**Конспект “Селекторы”**

Селекторы позволяют очень точно указывать к каким элементам применять CSS-свойства. Селекторы — это наш снайперский прицел, в этой главе мы будем учиться им пользоваться.

Так как глава посвящёна «стрельбе свойствами по элементам», то тема биатлона подходит как нельзя кстати. В каждом задании вам нужно будет с помощью селекторов закрывать мишени.

Но для начала представим, что механизма селекторов не существует. Как в этом случае задавать CSS-свойства элементам? С помощью атрибута style. Например:

<p style="color: red;">...</p>

**Селекторы по тегам**

Задавать атрибут style для каждого тега неудобно и долго. Особенно, если тот же результат можно получить с помощью единственного CSS-правила, в котором используется селектор для тега li.

С помощью селекторов по именам тегов можно задать стили для всех элементов списка, изображений, абзацев и так далее. Эти селекторы содержат имя тега без символов < и >. Например:

*li* {

/\* стили для элементов списка \*/

}

Одно правило может относиться сразу к нескольким селекторам, в таком случае селекторы перечисляются через запятую:

*a, img* {

/\* стили для ссылок и изображений \*/

}

**Селекторы по классам**

Класс — это один из атрибутов тегов. Выглядит он вот так:

<li class="first"></li>

Этот атрибут особенный, так как в CSS существует возможность выбирать элементы по классу. Делается это с помощью такого селектора: .*имя\_класса*. Например:

.first {

/\* стили для класса *first* \*/

}

Имена классов могут состоять из латинских символов, цифр и знаков - и \_. Имя класса должно начинаться с латинской буквы.

# Отрабатываем селекторы по классам

Синтаксис CSS позволяет выбирать элементы не только по одному классу или тегу. Можно, например, выбрать элемент одновременно по тегу и по классу или же элемент с двумя классами сразу. Для этого селектор составляется просто одной строкой из всех желаемых «частей» без пробелов. Давайте рассмотрим примеры.

В селекторе по тегу и классу первым пишется название тега, а потом идёт класс:

/\* выбор всех тегов ul с классом target \*/

ul.target {...}

<ul class="target"></ul>

Если у элемента задано несколько классов, в HTML и в CSS-селекторе они могут идти в разном порядке — это не будет влиять на выборку элементов:

/\* выбор элементов с двумя классами: text и green \*/

.text, .green {...}

<span class="text green"></span>

<p class="green text"></p>

Количество классов в селекторе может быть любым:

/\* выбор тегов span с четырьмя классами: underlined, red, big и text \*/

span.underlined, .red, .big, .text {...}

<span class="underlined red big text"></span>

# Контекстные селекторы

Селектор может состоять из нескольких частей, разделённых пробелом, например:

p strong { ... }

ul .hit { ... }

.footer .menu a { ... }

Такие селекторы называют контекстными или вложенными. Их используют для того, чтобы применить стили к элементу, только если он вложен в нужный элемент.

Например, селектор .menu a сработает для ссылки a только в том случае, если она расположена внутри элемента с классом .menu.

Читать их проще всего справа налево:

/\* выбрать все теги strong внутри тегов p \*/

p strong { ... }

ul .hit { ... } /\* выбрать все элементы с классом .hit внутри тегов ul \*/

/\* выбрать все ссылки внутри элементов с классом .menu,

которые лежат внутри элементов с классом .footer \*/

.footer .menu a { ... }

Таким образом, можно задавать элементам различные стили в зависимости от их контекста. Если ссылка расположена внутри меню, сделать её крупнее, а если внутри основного текста, то задать ей нужный цвет.

# Соседние селекторы

Контекстные селекторы используются для вложенных друг в друга элементов, а соседние — для расположенных рядом.

Например, теги <li> в списке являются соседними по отношению друг к другу и вложенными в тег <ul>.

Соседние селекторы записываются с помощью знака +, например, селектор1 + селектор2. Стили применятся к элементу, подходящему под селектор2, только если сразу перед ним расположен элемент, подходящий под селектор1.

Пример. Есть два элемента списка:

<ul>

<li class="hit"></li>

<li class="miss"></li>

</ul>

Селектор .hit + .miss применит стили к элементу с классом miss, так как перед ним есть элемент с классом hit.

Селектор .hit + li, а также селектор li + .miss, или даже li + li тоже применит стили к элементу с классом miss, то есть ко второму элементу списка.

А вот селектор .miss + .hit не сработает, так как элемент с классом miss находится после элемента с классом hit в разметке.

li { background-color: white; }

.first + li,

.fourth + li {

background-color: red;

}

# Контекстные и соседние селекторы

Селекторы в CSS можно очень гибко комбинировать. В частности, можно комбинировать контекстные и соседние селекторы.

Например, селектор .player-1 .hit + .miss сработает для тега с классом miss, если сразу перед ним расположен тег с классом hit и оба тега расположены внутри тега с классом player-1.

# Дочерние селекторы

Потомком называются любые элементы, расположенные внутри родительского элемента. А дочерними элементами называются ближайшие потомки. Взгляните на пример:

<ul>

<li><span>...</span></li>

<li><span>...</span></li>

</ul>

По отношению к списку <ul> элементы <li> являются дочерними элементами и потомками, а <span> — потомки, но не дочерние элементы.

Контекстные селекторы влияют на всех потомков, что не всегда удобно. Иногда необходимо задать стили только для дочерних элементов. Особенно это полезно при работе с [многоуровневыми списками](https://htmlacademy.ru/courses/38/run/5).

Для этого существует дочерний селектор, в котором используется символ >. Например: ul > li или ul > li > span.

# Псевдоклассы

Псевдоклассы — это дополнения к обычным селекторам, которые делают их ещё точнее и мощнее. Обычный селектор — это снайперский прицел, а с псевдоклассом он становится прибором ночного видения.

Псевдокласс добавляется к селектору c помощью символа :, вот так селектор:псевдокласс. Например:

a:visited { ... }

li:last-child { ... }

.alert:hover { ... }

Знакомство с псевдоклассами мы начнём с first-child и last-child.

Псевдокласс first-child позволяет выбрать первый дочерний элемент родителя, а last-child — последний дочерний элемент. Например:

li:last-child { ... }

Этот селектор выберет последний элемент списка.

# Псевдокласс :nth-child

Псевдоклассы из предыдущего примера относятся к семейству псевдоклассов, помогающих выбирать элементы по их расположению.

Вспомним [задание 4](https://htmlacademy.ru/courses/42/run/4). В нём каждому тегу <li> был задан собственный класс. Используя классы, мы могли выбрать любой из пяти тегов. Если бы тегов было десять, то пришлось бы использовать десять разных классов.

С помощью псевдокласса nth-child можно выбирать теги по порядковому номеру, не используя классы. Синтаксис псевдокласса: селектор:nth-child(выражение). Выражением может быть число или формула. Например:

1. li:nth-child(2) { ... }

2. li:nth-child(4) { ... }

3. li:nth-child(2n) { ... }

Первый селектор выберет второй элемент списка, второй селектор — четвёртый элемент списка, третий селектор — все чётные элементы.

# :nth-child и контекстные селекторы

Селекторы с псевдоклассами хорошо сочетаются с контекстными селекторами.

Например, следующий селектор выберет третий тег <li> внутри блока с классом shooter-2.

.shooter-2 li:nth-child(3) { ... }

Селектор не обязательно должен заканчивается псевдоклассом. Например, следующий селектор выберет элемент с классом .text в четвёртом элементе списка <li>.

li:nth-child(4) .text { ... }

# Псевдокласс :hover

Некоторые псевдоклассы позволяют выбирать элементы, с которыми взаимодействует пользователь. Сначала познакомимся с псевдоклассом :hover.

Этот псевдокласс позволяет выбрать элемент, когда на него наведён курсор мыши и кнопка мыши не нажата. Примеры:

1. a:hover { ... }

2. tr:hover { ... }

3 .menu-item:hover { ... }

Первый селектор выбирает ссылку, второй строку таблицы, третий элемент с классом menu-item, но только в том случае, если на них наведён курсор мыши.

Благодаря этому псевдоклассу можно добавлять в интерфейс динамику и интерактивность, т.к. элементы начинают реагировать на действия пользователя, изменяя свой внешний вид.

# Динамические эффекты с помощью :hover

Интересовались ли вы, как с помощью CSS создаются выпадающие меню?

Львиная доля динамических эффектов, создаваемых с помощью CSS, опираются на несколько псевдоклассов, главный из которых, конечно же, :hover. Весь секрет заключается в сочетании контекстных селекторов и псевдоклассов. Посмотрите на пример:

li.top ul.submenu {

display: none;

}

li.top:hover ul.submenu {

display: block;

}

Первое правило прячет список-подменю. Второе правило гласит: «если на верхний пункт меню, в котором находится подменю, наведут курсор, то надо показать подменю». Вот так всё просто.

Общий принцип такой: родительский элемент реагирует на наведение мыши и изменяет свойства элементов-потомков. То есть всё работает на контекстных селекторах вида селектор1:hover селектор2.

# Псевдоклассы :link, :visited и :active

Отвлечёмся от биатлона и познакомимся с псевдоклассами для ссылок.

* :link выбирает ещё не посещённые ссылки.
* :visited выбирает посещённые ссылки.
* :active выбирает активные ссылки (кнопка мыши зажата на ссылке).

Пример задания CSS-правил для ссылок:

a:link { ... }

a:visited { ... }

a:hover { ... }

a:active { ... }

Обратите внимание на порядок правил. Если их расположить по-другому, то некоторые могут не сработать.

# Псевдокласс :focus

Псевдокласс :focus позволяет выбрать элемент, который в данный момент в фокусе. Например, текстовое поле, в которое установлен курсор, находится в фокусе.

В фокусе могут быть не только текстовые поля. Если вы переключаетесь между элементами веб-страницы с помощью клавиши **tab**, то в фокус будут попадать ссылки.

Пример использования псевдокласса:

input:focus {

/\* стили для поля в фокусе \*/

}

# Селекторы атрибутов

Ранее мы познакомились с атрибутом class и специальными селекторами по классу. Существуют селекторы, которые позволяют выбирать элементы по любым атрибутам.

Чаще всего такие селекторы используются [при работе с формами](https://htmlacademy.ru/courses/50), так как поля форм имеют атрибут type с разными значениями.

Селекторы атрибутов записываются с использованием квадратных скобок: элемент[атрибут]. Примеры селекторов:

1. input[checked] { ... }

2. input[type="text"] { ... }

Первый селектор выберет поля формы, у которых есть атрибут checked, второй селектор выберет поля формы, у которых атрибут type имеет значение text.

**Селектор по id**

Существует ещё один HTML-атрибут, для которого существует специальный селектор. Этот атрибут id (идентификатор), а селектор записывается с помощью символа #, например, #some-id.

На значение id распространяются те же ограничения, что и на имя класса. Также id должен быть уникальным на странице. Пример:

HTML<p id="greeting">Приветствие!</p>

CSS#greeting {

...

}

Использование селекторов по id при оформлении считается плохой практикой. Существуют редкие исключения из этого правила, например, при оживлении слайдера на чистом CSS.