**Простые селекторы**

В свою очередь, простые селекторы делятся на:

* + Селектор по тегу,
  + Селектор по идентификатору,
  + Селектор по атрибуту,
  + Селектор по классу.

Самый используемый вариант — последний.

**Кроме того, к этой группе относится универсальный селектор, который обозначается звёздочкой \* и предназначен для работы со всеми элементами страницы.**

1

**Селектор по тегу** позволяет выбрать на странице все элементы какого-либо типа, например, параграфы, заголовки, блоки и так далее. Для использования этого типа простых селекторов надо записать тег без скобок, как вы уже видели в примере с покраской шрифта заголовка.

h1 {

**color**: red;

}

2

**Селектор по классу**— самый распространённый вариант в современной вёрстке. Он позволяет разработчику самостоятельно выбрать элементы, к которым будет применён стиль. Правда, сначала придётся внести изменения в *HTML*. Представим, что вы готовитесь к переезду, и для того, чтобы подсчитать бюджет, накидали общий список покупок.

Теперь же настала пора отправляться за покупками, и нужно как-то выделить то, что вы собираетесь приобрести в продовольственном, хозяйственном и магазине для животных. Для этого в*HTML* после каждого тега li до закрывающей скобки нужно написать class="food", "household" или "vet".

**К названиям классов есть следующие требования:**

* Класс не может начинаться с цифры.
* Класс не может содержать пробел.
* Класс нужно называть буквами латинского алфавита.
* Класс чувствителен к регистру: Food и food — два разных класса.
* Класс может начинаться и содержать дефис ‐ и/или нижнее подчеркивание \_.

После дополнения классами внешний вид списка никак не изменится. Но стоит применить *CSS*, как страница заиграет новыми красками. Допустим, все пункты списка, относящиеся к еде, станут оранжевыми, все хозяйственные принадлежности — зелёными, предметы первой необходимости для кролика — голубыми. Для этого мы применим селекторы: .food, .household и .vet. **Точка перед названием — специальное обозначение, которое даёт CSS понять, что мы применяем стиль к классу.**

После похода в ближайшие магазины выяснилось, что некоторые пункты так просто не добыть. Для того чтобы их отметить, можно ввести дополнительный класс. Например, not-bought. Добавлять его в *HTML* нужно к существующим классам через пробел в те же кавычки. После этого можно отметить недостающие покупки, допустим, красным цветом фона.

Обратите внимание: селекторы можно записывать последовательно, без пробела. Например:

**.class1.class2** /\* элемент с двумя классами class1 и class2 \*/

div**#some-id.class1** /\* div с идентификатором some-id и классом class1\*/

Из всего списка покупок неохваченной классами осталась только переобивка дивана. Она предполагает не поход в магазин, а обращение в сервисную службу. Порядок действий в данном случае будет несколько отличаться. Для таких особенных элементов существует понятие идентификатор.

3

**Селектор по идентификатору** определяет уникальное имя элемента, которое используется для изменения его стиля и обращения к нему через скрипты. В *HTML* он обозначается при помощи букв id:

**Идентификатор**

<li id="Sofa">Переобивка дивана</li>

В *CSS* для записи селектора по идентификатору используется решётка.

**#Sofa** {

**color**: red;

}

Имя идентификатора выбирают, следуя тем же правилам, что и для классов:

* Не может начинаться с цифры.
* Не может содержать пробел.
* Нужно называть буквами латинского алфавита.
* Чувствителен к регистру: Sofa и sofa — два разных идентификатора.
* Может начинаться и содержать дефис - и/или нижнее подчеркивание \_.

Учитывая, что основное свойство идентификатора — уникальность, его применение для целей назначения стилей должно быть очень ограниченным. Излишняя специфичность приведёт к большей трудоёмкости (только представьте, сколько стилей придётся назначать идентификаторам в большом проекте). Кроме того, именно у идентификатора самый большой приоритет среди селекторов, то есть стили, назначенные при помощи селектора по идентификатору, очень сложно менять.

Основное применение идентификатора вы уже рассматривали в предыдущем модуле. Речь про навигацию по странице и создание якорей. Список покупок при помощи этого метода можно переформатировать следующим образом.

После появления в списке навигации стало очень легко стилизовать отдельные маленькие списки. А вот как быть с самим меню? Можно использовать селектор по тегу а, но мы подразумеваем, что проекты, с которыми вам предстоит работать, будут больше списка покупок, так что ссылок в них окажется не четыре. Можно каждой ссылке присвоить класс, но в данном случае есть прямой путь.

4

**Селектор по атрибуту** позволяет обратиться к элементу *HTML* по значению его атрибутов, записывается при помощи квадратных скобок.

В нашем примере мы будем обращаться к ссылкам, используя атрибут href и его значение.  Например, для того, чтобы изменить цвет фона ссылки на продукты, используем селектор a[href="#Food"].

Можно обратиться по атрибуту, зная только часть значения. Допустим, вы забыли, как по-английски хозтовары. *House*... что-то там, да ещё и не помню, с какой буквы. Тогда селектор можно записать следующим образом a[href \*= "ouse"] или a[href \*= ouse], а фон сделаем зелёным. Оператор \*= означает, что искомое может располагаться где угодно в значении атрибута.

Есть отдельные операторы, которые позволяют обратиться к атрибуту, зная только окончание значения $=, или только его начало ^=.  С другими вариантами селекторов по атрибуту вы можете познакомиться в [документации](https://developer.mozilla.org/en-US/docs/Web/CSS/Attribute_selectors).

## ****Комбинированные селекторы****

Комбинированные селекторы образуются из простых селекторов по специальным правилам.

**Простой комбинатор (запятая)**выбирает все совпадающие узлы.

Синтаксис: селектор-1, селектор-2

Пример: div, p выберет оба элемента: и <div>, и <p>.

**Комбинатор потомков** **(пробел)** выбирает элементы, которые находятся внутри указанного элемента (вне зависимости от уровня вложенности).

Синтаксис: селектор-1 селектор-2 (через пробел)

Пример: div p выберет все элементы <p>, которые находятся внутри элемента <div>.

**Комбинатор дочерних элементов (>)**, в отличие от пробела, выбирает только дочерние элементы.

Синтаксис: селектор-1 > селектор-2

Пример: div > img выберет только дочерние, первого уровня вложенности, элементы <img>.

**Комбинатор соседних элементов (~)** выбирает элементы, которые находятся на этом же уровне вложенности, после указанного элемента, с тем же родителем.

Синтаксис: селектор-1 ~ селектор-2

Пример: p ~ .cat выберет все элементы с классом cat, которые находятся после элемента <p> внутри одного родителя.

**Комбинатор следующего соседнего элемента (+)** выбирает элемент, который находится сразу после указанного элемента, внутри одного родителя.

Синтаксис: селектор-1 + селектор-2

Пример: .header + .container выберет элемент с классом container, который находится непосредственно после элемента с классом header.