

Selector's क्या होते हैं?

हम किसी भी HTML एलिमेंट को CSS में जिस नाम से हैंडल करते हैं उसको CSS Selector कहते हैं। जैसे हमने HTML में लिखा `<h1>` या कोई Division "`<div>`" बनाते हैं तो CSS में उसको `div{}` या `h1{}` से हैंडल करेंगे।

Selector's की 5 कैटेगरी होती हैं।

1. Simple Selectors
2. Combinator Selectors
3. Attribute Selectors
4. Pseudo-Class Selectors
5. Pseudo-Element Selectors

1. Simple Selectors: सिम्पल सेलेक्टर को भी 5 भागों में बांटा गया है।

- a) Element Selector
 - b) ID Selector
 - c) Class Selector
 - d) Universal Selector
 - e) Grouping Selector
- a) Element Selector:- मान लीजिये हमने अपने Document में `<h1>` या `<h2>` या कोई `<p>` पैराग्राफ बनाया है। अब यदि इनको CSS में हैंडल करना है तो इनको `h1{ }`, `h2{ }` और `p{ }` से हैंडल करेंगे। यानी जैसे HTML में tag हैं वैसे के वैसे लगते हैं तो ये कहलाते हैं Element Selector.
- b) ID Selector:- मान लीजिये हम कोई Division बनाते हैं और उसको कोई एक ID दे देते हैं जैसे `<div id="Box">` यहाँ पर हमने डिविजन लगाया और उसमें एक ID दी box तो CSS में जब हम ID हैंडल करते हैं तो `#box{ }` से करते हैं और उसकी स्टाइलिंग करते हैं। तो जो हम "#" लगाकर किसी tag को हैंडल करते हैं वो होते हैं ID Selector.
- c) Class Selector:- इसी प्रकार किसी डिविजन को हम class देते हैं जैसे:- `<div class="Box">` तो CSS में हम इसको हैंडल करेंगे `div.box{ }` या `.box{ }` से, तो जो हम "." (Dot) के साथ लेते हैं वो होता है class selector.
- d) Universal Selector:- यूनिवर्सल selector को हम CSS में शुरू में ही लगा देते हैं इसको हम `*{ }` (Star) से लिखते हैं इसमें हम जो भी स्टाइलिंग करते हैं वो CSS में सभी एलिमेंट्स में Apply हो जाती है। इसमें हम ज्यादातर Margin और padding को ही यूज करते हैं और आपने देखा होगा margin और padding खतम करने के लिए इनको हम 0 लिखते हैं जिससे पूरे CSS की margin और padding 0 हो जाती है। तो ये होता है universal selector.
- e) Grouping Selector:- मान लीजिये हम CSS में एक से ज्यादा HTML एलिमेंट्स को एक साथ हैंडल करते हैं जैसे हमने `<h1>`, `<h2>`, `<h3>` और `<p>` चारों को एक साथ हैंडल करना है तो हम लिखेंगे `h1,h2,h3,p{ }` और जो भी स्टाइल करेंगे तो उसका रिज़ल्ट सभी पर एक साथ आता है, और सभी की स्टाइल एक जैसी हो जाएगी।
2. Combinator Selector:- इसको भी 4 भागों में बांटा गया है।
- a) Descendant Selector (Space) डिसेंडेंट (वंशज)
 - b) Child Selector (`>`)
 - c) Adjacent Sibling Selector (`+`) अड्जसेंट
 - d) General Sibling Selector (`~`)

- a) Descendant Selector:- इसका मतलब है वंशज यानि कोई भी टेक्स्ट या पैराग्राफ चाहे वो डिविजन के पैराग्राफ हो या डिविजन के अंदर डिविजन के डिविजन के पैराग्राफ हो या उसके child पैराग्राफ हो सभी के स्टाइल एक साथ change होगी जैसे हम HTML में लिखते हैं

```
<div class="Box">
  <p>paragraph one</p>
  <p>paragraph two</p>
  <div class="inbox">
    <p>inbox paragraph one</p>
    <p>inbox paragraph two</p>
  </div>
```

इसको CSS हम निम्न प्रकार से लिखेंगे:

```
div.box p { }
```

तो बॉक्स पैराग्राफ हो या बॉक्स के अंदर के child पैराग्राफ हो सभी की स्टाइल एक साथ change हो जाएगी। ये होता है Descendant selector.

- b) Child Selector:- इसको समझने के लिए हम उधारण के लिए उपर वाली tag को ऐसे ही रहने देते हैं और CSS में जाकर हैंडल करते हैं तो इसके लिए हम CSS में लिखेंगे

```
div.box > p { }
```

अब जो स्टाइलिंग करेंगे उससे बॉक्स के अंदर जो direct child है उसी की स्टाइलिंग change होगी। और जो उसके अंदर तो पैराग्राफ लिया है "inbox" उसकी स्टाइल change नहीं होगी। ये होता है child selector.

- c) Adjacent Sibling Selector:- इसको समझने के लिए मैं एक उधारण लेता हूँ। मान लो हमने HTML में 3-4 पैराग्राफ लिखे जैसे:

```
<h1>Sample Paragraph</h1>
<p>Paragraph one</p>
<p>Paragraph two</p>
<p>Paragraph two</p>
```

और CSS में मैं लिखूँ: **h1 + p { }** तो **<h1>** (heading1) के तुरंत बाद वाले पैराग्राफ की स्टाइल ही change होगी। तो ये होता है adjacent sibling selector जिसका मतलब ही होता है सटा हुआ या आसन्न भाई।

- d) General Sibling Selector:- इसके अंतर्गत यदि हम **h1 + p { }** में "+" के स्थान पर "~" (Tilde) लगा देते हैं तो **<h1>** (heading1) के तुरंत बाद वाले जितने पैराग्राफ होंगे उन सभी की स्टाइल change हो जाएगी।

3. Attribute Selector:- इसको समझने के लिए ये जान लो की HTML के प्रत्येक element की कोई न कोई विशेषता होती है। जैसे: **<input>** tag में type. मान लो हमने HTML में लिखा

```
<h1>Sample Form</h1>
Name: <input type="text"><br>
Father Name: <input type="text"><br>
Mail ID: <input type="mail"><br>
```

अब इसके बाद हम CSS में लिखें `input { }` तो ये तो हो गया सिम्पल सेलेक्टर क्योंकि हम Tag को हैंडल कर रहे हैं। ओर यदि हम `input` के आगे square bracket में लिख दें `type='text'` जैसे: `input [ type='text' ] { }` तो ये हो जाएगा विशेषता यानि attribute selector क्योंकि हमने इसमें विशेषता जोड़ दी तो `input` के अंदर जितने थी text वाली attribute होंगी उन सभी की स्टाइल change हो जाएगी।

4. Pseudo-class selector:- इसका मतलब होता है मिथ्या या आभासी ये एक स्पेशल कटेगरी है। इससे हम किसी भी element के लिए एक स्पेशल स्टेट define कर सकते हैं। (h1: hover) जैसे ही हम उस element के उपर माऊस ले कर जाएँ तो उस पर किस प्रकार की स्टाइल अप्लाई हो। इसकी मदद से हम किसी भी element की (div.box p:first-child { }) बिना ID या बिना Class define किए उसके hidden address से भी हम उसके उपर स्टाइल अप्लाई कर सकते हैं। उधारण के लिए मैं एक डिविजन में कोई 3-4 पैराग्राफ बनाता हूँ

```
<div class="box">
    <p>paragraph one</p>
    <p>paragraph two</p>
    <p>paragraph three</p>
    <p>paragraph four</p>
```

अब मैं चाहता हूँ की इसके सिर्फ पहले वाले पैराग्राफ पर कोई स्टाइल अप्लाई हो जाए तो उसको हम CSS में इस प्रकार लिखेंगे

`div.box p: first-child{ }` तो इस प्रकार से ये इसका hidden एड्रेस हो गया। इसी प्रकार हम last-child या nth-child लिख सकते हैं।

Pseudo-Classes हमारे HTML element के लिए 2 प्रकार से काम करती हैं। पहला तो हमने सीखा की ये हमारे HTML element के hidden एड्रेस से उसको ढूँढ कर उस पर स्टाइल अप्लाई करने का काम करता है। example;- `div.container p:nth-child{ color:red;}` ओर दूसरा ये हमारे HTML element के स्पेशल स्टेट पर स्टाइल अप्लाई करने का काम करता है। example: `div.container p:होवर{ color:white; background-color:red;}` Pseudo-class काफी बड़ा है इसलिए इसको को चार भागों में बाँट लेते

For Linking	For Common	For Form	For Variable
:link :hover :visited :active :target	:first-child :last-child :nth-child :nth-last-child() :nth-last-of-type() :nth-of-type() :only-child :only-of-type :first-of-type :last- of-type :empty :not	:focus :checked :disabled :enabled :required :optional :in-range :out-of-range :read-only :read-write :valid :default :invalid	:root

पहला सीखते हैं लिंक , जिस प्रकार एक सहर को दूसरे शहर से जोड़ने के लिए रास्ते होते हैं या सड़के होती हैं जो एक शहर या गाँव को दूसरे शहर या गाँव से जोड़ते हैं इसी प्रकार link से हम एक वैबसाइट को दूसरी वैबसाइट से जोड़ सकते हैं या ये कहे की एक वैबसाइट को दूसरी वैबसाइट से link कर सकते हैं। ओर

लिंक हमेशा HTML में ही generate होता है ना की CSS में। CSS में हम उस link की स्टाइलिंग कर सकते हैं। ओर इसके लिए अलग अलग 4 स्टेज होती हैं।

1. Link:- इसमें किसी भी लिंक पर एक बात भी क्लिक नहीं किया गया होता ये लिंक की पहली condition होती है। इसको CSS में `a:link { color:red }` से हैंडल करते हैं।
Output: **Google**
Clicked लिंक - Google Unclicked लिंक - Google
2. Visited:- किसी लिंक पर एक बात क्लिक हो जाता है इसका मतलब उस पर visit की जा चुकी है वो visited लिंक होते हैं दोनों प्रकार के लिंक उपर दिखाये गए हैं। इसको CSS में हम `a:visited{ color:blue }` इस प्रकार हैंडल करेंगे।
3. Hover:- जब हम किसी भी लिंक के उपर अपना माऊस ले कर जाते हैं तो क्लिक करने से पहले माऊस ले जाते समय जो प्रतिक्रिया (action play) होती है उसको hover कहते हैं। ओर लिंक पर जो बदलाव हो रहे होते हैं उन्हें Hover Effect कहते हैं। इसको CSS में `a:hover { color:red }` से हैंडल करते हैं।
CSS tab:- `a:hover{ color:blue; }`
Output:- माऊस ले जाने से पहले - **Google**
माऊस ले जाने के बाद - **Google**
4. Active:- एक समय पर (At the time) जब हम किसी लिंक पर मौजूद होते हैं उसको एक्टिव लिंक कहते हैं। इसको CSS में `a:active { color:gray }` से हैंडल करते हैं।
5. Target:-

अधिक Example के साथ सीखने के लिए विडियो की मदद ले सकते हैं।

Part 2 Common Selectors

Common selector में सबसे पहले आता है “:first-child” इसको समझते हैं, जैसे हम अपने HTML में एक Division बनाते हैं “container” `<div class="container">` ओर इस डिविजन के अंदर हम एक ओर Division ओर बनाते हैं `<div class="box1">` ओर इसमें कुछ पैराग्राफ ओर Heading बनाते हैं जैसे:-

```
<div class="container">
  <div class="box1">
    <h1>Heading one</h1>
    <p>paragraph one box1</p>
    <p>paragraph two box1</p>
    <p>paragraph three box1</p>
    <p>paragraph four box1</p>
  </div>
<p>paragraph one container</p>
<p>paragraph two container</p>
  <div class="box2">
    <p>paragraph one box2</p>
    <p>paragraph two box2</p>
    <p>paragraph three box2</p>
```

```
<p>paragraph four box2</p>
</div>
```

</div>

इसमें हमने एक डिविजन लिया container इसमें दो Division Box1 व Box2 और दो paragraph बनाए हैं तो ये चारो container के child हो गए और container इनका parent हो गया इसी प्रकार box1 के पाँच child हो गए एक heading और चार paragraph और box2 के चारों पैराग्राफ उसके चार child हो गए।

:First-child

अब हम CSS में जाकर लिखें `div.container p:first-child{ color:red;}` तो ये container में देखेगा की “p” paragraph first child है। यहाँ आप देखेंगे की container का first child box1 है पैराग्राफ second child है उसके बाद देखेगा box1 में तो इसका भी first child h1 है और पैराग्राफ second child है। अब देखेगा दूसरे division box2 में तो यहाँ पर first child पैराग्राफ मिल जाएगा और जैसे ही इसको first child पैराग्राफ मिला तो ये इसकी स्टाइल change कर देगा बाकी किसी पर कुछ फर्क नहीं पड़ेगा क्योंकि वो देखेगा की first child पैराग्राफ कहाँ पर है और उसी की स्टाइल change हो जाएगी

Output:-

Heading one

Paragraph one box1

Paragraph two box1

Paragraph three box1

Paragraph four box1

Paragraph one container

Paragraph two container

Paragraph one box2

Paragraph two box2

Paragraph three box2

Paragraph four box2

इसकी output में आप देखोगे box2 में जो पहला पैराग्राफ है उसका ही कलर रेड हुआ है।

:Last-child

इस प्रकार हम लिखें `div.container p:last-child{ color:blue}` तो ये container में जाएगा और देखेगा की किस parent का last-child paragraph है। और इसको जिस parent का last child पैराग्राफ मिलेगा उसकी स्टाइल change कर देगा। यहाँ उपर दिये गए उधारण के अंतर्गत container का last child है box2 और box1 का व box2 का last child पैराग्राफ है तो इनका कलर ब्लू कर देगा।

Output:-

Heading one

Paragraph one box1

Paragraph two box1

Paragraph three box1

Paragraph four box1

Paragraph one container

Paragraph two container

Paragraph one box2

Paragraph two box2

Paragraph three box2

Paragraph four box2

:First-of-type

ये किस प्रकार काम करता है, अब हम कर रहे हैं `div.container p:first-of-type{ color:red;}` तो ये टाइप देखेगा यानि “p” टाइप का जो भी फ़र्स्ट होगा उसकी स्टाइल change कर देगा यहाँ पर container में “p” टाइप का इसको पहला पैराग्राफ मिल रहा है और box1 में भी “p” टाइप मिल रहा है और box2 में भी तो ये तीनों पहले पैराग्राफ का कलर रेड कर देगा।

Output:-

Heading one

Paragraph one box1

Paragraph two box1

Paragraph three box1

Paragraph four box1

Paragraph one container

Paragraph two container

Paragraph one box2

Paragraph two box2

Paragraph three box2

Paragraph four box2

:Last-of-type

CSS में हम लिखें `div.container p:last-of-type{ color:red;}` तो ये भी first-of-type की तरह ये देखेगा की “p” टाइप का last कोनसा है। और उसको ढूँढ कर उसका कलर chane कर देगा।

Output:-

Heading one

Paragraph one box1

Paragraph two box1

Paragraph three box1

Paragraph four box1

Paragraph one container

Paragraph two container

Paragraph one box2

Paragraph two box2

Paragraph three box2

Paragraph four box2

:nth-child

इसके लिए हम CSS में लिखते हैं `div.container p:nth-child(1){ color:red;}` इसमें हमको एक नंबर बताना पड़ता है जैसे इसमें मैंने नंबर दे दिया “(1)” तो ये जिस भी parent में पहला child पैराग्राफ होगा उसकी style को change कर देगा। यहाँ पर इसको पैराग्राफ first child के रूप में box2 में मिल रहा है क्योंकि container का पहला child box1 है और box1 का पहला child h1 है।

Output:-

Heading one

Paragraph one box1
Paragraph two box1
Paragraph three box1
Paragraph four box1

Paragraph one container
Paragraph two container

Paragraph one box2
Paragraph two box2
Paragraph three box2
Paragraph four box2

और यदि हम `:nth-child "(2)"` नंबर डाल दें तो ये सभी parent का दूसरा child पैराग्राफ खोजेगा और उनका कलर चंगे करेगा इसको कहाँ कहाँ मिला आप output में देखें।

Output:-

Heading one

Paragraph one box1
Paragraph two box1
Paragraph three box1
Paragraph four box1

Paragraph one container
Paragraph two container

Paragraph one box2
Paragraph two box2
Paragraph three box2
Paragraph four box2

इसी प्रकार 3,4,5 या 6 नंबर लिखेंगे तो उसी हिसाब से उस पैराग्राफ की स्टाइल change कर देगा। और यदि हम नंबर की बजाय even और odd लिखें तो ये प्रत्यक parent में even number या odd number पर जो पैराग्राफ मिलेगा उसकी स्टाइल change करेगा। even और odd को हम `"(2n)"` और `"(2n+1)"` करके भी लिख सकते हैं

:nth-last-child

यदि हम CSS में लिखें `div.container p:nth-last-child(1){ color:red;}` तो एक nth-child के तरह काम करेगा फरक इतना होगा की ये नीचे से काम करेगा जैसे nth-child नंबर से और even व odd से उपर से खोजता है वैसे ही ये नीचे से खोजेगा।

:nth-of-type

यदि हम CSS में `div.container p:nth-of-type(1){ color:red;}` लिख दें तो ये उस टाइप को को देखेगा ना की child को ये भी दोनों तरफ से काम करता है यदि हम टाइप कर दें `:nth-last-of-type` तो ये नीचे से काम करेगा वरना उपर से काम करेगा।

:only-of-type

यदि हम CSS में कोई नाम न बता कर केवल colon लगाकर only-of-type लिख दें जैसे `div.container :only-of-type{ color:red;}` तो ये HTML में single element को खोजेगा। यानि container में अपने type की कोई इकलोती child है या नहीं उपर दिये गए उधारण में ऐसी single टाइप सिर्फ "h1" है तो ये उसकी स्टाइल change कर देगा।

Output:-

Heading one

Paragraph one box1
Paragraph two box1
Paragraph three box1
Paragraph four box1

Paragraph one container
Paragraph two container

Paragraph one box2
Paragraph two box2
Paragraph three box2
Paragraph four box2

:only-child

यदि हम सीएसएस में लिखें `div.container :only-child{ color:red;}` तो ये container में जीतने भी parent हैं उनमें सिंगल child को खोजेगा हमारे उधारण में कोई भी single child नहीं है तो ये किसी की भी स्टाइल change नहीं करेगा।

:empty

इसके लिए हम `div.container :empty{ color:red;}` लिखते हैं। तो ये देखेगा की HTML में कोई element है या नहीं यदि कोई element empty मिलता है यानि जिसमें कुछ ना लिखा हो जैसे `<h1></h1>` या `<p></p>` तो ये उसकी स्टाइल change कर देगा। उधारण के लिए हम box2 में एक empty पैराग्राफ बनाए और उसकी स्टाइल नीचे दिये अनुसार कर दें।

```
div.container :empty{  
    color : red;  
    background-color : yellow;  
    width : 100px;  
    height : 100px;  
}
```

तो box2 की output कुछ इस प्रकार की होगी

Output:

Paragraph one box2
Paragraph two box2
Paragraph three box2
Paragraph four box2

:not

इसके लिए हम CSS में जा कर इसको box1 पर अप्लाई करते हैं इसके लिए टाइप करेंगे `div.box1 :not(p) {color:red;}` तो ये HTML में जाकर box1 में देखेगा की “p” पैराग्राफ, उन को छोड़ के जो दूसरे element हैं उनकी स्टाइल change कर देगा। box1 में पैराग्राफ के अलावा दूसरा एलिमेंट एक ही है वो है “h1” तो ये उसी का कलर change कर देगा।

5. Pseudo-Element selector:- ये HTML element के लिए CSS में एक specific area define करता है और उस एरिया पर स्टाइल अप्लाई करता है। उधारण के लिए हमारे पास कोई एक पैराग्राफ है जैसे:

```
<h1>Sample Paragraph</h1>
```


<p>Video provides a powerful way to help you prove your point. When you click Online Video, you can paste in the embed code for the video you want to add. You can also type a keyword to search online for the video that best fits your document.

</p>

तो इसके लिए हम CSS में First Letter (p::first-letter{}) ले लिए स्टाइलिंग कर सकते हैं या first line (p::first-line {}) के लिए स्टाइलिंग कर सकते हैं। इसकी ओर वैल्यू हैं। ::after, ::selection, ::before, ::placeholder etc।