## Selectors in CSS

CSS selectors define the pattern to select elements to which a set of CSS rules are then applied.

1) .class .intro Selects all elements with class="intro"

<html>

<head>

<style>

.intro {

background-color: yellow;

}

</style>

</head>

<body>

<h1>Demo of the .class selector</h1>

<div class="intro">

<p>My name is Rajat Sharma.</p>

<p>I live in Delhi.</p>

</div>

<p>My best friend is Manish.</p>

<p class="intro">My best friend is Shyam.</p>

</body>

</html>

2) Multiple Class Selectors: .class1.class2

.name1.name2{}

Selects all elements with both name1 and name2 set within its class attribute.

<html>

<head>

<style>

.intro.info /\*This style will be applied on the element which have class="intro info"\*/

{

background-color: yellow;

}

</style>

</head>

<body>

<h1>Demo of the .class.class selector</h1>

<div class="intro info">

<p>My name is Jimmy.</p>

<p>I live in UIT.</p>

</div>

<p class="info">My best friend is ABRAR.</p>

<p class="intro">My best friend is Haji Ali.</p>

</body>

</html>

3) Descendant class selector: .class1 .class2

.name1 .name2{} Selects all elements with name2 that is a descendant of an element with name1

<html>

<head>

<style>

.intro .info

{

background-color: yellow;

}

</style>

</head>

<body>

<h1>Demo of the .class.class selector</h1>

<div class="intro">

<p class="info">My name is Jimmy.</p>

<p>I live in UIT.</p>

</div>

<p class="info">My best friend is ABRAR.</p>

<p class="intro">My best friend is Haji Ali.</p>

</body>

</html>

4) Id selector :#id

#firstname{} Selects the element with id="firstname"

<html>

<head>

<style>

#firstname {

background-color: yellow;

}

</style>

</head>

<body>

<h1>Demo of the #id selector</h1>

<div>

<p id="firstname">My name is Rajat Sharma.</p>

<p id="hometown">I live in Bikaner.</p>

</div>

</body>

</html>

5) Universal selector : \*

\*{} Selects all elements

<html>

<head>

<style>

\* {

background-color: yellow;

}

</style>

</head>

<body>

<h1>Demo of the \* selector</h1>

<div class="intro">

<p id="firstname">My name is Rajat.</p>

<p id="hometown">I live in Bikaner.</p>

</div>

<p>My best friend is Usman.</p>

</body>

</html>

6) Tag/Element selector: selects and element

element{}

p{} Selects all <p> elements

<html>

<head>

<style>

p {

background-color: yellow;

}

</style>

</head>

<body>

<h1>Demo of the element selector</h1>

<div>

<p id="firstname">My name is Rajat.</p>

<p id="hometown">I live in Churu.</p>

</div>

<p>My best friend is Aftab.</p>

</body>

</html>

7) element.class

p.intro{} Selects all <p> elements with class="intro"

<html>

<head>

<style>

p.intro {

background-color: yellow;

}

</style>

</head>

<body>

<h1>Demo of the element.class selector</h1>

<div class="intro">

<p>My name is Pankaj.</p>

<p>I live in Rampura.</p>

</div>

<p>My best friend is Eesar.</p>

<p class="intro">My best friend is Manish.</p>

</body>

</html>

8) Multiple element selector

element,element{}

div, p{} Selects all <div> elements and all <p> elements

<html>

<head>

<style>

h2, p {

background-color: yellow;

}

</style>

</head>

<body>

<h1>Demo of the element, element selector</h1>

<h2>Welcome to My Homepage</h2>

<div>

<p>My name is Nitin.</p>

<p>I live in Muktaprashad.</p>

</div>

<p>My best friend is Abrar.</p>

</body>

</html>

9) Descendant element selector

element element{}

div p{} Selects all <p> elements inside <div> elements

<html>

<head>

<style>

div p {

background-color: yellow;

}

</style>

</head>

<body>

<h1>Demo of the element element selector</h1>

<div>

<h2>My name is Diljeet</h2>

<p>I live in Jodhpur.</p>

</div>

<p>My best friend is Shubham.</p>

</body>

</html>

10) Child Selector

element>element{}

div > p{} Selects all <p> elements where the parent is a <div> element

<html>

<head>

<style>

div > p {

background-color: yellow;

}

</style>

</head>

<body>

<h1>Demo of the element > element selector</h1>

<div>

<h2>My name is Mohit</h2>

<p>I live in Suratgarh.</p>

</div>

<div>

<span><p>I will not be styled.</p></span>

</div>

<p>My best friend is Ujjawal.</p>

</body>

</html>

11) element+element {}

div + p{}

Selects the first <p> element that is placed immediately after <div> elements

<html>

<head>

<style>

div + p {

background-color: yellow;

}

</style>

</head>

<body>

<h1>Demo of the element + element selector</h1>

<div>

<h2>My name is Vikram</h2>

<p>I live in Jaipur.</p>

</div>

<p>My best friend is Manohar.</p>

<p>I will not be styled.</p>

</body>

</html>

12) element1~element2{}

The element1~element2 selector matches occurrences of element2 that are preceded by element1.

Both elements must have the same parent, but element2 does not have to be immediately preceded by element1.

p ~ ul{} Selects every <ul> element that is preceded by a <p> element

<html>

<head>

<style>

p ~ ul {

background: #ff0000;

}

</style>

</head>

<body>

<div>A div element.</div>

<ul>

<li>Coffee</li>

<li>Tea</li>

<li>Milk</li>

</ul>

<p>The first paragraph.</p>

<ul>

<li>Coffee</li>

<li>Tea</li>

<li>Milk</li>

</ul>

<h2>Another list</h2>

<ul>

<li>Coffee</li>

<li>Tea</li>

<li>Milk</li>

</ul>

</body>

</html>

13) Attribute selector: It selects the elements with a given attribute.

[attribute]{}

[target]{} Selects all elements with a target attribute

<html>

<head>

<style>

a[target] {

background-color: yellow;

}

</style>

</head>

<body>

<h1>Demo of the [attribute] selector</h1>

<p>Links with a target attribute is styled with a yellow background:</p>

<a href="https://www.cinstitute.co.in">C Institute</a><br>

<a href="https://www.google.com" target="\_blank">Google.com</a><br>

<a href="https://www.microsoft.com" target="\_blank">Microsoft.com</a>

</body>

</html>

14)[attribute=value] {}

[target=\_blank]{} Selects all elements with target="\_blank"

<html>

<head>

<style>

a[target=\_blank] {

background-color: yellow;

}

</style>

</head>

<body>

<h1>Demo of the [attribute=value] selector</h1>

<p>Links with target="\_blank" is styled with a yellow background:</p>

<a href="https://www.cinstitute.co.in">C Institute</a><br>

<a href=<https://www.google.com> target="\_blank">Google.com</a><br>

<a href=<https://www.microsoft.com> target="\_blank">Microsoft.com</a>

</body>

</html>

15) [attribute~=value]{}

[title~=flower]{}

Selects all elements with a title attribute containing the word "flower"

<html>

<head>

<style>

[title~=flower] {

border: 5px solid yellow;

}

</style>

</head>

<body>

<h1>Demo of the [attribute~=value] selector</h1>

<p title="All about flowers">

all is well

</p>

<p>

Be with us, be ahead

</p>

<p title="all about flower">

Flower is different from fool. This will be seleted.

</p>

</body>

</html>

16) [attribute|=value]{}

[lang|=en]{}

Selects all elements with a lang attribute value starting with "en"

**Note**: The value has to be a whole word, either alone, like lang="en", or followed by a hyphen( - ), like lang="en-us".

<!DOCTYPE html>

<html>

<head>

<style>

[lang|=en] {

background: yellow;

}

</style>

</head>

<body>

<h1>Demo of the [attribute|=value] selector</h1>

<p>Elements with a lang attribute with starting value "en" is styled with a yellow background:</p>

<p lang="en">Hello!</p>

<p lang="en-us">Happy!</p>

<p lang="en-gb">Sad!</p>

<p lang="us-en">Go!</p>

<p lang="no">Do!</p>

</body>

</html>

17) [attribute^=value] {}

[href^="https"]{}

Selects every element whose href attribute value begins with "https"

<html>

<head>

<style>

div[class^="test"]

{

background: #ffff00;

}

</style>

</head>

<body>

<div class="first\_test">The first div element.</div>

<div class="second">The second div element.</div>

<div class="test">The third div element.</div>

<div class="test-ex">The fourth div element.</div>

<p class="test\_ex">This is a paragraph.</p>

<p class="test\_ex">This is a paragraph.</p>

</body>

</html>

18) [attribute$=value] {}

[href$="pdf"]{} Selects every element whose href attribute value ends with "pdf"

Set a background color on all <div> elements that have a class attribute value that ends with "test":

<html>

<head>

<style>

div[class$="test"] {

background: #ffff00;

}

</style>

</head>

<body>

<div class="first\_test">The first div element.</div>

<div class="second.test">The second div element.</div>

<div class="test">The third div element.</div>

<p class="test">This is some text in a paragraph.</p>

<p class="mytest">This is some text in a paragraph.</p>

</body>

</html>

19) [attribute\*=value] {}

[class\*="test"] Selects every element whose class attribute value contains the substring "test"

<html>

<head>

<style>

[class\*="test"] {

background: #ffff00;

}

</style>

</head>

<body>

<div class="first\_test">The first div element.</div>

<div class="second">The second div element.</div>

<div class="test">The third div element.</div>

<p class="test">This is some text in a paragraph.</p>

</body>

</html>

20) :active{}

a:active{} Selects the active link

21) ::after{}

p::after{} Insert something after the content of each <p> element

<html>

<head>

<style>

p::after {

content: " - Got it";

font-weight:900;

color:white;

background-color:black;

}

</style>

</head>

<body>

<h1>::after selector</h1>

<p>My name is Tarun Verma</p>

<p>I live in Hindustan</p>

</body>

</html>

22) ::before{}

p::before{} Insert something before the content of each <p> element

<html>

<head>

<style>

p::before {

content: " Hello - ";

font-weight:900;

color:white;

background-color:black;

}

</style>

</head>

<body>

<h1>::after selector</h1>

<p>My name is Tarun Verma</p>

<p>I live in Hindustan</p>

</body>

</html>

23) :checked

input:checked Selects every checked <input> element

<html>

<head>

<style>

input:checked

{

height: 30px;

width: 30px;

}

</style>

</head>

<body>

<form action="">

<input type="radio" checked name="gender"> Male<br>

<input type="radio" name="gender"> Female<br>

<input type="checkbox" checked > I have a car<br>

<input type="checkbox" > I have a horse

</form>

</body>

</html>

24) :default{}

input:default{} Selects the default <input> element

The :default selector selects the default form element in a group of related elements.

The :default selector can only be used on <button>, <input type="checkbox">, <input type="radio">, and <option> elements

<html>

<head>

<style>

input:default {

box-shadow: 0 0 1px 1px red;

}

</style>

</head>

<body>

<p>The "female" radio button is checked by default:</p>

<form action="">

<input type="radio" name="gender" value="male"> Male<br>

<input type="radio" name="gender" value="female" checked> Female<br>

<input type="radio" name="gender" value="other"> Other

</form>

</body>

</html>

25) :disabled{}

input:disabled{} Selects every disabled <input> element

<html>

<head>

<style>

input[type=text]:enabled {

background: #ffff00;

}

input[type=text]:disabled {

background: #dddddd;

}

</style>

</head>

<body>

<form action="">

Username: <input type="text" value="tarrunverrma"><br>

Email: <input type="text" value="tarun@cinstitute.co.in"><br>

Country: <input type="text" disabled="disabled" value="Bharat">

</form>

</body>

</html>

26) :empty{}

p:empty{} Selects every <p> element that has no children (including text nodes)

<html>

<head>

<style>

p:empty {

width: 100px;

height: 20px;

background: #ff0000;

}

</style>

</head>

<body>

<p></p>

<p>A paragraph.</p>

<p>Another paragraph.</p>

</body>

</html>

27) :enabled{}

input:enabled{} Selects every enabled <input> element

<html>

<head>

<style>

input[type=text]:enabled {

background: #ffff00;

}

input[type=text]:disabled {

background: #dddddd;

}

</style>

</head>

<body>

<form action="">

Username: <input type="text" value="tarrunverrma"><br>

Email: <input type="text" value="tarun@cinstitute.co.in"><br>

Country: <input type="text" disabled="disabled" value="Bharat">

</form>

</body>

</html>

28) :first-child{}

p:first-child{} Selects every <p> element that is the first child of its parent

Select and style every <p> element that is the first child of its parent:

<html>

<head>

<style>

p:first-child {

background-color: yellow;

}

</style>

</head>

<body>

<p>This paragraph is the first child of its parent (body).</p>

<p>This paragraph is not the first child of its parent (body).</p>

<div>

<p>This paragraph is the first child of its parent (div).</p>

<p>This paragraph is not the first child of its parent (div).</p>

</div>

</body>

</html>

29) ::first-letter{}

p::first-letter{} Selects the first letter of every <p> element

<html>

<head>

<style>

p::first-letter {

font-size: 200%;

color: #8A2BE2;

}

</style>

</head>

<body>

<h1>Demo of the ::first-letter selector</h1>

<p>My name is Gurjeet.</p>

<p>I live in Anupgarh.</p>

<p>My best friend is Tickey.</p>

</body>

</html>

30) ::first-line{}

p::first-line{} Selects the first line of every <p> element

<html>

<head>

<style>

p::first-line {

background-color: yellow;

}

</style>

</head>

<body>

<h1>Demo of the ::first-line selector</h1>

<h2>WWF's Mission Statement</h2>

<p>To stop the degradation of the planet's natural environment and to build a future in which humans live in harmony with nature, by; conserving the world's biological diversity, ensuring that the use of renewable natural resources is sustainable, and promoting the reduction of pollution and wasteful consumption.</p>

</body>

</html>

31) :first-of-type{}

p:first-of-type{} Selects every <p> element that is the first <p> element of its parent

<html>

<head>

<style>

p:first-of-type {

background: red;

}

</style>

</head>

<body>

<div>

<p>The first paragraph.</p>

<p>The second paragraph.</p>

</div>

<p>The third paragraph.</p>

<p>The fourth paragraph.</p>

</body>

</html>

32) :focus{}

input:focus{} Selects the input element which has focus

<html>

<head>

<style>

input:focus {

background-color: yellow;

}

</style>

</head>

<body>

<h1>Demo of the :focus selector</h1>

<p>Click inside the text fields to see a yellow background:</p>

<form>

First name: <input type="text" name="firstname"><br>

Last name: <input type="text" name="lastname">

</form>

</body>

</html>

33) :fullscreen{}

:fullscreen{} Selects the element that is in full-screen mode

34) :hover{}

a:hover{} Selects links on mouse over

35) :in-range{}

input:in-range{} Selects input elements with a value within a specified range

<html>

<head>

<style>

input:in-range {

border: 2px solid yellow;

}

</style>

</head>

<body>

<h1>Demo of the :in-range selector</h1>

<input type="number" min="5" max="10" value="7">

<p>Try typing a number out of range (less than 5 or higher than 10), to see the styling disappear.</p>

</body>

</html>

36) :indeterminate{}

input:indeterminate{} Selects input elements that are in an indeterminate state

37) :invalid{}

input:invalid{} Selects all input elements with an invalid value

<html>

<head>

<style>

input:invalid {

border: 2px solid red;

}

</style>

</head>

<body>

<h1>Demo of the :invalid selector</h1>

<input type="email" value="supportEmail">

<p>Try typing a legal e-mail address, to see the styling disappear.</p>

</body>

</html>

38) :lang(language){}

p:lang(it){} Selects every <p> element with a lang attribute equal to "it" (Italian)

<html>

<head>

<style>

p:lang(it) {

background: yellow;

}

</style>

</head>

<body>

<h1>Demo of the :lang selector</h1>

<p>I live in Italy.</p>

<p lang="it">Athe koni italian!</p>

</body>

</html>

39) :last-child{}

p:last-child{} Selects every <p> element that is the last child of its parent

<html>

<head>

<style>

p:last-child {

background-color: pink;

}

</style>

</head>

<body>

<p>The first paragraph.</p>

<p>The second paragraph.</p>

<p>The third paragraph.</p>

<p>The last paragraph.</p>

</body>

</html>

40) :last-of-type{}

p:last-of-type{} Selects every <p> element that is the last <p> element of its parent

<html>

<head>

<style>

p:last-of-type

{

background: pink;

}

</style>

</head>

<body>

<h1>This is a heading</h1>

<p>The first paragraph.</p>

<p>The second paragraph.</p>

<p>The third paragraph.</p>

<p>The fourth paragraph.</p>

</body>

</html>

41) :link{}

a:link{} Selects all unvisited links

42) ::marker{}

::marker{} Selects the markers of list items

<html>

<head>

<style>

::marker {

color: blue;

font-size:1.5em;

font-weight:900;

}

</style>

</head>

<body>

<h1>Demo of the ::marker selector</h1>

<ul>

<li>Coffee</li>

<li>Tea</li>

<li>Milk</li>

</ul>

<ol>

<li>First</li>

<li>Second</li>

<li>Third</li>

</ol>

</body>

</html>

43) :not(selector){}

:not(p) Selects every element that is not a <p> element

<html>

<head>

<style>

p {

color: #000000;

}

:not(p) {

color: #ff0000;

}

</style>

</head>

<body>

<h1>This is a heading</h1>

<p>This is a paragraph.</p>

<p>This is another paragraph.</p>

<div>This is some text in a div element.</div>

</body>

</html>

44) :nth-child(n){}

The :nth-child(n) selector matches every element that is the nth child of its parent.

n can be a number, a keyword (odd or even), or a formula (like an + b).

p:nth-child(2){} Selects every <p> element that is the second child of its parent

<html>

<head>

<style>

/\* Selects the second element of div siblings \*/

div:nth-child(2) {

background: red;

}

/\* Selects the second li element in a list \*/

li:nth-child(2) {

background: lightgreen;

}

/\* Selects every third element among any group of siblings \*/

:nth-child(3) {

background: yellow;

}

</style>

</head>

<body>

<div>

<p>This is some text.</p>

</div>

<div>

<p>This is some text.</p>

</div>

<div>

<p>This is some text.</p>

</div>

<ul>

<li>First list item</li>

<li>Second list item</li>

<li>Third list item</li>

<li>Fourth list item</li>

<li>Fifth list item</li>

</ul>

</body>

</html>

45) :nth-last-child(n){}

The :nth-last-child(n) selector matches every element that is the nth child, regardless of type, of its parent, counting from the last child.

n can be a number, a keyword, or a formula.

p:nth-last-child(2){} Selects every <p> element that is the second child of its parent, counting from the last child

<html>

<head>

<style>

p:nth-last-child(2) {

background: pink;

}

</style>

</head>

<body>

<p>The first paragraph.</p>

<p>The second paragraph.</p>

<p>The third paragraph.</p>

<p>The fourth paragraph.</p>

</body>

</html>

46) :nth-last-of-type(n){}

The :nth-last-of-type(n) selector matches every element that is the nth child, of a particular type, of its parent, counting from the last child.

n can be a number, a keyword, or a formula.

p:nth-last-of-type(2){} Selects every <p> element that is the second <p> element of its parent, counting from the last child

<html>

<head>

<style>

p:nth-last-of-type(2) {

background: powderblue;

color:navy;

font-weight:900;

}

</style>

</head>

<body>

<p>The first paragraph.</p>

<p>The second paragraph.</p>

<p>The third paragraph but its 2nd child from last that is of type p.</p>

<div>This is 2nd child from last but its not a p</div>

<p>The fourth paragraph.</p>

</body>

</html>

47) :nth-of-type(n){}

p:nth-of-type(2){} Selects every <p> element that is the second <p> element of its parent

The :nth-of-type(n) selector matches every element that is the nth child, of the same type (tag name), of its parent.

n can be a number, a keyword (odd or even), or a formula (like an + b).

<html>

<head>

<style>

/\* Selects the second element of div siblings \*/

div:nth-of-type(2) {

background: pink;

}

/\* Selects the second li element in a list \*/

li:nth-of-type(2) {

background: lightred;

}

/\* Selects every third element among any group of siblings \*/

:nth-of-type(3) {

background: darkblue;

color:cyan;

}

</style>

</head>

<body>

<div>

<p>This is some text.</p>

</div>

<div>

<p>This is some text.</p>

</div>

<div>

<p>This is some text.</p>

</div>

<ul>

<li>First list item</li>

<li>Second list item</li>

<li>Third list item</li>

<li>Fourth list item</li>

<li>Fifth list item</li>

</ul>

</body>

</html>

48) :only-of-type{}

The :only-of-type selector matches every element that is the only child of its type, of its parent.

p:only-of-type{} Selects every <p> element that is the only <p> element of its parent

<html>

<head>

<style>

p:only-of-type {

background: cyan;

color:navy;

}

</style>

</head>

<body>

<div><p>This is a paragraph.</p></div>

<div><p>This is a paragraph.</p><p>This is a paragraph.</p></div>

</body>

</html>

49) :only-child{}

The :only-child selector matches every element that is the only child of its parent.

p:only-child{} Selects every <p> element that is the only child of its parent

<html>

<head>

<style>

p:only-child {

background: blue;

color:white;

}

</style>

</head>

<body>

<div><p>This is a paragraph.</p></div>

<div><span>This is a span.</span><p>This is a paragraph.</p></div>

</body>

</html>

50) :optional{}

The :optional selector selects form elements which are optional.

Form elements with no required attribute are defined as optional.

input:optional{} Selects input elements with no "required" attribute

<html>

<head>

<style>

input:optional {

background-color: lightpink;

}

</style>

</head>

<body>

<h1>Demo of the :optional selector</h1>

<p>An optional input element:<br><input></p>

<p>A required input element:<br><input required></p>

</body>

</html>

51) :out-of-range{}

input:out-of-range{} Selects input elements with a value outside a specified range

The :out-of-range selector selects all elements with a value that is outside a specified range.

**Note:** The :out-of-range selector only works for input elements with min and/or max attributes!

<html>

<head>

<style>

input:out-of-range {

background: red;

color:white;

}

</style>

</head>

<body>

<h1>Demo of the :out-of-range selector</h1>

<input type="number" min="5" max="10" value="17">

<p>Try typing a number within the given range (between 5 and 10), to see the styling disappear.</p>

</body>

</html>

52) ::placeholder{}

input::placeholder{} Selects input elements with the "placeholder" attribute specified

The ::placeholder selector selects form elements with placeholder text, and let you style the placeholder text.

The placeholder text is set with the placeholder attribute, which specifies a hint that describes the expected value of an input field.

<html>

<head>

<style>

::-webkit-input-placeholder { /\* Edge \*/

color: red;

}

:-ms-input-placeholder { /\* Internet Explorer \*/

color: red;

}

::placeholder {

color: red;

}

</style>

</head>

<body>

<p>Use the ::placeholder selector to change the color of the placeholder text:</p>

<input type="text" name="fname" placeholder="First name">

</body>

</html>

53) :read-only{}

input:read-only{} Selects input elements with the "readonly" attribute specified

The :read-only selector selects elements which are "readonly".

Form elements with a "readonly" attribute are defined as "readonly".

<html>

<head>

<style>

input:read-only {

background-color: yellow;

}

</style>

</head>

<body>

<h1>Demo of the :read-only selector</h1>

<p>The :read-only selector selects form elements with a "readonly" attribute:</p>

<p>A normal input element:<br><input value="hello"></p>

<p>A readonly input element:<br><input readonly value="hello"></p>

</body>

</html>

54) :read-write{}

input:read-write{} Selects input elements with the "readonly" attribute NOT specified

The :read-write selector selects form elements which are "readable" and "writeable".

Form elements with no "readonly" attribute, and no "disabled" attribute are defined as "read-" and "write-able".

<html>

<head>

<style>

input:read-write {

background-color: yellow;

}

</style>

</head>

<body>

<h1>Demo of the :read-write selector</h1>

<p>The :read-write selector selects form elements with no "readonly" attribute:</p>

<p>A normal input element:<br><input value="hello"></p>

<p>A readonly input element:<br><input readonly value="hello"></p>

</body>

</html>

55) :required{}

input:required {} Selects input elements with the "required" attribute specified

The :required selector selects form elements which are required.

Form elements with a required attribute are defined as required.

**Note**: The :required selector only applies to the form elements: input, select and textarea.

<html>

<head>

<style>

input:required {

background-color: yellow;

}

</style>

</head>

<body>

<h1>Demo of the :required selector</h1>

<p>An optional input element:<br><input></p>

<p>A required input element:<br><input required></p>

<p>The :required selector selects form elements with a "required" attribute.</p>

</body>

</html>

56) :root{}

:root{} Selects the document's root element

The :root selector matches the document's root element.

In HTML, the root element is always the html element.

<html>

<head>

<style>

:root {

background: #ff0000;

}

</style>

</head>

<body>

<h1>This is a heading</h1>

</body>

</html>

57) ::selection{}

::selection{} Selects the portion of an element that is selected by a user

The ::selection selector matches the portion of an element that is selected by a user.

Only a few CSS properties can be applied to the ::selection selector: color, background, cursor, and outline.

<html>

<head>

<style>

::selection {

color: red;

background: yellow;

}

</style>

</head>

<body>

<h1>Select some text on this page:</h1>

<p>This is a paragraph.</p>

<div>This is some text in a div element.</div>

</body>

</html>

58) :target{}

#news:target{} Selects the current active #news element (clicked on a URL containing that anchor name)

URLs with an # followed by an anchor name link to a certain element within a document. The element being linked to is the target element.

The :target selector can be used to style the current active target element.

<html>

<head>

<style>

:target {

background-color: navy; color:white;

}

</style>

</head>

<body>

<h1>This is a heading</h1>

<p><a href="#news1">Jump to New content 1</a></p>

<p><a href="#news2">Jump to New content 2</a></p>

<p>Click on the links above and the :target selector highlight the current active HTML anchor.</p>

<p id="news1"><b>New content 1...</b></p>

<p id="news2"><b>New content 2...</b></p>

</body>

</html>

59) :valid{}

input:valid{} Selects all input elements with a valid value

The :valid selector selects form elements with a value that validates according to the element's settings.

**Note**: The :valid selector only works for form elements with limitations, such as input elements with min and max attributes, email fields with a legal email, or number fields with a numeric value, etc.

<html>

<head>

<style>

input:valid {

background-color: #000000; color:smokewhite;

}

</style>

</head>

<body>

<h1>Demo of the :valid selector</h1>

<input type="email" value="support@example.com">

<p>Try typing an illegal e-mail address, to see the styling disappear.</p>

</body>

</html>

60) :visited{}

a:visited{} Selects all visited links