Selector's क्या होते हैं?

हम किसी भी HTML एलिमंट को CSS मे जिस नाम से हैंडल करते हैं उसको CSS Selector कहते हैं। जैसे हमने HTML मे लिखा <h1> या कोई Division " " बनाते हैं तो CSS मे 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 सेलेकटोर को भी 5 भागों मे बंता 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 मे <h1> या <h2> या कोई <p> पैराग्राफ बनाया है। अब यदि इनहि को CSS मे हैंडल करना है तो इनको h1{ }, h2{ } ओर p{ } से हैंडल करेंगे। यानि जैसे HTML मे tag हैं वैसे के वैसे लगते हैं तो ये कहलाते हैं Element Selector.

b) ID Selector:- मान लीजिये हम कोई Division बनाते हैं ओर उसको कोई एक ID दे देते हैं जैसे यहाँ पर हमने डिविजन लगाया ओर उसमे एक ID दी box तो CSS मे जब हम ID हैंडल करते हैं तो #box{ } से करते हैं ओर 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:- > तो CSS मे 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:- यूनिवरसल selector को हम CSS मे शुरू मे ही लगा देते हैं इसको हम *{ } (Star) से लिखते हैं इसमे हम जो भी स्टाइलिंग करते हैं वो CSS मे सभी एलिमेंट्स मे Apply हो जाती है। इसमे हम ज़्यादातर Margin ओर padding को ही यूज करते हैं ओर आपने देखा होगा margin ओर padding खतम करने के लिए इनको हम 0 लिखते हैं जिससे पूरे css की margin ओर padding 0 हो जाती है। तो ये होता है universal selector.

e) Grouping Selector:- मान लीजिये हम CSS मे एक से ज्यादा HTML एलिमेंट्स को एक साथ हैंडल करते हैं जैसे हमने <h1>, <h2>, <h3> ओर <p> चारों को एक साथ हैंडल करना है तो हम लिखेंगे h1,h2,h3,p{ } ओर जो भी स्टाइल करेंगे तो उसका रिज़ल्ट सभी पर एक साथ आता है, ओर सभी की स्टाइल एक जैसी हो जाएगी।

2. Combinator Selector:- इसको भी 4 भागों मे 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 मे लिखते है

```
<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 ओर CSS मे जाकर हैंडल करते हैं तो इसके लिए हम CSS मे लिखेंगे

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

अब जो स्टाइलिंग करेंगे उससे बॉक्स के अंदर जो direct child है उसी की स्टाइलिंग change होगी। ओर 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 मे 3-4 पैराग्राफ लिखे जैसे:

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

    <p>Paragraph one</p>

    <p>Paragraph two</p>

    <p>Paragraph two</p>
```

ओर CSS मे मैं लिखू: **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 अंतेर्गत यदि हम **h1 + p { }** मे "+" 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 प्रत्यक element की कोई न कोई विशेषता होती है। जैसे: <input> tag मे type. मान लो हमने HTML मे लिखा

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

अब इसके बाद हम CSS मे लिखें input { } तो ये तो हो गया सिम्पल सेलेकटोर क्योंकि हम Tag को हैंडल कर रहे हैं। ओर यदि हम input के आगे squre bracket मे लिख दें type='text' जैसे: input [type='text'] { } तो ये हो जाएगा विशेषता यानि attribute selector क्योंकि हमने इसमे 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 मे 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 मे इस प्रकार लिखेंगे

div.box p: first-child{ } तो इस प्रकार से ये इसका hidden एड्रैस 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 एड्रैस से उसको ढूंढ 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 भागो मे बाँट लेते

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

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

1. Link:- इसमे किसी भी लिंक पर एक बात भी क्लिक नहीं किया गया होता ये लिंक की पहली condition होती है। इसको CSS मे 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 मे 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 उन्हे Hover Effect कहते हैं। इसको CSS मे 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 मे 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 मे सबसे पहले आता है ":first-child" इसको समझते हैं,जैसे हम अपने HTML मे एक Division बनाते हैं "container" ओर इस डिविजन के अंडर हम एक ओर Division ओर बनाते हैं ओर इसमे 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>
```

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

:First-child

अब हम CSS मे जाकर लिखें div.container p:first-child{ color:red;} तो ये container मे 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 मे तो इसका भी first child h1 है ओर पैराग्राफ second child है। अब देखेगा दूसरे division box2 मे तो यहाँ पर first child पैराग्राफ मिल जाएगा ओर 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 ओर 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 मे आप देखोगे box2 मे जो पहला पैराग्राफ है उसका ही कलर रेड हुआ है।

:Last-child

इस प्रकार हम लिखें div.container p:last-chld{ color:blue} तो ये container मे जाएगा ओर देखेगा की किस parent का last-child paragraph है। ओर 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 ओर 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 करगे 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 मे “p” टाइप का इसको पहला पैराग्राफ मिल रहा है ओर box1 मे भी “p” टाइप मिल रहा है ओर box2 मे 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 मे 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। ओर उसको ढूंढ 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 मे लिखते हैं div.container p:nth-child(1){ color:red;} इसमे हमको एक नंबर बताना पड़ता है जैसे इसमे मैंने नंबर दे दिया “(1)” तो ये जिस भी parent मे पहला child पैराग्राफ होगा उसकी style को change कर देगा। यहाँ पर इसको पैराग्राफ first child के रूप मे box2 मे मिल रहा है क्योंकि container का पहला child box1 है ओर 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

ओर यदि हम :nth-child "(2)" नंबर डाल दें तो ये सभी parent का दूसरा child पैराग्राफ खोजेगा ओर उनका कलर चंगे करेगा इसको कहाँ कहाँ मिला आप output मे 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। ओर यदि हम नंबर की बजाय even ओर odd लिखें तो ये प्रत्यक parent मे even number या odd number पर जो पैराग्राफ मिलेगा उसकी स्टाइल change करेगा। even ओर odd को हम "(2n)" ओर "(2n+1)" करके भी लिख सकते हैं

:nth-last-child

यदि हम CSS मे 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 ओर even व odd से उपर से खोजता है वैसे ही ये नीचे से खोजेगा।

:nth-of-type

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

:only-of-type

यदि हम CSS मे कोई नाम न बता कर केवल colon लगाकर only-of-type लिख दे जैसे div.container :only-of-type{ color:red;} तो ये HTML मे single element को खोजेगा। यानि container मे अपने type की कोई इकलोती child है या नहीं उपर दिये गए उधारण मे 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 मे लिखें div.container :only-child{ color:red;} तो ये container मे जीतने भी parent हैं उनमे सिंगल child को खोजेगा हमारे उधारण मे 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 मे कोई element है या नहीं यदि कोई element empty मिलता है यानि जिसमे कुछ ना लिखा हो जैसे <h1></h1> या तो ये उसकी स्टाइल change कर देगा। उधारण के लिए हम box2 मे एक empty पैराग्राफ बनाए ओर 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 मे जा कर इसको box1 पर अप्लाई करते हैं इसके लिए टाइप करेंगे div.box1 :not(p) {color:red;} तो ये HTML मे जाकर box1 मे देखेगा की “p” पैराग्राफ, उन को छोड़ के जो दूसरे element हाँ उनकी स्टाइल change कर देगा। box1 मे पैराग्राफ के अलावा दूसरा एलिमंट एक ही है वो है “h1” तो ये उसी का कलर change कर देगा।

5. Pseudo-Element selector:- ये HTML element के लिए CSS मे एक specific area define करता है ओर उस एरिया पर स्टाइल अप्लाई करता है। उधारण के लिए हमारे पास कोई एक पैराग्राफ है जैसे:
   <h1>Sample Paragraph</h1>

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.

तो इसके लिए हम CSS मे 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।