**Селектори і їх комбінації**

* [A](https://webref.ru/css/selector/tag) – звернення до тегу по його назві.
* [#id](https://webref.ru/css/selector/id) – звернення елементу який містить дане id.
* [.class](https://webref.ru/css/selector/class) – звернення елементу який містить class.
* [.class1.class2](https://webref.ru/css/selector/multiclass) – звернення до елементу який містить цих 2 класи.
* [\*](https://webref.ru/css/selector/universal) – звернення до всіх елементів(універсальний селектор).
* [A   B](https://webref.ru/css/selector/descendant) – звернення то елементу B який є дочірнім для А, причому вкладеність може бути не строга.
* [A > B](https://webref.ru/css/selector/child) – звернення то елементу B який є дочірнім для А, причому вкладеність строга.
* [A + B](https://webref.ru/css/selector/adjacent) – звернення то елементу B який є сусідом для А, тобто йде зразу після нього.
* [A ~ B](https://webref.ru/css/selector/sibling) – звернення то елементів B який є сусідами для А, тобто йдуть зразу після нього.
* [[attr]](https://webref.ru/css/selector/attr) – встановлює стиль для елемента в тому випадку, якщо елемент містить даний атрибут.
* [[attr='value']](https://webref.ru/css/selector/attr2) – встановлює стиль для елемента в тому випадку, якщо елемент містить атрибут з даним значенням.
* [[attr^='value']](https://webref.ru/css/selector/attr3) – встановлює стиль для елемента в тому випадку, якщо значення атрибута починається з вказаного тексту.
* [[attr$='value']](https://webref.ru/css/selector/attr4) – встановлює стиль для елемента в тому випадку, якщо значення атрибута закінчується зазначеним текстом.
* [[attr\*='value']](https://webref.ru/css/selector/attr5) – встановлює стиль для елемента в тому випадку, якщо частиною його значення є деякий текст.

**HTML форми**

HTML форми можна використовувати для надсилання даних через Інтернет і часто використовуються для організації зворотного зв'язку з користувачами.

Основні теги, які використовуються для створення HTML форм, - це тег **<form>**, тег **<input>**, тег **<textarea>**, тег **<select>** і тег **<option>.**

Елемент **<form>** (від англ. Form - форма) встановлює форму на веб-сторінці. Форма призначена для обміну даними між користувачем і сервером. Область застосування форм не обмежена відправкою даних на сервер, за допомогою клієнтських скриптів можна отримати доступ до будь-якого елементу форми, змінювати його і застосовувати на власний розсуд.

<form>  
 ...  
</form>

**<input>** (від англ. Input - введення) є одним з різнобічних елементів форми і дозволяє створювати різні частини інтерфейсу і забезпечувати взаємодію з користувачем. Однак реальна дія цього елемента залежить від того, яке значення встановлено у його атрибута **type**. А він може набувати таких значень:

* text: звичайне текстове поле
* password: теж текстове поле, тільки замість символів, що вводять відображаються зірочки, тому в основному використовується для введення пароля
* radio: радіокнопка або перемикач. З групи радіокнопок можна вибрати тільки одну
* checkbox: елемент прапорець, який може знаходитися в зазначеному або невідзначеними стані
* hidden: приховане поле
* submit: кнопка відправки форми
* color: поле для введення кольору
* date: поле для введення дати
* datetime: поле для введення дати і часу з урахуванням часового поясу
* datetime-local: поле для введення дати і часу без обліку часового поясу
* email: поле для введення адреси електронної пошти
* month: поле для введення року і місяця
* number: поле для введення чисел
* range: повзунок для вибору числа з деякого діапазону
* tel: поле для введення телефону
* time: поле для введення часу
* week: поле для введення року і тижня
* url: поле для введення адреси url
* file: поле для вибору відправляється файлу
* image: створює кнопку у вигляді картинки

Елемент **<select>** (від англ. Selection - вибір) дозволяє створити елемент інтерфейсу у вигляді списку, а також список з одним або множинним вибором.

Елемент **<option>** (від англ. Option - варіант, вибір) визначає окремі пункти списку, що створюється за допомогою контейнера <select>. Ширина списку визначається самим широким текстом, зазначеним в <option>, а також може змінюватися за допомогою стилів. Якщо планується відправляти дані списку на сервер, то потрібно помістити елемент <select> всередину форми. Це також необхідно, коли до даних списку йде звернення через скрипти.

Елемент **<optgroup>** (від англ. Option group - група варіантів) являє собою контейнер, усередині якого розташовуються елементи <option>, об'єднані в одну групу. Особливістю <optgroup> є те, що він не виділяється як звичайний елемент списку, акцентується за допомогою жирного накреслення, а всі елементи, що входять до цей контейнер, зміщуються вправо від свого початкового положення.

<select>  
 <optgroup *label*="<текст>">  
 <option>...</option>  
 </optgroup>  
</select>

**<textarea>** є елементом форми для створення області, в яку можна вводити кілька рядків тексту. На відміну від елемента <input> в текстовому полі допустимо робити переноси рядків, вони зберігаються при відправці даних на сервер.

<textarea>  
 текст  
</textarea>

Елемент **<button>** (від англ. Button - кнопка) створює на веб-сторінці кнопки і за своєю дією нагадує результат, одержаний за допомогою <input> (з атрибутом type = "button | reset | submit"). На відміну від цього елемента, <button> пропонує розширені можливості по створенню кнопок. Наприклад, на подібній кнопці можна розміщувати будь-які елементи HTML, в тому числі зображення. Використовуючи стилі можна визначити вид кнопки шляхом зміни шрифту, кольору фону, розмірів і інших параметрів.

<button>...</button>

Елемент **<fieldset>** (від англ. Form field set - набір полів форми) призначений для групування елементів форми. Таке групування полегшує роботу з формами, що містять велику кількість даних, наприклад, один блок може бути призначений для введення текстової інформації, а інший - для прапорців.

Елемент **<legend>** (від анг. Legend - легенда, напис) застосовується для створення заголовка групи елементів форми, яка визначається за допомогою <fieldset>. Група елементів позначається в браузері за допомогою рамки, а текст, який розташовується усередині контейнера <legend>, вбудовується в цю рамку.

<fieldset>  
 <legend>Текст</legend>

...  
</fieldset>

**Псевдоелемет**

Псевдоелемент в CSS - це ключове слово, яке додається до селектора, яке дозволяє стилізувати певну частину обраного елемента. Наприклад, його можна використовувати для:

* Стиль першої літери або рядка елемента
* Вставити вміст до або після вмісту елемента

**::after**

В CSS, :: after створює псевдоелемент, який є останнім нащадком обраного елемента.

Часто використовується для додавання косметичного вмісту в елемент за допомогою властивості content. За замовчуванням є інлайновим.

**::before**

В CSS, :: before створює псевдоелемент, який є першим нащадком обраного елемента.

Часто використовується для додавання косметичного вмісту в елемент за допомогою властивості content. За замовчування є інлайновим.

**::first-letter**

CSS псевдоелемент ::first-letter застосовує стилі до першої букви першого рядка блокового елемента, але тільки якщо немає іншого попереднього вмісту (такого як зображення або інлайн таблиці).

**::first-line**

CSS псевдоелемент :: first-line застосовує стилі до першому рядку блочного елемента. Зверніть увагу, що довжина першого рядка залежить від багатьох факторів, включаючи ширину елемента, ширину документа і розмір шрифту тексту.

**::selection**

Псевдоелемент :: selection дозволяє застосувати стилі до частини документа, який був виділений користувачем (наприклад, за допомогою миші).

**:: placeholder**

CSS псевдоелемент :: placeholder є текст placeholder в <input> або <textarea> елементах.

**Псевдоклас**

Вони визначають динамічний стан елементів, яке змінюється за допомогою дій користувача, а також положення в дереві документа.

**Синтаксис**

Псевдоклас не може існувати сам по собі. Він повинен бути прикріплений до селектора. Псевдоклас визначатиме тільки певний стан цього селектора.

Синтаксис виглядає наступним чином:

selector:*pseudo-classes*{  
 ...  
}

Прогалини між селектором і псевдоклас немає, щоб показати, що вони пов'язані один з одним.

**:hover**

Наприклад, типовим використовуваним псевдоклас є: hover, який буде застосовувати стиль, коли на цільової елемент наводиться покажчик миші. Давайте перевіримо це на засланнях.

**HTML**

<p> Наведіть курсор <a *href*="#"> на це посилання </a> і побачите, як вона стає червоною. </p>

**CSS**

a {  
 color: blue;  
}  
a:*hover* {  
 color: red;  
}

Перший рядок у CSS визначає, що всі елементи <a> повинні виглядати синіми. Другий рядок визначає, що при наведенні покажчика миші <a> повинні виглядати червоними.

Другий рядок орієнтована на ті ж елементи HTML, але тільки, коли відбувається щось конкретне (в даному випадку, наведення).

**:visited**

Цей псевдоклас націлюється на посилання, які вже були відвідані. За замовчуванням посилання відображаються синіми і при відвідуванні стають фіолетовими. Результати Google працюють так само.

**HTML**

<a *href*="https://www.google.com ">Google</a>

<a *href*="https://twitter.com">Twitter</a>  
<a *href*="https://www.facebook.com">Facebook</a>  
<a *href*="https://www.mozilla.org">Mozilla</a>

**CSS**

a {  
 color: dodgerblue;  
}  
a:*visited* {  
 color: rebeccapurple;  
}

Міняти колір відвіданих посилань часто забувають, але користувачам так зручніше переглядати список результатів. Це допомагає їм легко визначати, де вони вже були.

**:focus**

Даний псевдоклас відбувається, коли елемент HTML отримує фокус. Це особливо корисно для полів форм.

**HTML**

<p>  
 <label *for*="username">  
 <input *class*="form-input" *placeholder*="Введіть ім’я" *id*="username">  
 </label>  
</p>

**CSS**

.*form-input* {  
 border: 2px solid grey;  
 padding: 5px;  
}  
.*form-input*:*focus* {  
 background: lightyellow;  
 border-color: blue;  
 outline: none;  
}

**:first-child і : last-child**

Ці псевдокласи пов'язані з ієрархією в HTML. Вони націлюються на елементи HTML в залежності від порядку, в якому вони з'являються в коді.

**HTML**

<ul>  
 <li>Один</li>  
 <li>Два</li>  
 <li>Три</li>  
 <li>Чотири</li>  
</ul>

**CSS**

li:*first-child* {  
 background: greenyellow;  
}  
li:*last-child* {  
 background: lightsalmon;  
}

**:nth-child**

Даний псевдоклас є глобальною версією: first-child і: last-child. За допомогою: nth-child ви можете вирахувати конкретний дочірній цільової елемент.

Наприклад, якщо ви хочете націлитися на другий дочірній елемент, то використовуйте: nth-child(2).

**HTML**

<ul>  
 <li>Один</li>  
 <li>Два</li>  
 <li>Три</li>  
 <li>Чотири</li>  
</ul>

**CSS**

li:*nth-child*(2) {  
 background: violet;  
}

**odd і even**

Крім використання чисел, :nth-child поставляється з двома ключовими словами:

* : nth-child(odd) націлена на кожен непарний елемент;
* : nth-child(even) націлена на кожен парний елемент.

**CSS**

li:*nth-child*(odd) {  
 background: gold;  
}

**Лічильник n**

Потужним аспектом: nth-child є те, що він може націлюватися на елементи, грунтуючись на розрахунках за допомогою ключового слова n.

Значення n підвищується від 0 до кількості представлених дочірніх елементів.

Що робити, якщо ви хочете націлитися на кожен третій елемент?

**CSS**

li:*nth-child*(3n) {  
 background: hotpink;  
}

У нашому випадку n починається з нуля і закінчується шістьма.

Комп'ютери починають відлік з нуля. І оскільки в нашому списку сім елементів, ми будемо вважати до шести, тому що 0-1-2-3-4-5-6 представляють сім пунктів.

Ви можете прочитати: nth-child (3n) як «націлитися на кожен елемент, положення якого ділиться на 3». У нашому випадку, це третій і шостий пункт списку:

* 3 помножити на 0 - дорівнює нулю;
* 3 помножити на 1 - третій пункт;
* 3 помножити на 2 - шостий пункт.

Що робити, якщо ви хочете націлитися на перший пункт і кожен третій пункт після нього - **n + 1**

**CSS**

li:*nth-child*(3n+1) {  
 background: hotpink;  
}

**Додаткові джерела:**

https://developer.mozilla.org/en-US/docs/Learn/HTML/Forms/Your\_first\_HTML\_form

<https://www.w3schools.com/html/html_forms.asp>

<https://developer.mozilla.org/en-US/docs/Web/CSS/Pseudo-elements>

https://developer.mozilla.org/en-US/docs/Web/CSS/Pseudo-classes