Конспект «Селекторы. Знакомство»

Селекторы позволяют точно указывать к каким элементам применять CSS-свойства.

Без использования селекторов стили можно задать при помощи атрибута `style`.

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

Селекторы по тегам

Селекторы по именам тегов задают стили для всех элементов списка, изображений, абзацев и так далее. Эти селекторы содержат имя тега без символов `<` и `>`. Например:

```
li {
  /* стили для элементов списка */
}
```

Если правило относится сразу к нескольким селекторам, то селекторы перечисляются через запятую:

```
a, img {
  /* стили для ссылок и изображений */
}
```

Селекторы по классам

Можно задавать стили по классу элемента. 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.

Синтаксис CSS позволяет выбирать элементы 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.

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

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

CSS
```
ul.target {...} /* выбор всех тегов ul с классом target */
```

Если у элемента задано несколько классов, в HTML и в CSS-селекторе они могут идти в разном порядке.

HTML
```
<span class="text green"></span>
<p class="green text"></p>
```

CSS
```
.text.green {...} /* выбор элементов с двумя классами: text и green */
```

Количество классов в селекторе может быть любым:

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

CSS
```
span.underlined.red.big.text {...} /* выбор тегов span с четырьмя классами: underlined, red, big и text */
```

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.

Читать их проще всего справа налево:

```
/* выбрать все теги strong внутри тегов p */
p strong { ... }

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

/* выбрать все ссылки внутри элементов с классом .menu,
   которые лежат внутри элементов с классом .footer */
.footer .menu a { ... }
```

Соседние селекторы

Соседние селекторы используются для расположенных рядом элементов.

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`.

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

Дочерние селекторы

Любые элементы, расположенные внутри родительского элемента называются потомками. А дочерними элементами являются 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.

Если нужно задать стили только для дочерних элементов используется 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.

Псевдокласс добавляется к селектору с помощью символа `:`. Например:

```
a:visited { ... }
li:last-child { ... }
.alert:hover { ... }
```

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

```
li:last-child { ... }
```

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

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

Селекторы с псевдоклассами хорошо сочетаются с контекстными селекторами.

Псевдоклассы состояний

Благодаря некоторым селекторам можно добавлять в интерфейс динамику и интерактивность.

Псевдокласс `:hover` позволяет выбрать элемент, когда на него наведён курсор мыши и кнопка мыши не нажата. Пример:

```
a:hover { ... }
```

Существуют специальные псевдоклассы для ссылок:

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

Псевдокласс `:focus` позволяет выбрать элемент, который в данный момент в фокусе.

Селекторы атрибутов

Селекторы атрибутов позволяют выбирать элементы по любым атрибутам. Они записываются с использованием квадратных скобок: `элемент[атрибут]`. Примеры селекторов:

```
input[checked] { ... }
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

Для атрибута `id` существует специальный селектор. Он 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.

HTML
```
<p id="greeting">Приветствие!</p>
```

CSS
```
#greeting { ... }
```

Продолжить