

Конспект «Селекторы. Погружение».

Раздел 1

Объединение селекторов

Для того, чтобы выбрать элементы, которые одновременно удовлетворяют сразу нескольким условиям в CSS есть запись, которая фактически выполняет операцию логического умножения, «И». Селекторы, применяемые к одному элементу, в этом случае пишутся без пробелов:

```
.class1.class2 { }
```

Стили будут применяться ко всем элементам, которые одновременно имеют класс `class1` и `class2`. Например:

```
<div class="class1 class2">Блок с двумя классами</div>
```

Псевдоклассы

Псевдокласс `:not`

Псевдокласс `:not(селектор)` является отрицающим селектором. С его помощью можно выбрать элементы, которые *НЕ* содержат указанный селектор:

```
li:not(:last-child) { }
```

Этот селектор выберет все теги ``, *НЕ* являющиеся последними в их родителе.

В качестве селектора могут указываться псевдоклассы, теги, идентификаторы, классы и селекторы атрибутов. Нельзя использовать двойной псевдокласс `:not`, то есть конструкция `:not(:not(...))` не работает.

Также в комбинации с `:not` не применяются:

- объединение селекторов: например, `li:not(.heart.jack)` — некорректный селектор;
 - псевдоэлементы: `li:not(:after)` — неправильная запись;
 - селекторы-потомки, групповые селекторы или комбинации: например, нельзя писать `li:not(a span)` или `li:not(a + span)`.

Селектор `:not` можно комбинировать с другими.

Псевдокласс :nth-last-child

Псевдокласс `:nth-last-child` используется для добавления стиля к элементам на основе нумерации в дереве элементов. В отличие от псевдокласса `:nth-child`, отсчет ведется не от первого элемента, а от последнего.

Псевдоклассы :first-of-type и :last-of-type

Псевдокласс `:first-of-type` выбирает первый дочерний элемент родителя с учетом типа элементов.

Псевдокласс `:last-of-type` выбирает *последний* дочерний элемент родителя с учетом типа.

Пример записи:

```
ul:last-of-type {  
  
    ...  
}
```

Псевдоклассы :nth-of-type и :nth-last-of-type

Псевдокласс `:nth-of-type` работает почти так же, как и `:nth-child`. Разница заключается в том, что он учитывает тип элемента. При помощи данного псевдокласса можно выбрать определённый по счёту дочерний элемент определённого типа.

Псевдокласс `:nth-last-of-type` также выбирает элементы по их расположению, учитывая их тип, но отсчет ведется от конца.

Конспект «Селекторы. Погружение».

Раздел 2

Селектор последующих элементов

Селектор `селектор1 ~ селектор2` применяет стили к элементу, подходящему под `селектор2`, только если перед ним расположен элемент, подходящий под `селектор1`. Оба элемента должны принадлежать одному родителю.

Между элементами `селектор1` и `селектор2` могут находиться другие элементы.

Псевдокласс :empty

Псевдокласс `:empty`, выбирает только те теги, у которых нет дочерних элементов (в том числе текстовых узлов).

Псевдокласс `:only-child`

Псевдокласс `:only-child` используется, когда нужно прописать индивидуальные стили для элемента, который является единственным дочерним элементом внутри родительского контейнера.

Псевдоклас `:only-of-type`

Псевдокласс `:only-of-type` используется, когда нужно прописать индивидуальные стили для элемента, который является единственным дочерним элементом указанного типа внутри родительского контейнера.

Псевдоэлементы `::before` и `::after`

Псевдоэлемент `before` позволяет с помощью CSS добавить *псевдотег* внутрь другого элемента и оформить его. Содержимое псевдотега задаётся с помощью свойства `content`.

Ведёт себя псевдотег так же, как обычный `` с текстом. Ему можно задавать дополнительные стили.

Чтобы псевдоэлемент появился, ему необходимо задать свойство `content`. Достаточно даже пустой строки в значении свойства — `content: ""`;

Псевдоэлемент `after` аналогичен `before`. Отличие заключается в том, что он добавляет псевдотег не в начало, а в конец элемента.

Псевдоэлементы `before` и `after` можно использовать одновременно.

Псевдоэлементы `::first-line` и `::first-letter`

Псевдоэлемент `first-line` задает стиль первой строки форматированного текста. В правилах стиля допустимо использовать только свойства, относящиеся к шрифту, изменению цвета текста и фона.

Пример использования:

```
p::first-line { }
```

Аналогично псевдоэлемент `first-letter` определяет стиль первого символа в тексте элемента, к которому добавляется. К этому псевдоэлементу могут применяться только

стилевые свойства, связанные со свойствами шрифта, полями, отступами, границами, цветом и фоном.

Пример использования:

```
p::first-letter { }
```