SIMPLE SELECTORS – used to define the element(s) to be styled.

1. **Type Selector**

Use: Selects all the elements that match the specified element name.

Syntax: p

Example: p will match any <p> element.

1. **Class Selector**

Use: Selects all the elements that have the specified class attribute preceded by “.”

Syntax: .classname

Example: .classname will match all elements that have a class name, “classname”

1. **ID Selector**

Use: Selects an element that has the specified id attribute preceded by “#”

Syntax: #idname

Example: #idname will match all elements that have a class name, “idname”

1. **Universal Selector**

Use: Selects all elements.

Syntax: \*

Example: div \* selects all elements within the <div>

1. **Attribute Selector**

Use: Selects all elements matching the given attribute name

Syntax: a[attributename]

Example: a[attributename] selects an element with an attribute name attributename

1. **Pseudo-Classes**

Use: Selects an element specifying a special state.

* 1. Dynamic Pseudo-Classes – classify elements on characteristics other than their name, attributes, or content.
     1. Link Pseudo-Classes – displays links depending on whether it is visited or not.
     2. User Action Pseudo-Classes – changes rendering in response to user actions
  2. Target Pseudo-Classes – selects an element with anchor identifier
  3. Language Pseudo-Classes – selects an element with a specified language  
     Syntax: :lang([language])  
     Example: :lang(en) selects a n element with language en(English).
  4. UI Element States Pseudo-Classes – selects an element depending on the state of that element
  5. Structur­al Pseudo-Classes – selects elements based on extra information that cannot be represented by other simple selectors or combinators

Source:

https://www.w3.org/TR/selectors-3/#the-link-pseudo-classes-link-and-visited

COMBINATORS – shows the relationship between simple selectors

1. **Descendant Combinator ( white space )**

Use: Selects nodes that are descendants of the first element

Syntax: selector1 selector2

Example: div p will match all <p> elements inside <div> elements

1. **Child Combinator ( > )**

Use: Selects nodes that are immediate children of the first element

Syntax: selector1 > selector2

Example: div > p will match all <p> elements that are immediate children of a <div> element

1. **Sibling Combinator** 
   1. **Adjacent Sibling Combinator ( + )**

Use: Selects the element that is an immediate sibling of a specified element.

Syntax: selector1 + selector2

Example: img + p will match all <p> elements that are immediate siblings of an <img> element

* 1. **General Sibling Combinator** **( ~ )**

Use: Selects all elements that are siblings of a specified element.

Syntax: selector1 ~ selector2

Example: img ~ p will match all <p> elements that share the same parent with an <img> element