

Unit - III

3.1 Forms

HTML இல், பயனரிடமிருந்து தரவைச் சேகரிக்க படிவங்கள் பயன்படுத்தப்படுகின்றன. இணைய உலாவியில் பயனரிடமிருந்து சேகரிக்கப்பட்ட தரவு செயலாக்கத்திற்காக serverக்கு அனுப்பப்படும்.

Syntax

```
<form attributes>
```

.

.

```
</form>
```

Attributes of Form tag

- **name:** படிவத்திற்கான பெயரை வழங்க பயன்படுகிறது.
- **id:** படிவத்திற்கு ஒரு தனிப்பட்ட பெயரை வழங்க பயன்படுகிறது.
- **method:** server தரவு அனுப்பப்படும் முறையை வழங்க பயன்படுகிறது.
- **action:** சர்வரில் உள்ள கோப்பிற்கான URL ஐ வழங்க பயன்படுகிறது.
- **autocomplete:** படிவத்தை தானாக நிறைவு செய்யும் தன்மையை அமைக்க பயன்படுகிறது.

Example

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

Form elements

```
</form>
```

Input Element

HTML இல், படிவங்களுக்கான ஊடாகும் கட்டுப்பாடுகளை உருவாக்க உள்ளீட்டு உறுப்பு குறிச்சொல் பயன்படுத்தப்படுகிறது. செயலாக்கத்திற்கான பயனர் தரவை சேகரிக்க அவை பயன்படுத்தப்படுகின்றன.

General Syntax

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

Attribute

- **readonly:** உறுப்பு படிக்க மட்டும் உள்ளதா இல்லையா என்பதைக் குறிப்பிடப் பயன்படுகிறது. உறுப்பு உள்ள தரவு படிக்க முடியும்.
- **disabled:** உறுப்பை முடக்கப் பயன்படுகிறது. உறுப்பில் உள்ள தரவை அணுக முடியாது.

Input Element Types

HTML பக்கத்தில் உள்ளீட்டு புலத்தைக் குறிப்பிட உள்ளீட்டு tag பயன்படுத்தப்படுகிறது. பயனரிடமிருந்து தரவைப் பெற இது பயன்படுகிறது. HTML படிவங்களில் இது ஒரு முக்கிய அங்கமாகும். Type attributeன் மதிப்பின் அடிப்படையில் வெவ்வேறு உள்ளீட்டு கூறுகள் காட்டப்படும். உள்ளீட்டு கூறுகளின் வகைகள் கீழே கொடுக்கப்பட்டுள்ளன,

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

- Submit
- Reset

Text Box

பயனரிடமிருந்து ஒற்றை வரி மதிப்பைப் பெற இது பயன்படுகிறது. இது 20 எழுத்துகளின் இயல்புநிலை அகலத்தைக் கொண்டுள்ளது.

Syntax

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

Label

<input> உறுப்புக்கான text லேபிளைக் குறிப்பிட இது பயன்படுகிறது.

Syntax

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

for

- உள்ளீட்டு உறுப்புக்கான லேபிளை ஒதுக்க இது பயன்படுகிறது. உள்ளீட்டு உறுப்பின் ஐடி மதிப்பு இந்தப் attributeல் கொடுக்கப்பட்டுள்ளது.

Usage of Label

HTML லேபிளை இரண்டு வழிகளில் பயன்படுத்தலாம்,

1. <input> உறுப்பில் "id" attribute அமைத்து, <label> குறிச்சொல்லில் உள்ள "for" attributeல் மதிப்பைக் குறிப்பிடவும்.

Example

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

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

2. <input> tag <label> உறுப்புக்குள் பயன்படுத்தப்படலாம்.

Example

```
<label>
```

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

```
</label>
```

Password

பயனர்களிடமிருந்து கடவுச்சொற்களைப் பெற இது பயன்படுத்தப்படுகிறது. இந்தப் புலத்தில் உள்ள text முன்னிருப்பாக மறைக்கப்பட்டுள்ளது.

Syntax

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

Hidden

பயனரிடமிருந்து மறைக்கப்பட்ட தரவைச் சேமிக்க இந்த element பயன்படுத்தப்படுகிறது. படிவங்களைத் திருத்துவதற்குத் தேவையான மதிப்புகளைச் சேமிக்க இது பொதுவாகப் பயன்படுத்தப்படுகிறது.

Syntax

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

Textarea

இது பல வரி text உள்ளீட்டு புலமாகும். இந்த உறுப்பின் அளவு rows மற்றும் cols பண்புக்கூறால் வரையறுக்கப்படுகிறது.

Syntax

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

```
</textarea>
```

Radio Button

பயனருக்கு பல மதிப்புகளை வழங்க இது பயன்படுகிறது, அதில் இருந்து அவர்கள் ஒரு மதிப்பை

சரியாக தேர்ந்தெடுக்க முடியும். ஒவ்வொரு ரேடியோ பட்டன் உறுப்புக்கும் ஒரு <label> உறுப்பு இருக்க வேண்டும்.

Syntax

<label>

<input type="radio" name="" value=""> Display Text

</label>

Checkbox

பயனருக்கு பல மதிப்புகளை வழங்க இது பயன்படுகிறது, அதில் இருந்து அவர்கள் ஒன்றுக்கு மேற்பட்ட மதிப்பைத் தேர்ந்தெடுக்கலாம். ஒவ்வொரு செக்பாக்ஸ் உறுப்புக்கும் ஒரு <label> உறுப்பு இருக்க வேண்டும்.

Syntax

<label>

<input type="checkbox" name="" value="">Display Text

</label>

Select Tag

பல மதிப்புகளைக் காட்டும் கீழ்தோன்றும் பட்டியலை உருவாக்க இது பயன்படுகிறது. இது பல மதிப்புகளைக் காட்ட <option> குறிச்சொல்லைக் கொண்டுள்ளது.

Syntax

<select name="">

<option value="">....</option>

<option value="">....</option>

</select>

File Upload

கணினியிலிருந்து ஒரு கோப்பைத் தேர்ந்தெடுத்து அதை சர்வரில் பதிவேற்ற இது பயன்படுகிறது.

Syntax

<input type="file" attribute>

Attributes

- **accept:** பதிவேற்றுவதற்கு ஏற்கப்பட வேண்டிய கோப்பு வகையைக் குறிப்பிடுகிறது.
- **multiple:** இது பல கோப்புகளைத் தேர்ந்தெடுக்க அனுமதிக்கிறது.

Submit and Reset (Action Buttons)

இந்த உறுப்புகள் உலாவியில் பொத்தான்களாக வழங்கப்படுகின்றன. ஒவ்வொரு உறுப்புக்கும் ஒரு குறிப்பிட்ட செயல்பாடு உள்ளது, படிவத்தில் பயனர் உள்ளிட்ட தரவை சேவையகத்திற்கு அனுப்ப submitம், படிவத்தில் உள்ள தரவை அழிக்க resetம் பயன்படுத்தப்படுகிறது.

Syntax

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

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

Image Button

ஒரு படத்தை பொத்தானாக வரையறுக்க இது பயன்படுகிறது. பட பாதை "src" பண்புக்கூறில் குறிப்பிடப்பட்டுள்ளது.

Syntax

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

- **src:** படத்தின் பாதையை வழங்குகிறது.
- **alt:** படம் கிடைக்காதபோது மாற்று உரையை வழங்கவும்.

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>
  <label for="txtcpt" >Department: </label>
  <input type="text" id="txtcpt">
</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

Photo No file chosen

Address:

3.2 CSS Introduction

CSS

90 களின் பிற்பகுதியில், வலைத்தள உருவாக்கம் கடினமானதாக இருந்தது, ஏனெனில் அதில் வடிவமைப்புகள் சேர்க்கப்பட்டன. இதை முறியடிக்க, "separating structure and presentation" என்ற கருத்து அறிமுகப்படுத்தப்பட்டது. 1996 ஆம் ஆண்டில் Cascaded Style Sheets W3C ஆல் உருவாக்கப்பட்டது.

Advantages

- வேகமான இணையப் பக்க ஏற்றம்
- ஒவ்வொரு பக்கத்திலும் குறைவான குறியீடு
- இணையதள தோற்றம் சீரானது
- எளிதான வடிவமைப்பு புதுப்பித்தல்
- எளிதான தள பராமரிப்பு

CSS3 Syntax

CSS ஆனது உலாவியால் விளக்கப்படும் style விதிகளைக் கொண்டுள்ளது. உலாவி தொடர்புடைய கூறுகளுக்கு பாணியைப் பயன்படுத்துகிறது. Style விதி மூன்று பகுதிகளால் ஆனது.

Syntax

selector { property: value; }

- **Selector:** HTML பக்கத்தில் ஒரு உறுப்பைத் தேர்ந்தெடுக்க இது பயன்படுகிறது. தேர்வாளர்களின் வகைகள்
 - HTML tag selector
 - Universal Selector
 - Grouping Selector
 - Child selector
- **Property:** இது ஒரு வகை HTML பண்புக்கூறு.
- **Value:** அந்தச் propertyக்கு ஒதுக்கப்பட்ட மதிப்பு.

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>
<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



Adding CSS

Embedded Style sheets

இந்த வகை CSS ஸ்டைலிங் நேரடியாக HTML ஆவணத்தின் <head> பிரிவில் உள்ள <style> உறுப்புக்குள் வைக்கப்படுகிறது.

External style sheets

ஒரு HTML ஆவணத்தில் நேரடியாக CSS ஐ உட்பொதிப்பதற்குப் பதிலாக (இன்லைன் ஸ்டைல்கள் அல்லது <style> உறுப்பைப் பயன்படுத்தி), வெளிப்புற நடைத் தாள்கள் ".css" நீட்டிப்புடன் தனி CSS கோப்பைப்

பயன்படுத்துகின்றன. இது <head> பிரிவில் உள்ள <link> குறிச்சொல்லைப் பயன்படுத்தி HTML ஆவணத்துடன் இணைக்கப்பட்டுள்ளது.

Imported style sheets

வெளிப்புற அல்லது உட்பொதிக்கப்பட்ட CSS கோப்பில் @import விதியைப் பயன்படுத்தி HTML ஆவணத்திற்கு CSS விதிகள் இறக்குமதி செய்யப்படுகின்றன. @import விதியானது பல CSS கோப்புகளை இறக்குமதி செய்து இணைக்க உங்களை அனுமதிக்கிறது

Inline style sheets

HTML இல் உள்ள இன்லைன் ஸ்டைல்கள் என்பது CSS (கேஸ்கேடிங் ஸ்டைல் ஷீட்ஸ்) தனிப்பட்ட HTML உறுப்புகளுக்கு ஸ்டைல் பண்புக்கூறைப் பயன்படுத்தி நேரடியாகப் பயன்படுத்துவதற்கான ஒரு வழியாகும்.

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 implementation 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

CSS இன் பயன்பாட்டை நிர்வகிக்கும் நான்கு முக்கிய விதிகள் உள்ளன

- Order of Precedence

Dr. D. NATARAJASIVAN/TNPT

- Specificity
- Inheritance
- Important

Order of Precedence

ஒரே தேர்விக்கு ஒன்றுக்கு மேற்பட்ட விதிகள் பயன்படுத்தப்படும் போது, தேர்வாளர் மேல் கீழ் வரிசையில் இருந்து CSS உடன் பொருந்தும். வெளிப்புற நடை தாள்களைப் பயன்படுத்தும் போது இந்த விதி பயன்படுத்தப்படுகிறது.

Specificity

ஒரே உறுப்பு வெவ்வேறு தேர்வாளர்களால் இலக்காகும்போது, குறிப்பிட்ட விதி மிகவும் குறிப்பிட்ட தேர்வாளர் வழங்கும் விதியைப் பயன்படுத்துகிறது. குறிப்பிட்ட வரிசை

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

Inheritance

ஒரு குறிச்சொல் CSS பயன்படுத்தப்படும் போது, பெற்றோர் குறிச்சொல்லில் உள்ள குறிச்சொற்கள் பெற்றோர் குறிச்சொல்லின் அனைத்து CSS பண்புகளையும் பெறுகின்றன. இது அனைத்து பண்புகளுக்கும் பொருந்தாது

Important

ஒரு விதி முக்கியமானதாகக் குறிக்கப்படும் போது, அது ஒரு சாதாரண விதியை மாற்றிவிடும். இது ஒரு விதியுடன் சேர்த்து பயன்படுத்தப்படுகிறது.

Syntax

!important

3.3 CSS Selector

CSS தேர்விகள் (selectors) HTML கூறுகளைத் தேர்வு செய்வதற்காக பயன்படுத்தப்படுகின்றன, அதில் ஸ்டைல்கள் பயன்படுத்தப்படுகின்றன. சில பொதுவான CSS தேர்விகள் கீழே கொடுக்கப்பட்டுள்ளன:

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

Type Selector

CSS வகை தேர்வி (Type Selector) என்பது ஒரு எலிமென்ட் தேர்வியாகவும் அறியப்படுகிறது. இது HTML ஆவணத்தில் கொடுக்கப்பட்ட ஒரு வகையின் அனைத்து கூறுகளையும் தேர்வு செய்ய பயன்படுத்தப்படுகிறது. இது எலிமென்ட் பெயரைப் பயன்படுத்தி அல்லது அந்த எலிமென்ட்டின் பண்புக்கூறின் அடிப்படையில் செயல்படுத்தப்படலாம்.

Element Name Selector

இந்த வகையில், HTML எலிமென்ட் பெயரைப் பயன்படுத்தி அந்த குறிப்பிட்ட எலிமென்டிற்கான ஸ்டைலைத் தேர்வு செய்து பயன்படுத்தப்படுகிறது.

Syntax

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

Attribute Selector

இதில், HTML எலிமென்ட், அந்த எலிமென்ட்டில் உள்ள பண்புக்குறி (attribute) மதிப்பின் அடிப்படையில் தேர்வு செய்யப்படுகிறது.

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

இந்த வகை தேர்வியைப் பயன்படுத்தி, பல எலிமெண்டுகளை ஒரு பொதுவான CSS ஸ்டைல் பயன்படுத்தக் குழுவாக்கலாம். இதை தேர்விகளை commaவால் பிரித்து செய்ய முடியும்.

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

யுனிவர்சல் தேர்வி என்பது ஒரு சிறப்பு வகை தேர்வியாகும், இது ஒரு வலைப்பக்கத்தில் உள்ள அனைத்து எலிமெண்டுகளையும் குறிவைக்கிறது. இது நட்சத்திர சின்னம் (*) மூலம் பிரதிநிதித்துவப்படுத்தப்படுகிறது மற்றும் பக்கத்தில் உள்ள ஒவ்வொரு எலிமென்டிற்கும் ஸ்டைல்களைப் பயன்படுத்துவதற்கு பயன்படுத்தப்படுகிறது.

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>
```

Output

Universal Selector

This example shows the implementation of Universal Selector

Descendant Selector

Descendant selector என்பது மற்றொரு உறுப்பின் தொடர்ச்சியான உறுப்புகளை குறிவைக்கும் ஒரு வகையான தேர்ந்தெடுப்பான் ஆகும். ஒரு descendant என்பது மற்றொரு உறுப்பின் உள்ளே நேரடியாகவோ அல்லது மறைமுகமாகவோ உள்ள ஒரு உறுப்பு ஆகும். ஒரு descendant selector என்பது, பெற்றோர் உறுப்புக்கும் தொடர்ச்சியான உறுப்புக்கும் இடையே இடைவெளி வைத்து எழுதப்படுகிறது.

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

Descendant Selector

Descendant of Article

Descendant of Article

Descendant Selector

Descendant of Article

Child Selector

Child selector (>) என்பது மற்றொரு உறுப்பின் நேரடி குழந்தை உறுப்புகளை குறிவைக்கும் ஒரு தேர்ந்தெடுப்பான் ஆகும். அனைத்து தொடர்ச்சியான உறுப்புகளையும்

பொருத்தும் descendant selector-ஐ விட இது மிகவும் குறிப்பிட்டது.

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

Child Selector

Child of DIV

Child Selector

Descendant of DIV

Descendant of DIV

Dynamic Pseudo Classes

Dynamic pseudo-classes என்பது CSS தேர்விகள் ஆகும், இது எலிமெண்டுகளை அவற்றின் தற்போதைய நிலை அல்லது பயனர் தொடர்பின் அடிப்படையில் ஸ்டைலிங் செய்ய உதவுகின்றன. இவை "dynamic" என அழைக்கப்படுகின்றன, ஏனெனில் பயனர் தொடர்புகளுக்கான மறுமொழியாகவே அவை செயல்படுத்தப்படுகின்றன. சில பரவலாகப் பயன்படும் dynamic pseudo-classes இவை:

- **hover:** பயனர் எலிமென்டின் மீது மவுஸ் அசைவது அல்லது ஆவணத்துக்குள் குறி வைக்கிறதற்கான நிலை.
- **focus:** பயனர் ஒரு உள்ளீட்டு வயரின் மீது செல்வது அல்லது அதில் எழுதுவது போன்ற நிலையில்.
- **active:** பயனர் ஒரு உறுப்பு மீது கிளிக் செய்யும் போது பயன்படுத்தப்படும்.

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

Focus

General Input |

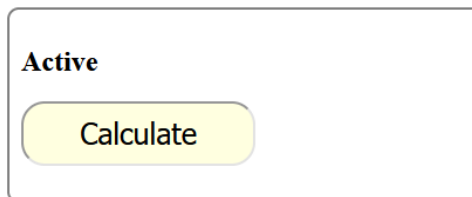
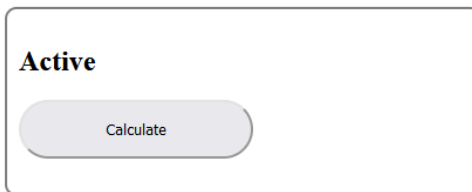
Focus

General Input

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

Class selector என்பது CSS இல் குறிப்பிட்ட ஒரு வகுப்பு பண்பு கொண்ட அனைத்து உறுப்புகளையும் தேர்ந்தெடுக்கப் பயன்படுகிறது. ஒரு **class selector** என்பது ஒரு புள்ளி (.) பின்னால் வகுப்பு பெயருடன் குறிக்கப்படுகிறது.

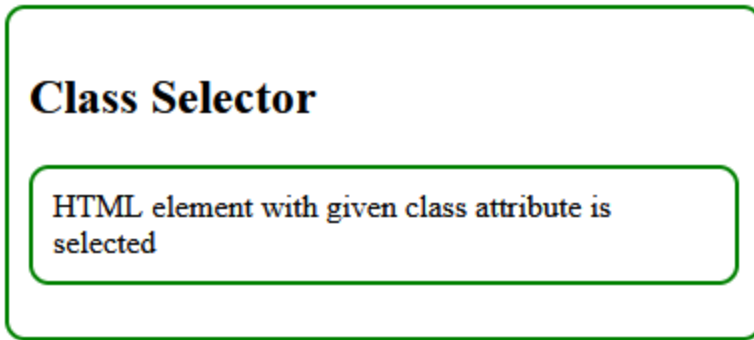
Syntax

```
.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



```
</style>
<div id="cont">
  <h2>ID Selector </h2>
  <p>
    HTML element with
  </p>
</div>
```

ID Selector

CSS ID selector என்பது HTML ஆவணத்தில் ஒற்றை, தனித்துவமான உறுப்பைத் தேர்ந்தெடுக்கப் பயன்படுகிறது. இது உறுப்பின் id பண்பு மூலம் அடையாளம் காணப்படுகிறது. ஒரு ID selector என்பது ஒரு ஹேஷ் சின்னம் (#) பின்னால் ID பெயருடன் குறிக்கப்படுகிறது.

Syntax

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

Example

```
<style>
#cont{
  width: auto;
  border: 4px dotted red;
  padding: 10px;
  border-radius: 10px;
  margin:5px;
}
```

Output

