

Unit - III

3.1 Forms

In HTML, Forms are used for collecting data from the user. The data collected from the user on the web browser is sent to the server for processing

Syntax

```
<form attributes>
```

.

.

```
</form>
```

Attributes of Form tag

- **name**: used to provide name for the Form.
- **id**: used to provide a unique name for the Form.
- **method**: Used to provide the method by which the data is sent to the server.
- **action**: used to provide the URL for the file on the server.
- **autocomplete**: used to set the Form autocomplete property.

Example

```
<form action="URL" method="POST/GET">  
  Form elements  
</form>
```

Input Element

In HTML, input element tag is used to create interactive controls for the FORMS. They are used to collect user data for processing.

General Syntax

```
<input type="" id="" name="" attribute>
```

Attribute

- **readonly**: Used to specify if the element is readonly or not. Data in the element can be read.
- **disabled**: Used to disable the element. Data in the element cannot be accessed.

Input Element Types

Input tag is used to specify an input field in the HTML page. It is used to get data from the user. It is an important element in HTML Forms. Based on the value of type attribute different Input elements are displayed. Types of Input elements are given below,

- Text Box
- Label
- Password
- Hidden
- Textarea
- Radio Button
- Checkbox
- Select: Pull Down List
- File Upload
- Submit
- Reset

</label>

Text Box

This is used to get a single line value from the user. It has a default width of 20 Characters.

Syntax

```
<input type="text" id="" name="" value="">
```

Label

This is used to specify a text label for an <input> element.

Syntax

```
<label for = "">Text Value</label>
```

for

- This is used to assign the label for an input element. The id value of the input element is given in this attribute.

Usage of Label

HTML label can be used in two ways,

1. Set the "id" attribute in <input> element and specify the value in the "for" attribute inside the <label> tag.

Example

```
<label for="txtname" >Enter Name: </label>
```

```
<input type="text" id="txtname" value="Rajesh" readonly>
```

2. <input> tag can be used within the <label> element.

Example

```
<label>
```

```
Enter Name: <input type="text" id="txtname" required>
```

Password

This is used to get passwords from the users. The text in this field is hidden by default.

Syntax

```
<input type="password" name="" value="">
```

Hidden

This element is used to store data that is hidden from the user. This is generally used to store values that are required for editing forms.

Syntax

```
<input type="hidden" name="" value="">
```

Textarea

This is a multi-line text input field. The size of this element is defined by the rows and cols attribute.

Syntax

```
<textarea rows="" cols="" name="" id="">
```

```
</textarea>
```

Radio Button

This is used to provide multiple values to the user from which they can select exactly one value. Each Radio Button element must have a <label> element.

Syntax

```
<label>
```

```
<input type="radio" name="" value=""> Display Text
</label>
```

Checkbox

This is used to provide multiple values to the user from which they can select more than one value. Each Checkbox element must have a <label> element.

Syntax

```
<label>
  <input type="checkbox" name="" value="">Display Text
</label>
```

Select Tag

This is used to create a drop-down list that shows multiple values. This consists of <option> tag to show multiple values.

Syntax

```
<select name="">
  <option value="">....</option>
  <option value="">....</option>
</select>
```

File Upload

This is used to select a file from the computer and upload it to the server.

Syntax

```
<input type="file" attribute>
```

Attributes

- **accept**: Specifies the file type to be accepted for uploading.
- **multiple**: This allows form multiple file selection.

Submit and Reset (Action Buttons)

These elements are rendered as buttons in the browser. Each element has a specific function, Submit is used to send the data entered by the user on the Form to the server and Reset is used to clear the data in the form.

Syntax

```
<input type="submit" value="">
<input type="reset" value="">
```

Image Button

This is used to define an image as a button. Image path is specified in "src" attribute.

Syntax

```
<input type="image" src="src" alt="alt" >
```

- src: Provides the path of the image.
- alt: Provide an alternate text when the image is not available.

Form Example

```
<form id="myform">
  <div>
    <label for="txtname" >Name: </label>
    <input type="text" id="txtname" required>
  </div>
  <div>
    <label>Gender</label>
    <label>
      <input type="radio" name="gender" value="M">Male
    </label>
  </div>
</div>
```

```

        <label></label>
        <label>
            <input type="radio" name="gender" value="F"> Female
        </label>
    </div>
</div>
<div>
    <label for="txtdpt" >Department: </label>
    <input type="text" id="txtdpt">
</div>
<div>
    <label>Language</label>
    <label><input type="checkbox" name="chktam"
value="tamil"> Tamil
    </label>
</div>
<div>
    <label></label>
    <label>
        <input type="checkbox" name="chkeng" value="english">
            English
    </label>
</div>
<div>
    <label for="selnation">Nationality</label>
    <select id="selnation">
        <option value="Indian">Indian</option>
        <option value="Other">Other</option>
    </select>
</div>
<div>
    <label for="upld">Photo</label>

```

```

        <input type="file" accept="image/*" multiple name="upld">
    </div>
</div>
    <label for="taradd" >Address: </label>
    <textarea id="taradd" rows="3" cols="20">
        Enter Your Address here
    </textarea>
</div>
<div>
    <input type="submit" value="Submit">
    <input type="reset" value="Clear">
    <input type="image" src="submit.png" alt="Image Button">
</div>
</form>

```

Output

HTML Form Submit and Reset

Name:

Gender
☐ Male
☐ Female

Department:

Language
☐ Tamil
☐ English

Nationality
Indian

Photo
Choose Files No file chosen

Address:
Enter Your Address here

Submit Clear

Submit

3.2 CSS Introduction

CSS

In the late 90's website creation was tedious as designs were added into it. To overcome this the concept of "separating structure and presentation" was introduced. In 1996 Cascaded Style Sheets was developed by W3C.

Advantages

- Faster web page load
- Less code in each page
- Website look is consistent
- Easy design updating
- Easy site maintenance

CSS3 Syntax

CSS contains style rules which are interpreted by the browser. The browser applies the style to the corresponding elements. Style rule is made of three parts

Syntax

selector { property: value; }

- Selector: This is used to select an element on the HTML page. Types of Selectors are
 - HTML tag selector
 - Universal Selector
 - Grouping Selector
 - Child selector
- Property: It is a type of HTML attribute.
- Value: The Value assigned for that property.

Example

```
<!DOCTYPE html>
<html>

  <head>
    <title>CSS</title>
  </head>
  <style>
    header {
      text-align:center;
      height: 100px;
      background-color: #60dd5a;
      padding: 10px;
      border-radius: 10px;
    }
    footer {
      text-align:center;
      font-size: 12px;
      height: 50px;
      line-height: 50px;
      background-color: #f1dc73;
      padding: 10px;
      border-radius: 10px;
    }
    article{
      width: auto;
      border: 2px solid gray;
      padding: 10px;
      border-radius: 10px;
      margin:5px;
    }
  </style>
</html>
```

```
</style>
<body>
<header>
<h1>CSS HTML Tag Selector</h1>
</header>
<article>
<p>
```

This example shows the implementation of HTML Tag Selector

```
</p>
</article>
<footer>
Developed by Dr. D. Natarajasivan
</footer>
</body>
```

```
</html>
```

Output

CSS HTML Tag Selector

This example shows the implementation of HTML Tag Selector

Developed by Dr. D. Natarajasivan

Adding CSS

Embedded Style sheets

This type of CSS styling is placed directly within the <style> element in the HTML document's <head> section.

External style sheets

Instead of embedding CSS directly within an HTML document (using inline styles or the <style> element), external style sheets use a separate CSS file with the ".css" extension. It is linked to the HTML document using the <link> tag in <head> section.

Imported style sheets

CSS rules are imported to an HTML document by using the @import rule within an external or embedded CSS file. The @import rule allows you to import and combine multiple CSS files

Inline style sheets

Inline styles in HTML are a way of applying CSS (Cascading Style Sheets) directly to individual HTML elements using the style attribute.

Example

```
<!DOCTYPE html>
<html>
<head>
<title>CSS</title>
</head>
<!--External Style -->
<link href="css/style.css" rel="stylesheet">
<!--Embedded Style-->
```

```

<style>
  @import "css/style1.css" screen and (max-width:780px);

  header {
    text-align:center;
    height: 100px;
    background-color: #60dd5a;
    padding: 10px;
    border-radius: 10px;
  }
  footer {
    text-align:center;
    font-size: 12px;
    height: 50px;
    line-height: 50px;
    background-color: #f1dc73;
    padding: 10px;
    border-radius: 10px;
  }
  article{
    width: auto;
    border: 2px solid gray;
    padding: 10px;
    border-radius: 10px;
    margin:5px;
  }
</style>
<body>
  <header>
    <h1>CSS HTML Style Sheet Types</h1>
    </header>

```

```

<article>
  <!--Inline CSS Style-->
  <h2 style="color:GREEN">Style Sheet Types</h2>
  <p>
    This example shows the <span>implementation</span> of
    Style Sheet Types
  </p>
</article>
<footer>
  Developed by Dr. D. Natarajasivan
</footer>
</body>
</html>

```

style.css

```

span {
  color:RED;
  font-size: 20px;
}

```

Output

CSS HTML Style Sheet Types

Style Sheet Types

This example shows the **implementation** of Style Sheet Types

Developed by Dr. D. Natarajasivan

Cascading Rules

There are four major rules that govern usage of CSS they are

- Order of Precedence
- Specificity
- Inheritance
- Important

Order of Precedence

When more than one rule for the same selector is used the selector matches CSS from Top down order. This rule is applied when using external style sheets also.

Specificity

When the same element is targeted by different selectors. The rule of specificity uses the rule that a most specific selector provides. Order of specificity

- Element name – not specific
- Classes – more specific
- ID – very specific

Inheritance

When a tag is applied CSS the tags within the parent tag gets all the CSS properties of the parent tag. This is not applied for all the properties

Important

When a rule is marked as important it takes over a normal rule. It is used along with a rule.

Syntax

!important

Dr. D. NATARAJASIVAN/TNPT

3.3 CSS Selector

A CSS selector is used to select the HTML element for which style is to be applied. Some of the CSS selectors available are

- Type Selector
- Grouping
- Universal Selector
- Descendant Selector
- Child Selector

Type Selector

A CSS type selector is also known as an element selector. This is used to select all the elements of a given type in the HTML document. This can be implemented using element name or based on the attribute of the element.

Element Name Selector

In this type HTML element name is used to select and apply the given style.

Syntax

```
tag{  
    Property: rule;  
}
```


Attribute Selector

In this the HTML element is selected based on the attribute value available in the element.

Syntax

```
[attribute=value]{  
    Property: rule;  
}
```

Example

```
<style>  
article{  
    width: auto;  
    border: 2px solid gray;  
    padding: 10px;  
    border-radius: 10px;  
    margin:5px;  
}  
[id=arthead]{  
    background-color:lightgreen;  
}  
</style>  
<article>  
    <h2>Tag Selector </h2>  
    <p id="artpara">  
        This is attribute selector  
    </p>  
</article>
```

Output

Tag Selector

This is attribute selector

Grouping

Using this type of selector multiple elements can be grouped to apply a common CSS. This is achieved by separating the selectors with commas.

Syntax

```
tag1, tag2, tag3{  
    Property: rule;  
}
```

Example

```
<style>  
h2, p{  
    font-family: "Arial Black";  
    color: #000;  
}  
</style>  
<article>  
    <h2>Grouping Selector </h2>  
    <p>This example shows the implementation of Grouping  
        Selector  
    </p>  
</article>
```

Output

Grouping Selector

This example shows the implementation of Grouping Selector

Universal Selector

The universal selector is a special type of selector that targets all elements on a webpage. It is represented by an asterisk (*) and is used to apply styles to every element on the page.

Syntax

```
*{  
    Property: rule;  
}
```

Example

```
<style>  
    *{  
        font-family: "Helvetica";  
        color: BLUE;  
    }  
</style>  
<article>  
    <h1>Universal Selector</h1>  
    <p>  
        This example shows the implementation of Universal Selector  
    </p>  
</article>
```

Dr. D. NATARAJASIVAN/TNPT

Output

Universal Selector

This example shows the implementation of Universal Selector

Descendant Selector

A descendant selector is a type of selector that targets elements that are descendants of another element. A descendant is an element that is contained within another element, either directly or indirectly. A descendant selector is written by separating the parent element and the descendant element with a space.

Syntax

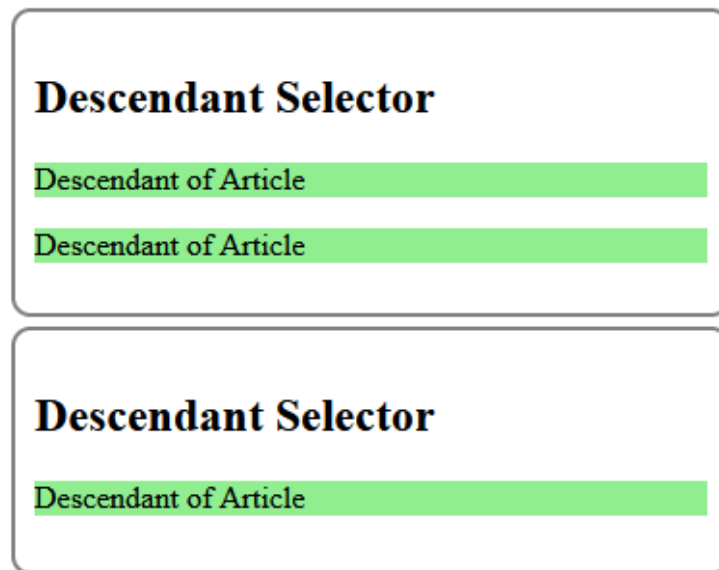
```
tag1 tag2 {  
    Property: rule;  
}
```

Example

```
<style>  
    article p{  
        background-color:lightgreen;  
    }  
</style>  
<article>  
    <h2>Descendant Selector</h2>  
    <p>Descendant of Article</p>  
</article>
```

```
<article>
  <h2>Descendant Selector </h2>
  <p>Descendant of Article</p>
</article>
```

Output



Child Selector

The child selector, represented by the greater-than sign (>), targets elements that are *direct children* of another element. It's more specific than the descendant selector, which matches all elements that are descendants of a specified element.

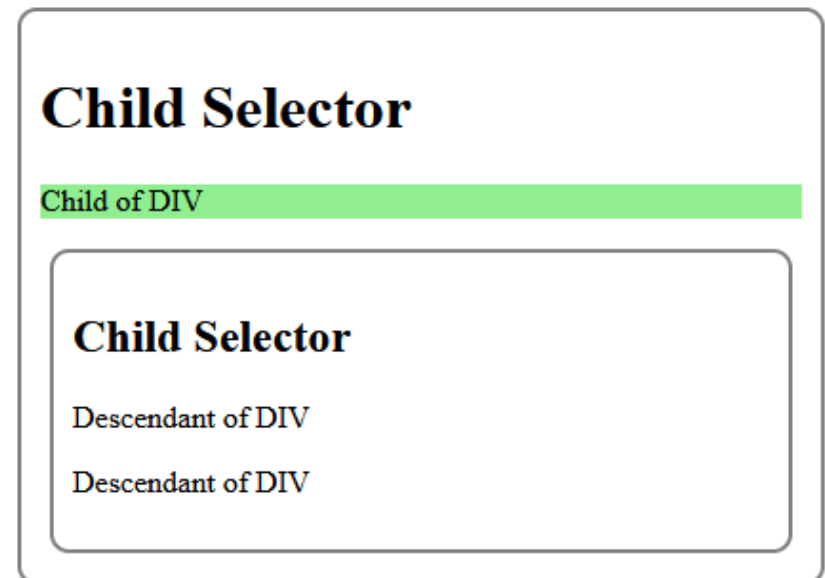
Syntax

```
tag1 > tag2{
  Property: rule;
}
```

Example

```
<style>
  div > p{
    background-color:lightgreen;
  }
</style>
<div>
  <h1>Child Selector</h1>
  <p>Child of DIV</p>
  <article>
    <h2>Child Selector</h2>
    <p>Descendant of DIV</p>
    <p>Descendant of DIV</p>
  </article>
</div>
```

Output



Dynamic Pseudo Classes

Dynamic pseudo-classes are CSS selectors that allow styling elements based on their current state or user interaction. These pseudo-classes are called "dynamic" because they are applied dynamically, meaning that they are applied in response to user interactions. Some of the pseudo classes are

- hover: Applied when the user moves the mouse over an element.
- focus: Applied when an element receives focus, such as when a user clicks on a form input.
- active: Applied when the user clicks on an element.

Syntax

```
tag:pseudoClass{  
    Property: rule;  
}
```

Example - HOVER

```
<style>  
  p:hover{  
    font-size: 125%;  
    background-color:#60dd5a;  
  }  
</style>  
<article>  
  <h2>Hover</h2>  
  <p>HTML element with given class attribute is selected</p>  
</article>
```

Output

Hover

HTML element with given class attribute is selected

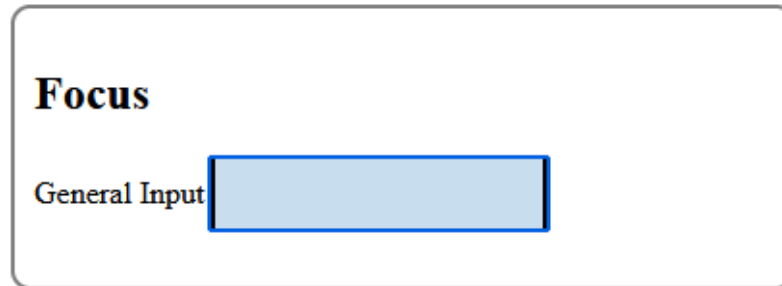
Hover

HTML element with given class attribute is selected

Example: Focus

```
<style>  
  input[type=text]{  
    border: none;  
    border-right: 2px solid black;  
    border-left: 2px solid black;  
  }  
  input[type=text]:focus{  
    padding: 10px;  
    background-color:rgba(146, 188, 222,0.5);  
  }  
</style>  
<article>  
  <h2>Focus</h2>  
  <label for="input">General Input</label>  
  <input type="text" id="input">  
</article>
```

Output

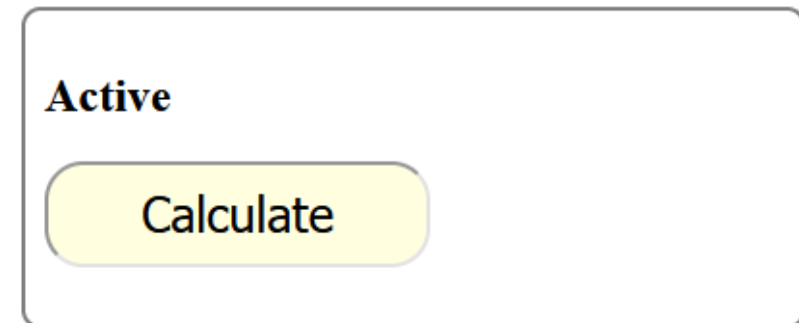
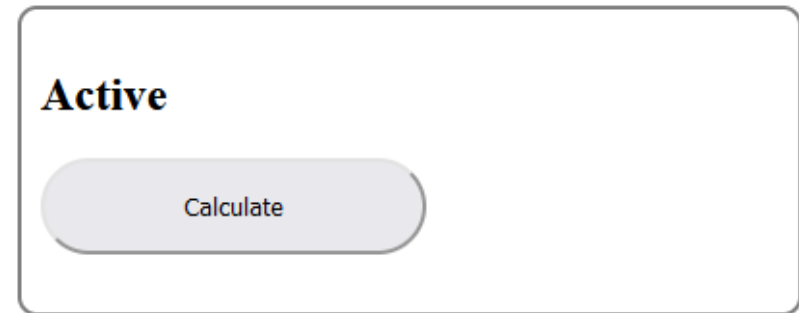


Example-ACTIVE

```
<style>
  input[type=button]{
    padding: 10px;
    width: 200px;
    height: 50px;
    border-radius: 40px;
  }
  input[type=button]:active{
    padding: 10px;
    font-size: 25px;
    background-color:lightyellow;
    border-radius: 20px;
    height: 55px;
  }
</style>
```

```
<article>
  <h2>Active</h2>
  <input type="button" value="Calculate">
</article>
```

Output



Class Selector

A **class selector** in CSS is used to select all elements with a specific class attribute. A class selector is denoted by a dot (.) followed by the class name.

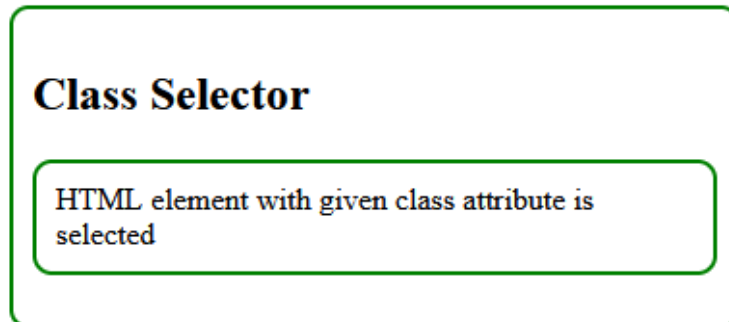
Syntax

```
.className{
  Property: rule;
}
```

Example

```
<style>
  .article{
    width: auto;
    border: 2px solid green;
    padding: 10px;
    border-radius: 10px;
    background-color: rgba(242, 204, 109)
    margin:5px;
  }
</style>
<div class="article">
  <h2>Class Selector</h2>
  <p class="article">
    HTML element with given class attribute is selected
  </p>
</div>
```

Output



ID Selector

A **CSS ID selector** is used to select a single, unique element within an HTML document. It's identified by the **id** attribute of the element. An ID selector is denoted by a hash symbol (#) followed by the ID name.

Syntax

```
#Tag-ID{
  Property: rule;
}
```

Example

```
<style>
  #cont{
    width: auto;
    border: 4px dotted red;
    padding: 10px;
    border-radius: 10px;
    margin:5px;
  }
</style>
<div id="cont">
  <h2>ID Selector </h2>
  <p>
    HTML element with
  </p>
</div>
```

Output

ID Selector

HTML element with