inline CSS</title>        </head>

    <body>
        <p style = "color:#009900; font-size:50px; font-style:italic; text-align:center;">
            Web Technology
        </p>
    </body>
</html>
```

Web Technology



University Estd. in Karnataka State by Act No. 41 of 2013



Cascading Style Sheet(CSS)

Internal or Embedded CSS:

- This can be used when a single HTML document must be styled uniquely.
- It is defined in <head> section of the HTML page inside the <style> tag.

Cascading Style Sheet(CSS)

External CSS:

- The external style sheet is generally used when you want to make changes on multiple pages.
- It is ideal for this condition because it facilitates you to change the look of the entire web site by changing just one file.
- It uses the <link> tag on every pages and the <link> tag should be put inside the head section.

Example:

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

- The external style sheet may be written in any text editor but must be saved with a **.css extension**. This file should not contain HTML elements.

Cascading Style Sheet(CSS)

External CSS:

- Let's take an example of a style sheet file named "**mystyle.css**".

```
body {  
    background-color: lightblue;  
}  
  
h1 {  
    color: navy;  
    margin-left: 20px;  
}
```

Cascading Style Sheet(CSS)

External CSS:

```
<!DOCTYPE html>
<html>
<head>
    <link rel="stylesheet" type="text/css" href="mystyle.css">
</head>
<body>
    <h1>The External style sheet is applied on this heading.</h1>
    <p>This paragraph will not be affected.</p>
</body>
</html>
```



Cascading Style Sheet(CSS)

Comments

- CSS comments are generally written to explain your code.
- It is very helpful for the users who reads your code so that they can easily understand the code.
- Comments are **ignored by browsers**.
- Comments are single and multiple lines statement, written within `//` and `/*.....*/` respectively.

Cascading Style Sheet(CSS)

Comments

```
<!DOCTYPE html>
<html>
<head>
<style>
p {
    color: blue;
    /* This is a single-line comment */
    text-align: center;
}
/* This is
a multi-line
comment */
</style>
</head>
<body>
<p>Hello Web Technology</p>
</html>
```

Hello Web Technology

Selectors

A selector identifies which element or elements in the HTML document will be selected by the declarations in the rule. They are a pattern that is used by the browser to select the HTML elements that will receive the style.

Element Selectors

Class Selectors

Attribute Selectors

Pseudo-Element and Pseudo-Class Selectors

Contextual Selectors

Element Selectors

- **Element selectors** select an element or group of elements of the HTML document, and the properties are applied on it.
- **Group selector** - Group of elements are separated using commas is called group selector.
- **Universal element selector** - All elements of the document can be selected by using the * (asterisk) character.

Example :

```
p{ font-style:italic; font-weight:bold;}
```

```
h1,h2{font-weight:bold; color:red;}
```

```
{ color:blue;}
```

Example

```
<title>Student details </title>
<style>
*{ color:blue;}
h1{color: red;}
</style>
>

<h1 >Student Info</h1>
<p >Amith</p>
<p>Easy to learn.</p>
<hr/>
<p >Bhushan</p>
<p>Very much special.</p>
<hr/>
>
```

Student Info

[Amith](#)

[Easy to learn.](#)

[Bhushan](#)

[Very much special.](#)

Class Selectors

A **class selector** allows to simultaneously target different HTML elements. The HTML elements with the same class attribute value, can be styled by using a class selector.

Syntax: period(.)classname{ styles;}

Example:

```
first {  
font-style: italic;  
color: red;  
  
cen {text-align: center;}}
```

Example

```
<head>
<title>Student details </title>
<style>
first {
font-style: italic;
color: red;

</style>
</head>
<body>
<h1 class="first">Student Info</h1>
<div>
<p class="first">Amith</p>
<p>Easy to learn.</p>
</div>
<hr/>
<div>
<p class="first">Bhushan</p>
<p>Very much special.</p>
</div>
<hr/>
</body>
</html>
```

Student Info

Amith

Easy to learn.

Bhushan

Very much special.

Id Selectors

An **id selector** allows to assign style to a specific element by its id attribute.

Syntax: hash (#)id name

Eg:

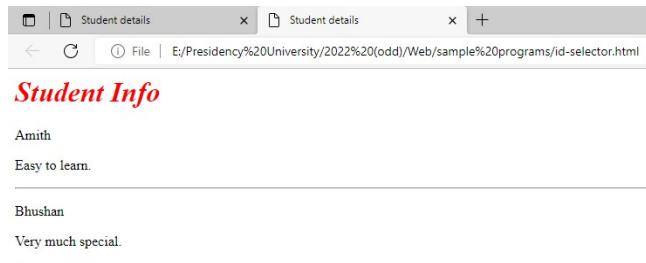
```
<head>
<title>Student details </title>
<style>
#first {
font-style: italic;
color: red;

</style>
</head>
<body>
<h1 id="first">Student Info</h1>
<div>
```

Example

```
<head>
<title>Student details </title>
<style>
#first {
    font-style: italic;
    color: red;
}

</style>
</head>
<body>
<h1 id="first">Student Info</h1>
<p> Amith</p>
<p>Easy to learn</p>
<hr/>
<p >Bhushan</p>
<p>Very much special.</p>
<hr/>
</body>
</html>
```



The screenshot shows a web browser window with two tabs both titled "Student details". The URL in the address bar is "E:/Presidency%20University/2022%20(odd)/Web/sample%20programs/id-selector.html". The content of the page is titled "Student Info" in red. It contains two entries: "Amith" with the note "Easy to learn." and "Bhushan" with the note "Very much special.". Both entries are displayed in black text.



Attribute Selectors

An **attribute selector** provides a way to select HTML elements either by the presence of an element attribute or by the value of an attribute.

Eg: [src], [src\$=".jpg"] , a[href*="gala"] etc.

[src] – selects all the elements which have ‘src’ as an attribute

[src\$=".jpg"] – selects all the elements with ‘src’ value ending with .jpg

a[href*="gala"] – selects <a> tag with ‘href’ value having text ‘gala’.

Suppose, we want special attention of user when a pop-up tooltip is available for a link or image. This can be done by using the following attribute selector: [title] { ... }

Example -

```
title {  
    cursor: help;  
    padding-bottom: 3px;  
    border-bottom: 2px dotted blue;  
}
```

```
<head>
<title>Student activities</title>
<style>
[title] {
cursor: help;
padding-bottom: 3px;
border-bottom: 2px dotted blue;
}
</style>
</head>
<body>
<div>

<a href = "s1.jpg" title= "link to photo"> click </a>
</div>
</body>
```



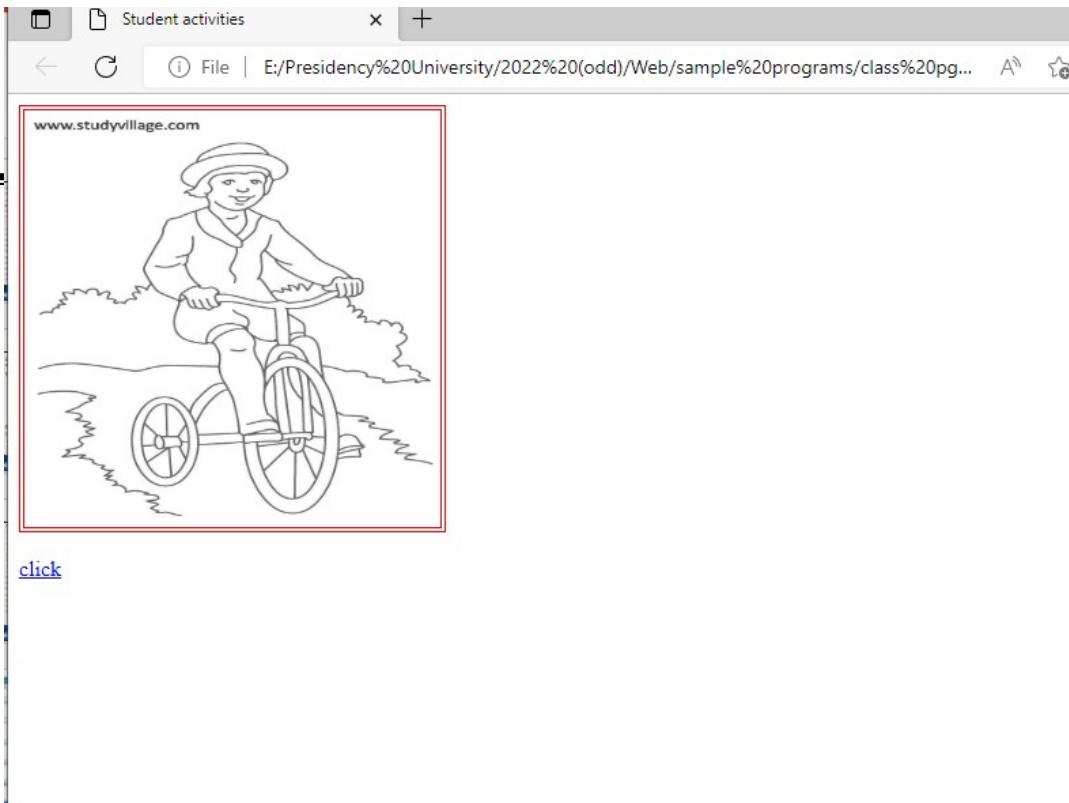
click

selector	Matches	Example
=]	A specific attribute.	[title] Matches any element with a title attribute
-=]	A specific attribute with a specific value.	a[title="posts from this country"] Matches any <a> element whose title attribute is exactly "posts from this country"
~=]	A specific attribute whose value matches at least one of the words in a space-delimited list of words.	[title~="Countries"] Matches any title attribute that contains the word "Countries"
^=]	A specific attribute whose value begins with a specified value.	a[href^="mailto"] Matches any <a> element whose href attribute begins with "mailto"
=]	A specific attribute whose value contains a substring.	img[src="flag"] Matches any element whose src attribute contains somewhere within it the text "flag"
\$=]	A specific attribute whose value ends with a specified value.	a[href\$=".pdf"] Matches any <a> element whose href attribute ends with the text ".pdf"

```
<title>Student activities</title>
<style>
src$=".jpg"] {
cursor: help;
padding: 3px;
border: 4px double red;

</style>
</head>
<body>
<div>

<br /><br />
<a href = "s1.jpg" title= "link to photo"> click </a>
</div>
</body>
```



Pseudo-classes

A pseudo-class is used to define a special state of an element that is recognizable.

For example, it can be used to:

- Style an element when a user moves the mouse over it
- Style visited and unvisited links differently
- Style an element when it gets focus
- Style the first letter of a paragraph etc.

	Type	Description
or	pseudo-class	Selects links that have not been visited
x	pseudo-class	Selects links that have been visited
ited	pseudo-class	Selects elements (such as text boxes or list boxes) that have the input focus.
s	pseudo-class	Selects elements that the mouse pointer is currently above.
r	pseudo-class	Selects an element that is being activated by the user. A typical example is a link that is being clicked.
ve	pseudo-class	Selects a form element that is currently checked. A typical example might be a radio button or a check box.
cked	pseudo-class	Selects an element that is the first child of its parent. A common use is to provide different styling to the first element in a list.
t-child	pseudo-class	Selects the first letter of an element. Useful for adding drop-caps to a paragraph.
t-letter	pseudo-element	Selects the first line of an element.
t-line	pseudo-element	

Pseudo class (examples)

```
html>
head>
style>
:link { color: red; }
:visited { color: green; }
:hover { color: hotpink; }
/style>
/head>
body>
h2>Styling a link depending on state</h2>
p><b><a href="https://www.google.com" target="_blank">This is a link</a></b></p>
/body>
/html>
```

Styling a link depending on state

This is a link

Styling a link depending on state

This is a link

Styling a link depending on state

This is a link

Pseudo-Elements

- A CSS pseudo-element is used to style specified parts of an element.
- For example, it can be used to:
 - Style the first letter, or line, of an element
 - Insert content before, or after, the content of an element

The ::first-line Pseudo-element

```
<html>
<head>
<style>
p::first-line {
  color: #ff0000;
  font-variant: small-caps;
}

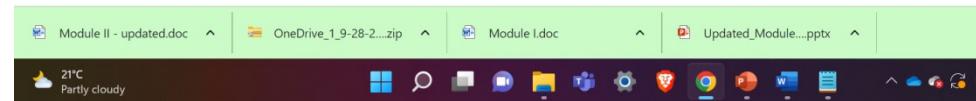
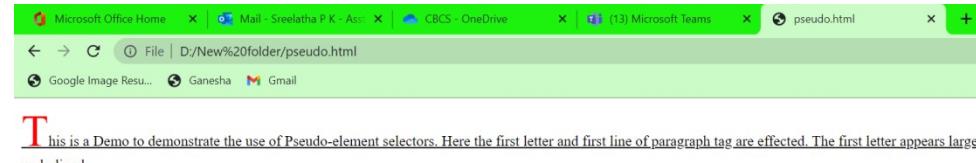
</style>
</head>
<body>
<p>You can use the ::first-line pseudo-element to add a special effect to the first line of a
ext. Some more text. And even more, and more.</p>

</body>
</html>
```

YOU CAN USE THE ::FIRST-LINE PSEUDO-ELEMENT TO ADD A SPECIAL EFFECT TO
the first line of a text. Some more text. And even more, and more.

Pseudo element (examples)

```
<html>
<head>
<style>
  p:first-letter { font-size: 300%; color: red;}
  p:first-line {text-decoration: underline;}
</style>
</head>
<body>
<p> This is a Demo to demonstrate the use of Pseudo-element selectors. Here the first letter and first line of paragraph tag are effected. The first letter appears larger and is red colored. the first line is underlined.... </p>
</body>
</html>
```



The ::first-letter Pseudo-element

```
<style>
  p::first-letter {
    color: #ff0000;
    font-size: xx-large;
```

You can use the ::first-letter pseudo-element to add a special effect to the first character of a text!

```
<style>
  p.intro::first-letter {
    color: #ff0000;
    font-size: 200%;
```

This is an introduction.

```
<p class="intro">This is an introduction.</p>
```

A **contextual selector** (in CSS3 also called **combinators**) allows to select elements based on their ancestors, descendants, or siblings. It selects elements based on their context or relation to other elements in the document tree.

Eg – Descendant selector matches the specified element that is contained within another element

div p – selects <p> tag that is contained within <div> tag.

Format	Example	Description
Descendant Selector - element element	div p	Selects all <p> elements inside <div> elements
Child Selector - element > element	div > p	Selects all <p> elements where the parent is a <div> element
Immediate Adjacent Sibling Selector element + element	div + p	Selects the first <p> element that are placed immediately after <div> elements
General Sibling Selector element ~ element	div ~ p	Selects every <p> element that are placed after <div> element

Advanced CSS: Layout

Normal Flow

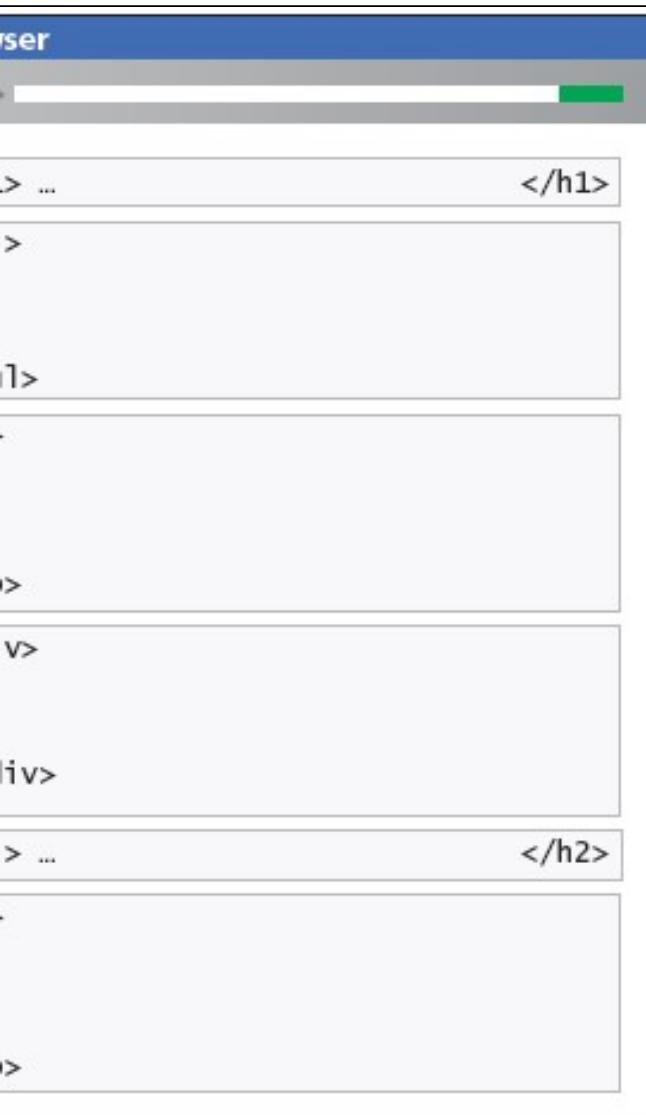
The browser will normally display block-level elements and inline elements from left to right and from top to bottom.

Block-level elements such as `<p>`, `<div>`, `<h2>`, ``, and `<table>` are elements that are contained on their own line, because block-level elements begin with a line break (new line).

Two block-level elements will not exist on the same line, without styling.

Inline elements such as ``, `<u>`, `<sub>`, `<sup>`, ``, `<i>` etc. are displayed within the same line and do not form their own blocks.

Block-level elements

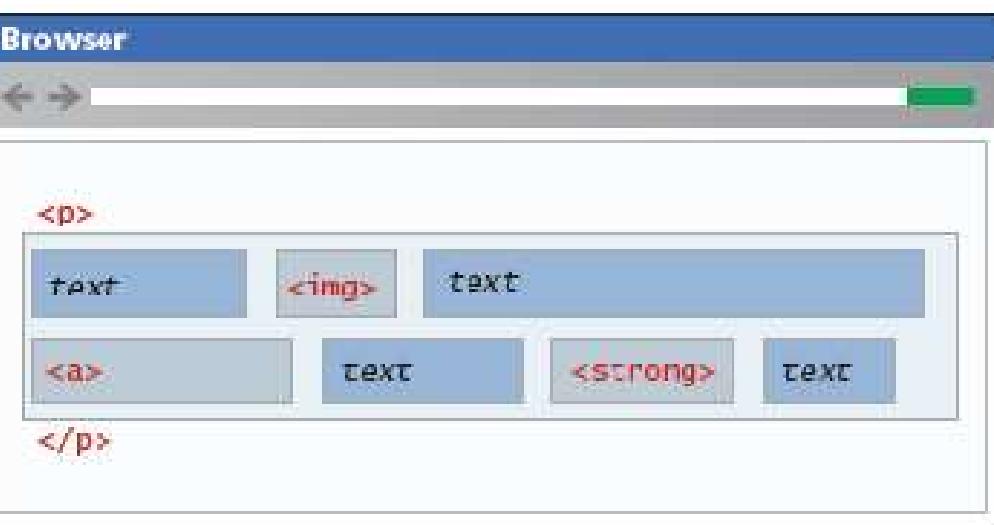


Properties of block-level elements -

- Each block exists on its own line.
- It is displayed in normal flow from the browser window's top to its bottom.
- By default each block level element fills up the entire width of its parent (browser window).
- CSS box model properties can be used to customize, for instance, the width of the box and the margin space between other block level elements.

Inline elements

Inline elements line up next to one another horizontally from left to right on the same line, when there is no enough space on the line, the content moves to a new line. Example - , <i>, , <u>,<a> etc.



Properties of inline elements are –

- Inline element is displayed in normal flow from its container's left to right.
- When a line is filled with content, the next line will receive the remaining content, and so on.
- If the browser window resizes, then inline content will be “re-flowed” based on the new width.

Types of inline elements

Replaced and Nonreplaced inline elements

Replaced inline elements are elements whose content and appearance is defined by some external resource, such as `` and the various form elements.

Nonreplaced inline elements are those elements whose content is defined within the document. Eg: `<a>`,``,`<i>`,``.

Note: A block-level or inline element is converted to another by using the CSS ‘display’ property.

```
span { display: block; }
```

```
i { display: inline; }
```

These two rules will make all `` elements behave like block-level elements and all `` elements like inline (that is, each list item will be displayed on the same line).

```
<!DOCTYPE html>
<html>
<head>
<style>
span{color:blue; display:block;}
</style>
</head>
<body>

<h1>The span element</h1>

<p>My mother has <span>blue</span> eyes and
my father has <span>dark green</span> eyes.</p>

</body>
</html>
```

The span element

My mother has
blue
eyes and my father has
dark green
eyes.

Positioning Elements

The position property of CSS is used to move an item from its regular position in the normal flow. An element can also be fixed to a position, so that it is always visible while the rest of the content scrolls.

The possible values for position property are-

Type	Description
relative	The element is moved relative to where it would be in the normal flow.
absolute	The element is removed from normal flow and positioned in relation to its nearest positioned ancestor.
fixed	The element is fixed in a specific position in the window even when the document is scrolled
static	The element is positioned according to the normal flow. This is the default.

The left, right, top, and bottom properties are used to indicate the distance the element will move.

Relative Positioning

In **relative positioning** an element is displaced out of its normal flow position and moved relative to where it would have been placed normally. The other contents around the relatively positioned element remain in its old position in the flow. the space the element would have occupied is preserved.

Eg –

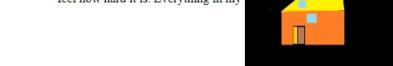
```
figure {  
position: relative;  
top: 150px;  
left: 200px;
```

The contents of block tag (figure) has to be placed at 150px,200px from its actual position.

```
<html>
<head>
<style>
figure {
position: relative;
top: 150px;
left: 200px;
```

```
</style>
</head>
<body>
```

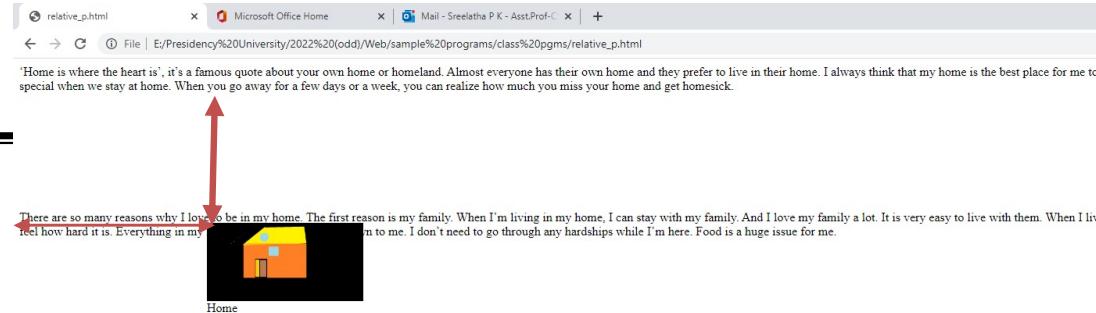
```
<p>'Home is where the heart is', it's a famous quote about your own home or homeland. Almost everyone has their own home and they prefer to live in their home. I always think that my home is the best place for me to live in this world. We all feel special when we stay at home. When you go away for a few days or a week, you can realize how much you miss your home and get homesick. </p>
```

```
<figure>

<figcaption>Home</figcaption>
</figure>
```

```
<p>There are so many reasons why I love to be in my home. The first reason is my family. When I'm living in my home, I can stay with my family. And I love my family a lot. It is very easy to live with them. When I live outside of my home, I can feel how hard it is. Everything in my home is very familiar and known to me. I don't need to go through any hardships while I'm here. Food is a huge issue for me.</p>
```

```
</body>
```

```
</html>PRESIDENCY  
UNIVERSITY  
University Estd. in Karnataka State by Act No. 41 of 2013
```



Absolute Positioning

In **absolute positioning** an element is completely removed from normal flow. Hence space is not left for the moved element, as it is no longer in the normal flow. Its position is moved in relation to its container block.

Eg –

```
figure {  
position: absolute;  
top: 60px;  
left: 200px;
```

With this positioning, the figure tag is placed at a distance of 60px from top and 200 px from left with respect to its container block. (Update this property in the above code, to view the difference).

Fixed Positioning

The element is positioned in relation to the viewport (i.e., to the browser window). Elements with **fixed positioning** do not move when the user scrolls up or down the page.

The fixed position is used to ensure that navigation elements or **advertisements are always visible.**

Eg –

```
figure {  
position: fixed;  
top: 0px;  
left: 0px;
```

With this positioning, the figure tag is placed at the top left most corner. (Update this property in the above code, to view the difference).

Z-index

Each positioned element has a stacking order defined by the z-index property (named for the z-axis). Items closest to the viewer (and thus on the top) have a larger **z-index** value, as shown in the example below.

```
<head>
  <style>
    figure {
      position: absolute;
      top: 60px;
      left: 200px;
      z-index:-1;
    }
  </style>
</head>
```

```
body{
  z-index:1;
```



Floating Elements

CSS float property is to displace an element out of its position in the normal flow.

When an item is floated, it is moved all the way to the far left or far right of its containing block and the rest of the content is “re-flowed” around the floated element.

Note: Absolutely positioned **elements** ignore the **float property!**

Float

```
<head>
<style>
img {
    float: right;
}

</style>
</head>
<body>
<h1>The float Property</h1>
<p>The image will float to the right in the text</p>
<p>
    Lorem ipsum.</p>
</body>
</html>
```

The float Property

The image will float to the right in the text

Lorem ipsum.



Cont'd

```
<style>  
img {  
    float: left;  
}</style>
```

```
<style>  
img {  
    float: none;  
}  
</style>
```

The float Property

The image will float to the right in the text



 Lorem ipsum.

The float Property

The image will float to the right in the text



 Lorem ipsum.



University Estd. in Karnataka State by Act No. 41 of 2013



Constructing Multicolumn Layouts

The space is divided into number of columns as mentioned by **column-count** property.

```
<html>
<head>
<style>
newspaper {
  column-count: 3;
}
</style>
</head>
<body>
  <h1>Create Multiple Columns</h1>
  <div class="newspaper">
    Cascading Style Sheets is a style
    sheet language used for describing
    the presentation of a document
    written in a markup language such
    as HTML. CSS is a cornerstone
    technology of the World Wide Web
    alongside HTML and JavaScript.
  </div>
</body>
</html>
```

Cascading Style Sheets is a style sheet language used for describing the presentation of a document written in a markup language such as HTML. CSS is a cornerstone technology of the World Wide Web alongside HTML and JavaScript.

Create Multiple Columns

Cascading Style Sheets is a style sheet language used for describing the presentation of a document written in a markup language such as HTML. CSS is a cornerstone technology of the World Wide Web alongside HTML and JavaScript.

Column-gap

```
<style>  
newspaper {  
    column-count: 3;  
    column-gap: 5px;  
  
</style>
```

Specify the Gap Between Columns

Cascading Style Sheets is a document written in a markup language such as HTML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript.

```
<style>  
.newspaper {  
    column-count: 3;  
    column-gap: 30px;  
}  
</style>
```

Specify the Gap Between Columns

Cascading Style Sheets is a style sheet language used for describing the presentation of a document written in a markup language such as HTML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript.

Column width

```
<style>  
newspaper {  
    column-width: 100px;
```

Specify The Column Width

Cascading Style Sheets is a style sheet language used for describing the presentation of a document written in a markup language such as HTML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript.

The following topics ppt will be sent in updated ppt----

Approaches to CSS Layout, Responsive Design, CSS Frameworks

• **XML:** Basics, demonstration of applications using XML

Approaches to CSS Layout

- The size of the screen used to view the page can vary
- Like some users will visit a website on a 21-inch wide screen, while few others on 120 inches screen.
- Users with the large monitor might expect a site to take advantage of the extra size; users with the small monitor will expect the site to scale to the smaller size and still be usable (clear).
- This problem can be dealt in two basic ways - **Fixed Layout and Liquid Layout.**

Fixed Layout

width is fixed by the designer.

Common width, that fits normal desktop monitor (1024 × 768), is considered.

Content may be positioned on the left or the right of the monitor.

Advantage of a fixed layout –

to produce

stable visual result

optimized for typical desktop monitors



The disadvantage of a fixed layout –

- For larger screens, there may be an **excessive amount of blank space** to the left and/or right of the content.
- When the browser window is less than the fixed width; the user will have to **horizontally scroll** to see all the content.
- If **smaller mobile** devices are used, more horizontal scrolling has to be done.

```
DOCTYPE html>
head>
<style>
div#left{
    width: 600px;
    float: left;
    font-size: 20px;
}
div#right{
    width: 300px;
    float: right;
    font-size: 20px;
}
</style>
<title>Fixed Layout</title>
<head>
<body>
<div style="text-align: center; color:green; font-size:x-large">
    Fixed Layout Demo
</div>

<div id="left">
<p>In around the 1990s, During the early age of web development, developers and designers used fixed-width designs to design their websites. Fixed Layout is a layout in which the width of main container is fixed ( in pixels). Popular Fixed width layouts are 1200px and 960px (used earlier).
</p>
<p>Properties of Fixed Layout Fixed width in pixels. Text doesn't scroll down when browser windows in minimized. Independent of screen size. Horizontal Scroll will come when screen size is less than width of main container
</p>
</div>
```

d="right">
his layout is defined with fixed pixels.
screen doesnot change, when the
n size / broswer size is reduced or increased.

ixed Layout is of same size on all screens.

y>

|>

A screenshot of a Microsoft Edge browser window titled "Fixed Layout". The address bar shows the file path: E:/Presidency%20University/2022%20(odd)/Web/sample%20programs/class%20pgms/fixed_layout.html. The main content area displays the text: "Fixed Layout Demo". Below it, there is a paragraph about the history of fixed-width layouts and their properties. The browser interface includes standard toolbar icons and a tab bar with other open tabs like Gmail, YouTube, Maps, and Microsoft Office Home.

In around the 1990s, During the early age of web development, developers and designers used fixed-width designs to design their websites. Fixed Layout is a layout in which the width of main container is fixed (in pixels). Popular Fixed width layouts are 1200px and 960px (used earlier).

Properties of Fixed Layout Fixed width in pixels. Text doesn't scroll down when browser windows in minimized. Independent of screen size.
Horizontal Scroll will come when screen size is less than width of main container

This layout is defined with pixels. The screen doesnot change, when the screen size / broswer size is reduced or increased.

Fixed Layout is of same size on all screens.

A screenshot of a Microsoft Edge browser window titled "Fixed Layout". The address bar shows the file path: E:/Presidency%20University/2022%20(odd)/... The main content area displays the text: "Fixed Layout Demo". Below it, there is a paragraph about the history of fixed-width layouts and their properties. The browser interface includes standard toolbar icons and a tab bar with other open tabs like Gmail, YouTube, Maps, and Microsoft Office Home.

In around the 1990s, During the early age of web development, developers and designers used fixed-width designs to design their websites. Fixed Layout is a layout in which the width of main container is fixed (in pixels). Popular Fixed width layouts are 1200px and 960px (used earlier).

Properties of Fixed Layout Fixed width in pixels. Text doesn't scroll down when browser windows in minimized. Independent of screen size.
Horizontal Scroll will come when screen size is less than width of main container

This layout is defined with fixed pixels. The screen doesnot change, when the screen size / broswer size is reduced or increased.

Fixed Layout is of same size on all screens.

Liquid Layout

widths are not specified using pixels, but percentage values

As widths are expressed as percentages, the webpage will adapt to any browser size.

Eg: width: 50%;

The advantage of a liquid layout –

Adapts to different browser sizes, so there is neither wasted white space nor any need for horizontal scrolling.



Fluid layout
on the browser
window.

However,
can get to
as browser

The disadvantage of a liquid layout –

more difficult to create because some elements, such as images, have fixed pixel sizes.

- The screen may grow or shrink dramatically.

```
CTYPE html>
d>
<style>
div#left{
    width: 66%;
    float: left;
    font-size: 20px;
}
div#right{
    width: 33%;
    float: right;
    font-size: 20px;
}
</style>
<title>Liquid Layout</title>
d>
y>
<div style="text-align: center; color:green; font-size:x-large">
    Liquid Layout Demo
</div>

<div id="left">
<p>In around the 1990s, During the early age of web development, developers and designers used fixed-width designs to design their websites. Which only looks good in one specified width. While most developers were using fixed-width design, some were also using a technique called "Liquid Layout".</p>
```

<p>The liquid layout means:
Instead of using a fixed width for your layouts
you could make a flexible layout using percentages
for your column width.

</p>
</div>

<div id="right">
<p>This layout which we define with percentages
instead of fixed pixels works in more situations
than fixed-width design. But the Liquid layout
also has a weakness, while it
will look good on a wide variety of screens but it
will not look good on very large screens or on very
small screens.
</p>

<p>On a very large screen, our website's content
will look stretched and on a very small screen, our
website's content will look squashed. And in both
the site doesn't look good.

</p>
</div>

Liquid Layout Demo

In around the 1990s, During the early age of web development, developers and designers used fixed-width designs to design their websites. Which only looks good in one specified width. While most developers were using fixed-width design, some were also using a technique called "Liquid Layout".

The liquid layout means: Instead of using a fixed width for your layouts you could make a flexible layout using percentages for your column width.

This layout which we define with percentages instead of fixed pixels works in more situations than fixed-width design. But the Liquid layout also has a weakness, while it will look good on a wide variety of screens but it will not look good on very large screens or on very small screens.

On a very large screen, our website's content will look stretched and on a very small screen, our website's content will look squashed. And in both situations, the site doesn't look good.

Liquid Layout Demo

In around the 1990s, During the early age of web development, developers and designers used fixed-width designs to design their websites. Which only looks good in one specified width. While most developers were using fixed-width design, some were also using a technique called "Liquid Layout".

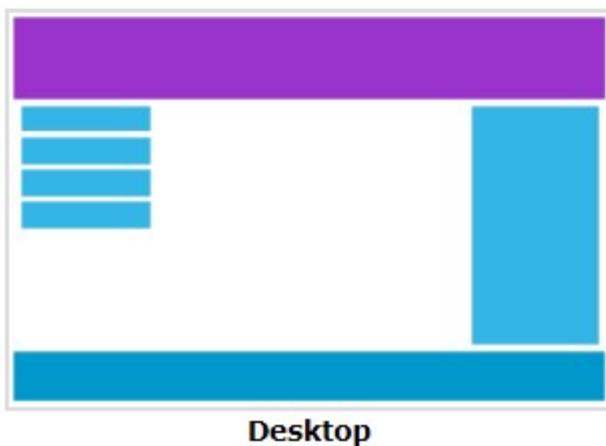
The liquid layout means: Instead of using a fixed width for your layouts you could make a flexible layout using percentages for your column width.

This layout which we define with percentages instead of fixed pixels works in more situations than fixed-width design. But the Liquid layout also has a weakness, while it will look good on a wide variety of screens but it will not look good on very large screens or on very small screens.

On a very large screen, our website's content will look stretched and on a very small screen, our website's content will look squashed. And in both situations, the site doesn't look good.

Responsive Design

- Makes your web page look good on all devices, using HTML and CSS.
- Web pages should not leave out information to fit smaller devices, but rather adapt its content to fit any device:



Desktop



Tablet



Phone

The four key components that make responsive design work are -

- Liquid layouts
- Scaling images to the viewport size
- Setting viewports via the <meta> tag
- Customizing the CSS for different viewports using media queries

Scaling images to the viewport size

- The viewport is the user's visible area of a web page.
- The viewport varies with the device, and will be smaller on a mobile phone than on a computer screen.

```
<meta name="viewport" content="width=device-width" />
```

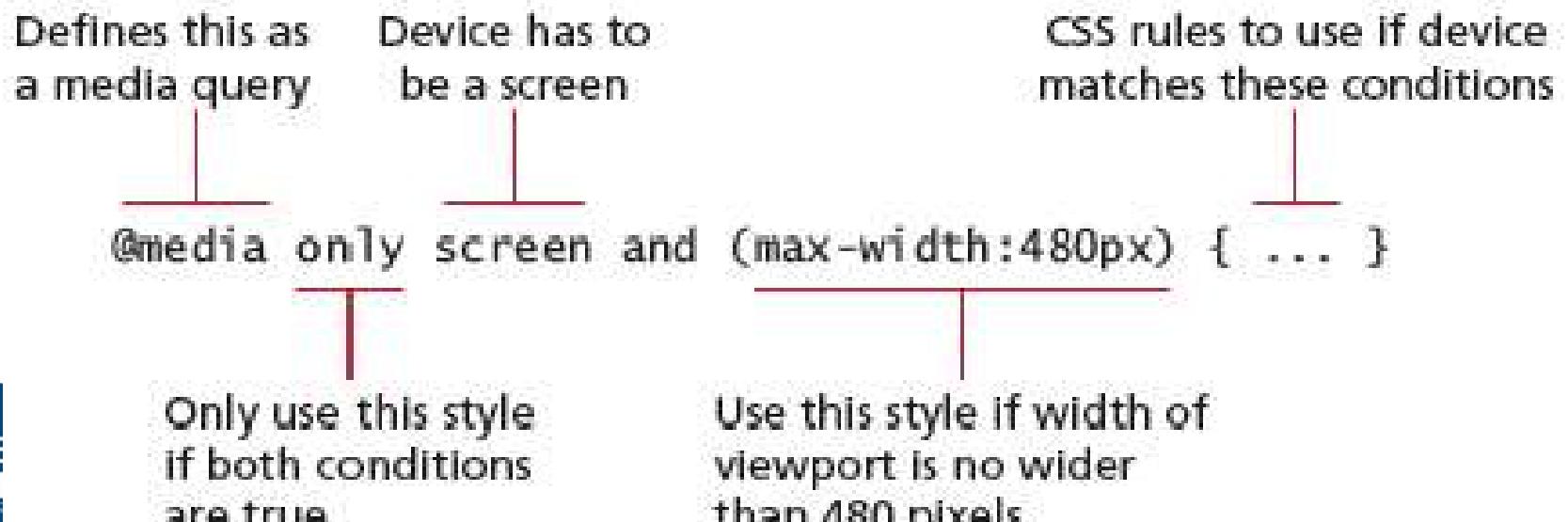
This makes the viewport as many pixels wide as the device screen width. This means that if the device has a screen that is 320 px wide, the viewport width will be 320 px; if the screen is 480 px, then the viewport width will be 480 px.

Media Queries

A media query is a way to apply style rules based on the medium that is displaying the file. It uses the @media rule to include a block of CSS properties only if a certain condition is true. You can use these queries to look at the capabilities of the device, and then define CSS rules.

Example-

`@media only screen and (max-width: 480px) {.....}` //This set of rule is applied when screen is used. Like set font-size, left and right margin etc.



CSS Frameworks

CSS framework is a pre-created set of CSS classes or other software tools that make it easier to use and work with CSS.

There are two main types of CSS framework: grid systems and CSS preprocessors.

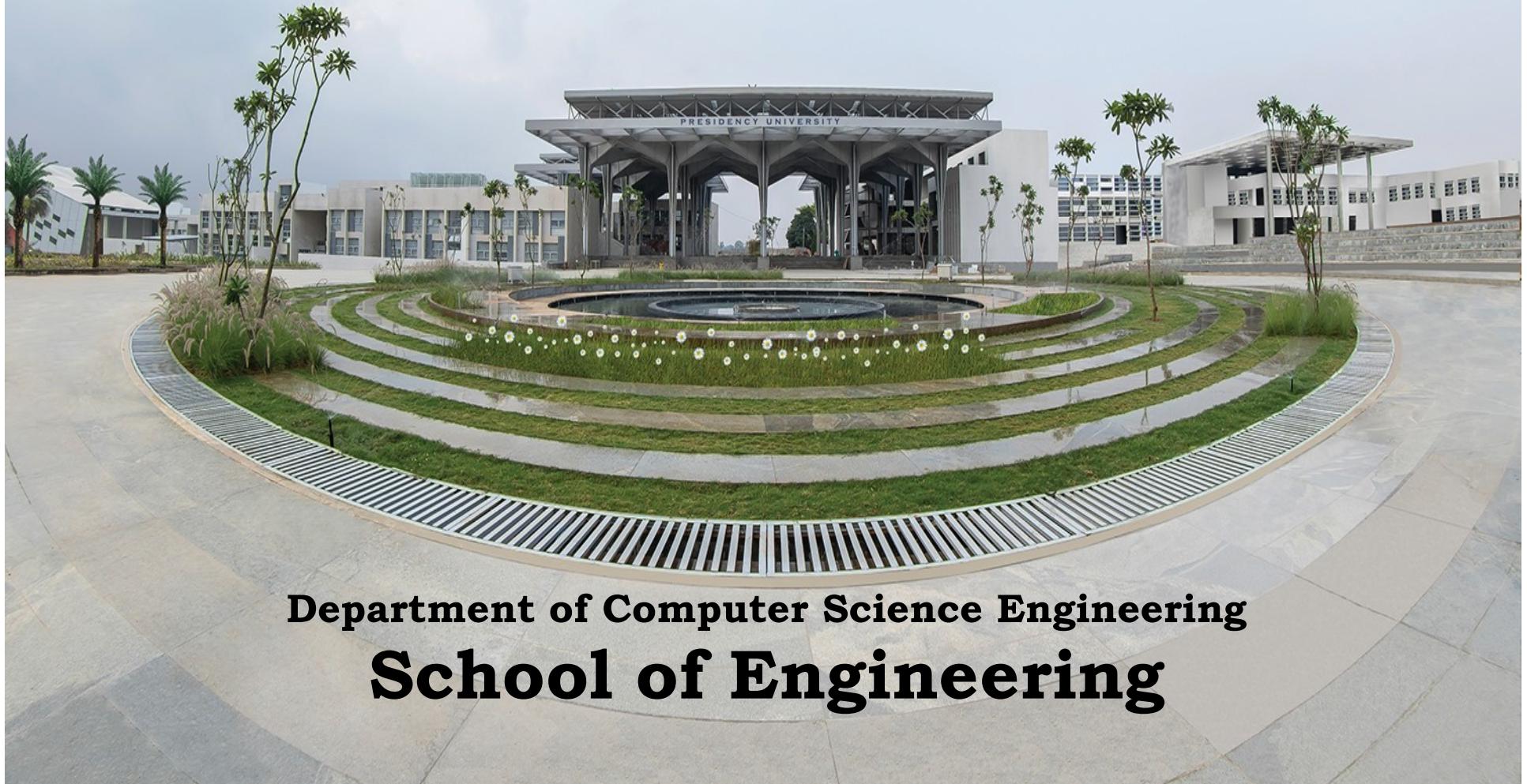
Grid Systems – Grids are used to achieve visual uniformity in a design. The screen is virtually divided into 5- or 7- or 12-column grid. Then, the text or graphics of the document is aligned and sized according to the grid.

CSS Preprocessor – this is a tool that takes code written in some type of preprocessed language and then converts that code into normal CSS. The preprocessed language uses programming identities such as variables, inheritance, calculations, and functions. Ex - LESS, SASS, and Stylus.

OVER
40
YEARS
OF ACADEMIC
WISDOM



PRESIDENCY UNIVERSITY



**Department of Computer Science Engineering
School of Engineering**

Introduction to HTML

HTML stands for **Hyper Text Markup Language**, which is the most widely used language on Web to develop web pages.

In late 1994, Tim Berners-Lee (who developed the initial version of HTML) started the World Wide Web Consortium (W3C), which had as one of its primary purposes to develop and distribute standards for Web technologies.

HTML 2.0 was the first standard HTML specification which was published in 1995.

HTML 4.01 was a major version of HTML and it was published in late 1999.

Currently we are having **HTML-5** version which was published in 2012.



HTML stands for **Hypertext Markup Language**, and it is the most widely used language to write Web Pages.

Hypertext refers to the way in which Web pages (HTML documents) are linked together. Thus, the link available on a webpage is called Hypertext.

As its name suggests, HTML is a **Markup Language** which means we use HTML to simply "mark-up" a text document with tags that tell a Web browser how to structure it to display.



Basic HTML Document

```
<html>
  <head>
    <title>This is document title</title>
  </head>
  <body>
    <h1>This is a heading</h1>
    <p>Document content goes here.....</p>
  </body>
</html>
```

output

This is a heading
Document content goes here.....



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



CSE2067 Web Technologies

HTML Tags

HTML is a markup language and makes use of various tags to format the content.

These tags are enclosed within angle braces **<Tag Name>**.

Tags have their corresponding closing tags.

For example,

<html> has its closing tag **</html>**

<body> tag has its closing tag **</body>** tag etc.



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



CSE2067 Web Technologies

Heading Tags

Any document starts with a heading.

We can use different sizes for your headings.

HTML also has six levels of headings, which use the elements **<h1>**, **<h2>**, **<h3>**, **<h4>**, **<h5>**, and **<h6>**.

While displaying any heading, browser adds one line before and one line after that heading.



```
<html>  
  
<head>  
    <title>Heading Example</title>  
</head>  
  
<body>  
    <h1>This is heading 1</h1>  
    <h2>This is heading 2</h2>  
    <h3>This is heading 3</h3>  
    <h4>This is heading 4</h4>  
    <h5>This is heading 5</h5>  
    <h6>This is heading 6</h6>  
</body>  
  
</html>
```

output

This is heading 1

This is heading 2

This is heading 3

This is heading 4

This is heading 5

This is heading 6



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



CSE2067 Web Technologies

Paragraph Tag

The **< p >** tag offers a way to structure your text into different paragraphs.

Each paragraph of text should go in between an opening **< p >** and a closing **< /p >** tag

```
<html>
```

```
  <head>
    <title>Paragraph Example</title>
  </head>
```

```
  <body>
    <p>Here is a first paragraph of text.</p>
    <p>Here is a second paragraph of text.</p>
    <p>Here is a third paragraph of text.</p>
  </body>
```

```
</html>
```

Output

Here is a first paragraph of text.

Here is a second paragraph of text.

Here is a third paragraph of text.



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



CSE2067 Web Technologies

Line Break Tag

Whenever you use the **
** element, anything following it starts from the next line.

The **
** tag has a space between the characters **br** and the forward slash.

If you omit this space, older browsers will have trouble rendering the line break, while if you miss the forward slash character and just use **
** it is not valid in XHTML.



```
<html>
```

```
  <head>
    <title>Line Break Example</title>
  </head>
```

```
  <body>
    <p>Hello<br />
      You delivered your assignment on time.<br />
      Thanks<br />
      Ramesh</p>
  </body>
```

```
</html>
```

Output

Hello
You delivered your assignment on
time.
Thanks
Ramesh



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



CSE2067 Web Technologies

Centering Content

You can use **<center>** tag to put any content in the center of the page or any table cell.

```
<html>  
  <head>  
    <title>Centring Content Example</title>  
  </head>  
  
  <body>  
    <p>This text is not in the center.</p>  
  
    <center>  
      <p>This text is in the center.</p>  
    </center>  
  </body>  
  
</html>
```

Output

This text is not in the center.

This text is in the center.

Horizontal Lines

Horizontal lines are used to visually break-up sections of a document.

The **<hr>** tag creates a line from the current position in the document to the right margin and breaks the line accordingly.

```
<html>
  <head>
    <title>Horizontal Line Example</title>
  </head>
  <body>
    <p>This is paragraph one and should be on top</p>
    <hr />
    <p>This is paragraph two and should be at bottom</p>
  </body>
</html>
```

This is paragraph one and should be on top

This is paragraph two and should be at bottom



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



CSE2067 Web Technologies

HTML comments are placed in between `<!-- ... -->` tags. So, any content placed with-in `<!-- ... -->` tags will be treated as comment and will be completely ignored by the browser.

```
<html>
  <head> <!-- Document Header Starts -->
    <title>This is document title</title>
  </head> <!-- Document Header Ends -->

  <body>
    <p>Document content goes here.....</p>
  </body>

</html>
```



Preserve Formatting

Sometimes, we want your text to follow the exact format of how it is written in the HTML document. In these cases, you can use the preformatted tag **<pre>**.

Any text between the opening **<pre>** tag and the closing **</pre>** tag will preserve the formatting of the source document.

```
<html>
  <head>
    <title>Preserve Formatting Example</title>
  </head>

  <body>
    <pre>
      function testFunction( strText ){
        alert (strText)
      }
    </pre>
  </body>

</html>
```

Output

```
function testFunction( strText ){
  alert (strText)
}
```

align attribute: left, center and right.

```
<html>  
  
<head>  
    <title>Align Attribute Example</title>  
</head>  
  
<body>  
    <p align = "left">This is left aligned</p>  
    <p align = "center">This is center aligned</p>  
    <p align = "right">This is right aligned</p>  
</body>  
  
</html>
```

Output

This is left aligned

This is center aligned

This is right aligned

Valign-top, middle, bottom: Vertically aligns tags within an HTML element.



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



CSE2067 Web Technologies

The title Attribute

The **title** attribute gives a suggested title for the element.

The behavior of this attribute will depend upon the element that carries it, although it is often displayed as a tooltip when cursor comes over the element or while the element is loading.

```
<html>
  <head>
    <title>The title Attribute Example</title>
  </head>

  <body>
    <h3 title = "Hello HTML!">Titled Heading Tag Example</h3>
  </body>

</html>
```



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



CSE2067 Web Technologies

The dir Attribute

The **dir** attribute allows you to indicate to the browser about the direction in which the text should flow.

The dir attribute can take one of two values:ltr and rtl

```
<html dir = "rtl">

    <head>
        <title>Display Directions</title>
    </head>

    <body>
        This is an example for right-to-left directed text.
    </body>

</html>
```



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



CSE2067 Web Technologies

Background color

bgcolor-> numeric or hexadecimal or RGB values: Places a background color behind an element

Bold Text

Anything that appears within **...** element, is displayed in bold

Italic Text

Anything that appears within *<i>...</i>* element is displayed in italicized

Underlined Text

Anything that appears within <u>...</u> element, is displayed with underline



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



CSE2067 Web Technologies

Strike Text

Anything that appears within **<strike>...</strike>** element is displayed with strikethrough

Monospaced Font

The content of a **<tt>...</tt>** element is written in monospaced font. Most of the fonts are known as variable-width fonts because different letters are of different widths (for example, the letter 'm' is wider than the letter 'i'). In a monospaced font, however, each letter has the same width.

Superscript Text

The content of a **^{...}** element is written in superscript; the font size used is the same size as the characters surrounding it but is displayed half a character's height above the other characters.

Subscript Text

The content of a **_{...}** element is written in subscript; the font size used is the same as the characters surrounding it, but is displayed half a character's height beneath the other characters.

Inserted Text

Anything that appears within **<ins>ghg</ins>** element is displayed as inserted text.

Deleted Text

Anything that appears within **. ** element, is displayed as deleted text.

Larger Text

The content of the **<big>...</big>** element is displayed one font size larger than the rest of the text surrounding it

Smaller Text

The content of the **<small>...</small>** element is displayed one font size smaller than the rest of the text surrounding it



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



CSE2067 Web Technologies

```
<!DOCTYPE html>
<html>
<!-- body tag starts here -->

<body bgcolor="green">
    <h2>GeeksforGeeks</h2>

    <p> It is a Computer Science portal For Geeks </p>

</body>
<!-- body tag ends here -->

</html>
```



Grouping Content

The **<div>** and **** elements allow you to group together several elements to create sections or subsections of a page.

For example, you might want to put all of the footnotes on a page within a **<div>** element to indicate that all of the elements within that **<div>** element relate to the footnotes. You might then attach a style to this **<div>** element so that they appear using a special set of style rules.

The **** element, on the other hand, can be used to group inline elements only. So, if you have a part of a sentence or paragraph which you want to group together, you could use the ****



HTML tag is used as a generic container of inline elements. It is used for styling purpose to the grouped inline elements (using class and id attribute or inline style).
The tag does not have any default meaning or rendering.
The tag can be useful for the following task:

- To change the color, font, background of a part of text using CSS
- To apply the scripts to the particular part of the text.
- To change the language of a part of the text.



```
<html>

<head>
    <title>Div Tag Example</title>
</head>

<body>
    <div id = "menu" align = "middle" >
        <a href = "/index.htm">HOME</a> |
        <a href = "/about/contact_us.htm">CONTACT</a>
        <a href = "/about/index.htm">ABOUT</a>
    </div>

    <div id = "content" align = "left" bgcolor = "white">
        <h5>Content Articles</h5>
        <p>Actual content goes here.....</p>
    </div>
</body>

</html>
```

```
<html>

<head>
    <title>Span Tag Example</title>
</head>

<body>
    <p>This is the example of <span style =
"color:green">span tag</span>
        and the <span style = "color:red">div
tag</span> alongwith CSS</p>
</body>

</html>
```



PRESIDENCY
UNIVERSITY



Private University Estd. in Karnataka State by Act No. 41 of 2013

-
- **html>**
 - **<head>**
 - **<title>Span Tag</title>**
 - **</head>**
 - **<body>**
 - **<h2>Example of span tag</h2>**
 - **<p>I have choosen only**
 - **red,**
 - **blue, and**
 - **green colors for my painting.**
 - **</p>**
 - **</body>**
 - **</html>**



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



CSE2067 Web Technologies

Emphasized Text

Anything that appears within `...` element is displayed as emphasized text.

Marked Text

Anything that appears with-in `<mark>...</mark>` element, is displayed as marked with yellow ink.

Strong Text

Anything that appears within `...` element is displayed as important text.

Text Direction

The `<bdo>...</bdo>` element stands for Bi-Directional Override and it is used to override the current text direction.

```
<bdo dir="rtl">
```

This text will go right-to-left.

```
</bdo>
```

Short Quotations

The `<q>...</q>` element is used when you want to add a double quote within a sentence.

Address Text

The `<address>...</address>` element is used to contain any address.



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



CSE2067 Web Technologies

List

HTML has several list elements. Most list elements are composed of one or more (List Item) elements.

There are three types of List

1. Unordered List
2. Ordered List
3. Definition List

1) UL : Unordered List. Items in this list start with a list mark such as a bullet. Browsers will usually change the list mark in nested lists.

 List item ...

 List item ...



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



CSE2067 Web Technologies

Choice of three bullet types: **disc(default), circle, square**.

These are controlled by the “TYPE” attribute for the element.

```
<UL TYPE="square">  
<LI> List item ...</LI>  
<LI> List item ...</LI>  
<LI> List item ...</LI>  
</UL>
```



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



CSE2067 Web Technologies

2) OL: Ordered List. Items in this list are numbered automatically by the browser.

 List item ...
 List item ...
 List item ...

Choice of setting the TYPE Attribute to one of five numbering styles.

8446075868



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



CSE2067 Web Technologies

Numbering Styles in Ordered List [By default – Arabic numbers]

TYPE	Numbering Styles	
1	Arabic numbers	1,2,3,
a	Lower alpha	a, b, c,
A	Upper alpha	A, B, C,
i	Lower roman	i, ii, iii,
I	Upper roman	I, II, III,



specify a starting number for an ordered list.

<OL TYPE =“i”>

 List item1 ...
 List item2 ...

<P> text</P>

<OL TYPE=“i” START=“3”>
 List item3 ...

Output:

i. List item1 ...
ii. List item2 ...

text

iii. List item3...



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



CSE2067 Web Technologies

DL: Definition List. This kind of list is different from the others. Each item in a DL consists of one or more **Definition Terms (DT elements)**, followed by one or more **Definition Description (DD elements)**.

```
<DL>
<DT> HTML </DT>
<DD> Hyper Text Markup Language </DD>
<DT> DOG </DT>
<DD> A human's best friend!</DD>
</DL>
```

Output:
HTML

DOG

Hyper Text Markup Language
A human's best friend!



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013

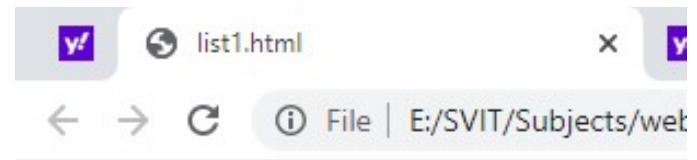


CSE2067 Web Technologies

can nest lists by inserting a UL, OL, etc., inside a list item (LI).

Example

```
<UL TYPE = "square">
<LI> List item ...</LI>
<LI> List item ...
<OL TYPE="i" START="3">
<LI> List item ...</LI>
</OL>
</LI>
<LI> List item ...</LI>
</UL>
```



- List item ...
- List item ...
- 3. List item ...
- 4. List item ...
- 5. List item ...
- 6. List item ...
- 7. List item ...
- List item ...



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



CSE2067 Web Technologies

```
<H1 ALIGN="CENTER">SAFETY TIPS FOR </H1>
<OL TYPE="a" START="2">
<LI>Be able to swim </LI>
<LI>Wear a life jacket at all times </LI>
<LI>Don't stand up or move around. If canoe tips,
    <UL>
        <LI>Hang on to the canoe </LI>
        <LI>Use the canoe for support and </LI>
        <LI>Swim to shore
    </UL> </LI>
<LI>Don't overexert yourself </LI>
<LI>Use a bow light at night </LI>
</OL>
```



TABLE Tag

The <TABLE></TABLE> element has four sub-elements:

Table Row <TR></TR>.

Table Header <TH></TH>.

Table Data <TD></TD>.

Caption <CAPTION></CAPTION>.

The table row elements usually contain table header elements or table data elements.



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



CSE2067 Web Technologies

```

<table border="1">

<tr>
<th> Column 1 header </th>
<th> Column 2 header </th>
</tr>

<tr>
<td> Row1, Col1 </td>
<td> Row1, Col2 </td>
</tr>

<tr>
<td> Row2, Col1 </td>
<td> Row2, Col2 </td>
</tr>
</table>

```

Column 1 Header	Column 2 Header
Row1, Col1	Row1, Col2
Row2, Col1	Row2, Col2



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



CSE2067 Web Technologies

TABLE ATTRIBUTES

BGColor: Some browsers support background colors in a table.

Width: you can specify the table width as an absolute number of pixels or a percentage of the document width. You can set the width for the table cells as well.

Border: You can choose a numerical value for the border width, which specifies the border in pixels.

CellSpacing: Cell Spacing represents the space between cells and is specified in pixels.



TABLE ATTRIBUTES

CellPadding: Cell Padding is the space between the cell border and the cell contents and is specified in pixels.

Align: tables can have left, right, or center alignment.

Background: Background Image, will be titled in IE3.0 and above.

BorderColor, BorderColorDark:



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



CSE2067 Web Technologies

A table caption allows you to specify a line of text that will appear centered above or below the table.

<TABLE BORDER=1 CELLPADDING=2>

<CAPTION ALIGN="BOTTOM"> Label For My Table </CAPTION>

The Caption element has one attribute ALIGN that can be either TOP (Above the table) or BOTTOM (below the table).



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



CSE2067 Web Technologies

```
<html>

    <head>
        <title>HTML Table Cellpadding</title>
    </head>

    <body>
        <table border = "1" cellpadding = "5" cellspacing = "5">
            <tr>
                <th>Name</th>
                <th>Salary</th>
            </tr>
            <tr>
                <td>Ramesh Raman</td>
                <td>5000</td>
            </tr>
            <tr>
                <td>Shabbir Hussein</td>
                <td>7000</td>
            </tr>
        </table>
    </body>

</html>
```



Colspan and Rowspan Attributes

We will use **colspan** attribute if you want to merge two or more columns into a single column. Similar way use **rowspan** if we want to merge two or more rows.

```
<html>
  <head>
    <title>HTML Table Colspan/Rowspan</title>
  </head>

  <body>
    <table border = "1">
      <tr>
        <th>Column 1</th>
        <th>Column 2</th>
        <th>Column 3</th>
      </tr>
      <tr>
        <td rowspan = "2">Row 1 Cell 1</td>
        <td>Row 1 Cell 2</td>
        <td>Row 1 Cell 3</td>
      </tr>
```

```
<tr>
  <td>Row 2 Cell 2</td>
  <td>Row 2 Cell 3</td>
</tr>
<tr>
  <td colspan = "3">Row 3 Cell 1</td>
</tr>
</table>
</body>
```

Column 1	Column 2	Column 3
Row 1 Cell 1	Row 1 Cell 2	Row 1 Cell 3
	Row 2 Cell 2	Row 2 Cell 3
Row 3 Cell 1		



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



Table Height and Width

You can set a table width and height using width and height attributes.

```
<html>
  <head>
    <title>HTML Table Width/Height</title>
  </head>
  <body>
    <table border = "1" width = "400" height
= "150">
      <tr>
        <td>Row 1, Column 1</td>
        <td>Row 1, Column 2</td>
      </tr>
    </table>
  </body>
</html>
```



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



CSE2067 Web Technologies

Table Header, Body, and Footer

Tables can be divided into three portions – a header, a body, and a foot.

<thead> – to create a separate table header.

<tbody> – to indicate the main body of the table.

<tfoot> – to create a separate table footer.

CSS can be applied on these sections separately.



Reference tag/anchor tag

The tags used to produce links are the **<A>** and ****.

The **<A>** tells where the link should start and
the **** indicates where the link ends.

Everything between these two will work as a link.

The example below shows how to make the word **Here** work as a link to yahoo.

Click **here**



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



CSE2067 Web Technologies

Internal Links : Links can also be created inside large documents to simplify navigation.

Select some text at a place in the document that you would like to create a link to, then add an anchor to link to like this:

``

The Name attribute of an anchor element specifies a location in the document that we link to shortly. All NAME attributes in a document must be unique.

Next select the text that you would like to create as a link to the location created above.

`Go To Book Mark`



Insert Image – img Tag

You can insert any image in your web page by using tag.

```
<html>
  <head>
    <title>Using Image in Webpage</title>
  </head>

  <body>
    <p>Simple Image Insert</p>
    <img src = "/html/images/test.png" />
  </body>
</html>
```

Set Image Width/Height

You can set image width and height based on your requirement using **width and height attributes**.



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



CSE2067 Web Technologies

Set Image Border

By default, image will have a border around it, you can specify border thickness in terms of pixels using border attribute.

Set Image Alignment

By default, image will align at the left side of the page, but you can use **align** attribute to set it in the center or right.

```
<html>
```

```
<head>
```

```
 <title>Set Image Alignment</title>
```

```
</head>
```

```
<body>
```

```
 <p>Setting image Alignment</p>
```

```
 <img src =
"/html/images/test.png" alt = "Test
Image" border = "3" align = "right"/>
</body>
```

```
</html>
```



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



CSE2067 Web Technologies

Image as a Link

```
<a href="basic.html">   
</a>
```



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



CSE2067 Web Technologies

HTML Email Tag

HTML **< a >** tag provides you option to specify an email address to send an email. While using **< a >** tag as an email tag, you will use **mailto: email address** along with **href** attribute.

```
<a href = "mailto: abc@example.com">Send Email</a>
```



HTML forms

HTML Forms are required, when you want to collect some data from the site visitor / customer.

For example, we would like to collect information such as name, email address, credit card, etc.

There are various form elements available like text fields, textarea fields, drop-down menus, radio buttons, checkboxes, etc.



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



The HTML <form> tag is used to create an HTML form and it has following syntax

<form>

form elements like input, textarea etc.

</form>



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



HTML Form Controls

There are different types of form controls that you can use to collect data using HTML form

- Text Input Controls
- Checkboxes Controls
- Radio Box Controls
- Select Box Controls
- Textarea Controls
- Submit and Reset Button etc.

All the above listed controls except Reset and Submit also require **name** attribute, which becomes the name of the control within the form data.



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



Input Element

One commonly used attribute of <input> is **type**

The type of control is mentioned in this attribute.. The text, password, checkboxes, and radio controls come under <input> tag.

Another attribute of <input> is **name** – it is used to give a name for the specific control.



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



CSE2067 Web Technologies

Single-line text input controls – This control is used for items that require only one line of user input, such as search boxes or names. They are created using HTML <input> tag.

```
<body>
  <form >
    First name: <input type = "text" name = "first_name" />
    <br>
    Last name: <input type = "text" name = "last_name" />
  </form>
</body>
```



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



Multiple-Line Text Input: used when the user is required to give details that may be longer than a single sentence. Multi-line input controls are created using HTML **<textarea>** tag.

```
<body>
  <form>
    Description : <br />
    <textarea rows = "5" cols = "50" >
      Enter description here...
    </textarea>
  </form>
</body>
```



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



Checkbox Control:-Checkboxes are used when more than one option is required to be selected.

<form>

```
<input type="checkbox" value="bike" > I have a bike<br>
<input type="checkbox" value="car" > I have a car
</form>
```



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



Radio Button Control:-Radio buttons are used when out of many options, just one option is required. The name is common for the radio buttons of a group, so that only one can be selected at a time.

```
<form>
  <input type="radio" name="gender" value="male"
checked="checked"> Male<br>
  <input type="radio" name="gender" value="female">
Female<br>
  <input type="radio" name="gender" value="other"> Other
</form>
```



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



```
<body>
  <p>
    Age Category
  </p>
  <form action = "">
    <p>
      <label><input type = "radio" name = "age"
                  value = "under20" checked = "checked" />
          0-19 </label>
      <label><input type = "radio" name = "age"
                  value = "20-35" /> 20-35 </label>
      <label><input type = "radio" name = "age"
                  value = "36-50" /> 36-50 </label>
      <label><input type = "radio" name = "age"
                  value = "over50" /> Over 50 </label>
    </p>
  </form>
</body>
```

Age Category

0-19 20-35 36-50 Over 50



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



CSE2067 Web Technologies

Select Box Control:- A select box, also called drop down box which provides option to list down various options in the form of drop down list, from where a user can select one or more options.

```
<select name="cars">
    <option value="volvo">Volvo</option>
    <option value="benz">Benz</option>
    <option value="fiat">Fiat</option>
    <option value="audi">Audi</option>
</select>
```



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



To define a pre-selected option, add the selected attribute to the option:

```
<option value="fiat"  
selected="selected">Fiat</option>
```

Use the size attribute to specify the number of visible values:

```
<select name="cars" size="3">
```

Use the multiple attribute to allow the user to select more than one value:

```
<select name="cars" size="4" multiple="multiple">
```

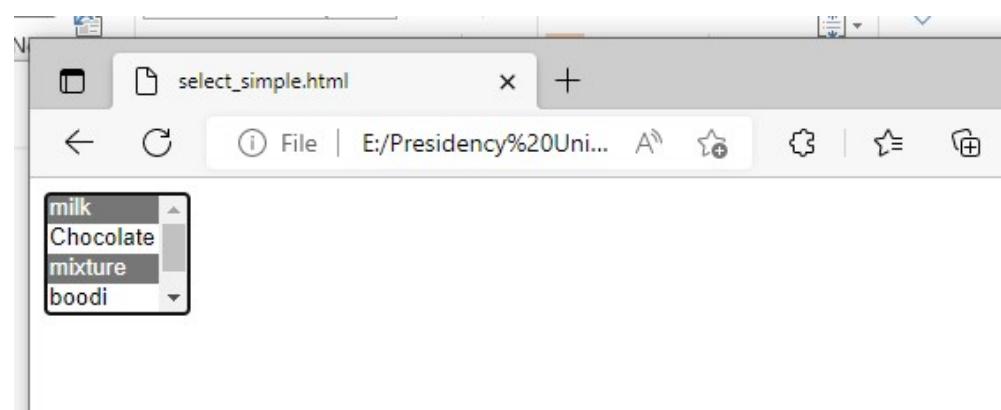


**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



```
<form>
<p>
<select name="list" multiple="multiple">
<option selected="selected">milk</option>
<option>Chocolate</option>
<option>mixture</option>
<option> boodi</option>
<option>jelebi</option>
</select>
</p>
```



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



CSE2067 Web Technologies

Button Controls

There are various ways in HTML to create clickable buttons. We can create a clickable button using `<input>` tag by setting its type attribute to button.

```
<input type = "button" name = "ok" value = "OK" />
```

Other options

```
<button type="button" onclick="alert('Hello World!')">Click  
Me!</button>
```

```
<input type = "image" name = "imagebutton" src = "logo.png" />
```



```
<input type="password" name="psw"> // defines password filed
```

```
<input type="submit" value="Submit"> // submit data form to an  
action handler
```

```
<input type="reset"> // reset all form values to default values
```

```
<input type="color"> // used for input a color.
```



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



`<input type="date">` // used for input a date.

`<input type="email">` // used for input an e-mail address.

`<input type="file">` defines a file-select field and a "Browse" button for file uploads.



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



`<input type="number">` defines a numeric input field.

`<input type="time">` allows the user to select a time (no time zone).

`<input type="url">` is used for input fields that should contain a URL address.



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



<input type="range"> defines a control for entering a number whose exact value is not important (like a slider control).

Default range is 0 to 100.
However, we can set restrictions on what numbers are accepted with the min, max, and step attributes:

<input type="range" min="0" max="10">



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



HTML Input Attributes

The **value** attribute specifies the initial value for an input field:

```
<input type="text" name="firstname" value="John">
```

The **readonly** attribute specifies that the input field is read only (cannot be changed):

```
<input type="text" name="firstname" value="John" readonly="readonly">
```

The **disabled** attribute specifies that the input field is disabled.

```
<input type="text" name="firstname" value="John" disabled="disabled">
```



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



The **size** attribute specifies the size (in characters) for the input field:

```
<input type="text" name="firstname" size="40">
```

The **maxlength** attribute specifies the maximum allowed length for the input field:

```
<input type="text" name="firstname" maxlength="10">
```

The **autofocus** attribute specifies that the input field should automatically get focus when the page loads.

```
<input type="text" name="fname" autofocus>
```



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



The **form method** attribute defines the HTTP method for sending form-data to the action URL.

<form action="/action_page.php" method="get">

The **pattern** attribute specifies a regular expression that the <input> element's value is checked against.

<input type="text" name="country_code" pattern="[A-Za-z]{3}">



The **placeholder** attribute specifies a hint that describes the expected value of an input field (a sample value or a short description of the format).

```
<input type="text" name="fname" placeholder="First  
name">
```

The **required** attribute specifies that an input field must compulsorily be filled out before submitting the form.

```
<input type="text" name="username" required="required">
```



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



The Action Attribute

The action attribute defines the action to be performed when the form is submitted. Normally, the form data is sent to a web page on the server when the user clicks on the submit button.

```
<form action="/action_page.php">  
    First name:<br>  
    <input type="text" name="firstname"  
    value="Mickey"><br>  
    Last name:<br>  
    <input type="text" name="lastname" value="Mouse">  
    <br><br>  
    <input type="submit" value="Submit">  
</form>
```



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



The Target Attribute

The target attribute specifies if the submitted result will open in a new browser tab, a frame, or in the current window.

```
<form action="/action_page.php" target="_blank">
```



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



The Method Attribute

The method attribute specifies the HTTP method (GET or POST) to be used when submitting the form data:

```
<form action="/action_page.php" method="get">  
<form action="/action_page.php" method="post">
```



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



Notes on GET:

- Appends form-data into the URL in name/value pairs
- The length of a URL is limited (about 3000 characters)
- Never use GET to send sensitive data! (will be visible in the URL)
- Useful for form submissions where a user wants to bookmark the result
- GET is better for non-secure data, like query strings in Google

Notes on POST:

- POST has no size limitations, and can be used to send large amounts of data.
- Form submissions with POST cannot be bookmarked

Note: more details will be discussed during PHP session



Grouping Form Data with <fieldset>

The <fieldset> element is used to group related data in a form.

The <legend> element defines a caption for the <fieldset> element.

```
<form action="/action_page.php">
  <fieldset>
    <legend>Personal information:</legend>
    First name:<br>
    <input type="text" name="firstname" value="Mickey"><br>
    Last name:<br>
    <input type="text" name="lastname" value="Mouse"><br><br>
    <input type="submit" value="Submit">
  </fieldset>
</form>
```



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



Frames

- Frames are rectangular sections of the display window, each of which can display a different document
- The `<frameset>` tag specifies the number of frames and their layout in the window
- `<frameset>` takes the place of `<body>`
- Cannot have both!
- `<frameset>` must have either a `rows` attribute or a `cols` attribute, or both (usually the case)
- Default is 1
- The possible values for `rows` and `cols` are numbers, percentages, and asterisks



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



CSE2067 Web Technologies

The <frame> tag specifies the content of a frame

The first <frame> tag in a <frameset> specifies the content of the first frame, etc.
Row-major order is used

Frame content is specified with the src attribute

Without a src attribute, the frame will be empty (such a frame CANNOT be filled later)

If <frameset> has fewer <frame> tags than frames, the extra frames are empty



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



CSE2067 Web Technologies

Frames (continued)

- An asterisk after some other specification gives the remainder of the height of the window
- Examples:

```
<frameset rows = "150, 200, 300">
```

```
<frameset rows = "25%, 50%, 25%">
```

```
<frameset rows = "50%, 20%, *" >
```

```
<frameset rows = "50%, 25%, 25%" cols = "40%, *">
```

- A number value specifies the row height in pixels - Not terribly useful!
- A percentage specifies the percentage of total window height for the row - Very useful!



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



CSE2067 Web Technologies

Creating Frames

To use frames on a page we use <frameset> tag instead of <body> tag.

The <frameset> tag defines, how to divide the window into frames.

The **rows** attribute of <frameset> tag defines horizontal frames and **cols** attribute defines vertical frames.

Note: HTML 5 is not supporting frame tag



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



CSE2067 Web Technologies

```
<html>

    <head>
        <title>HTML Frames</title>
    </head>

    <frameset rows = "10%,80%,10%">
        <frame name = "top" src = "/html/top_frame.htm" />
        <frame name = "main" src = "/html/main_frame.htm" />
        <frame name = "bottom" src = "/html/bottom_frame.htm" />

    <noframes>
        <body>Your browser does not support frames.</body>
    </noframes>

    </frameset>

</html>
```



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



CSE2067 Web Technologies

We can define an inline frame with HTML tag **<iframe>**.

The **<iframe>** tag is not somehow related to **<frameset>** tag, instead, it can appear anywhere in your document.

The **<iframe>** tag defines a rectangular region within the document in which the browser can display a separate document, including scrollbars and borders.

An inline frame is used to embed another document within the current HTML document.



```
<html>

    <head>
        <title>HTML Iframes</title>
    </head>

    <body>
        <p>Document content goes here...</p>

        <iframe src = "/html/menu.htm" width = "555" height = "200">
            Sorry your browser does not support inline frames.
        </iframe>

        <p>Document content also go here...</p>
    </body>

</html>
```



Set Font Size

You can set content font size using **size** attribute. The range of accepted values is from 1(smallest) to 7(largest). The default size of a font is 3.

```
<html>  
  <head>  
    <title>Setting Font Size</title>  
  </head>  
  
  <body>  
    <font size = "1">Font size = "1"</font><br />  
    <font size = "2">Font size = "2"</font><br />  
    <font size = "3">Font size = "3"</font><br />  
    <font size = "4">Font size = "4"</font><br />  
    <font size = "5">Font size = "5"</font><br />  
    <font size = "6">Font size = "6"</font><br />  
    <font size = "7">Font size = "7"</font>  
  </body>  
  
</html>
```



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



CSE2067 Web Technologies

The <marquee> Tag

An HTML marquee is a scrolling piece of text displayed either horizontally across or vertically down your webpage depending on the settings.

```
<html>  
  <head>  
    <title>HTML marquee Tag</title>  
  </head>  
  
  <body>  
    <marquee>This is basic example of marquee</marquee>  
  </body>  
  
</html>
```



```
<body>
  <marquee direction = "right">This text will scroll from left to right</marquee>
</body>
```

```
<body>
  <marquee direction = "up">This text will scroll from bottom to up</marquee>
</body>
```



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



CSE2067 Web Technologies

The HTML Style Attribute

```
<tagname style="property:value;">
```

```
<body style="background-color:powderblue;">
<h1 style="color:blue;">This is a heading</h1>
<p style="color:red;">This is a paragraph.</p>
<h1 style="font-family:verdana;">This is a heading</h1>
<p style="font-family:courier;">This is a paragraph.</p>
<h1 style="font-size:300%;">This is a heading</h1>
<p style="font-size:160%;">This is a paragraph.</p>
<h1 style="text-align:center;">Centered Heading</h1>
<p style="text-align:center;">Centered paragraph.</p>
```



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



HTML Colors

```
<h1 style="background-color:DodgerBlue;">Hello World</h1>
<p style="background-color:Tomato;">Lorem ipsum...</p>
```

```
<h1 style="color:Tomato;">Hello World</h1>
<p style="color:DodgerBlue;">Lorem ipsum...</p>
<p style="color:MediumSeaGreen;">Ut wisi enim...</p>
```

```
<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>
```



Image Floating

```
<p>   
The image will float to the right of the text.</p>
```

```
<p>  
The image will float to the left of the text.</p>
```



Image Maps

The <map> tag defines an image-map. An image-map is an image with clickable areas.

In the image below, click on the computer, the phone, or the cup of coffee:



```

```

```
<map name="workmap">
```

```
    <area shape="rect" coords="34,44,270,350" alt="Computer"  
        href="computer.htm">
```

```
    <area shape="rect" coords="290,172,333,250" alt="Phone"  
        href="phone.htm">
```

```
    <area shape="circle" coords="337,300,44" alt="Coffee"  
        href="coffee.htm">
```

```
</map>
```



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



CSE2067 Web Technologies

Background Image

```
<body style="background-image:url('clouds.jpg');">
```

```
<h2>Background Image</h2>
```

```
</body>
```

```
<p style="background-image:url('clouds.jpg');">
```

```
...
```

```
</p>
```



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



CSE2067 Web Technologies

The <div> Element

The <div> element is often used as a container for other HTML elements.

```
<div style="background-color:black;color:white;padding:20px;">
    <h2>London</h2>
    <p>London is the capital city of England. It is the most populous
       city in the United Kingdom, with a metropolitan area of over 13
       million inhabitants.</p>
</div>
```



The Element

The element is often used as a container for some text.

```
<h1>My <span style="color:red">Important</span>  
Heading</h1>
```



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



CSE2067 Web Technologies

HTML Iframes

An iframe is used to display a web page within a web page.

```
<iframe src="URL"></iframe>
```

```
<iframe src="demo_iframe.htm" height="200"  
width="300"></iframe>
```

```
<iframe src="demo_iframe.htm"  
style="height:200px;width:300px;"></iframe>
```



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



CSE2067 Web Technologies

```
<iframe src="demo_iframe.htm"  
style="border:none;"></iframe>
```

```
<iframe src="demo_iframe.htm" style="border:2px solid  
red;"></iframe>
```



PRESIDENCY
UNIVERSITY

Private University Estd. in Karnataka State by Act No. 41 of 2013



CSE2067 Web Technologies

Iframe - Target for a Link

An iframe can be used as the target frame for a link.

The target attribute of the link must refer to the name attribute of the iframe:

```
<iframe src="demo_iframe.htm"  
name="iframe_a"></iframe>
```

```
<p><a href="https://www.google.com"  
target="iframe_a">Load new web page</a></p>
```



HTML Entities

Result	Description	Entity Name
	non-breaking space	
<	less than	<
>	greater than	>
&	ampersand	&
"	double quotation mark	"
'	single quotation mark (apostrophe)	'
¢	cent	¢
£	pound	£
¥	yen	¥
€	euro	€
©	copyright	©
®	registered trademark	®



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



CSE2067 Web Technologies

HTML - Embed Multimedia

Sometimes you need to add music or video into your web page. The easiest way to add video or sound to your web site is to include the special HTML tag called <embed>. This tag causes the browser itself to include controls for the multimedia automatically provided browser supports <embed> tag and given media type.

You can also include a <noembed> tag for the browsers which don't recognize the <embed> tag. You could, for example, use <embed> to display a movie of your choice, and <noembed> to display a single JPG image if browser does not support <embed> tag.



```
<html>

    <head>
        <title>HTML embed Tag</title>
    </head>

    <body>
        <embed src = "/html/yourfile.mid" width = "100%" height = "60" >
        <noembed><img src = "yourimage.gif" alt = "Alternative Media"
    ></noembed>
        </embed>
    </body>

</html>
```



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



CSE2067 Web Technologies

Sr No	Attribute & Description
1	align Determines how to align the object. It can be set to either center, <i>left</i> or <i>right</i> .
2	autoplay This boolean attribute indicates if the media should start automatically. You can set it either true or false.
3	loop Specifies if the sound should be played continuously (set loop to true), a certain number of times (a positive value) or not at all (false)
4	playcount Specifies the number of times to play the sound. This is alternate option for <i>loop</i> if you are usiong IE.
5	hidden Specifies if the multimedia object should be shown on the page. A false value means no and true values means yes.
6	width Width of the object in pixels
7	height Height of the object in pixels
8	name A name used to reference the object.
9	src URL of the object to be embedded.
10	volume Controls volume of the sound. Can be from 0 (off) to 100 (full volume).



Background Audio

You can use HTML <bgsound src=" " > tag to play a soundtrack in the background of your webpage. This tag is supported by Internet Explorer only and most of the other browsers ignore this tag. It downloads and plays an audio file when the host document is first downloaded by the user and displayed. The background sound file also will replay whenever the user refreshes the browser.



8 Best Front End Programming Languages that You Should Know About

What Is Frontend Development?

1. HyperText Markup Language (HTML)
2. Cascading Style Sheets (CSS)
3. JavaScript
4. React
5. Angular
6. Vue
7. jQuery
8. Swift



Back end web technologies

JAVASCRIPT....

RUBY. ...

PHP. ...

JAVA. ...

GOLANG. ...

C# ...

PERL.



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



CSE2067 Web Technologies

103



Front End

- Markup and web languages such as HTML, CSS and Javascript
- Asynchronous requests and Ajax
- Specialized web editing software
- Image editing
- Accessibility
- Cross-browser issues
- Search engine optimisation



Back End

- Programming and scripting such as Python, Ruby and/or Perl
- Server architecture
- Database administration
- Scalability
- Security
- Data transformation
- Backup



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



CSE2067 Web Technologies

Web Designer vs Web Developer

Job Title	
Web Designer	Web Developer
Focuses on look and feel of a website	Creates the inner workings of a website
HTML, CSS, and JavaScript	PHP, .NET, Python, C, Ruby
Right-brain: Strong intuition, creativity and imagination	Left-brain: Logic, linear thinking and technical
Salary	
\$64,110	\$70,120
Freelance	
Yes	Yes



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



FRONTEND DEVELOPER

BACKEND DEVELOPER

How Things Look

Markup

Web Languages

Web Editing

Image Editing

Accessibility

Usability

Browser Issues

SEO

How Things Work

Programming

Scripting

Server Architecture

Database Admin.

Security

Scalability

Data Transformation

Backup



Difference between HTML & XHTML

Sl. No.	HTML	XHTML
1	HTML stands for Hypertext Markup Language.	XHTML stands for Extensible Hypertext Markup Language.
2	It was developed by Tim Berners-Lee	It was developed by W3C ie. World Wide Web Consortium.
3	It was developed in 1991	It was released in 2000.
4	All tags and attributes are not necessarily to be in specific case.	Every tag and attribute should be in lower case.
5.	It is not necessary to close the tags in the order they are opened.	It is necessary to close the tags in the same order they are opened.
6.	While using the attributes it is not necessary to mention quotes.	While using the attributes it is mandatory to mention quotes. Eg - <p bgcolor = "yellow">
7.	Filename extension used are .html, .htm	Filename extension used are .xhtml, .xht, .xml



THANK YOU



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



CSE2067 Web Technologies

108