**1.css selector, pseudo class, pseudo elements: -**

A CSS selector selects the HTML element(s) you want to style.

All CSS Simple Selectors

|  |  |  |
| --- | --- | --- |
| **Selector** | **Example** | **Example description** |
| [#*id*](https://www.w3schools.com/cssref/sel_id.asp) | #firstname | Selects the element with id="first name" |
| [.*class*](https://www.w3schools.com/cssref/sel_class.asp) | .intro | Selects all elements with class="intro" |
| [*element.class*](https://www.w3schools.com/cssref/sel_element_class.asp) | p.intro | Selects only <p> elements with class="intro" |
| [\*](https://www.w3schools.com/cssref/sel_all.asp) | \* | Selects all elements |
| [*element*](https://www.w3schools.com/cssref/sel_element.asp) | p | Selects all <p> elements |
| [*element,element,..*](https://www.w3schools.com/cssref/sel_element_comma.asp) | div, p | Selects all <div> elements and all <p> elements |

All CSS Pseudo Classes: -

A pseudo-class is used to define a special state of an element.

For example, it can be used to:

* Style an element when a user mouses over it
* Style visited and unvisited links differently
* Style an element when it gets focus

|  |  |  |
| --- | --- | --- |
| **Selector** | **Example** | **Example description** |
| [:active](https://www.w3schools.com/cssref/sel_active.asp) | a:active | Selects the active link |
| [:checked](https://www.w3schools.com/cssref/sel_checked.asp) | input:checked | Selects every checked <input> element |
| [:disabled](https://www.w3schools.com/cssref/sel_disabled.asp) | input:disabled | Selects every disabled <input> element |
| [:empty](https://www.w3schools.com/cssref/sel_empty.asp) | p:empty | Selects every <p> element that has no children |
| [:enabled](https://www.w3schools.com/cssref/sel_enabled.asp) | input:enabled | Selects every enabled <input> element |
| [:first-child](https://www.w3schools.com/cssref/sel_firstchild.asp) | p:first-child | Selects every <p> elements that is the first child of its parent |
| [:first-of-type](https://www.w3schools.com/cssref/sel_first-of-type.asp) | p:first-of-type | Selects every <p> element that is the first <p> element of its parent |
| [:focus](https://www.w3schools.com/cssref/sel_focus.asp) | input:focus | Selects the <input> element that has focus |
| [:hover](https://www.w3schools.com/cssref/sel_hover.asp) | a:hover | Selects links on mouse over |
| [:in-range](https://www.w3schools.com/cssref/sel_in-range.asp) | input:in-range | Selects <input> elements with a value within a specified range |
| [:invalid](https://www.w3schools.com/cssref/sel_invalid.asp) | input:invalid | Selects all <input> elements with an invalid value |
| [:lang(*language*)](https://www.w3schools.com/cssref/sel_lang.asp) | p:lang(it) | Selects every <p> element with a lang attribute value starting with "it" |
| [:last-child](https://www.w3schools.com/cssref/sel_last-child.asp) | p:last-child | Selects every <p> elements that is the last child of its parent |
| [:last-of-type](https://www.w3schools.com/cssref/sel_last-of-type.asp) | p:last-of-type | Selects every <p> element that is the last <p> element of its parent |
| [:link](https://www.w3schools.com/cssref/sel_link.asp) | a:link | Selects all unvisited links |
| [:not(selector)](https://www.w3schools.com/cssref/sel_not.asp) | :not(p) | Selects every element that is not a <p> element |
| [:nth-child(n)](https://www.w3schools.com/cssref/sel_nth-child.asp) | p:nth-child(2) | Selects every <p> element that is the second child of its parent |
| [:nth-last-child(n)](https://www.w3schools.com/cssref/sel_nth-last-child.asp) | p:nth-last-child(2) | Selects every <p> element that is the second child of its parent, counting from the last child |
| [:nth-last-of-type(n)](https://www.w3schools.com/cssref/sel_nth-last-of-type.asp) | p:nth-last-of-type(2) | Selects every <p> element that is the second <p> element of its parent, counting from the last child |
| [:nth-of-type(n)](https://www.w3schools.com/cssref/sel_nth-of-type.asp) | p:nth-of-type(2) | Selects every <p> element that is the second <p> element of its parent |
| [:only-of-type](https://www.w3schools.com/cssref/sel_only-of-type.asp) | p:only-of-type | Selects every <p> element that is the only <p> element of its parent |
| [:only-child](https://www.w3schools.com/cssref/sel_only-child.asp) | p:only-child | Selects every <p> element that is the only child of its parent |
| [:optional](https://www.w3schools.com/cssref/sel_optional.asp) | input:optional | Selects <input> elements with no "required" attribute |
| [:out-of-range](https://www.w3schools.com/cssref/sel_out-of-range.asp) | input:out-of-range | Selects <input> elements with a value outside a specified range |
| [:read-only](https://www.w3schools.com/cssref/sel_read-only.asp) | input:read-only | Selects <input> elements with a "readonly" attribute specified |
| [:read-write](https://www.w3schools.com/cssref/sel_read-write.asp) | input:read-write | Selects <input> elements with no "readonly" attribute |
| [:required](https://www.w3schools.com/cssref/sel_required.asp) | input:required | Selects <input> elements with a "required" attribute specified |
| [:root](https://www.w3schools.com/cssref/sel_root.asp) | root | Selects the document's root element |
| [:target](https://www.w3schools.com/cssref/sel_target.asp) | #news:target | Selects the current active #news element (clicked on a URL containing that anchor name) |
| [:valid](https://www.w3schools.com/cssref/sel_valid.asp) | input:valid | Selects all <input> elements with a valid value |
| [:visited](https://www.w3schools.com/cssref/sel_visited.asp) | a:visited | Selects all visited links |

All CSS Pseudo Elements: -

A CSS pseudo-element is used to style specified parts of an element.

For example, it can be used to:

* Style the first letter, or line, of an element
* Insert content before, or after, the content of an element

|  |  |  |
| --- | --- | --- |
| **Selector** | **Example** | **Example description** |
| [::after](https://www.w3schools.com/cssref/sel_after.asp) | p::after | Insert content after every <p> element |
| [::before](https://www.w3schools.com/cssref/sel_before.asp) | p::before | Insert content before every <p> element |
| [::first-letter](https://www.w3schools.com/cssref/sel_firstletter.asp) | p::first-letter | Selects the first letter of every <p> element |
| [::first-line](https://www.w3schools.com/cssref/sel_firstline.asp) | p::first-line | Selects the first line of every <p> element |
| [::selection](https://www.w3schools.com/cssref/sel_selection.asp) | p::selection | Selects the portion of an element that is selected by a user |

**HTML Forms:-**

An HTML form is used to collect user input. The user input is most often sent to a server for processing.

## Form controls and form fields [:-](https://web.dev/learn/forms/#form-controls-and-form-fields)

A **form control** is an element that enables user interaction and data entry or selection: an <input>, <select>, <textarea> or <button>.

Sometimes **form field** is used to refer to form controls, in particular elements for text entry: <input> and <textarea>.

**array iteration method: -**

Array iteration methods operate on every array item.

Function(value,index,arr)

# **Array Methods Reference**

The following table lists all the Array methods.

| Method | Description |
| --- | --- |
| concat() | Returns new array by combining values of an array that is specified as parameter with existing array values. |
| every() | Returns true or false if every element in the specified array satisfies a condition specified in the callback function. Returns false even if single element does not satisfy the condition. |
| filter() | Returns a new array with all the elements that satisfy a condition specified in the callback function. |
| forEach() | Executes a callback function for each elements of an array. |
| indexOf() | Returns the index of the first occurrence of the specified element in the array, or -1 if it is not found. |
| join() | Returns string of all the elements separated by the specified separator |
| lastIndexOf() | Returns the index of the last occurrence of the specified element in the array, or -1 if it is not found. |
| map() | Creates a new array with the results of calling a provided function on every element in this array. |
| pop() | Removes the last element from an array and returns that element. |
| push() | Adds one or more elements at the end of an array and returns the new length of the array. |
| reduce() | Pass two elements simultaneously in the callback function (till it reaches the last element) and returns a single value. |
| reduceRight() | Pass two elements simultaneously in the callback function from right-to-left (till it reaches the last element) and returns a single value. |
| reverse() | Reverses the elements of an array. Element at last index will be first and element at 0 index will be last. |
| shift() | Removes the first element from an array and returns that element. |
| slice() | Returns a new array with specified start to end elements. |
| some() | Returns true if at least one element in this array satisfies the condition in the callback function. |
| sort() | Sorts the elements of an array. |
| splice() | Adds and/or removes elements from an array. |
| toString() | Returns a string representing the array and its elements. |
| unshift() | Adds one or more elements to the front of an array and returns the new length of the array. |

**string method:-**

String methods help you to work with strings.

## String Methods

| Method | Description |
| --- | --- |
| charAt(position) | Returns the character at the specified position (in Number). |
| charCodeAt(position) | Returns a number indicating the Unicode value of the character at the given position (in Number). |
| concat([string,,]) | Joins specified string literal values (specify multiple strings separated by comma) and returns a new string. |
| indexOf(SearchString, Position) | Returns the index of first occurrence of specified String starting from specified number index. Returns -1 if not found. |
| lastIndexOf(SearchString, Position) | Returns the last occurrence index of specified SearchString, starting from specified position. Returns -1 if not found. |
| localeCompare(string,position) | Compares two strings in the current locale. |
| match(RegExp) | Search a string for a match using specified regular expression. Returns a matching array. |
| replace(searchValue, replaceValue) | Search specified string value and replace with specified replace Value string and return new string. Regular expression can also be used as searchValue. |
| search(RegExp) | Search for a match based on specified regular expression. |
| slice(startNumber, endNumber) | Extracts a section of a string based on specified starting and ending index and returns a new string. |
| split(separatorString, limitNumber) | Splits a String into an array of strings by separating the string into substrings based on specified separator. Regular expression can also be used as separator. |
| substr(start, length) | Returns the characters in a string from specified starting position through the specified number of characters (length). |
| substring(start, end) | Returns the characters in a string between start and end indexes. |
| toLocaleLowerCase() | Converts a string to lower case according to current locale. |
| toLocaleUpperCase() | Converts a sting to upper case according to current locale. |
| toLowerCase() | Returns lower case string value. |
| toString() | Returns the value of String object. |
| toUpperCase() | Returns upper case string value. |
| valueOf() | Returns the primitive value of the specified string object. |

browser object model: -

he Browser Object Model (BOM) is the core of JavaScript in web browser. It consists of objects that provide browser functionality independent of the web page content.

### 1. **Window Object**

* window.open() - open a new window
* window.close() - close the current window
* window.moveTo() - move the current window
* window.resizeTo() - resize the current window

# 2.**Window Screen**

* screen.width
* screen.height
* screen.availWidth
* screen.availHeight
* screen.colorDepth
* screen.pixelDepth

# 3.**Window Location**

* window.location.href returns the href (URL) of the current page
* window.location.hostname returns the domain name of the web host
* window.location.pathname returns the path and filename of the current page
* window.location.protocol returns the web protocol used (http: or https:)
* window.location.assign() loads a new document

# 4.**Window History**

* history.back() - same as clicking back in the browser
* history.forward() - same as clicking forward in the browser

# **5**.**Window Navigator**

* navigator.appName
* navigator.appCodeName
* navigator.platform

# 6.**Popup Boxes**

## Alert Box

## Confirm Box

**7.Cokkies**

Cookies are data, stored in small text files, on your computer.

When a web server has sent a web page to a browser, the connection is shut down, and the server forgets everything about the user.

Cookies were invented to solve the problem "how to remember information about the user":

* When a user visits a web page, his/her name can be stored in a cookie.
* Next time the user visits the page, the cookie "remembers" his/her name.

**bootstrap grid explanation: -**

## Grid Classes

The Bootstrap grid system has four classes:

* xs (for phones - screens less than 768px wide)
* sm (for tablets - screens equal to or greater than 768px wide)
* md (for small laptops - screens equal to or greater than 992px wide)
* lg (for laptops and desktops - screens equal to or greater than 1200px wide)

**block level, inline, inline block in html:-**

# Block and Inline Elements

In HTML 4.01, there were two basic categories of HTML elements:

* Block level elements
* Inline elements

## Block Level Elements

Block level elements take up as much space as possible by default. Each block level element will start a new line on the page, stacking down the page. In addition to stacking vertically, block level elements will also take up as much horizontal space as possible.

The p element is an example of a block level element. Each new paragraph tag will appear on its own line vertically. Paragraphs with longer content will stretch all the way to the edge of the page.

Examples of block level elements:

* <p>
* <ol>, <ul>, <dl>
* All headings
* <article>, <section>, <div>

## Inline Elements

Inline elements display in a line. They do not force the text after them to a new line.

An anchor (or link) is an example of an inline element. You can put several links in a row, and they will display in a line.

Examples of inline elements:

* <a>
* <strong>, <em>, <b>, <i>, <q>, <mark>
* <span>

**local storage session storage: -**

**let var constant difference: -**

### Difference between Var, Let and Const

The following table briefs the difference between let and var and const in javascript:

| **var** | **let** | **const** |
| --- | --- | --- |
| var has the function or global scope. | let have the block scope. | const variable has the block scope. |
| It gets hoisted to the top of its scope and initialized undefined. | It also got hoisted to the top of its scope but didn't initialize. | It also got hoisted to the top of its scope but didn't initialize. |
| It can be updated or re-declared. | It can only be updated and can't be re-declared. | It can't be updated or re-declared. |
| It's an old way to declare a variable. | It's a new way to declare variables introduced in ES6. | It's also a new way to declare a variable, which introduces in ES6. |





 