CSS - Reference

Simple selectors:

ID selector

Selects elements based on the value of the `id` attribute.
Syntax: `#idname`
Example: `#gb78` will match the elements with ID "gb78".

Class selector

Selects elements that have the given `class` attribute.
Syntax: `.classname`
Example: `.gb_0` will match the elements with class "gb_0".

Attribute selector

Selects elements that have the given attribute or based on the value of the given attribute.
Syntax: `[attr]` `[attr=value]`
Example: `[href]` will match the elements having `href` attribute set (to any value).
`[href="https://play.google.com/?hl=en&tab=w8"]` will match elements having `href` attribute set to given value.

Universal selector

Selects all elements. Optionally, it may be restricted to a specific namespace or to all namespaces.
Syntax: `*`
`ns|*`
Example: `*` will match all the elements of the document.

Wildcards:

`[attr^=value]` Represents elements with an attribute name of *attr* whose value is prefixed (preceded) by *value*.
Example:
```
[href^="https://play.google.com/"]
a[href^="https://play.google.com/"]
```

`[attr*=value]` Represents elements with an attribute name of *attr* whose value contains at least one occurrence of *value* within the string.
Example:
```
[href*="play.google.com"]
a[href*="play.google.com"]
```

`[attr$=value]` Represents elements with an attribute name of *attr* whose value is suffixed (followed) by *value*.
Example:
```
[href$="hl=en&tab=w8"]
a[href$="hl=en&tab=w8"]
```

[*attr~=value*] Represents elements with an attribute name of *attr* whose value is a whitespace-separated list of words, one of which is exactly *value*.
Example: `[title~="apps"]`
`a[title~="apps"]`

Multiple Selectors:

[*attr1=value*] [*attr2=value*] Represents elements with attributes of *attr1* AND *attr2* along with given values.
Example: `[class="gb_O"][data-pid="78"]`

Combinators:

Child combinator

The > combinator selects nodes that are direct children of the first element.
Syntax: `A > B`
Example: `a[href*="play.google"]>span.gb_W` will match all `span.gb_W` elements that are nested directly inside a `a[href*="play.google"]` element.

Descendant combinator

The (space) combinator selects nodes that are descendants(children, grandchildren and more) of the first element.
Syntax: `A B`
Example: `ul.gb_da.gb_6 a[href*="play.google"]` will match all `a[href*="play.google"]` elements that are inside a `ul.gb_da.gb_6` element.

Adjacent sibling combinator

The + combinator selects adjacent siblings. This means that the second element directly follows the first, and both share the same parent.
Syntax: `A + B`
Example: `ul.gb_da.gb_6+a` will match all `a` elements that directly follow an `ul.gb_da.gb_6`.

General sibling combinator

The ~ combinator selects siblings. This means that the second element follows the first (though not necessarily immediately), and both share the same parent.
Syntax: `A ~ B`
Example: `ul.gb_da.gb_6~ul` will match all `ul` elements that follow a `ul.gb_da.gb_6`.

For Further Reading: https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Selectors