

## **CSS Selectors**

- “Selector” is a syntax to select, it is used to select the desired elements in the webpage.
- Selector is nothing but reusable styles.
- When we use a selector, the browser searches the entire webpage for the matching elements and returns the matching elements; and we apply styles only for those matching elements.
- First, we have to select the element/elements, and then only we can apply some styles to it.
- Selectors we can use in internal & external CSS only.

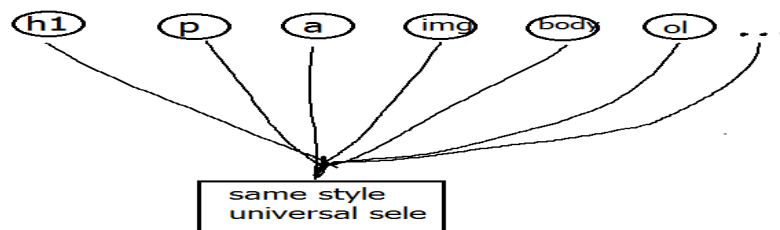
### **Types of Selectors:**

1. Universal selector
2. Tag selector
3. ID selector
4. Class selector
5. Compound selector
6. Grouping selector
7. Child selector
8. Direct Child selector
9. Attribute selector
10. Pseudo selector
11. Pseudo elements
- Etc...
- 12.**

## Universal selector

- It selects all the tags in the webpages, include html, body, head etc...
- Used to define common properties for all tags (global styles).

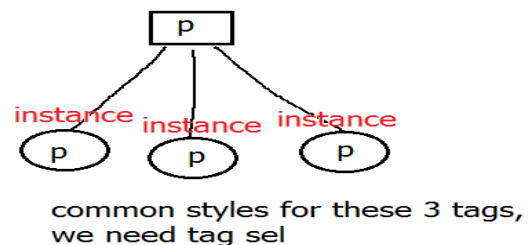
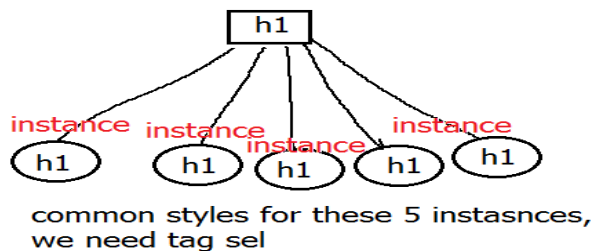
Syn: **\*{**  
    **property:value;**  
    ...  
    **}**



## Tag selector

- It is used to define common styles for all the instances of the specified tag.
- We can define any no.of tag selectors

Syn: **tag-name{**  
    **property:value;**  
    ...  
    **}**



## ID selector

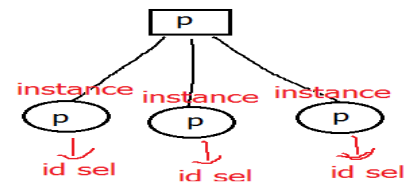
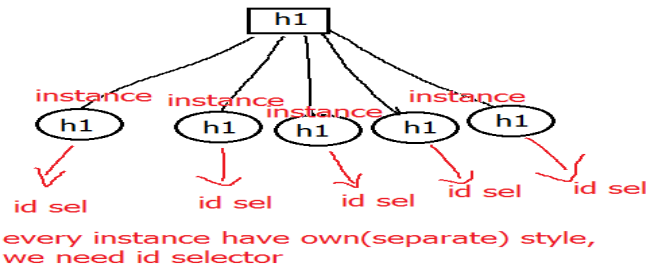
- It selects all the instances of the specified tag, means it used to specify common attributes of multiple tags.
- Id is "identification name"
- Id should be unique in the web page.
- # is symbol of ID selector.

Syn: **#id-name{**  
    **property:value;**  
    ...  
    **}**

Mapping Syn:

`<tag1 id="id-name" ...>`

`<tag2 id="id-name" ...>`



### Class selector

- It selects one or more elements, based on the class name, means it used to specify common attributes of multiple tags.
- We use same class for similar elements/tags.
- “.” is symbol of Class selector.
- A tag they can use multiple classes

Syn: `.ClassName{`  
    **property:value;**  
    ...  
}

Calling Syn:

`<tag1 class="ClassName" ...>`

`<tag2 class="ClassName1 ClassName2 ..." >`

### Compound selector

- It selects the instances of specific tag, which have specified class name.
- Its combination of “tag” selector and “class” selector.

Syn: `tagname.classname{`  
    **property:value;**  
    ...  
}  
`Tag#ID{`  
    **property:value;**  
    ...  
}

Calling Syn:

`<tag class="ClassName" ...>`

### Grouping selector

- It selects the specified group of tags/elements, means to set common properties for different tags.
- “,” is the symbol of grouping selector.

**Syn:** tag1, tag2, tag3, ...{  
    **property:value;**  
    ...  
}

### Child selector

- It selects all the child tags/elements (including grandchild) of the specified parent tag,
- “space” is the symbol of child selector.

**Syn:** p-tag ch-tag {  
    **property:value;**  
    ...  
}

### Direct Child selector

- It selects only the direct child tags/elements (excluding the grandchild) of the specified parent tag,
- “>” is the symbol of direct child selector.

**Syn:** p-tag >ch-tag {  
    **property:value;**  
    ...  
}

### Attribute selector

- It selects all the tags/elements that are having specified attribute,
- “[ ]” is the symbol of attribute selector.

**Syn:** tag [attribute="value"] {  
    **property:value;**  
    ...  
}

## Pseudo classes

- All pseudo selector/classes should be represented with ":"symbol.

### link selector

- It used to change the default look of a hyperlinks.

```
Syn: a:link{  
    property:value;  
    ...  
}
```

### visited selector

- It used to change the default look of an already opened/visited hyperlinks.

```
Syn: a:visited{  
    property:value;  
    ...  
}
```

### active selector

- It used to change the default look of a hyperlink @the moment of mouse clicked.

```
Syn: a:active{  
    property:value;  
    ...  
}
```

### Hover selector

- It applies the style only when the user places the mouse pointer on the element, at run time.
- It automatically removes the style, if mouse pointer is coming out of element (now id displaying with original settings).

```
Syn: tag:hover{  
    property:value;  
    ...  
}
```

### focus selector

- It applies the style only when the focus (cursor) is comes on to the element.
- It automatically removes the style, if mouse pointer wascoming out of element (now element displaying with original styles).

```
Syn: tag:focus{  
    property:value;  
    ...  
}
```

}

**Note:** it is applicable only for which control/element allows cursor.

### first-child selector

- It is used to apply styles for first positioned child element.

**Syn:** `:first-child {  
 property:value;  
 ...  
}`

### last-child selector

- It is used to apply styles for last positioned child element.

**Syn:** `:last-child {  
 property:value;  
 ...  
}`

### nth-child selector

- It is used to change the style of child html elements
- Even is a keyword, it represents all even positioned child elements

**Syn:** `:nth-child(even){  
 property:value;  
 ...  
}`

- Odd is a keyword, it represents all odd positioned child elements.

**Syn:** `:nth-child(odd){  
 property:value;  
 ...  
}`

- N represents n<sup>th</sup> positioned child element.

**Syn:** `td:nth-child(N){  
 property:value;  
 ...  
}`

## CSS Pseudo Elements

Selector	Meaning
----------	---------

<b>::after</b>	Insert content(either text or image) after tag/element
----------------	--

<b>::before</b>	Insert content(either text or image) before tag/element
-----------------	---

<b>::first-letter</b>	Selects the first letter of every tag/element
-----------------------	---

<b>::first-line</b>	Selects the first line of every element
---------------------	---

<b>::selection</b>	Selects the portion of an element that is selected by a user
--------------------	--

**::file-selector-button** selects file control button of an <input> of type="file"

**::marker** CSS pseudo-element selects the marker box of a list item, which typically contains a bullet or number. It works on any element or pseudo-element set to display: list-item, such as the <li> and <summary> elements.

**::placeholder** CSS pseudo-element represents the placeholder text in an <input> or <textarea> element.

### **Note:**

the double colon notation:: after **Vs** :after

The double colon replaced the single-colon notation for pseudo-elements in CSS3. distinguish between pseudo-classes and pseudo-elements.

The single-colon was used for both pseudo-classes and pseudo-elements in CSS2 and CSS1 versions.

In css1 & css2 → :selector                      :element

Since css3 → :selector                                      ::element

## CSS Precedence

- Ccss styles are applied in the following order (lower priority to higher priority).
- The higher priority style overrides the same property's value of the lower priority.
  1. Browser default style
  2. Tag selector
  3. Direct child selector
  4. Child selector
  5. Class selector
  6. Attribute selector
  7. ID selector