Селекторы и их комбинации

Селекторы типа В качестве селектора может выступать любой элемент HTML, для которого определяются правила форматирования, такие как: цвет, фон, размер и др.

*E { Описание правил стиля }*

Здесь E имя произвольного элемента. Следует понимать, что хотя правила стиля можно применить к любому элементу, результат будет заметен только для элементов, которые непосредственно отображаются в контейнере <body>.

***p {   
 text-align: justify; /\* Выравнивание по ширине \*/  
 color: green; /\* Зелёный цвет текста \*/  
 }***

Идентификаторы Идентификатор определяет уникальное имя элемента, которое используется для изменения его стиля и обращения к нему через скрипты.

*#<Имя идентификатора> { Описание правил стиля }*

При описании идентификатора вначале указывается символ решётки (#), затем идёт имя идентификатора. Оно должно начинаться с латинского символа и может содержать в себе символ дефиса (-) и подчеркивания (\_). Использование русских букв в именах идентификатора недопустимо. В отличие от классов идентификаторы должны быть уникальны, иными словами, встречаться в коде документа только один раз.

Обращение к идентификатору происходит с помощью атрибута id, значением которого выступает имя идентификатора (id="header"). Символ решётки при этом уже не указывается.

***#help {  
 position: absolute; /\* Абсолютное позиционирование \*/  
 left: 160px; /\* Положение элемента от левого края \*/  
 top: 50px; /\* Положение от верхнего края \*/  
 width: 225px; /\* Ширина блока \*/  
 padding: 5px; /\* Поля вокруг текста \*/  
 background: #f0f0f0; /\* Цвет фона \*/   
 }***

Классы Классы применяют, когда необходимо определить стиль для одного или нескольких элементов веб-страницы. Элементы при этом могут быть разными, но иметь частично или полностью похожие элементы оформления. В коде HTML имя класса определяется через атрибут class.

*E.<Имя класса> { Описание правил стиля }  
.<Имя класса> { Описание правил стиля }*

Здесь E — обозначает любой элемент. Имена классов должны начинаться с латинского символа и могут содержать в себе символ дефиса (-) и подчёркивания (\_). Использование русских букв в именах классов недопустимо. Чтобы указать в коде HTML, что элемент используется с определённым классом, к тегу добавляется атрибут class.

***p { /\* Обычный абзац \*/   
 text-align: justify; /\* Выравнивание текста по ширине \*/}  
p.cite { /\* Абзац с классом cite \*/  
 color: navy; /\* Синий цвет текста \*/  
 margin-left: 20px; /\* Отступ слева \*/  
 border-left: 1px solid navy; /\* Граница слева от текста \*/  
 padding-left: 15px; /\* Расстояние от линии до текста \*/ }***

Мультиклассы К любому элементу одновременно можно добавить несколько классов, перечисляя их в атрибуте class через пробел. В этом случае к элементу применяется стиль, описанный в правилах для каждого класса. Поскольку при добавлении нескольких классов они могут содержать одинаковые стилевые свойства, но с разными значениями, то берётся значение у класса, который описан в коде ниже.

*<E class="class1 class2 ...">*

Здесь E — обозначает любой элемент.

В стилях допустимо использовать запись следующего вида.

*.class1.class2 { Описание правил стиля }*

В таком случае стиль применяется только для элементов, у которых одновременно заданы классы class1 и class2, т. е. в коде HTML используется конструкция <E class="class1 class2>.

***.level1 { font-size: 1em; }  
 .level2 { font-size: 1.2em; }  
 .level3 { font-size: 1.4em; }  
 .level4 { font-size: 1.6em; }  
 .level5 { font-size: 1.8em; }  
 .level6 { font-size: 2em; }  
 a.tag { color: #468be1; /\* Цвет ссылок \*/ }***

Универсальный селектор Иногда требуется установить одновременно один стиль для всех элементов веб-страницы, например, задать шрифт или начертание текста. В этом случае поможет универсальный селектор, который соответствует любому элементу веб-страницы.

В CSS3 универсальный селектор применяется также в комбинации с пространством имён.

*ns|\* — все элементы в пространстве имён ns.  
\*|\* — все элементы во всех пространствах имён.  
|\* — все элементы без явного указания пространства имён.*

*\* { Описание правил стиля }*

Для обозначения универсального селектора применяется символ звёздочки (\*). В некоторых случаях указывать универсальный селектор не обязательно. Так, например, записи \*.class и .class являются идентичными по своему результату.

***\* {   
 margin: 0;  
 padding: 0; /\* Убираем отступы и поля для всех элементов \*/  
 }***

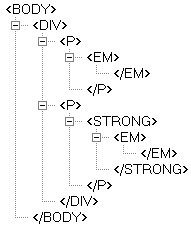
Вложенные селекторы При создании веб-страницы часто приходится вкладывать одни элементы внутрь других. Чтобы стили для этих элементов использовались корректно, помогут вложенные селекторы. Например, задать стиль для <b> только когда он располагается внутри контейнера <p>. Таким образом можно одновременно установить стиль для отдельного элемента, а также для элемента, который находится внутри другого.

*E F { Описание правил стиля }*

Здесь E это родительский элемент, а F — дочерний элемент, расположенный в контейнере <E>. В этом случае стиль будет применяться к элементу <F>, когда соблюдается следующий код <E><F></F></E>. Не обязательно должно быть два элемента, допускается произвольный уровень вложения. При этом конструкция может записываться так: div div ul li {...}.

p b {  
 font-family: Times, serif; /\* Семейство шрифта \*/  
 font-weight: bold; /\* Жирное начертание \*/  
 color: navy; /\* Синий цвет текста \*/ }

Дочерние селекторы называется элемент, который непосредственно располагается внутри родительского элемента. Какой элемент выступает родителем, а какой его потомком легко выяснить с помощью дерева элементов — так называется структура отношений элементов документа между собой (рис. 1).



Здесь представлена вложенность элементов и их иерархия. Здесь дочерним элементом по отношению к <div> выступает элемент <p>. Вместе с тем <strong> не является дочерним для элемента <div>, поскольку он расположен в контейнере <p>.

*E > F { Описание правил стиля }*

Стиль применяется к элементу F, только когда он является дочерним для элемента E.

#menu { margin: 0; padding: 0; /\* Убираем отступы \*/ }  
#menu > li { list-style: none; /\* Убираем маркеры списка \*/  
width: 100px; /\* Ширина элемента в пикселах \*/  
background: #b3d9d2; /\* Цвет фона \*/  
 float: left; /\* Располагаем элементы по горизонтали \*/ }  
li > ul { list-style: none; /\* Убираем маркеры списка \*/  
margin: 0; padding: 0; /\* Убираем отступы вокруг элементов списка \*/  
border-bottom: 1px solid #666; /\* Граница внизу \*/  
padding-top: 5px; /\* Добавляем отступ сверху \*/ }  
li > a {display: block; /\* Ссылки отображаются в виде блока \*/  
font-weight: normal; /\* Нормальное начертание текста \*/  
font-size: 90%; /\* Размер шрифта \*/  
background: #fff; /\* Цвет фона \*/}

Соседние селекторы Соседними называются элементы веб-страницы, когда они следуют непосредственно друг за другом в коде документа.

*E + F { Описание правил стиля }*

Для управления стилем соседних элементов используется символ плюса (+), который устанавливается между двумя селекторами E и F. Пробелы вокруг плюса не обязательны. Стиль при такой записи применяется к элементу F, но только в том случае, если он является соседним для элемента E и следует сразу после него. Рассмотрим несколько примеров.

***<p>Lorem ipsum <b>dolor</b> sit amet.</p>***

Элемент <b> является дочерним по отношению к <p>, поскольку он находится внутри этого контейнера. Соответственно <p>выступает в качестве родителя <b>.

***<p>Lorem ipsum <b>dolor</b> <var>sit</var> amet.</p>***

Элементы <var> и <b> никак не перекрываются и представляют собой соседние элементы. То, что они расположены внутри контейнера <p>, не влияет на их отношение.

***<p>Lorem <b>ipsum </b> dolor sit amet, <i>consectetuer</i> adipiscing <tt>elit</tt>.</p>***

Соседними здесь являются элементы <b> и <i>, а также <i> и <tt>. При этом <b> и <tt> к соседним элементам не относятся из-за того, что между ними расположен контейнер <i>.

***b + i {  
color: red; /\* Красный цвет текста \*/  
}***

Родственные селекторы Родственные селекторы по своему поведению похожи на соседние селекторы (запись вида E + F), но в отличие от них стилевые правила применяются ко всем близлежащим элементам. К примеру, для селектора h1~p стиль будет применяться ко всем элементам <p>, располагающихся после заголовка <h1>. При этом <h1> и <p>должны иметь общего родителя, так что если <p> вставить внутрь <div>, то стили применяться уже не будут.

Здесь красный цвет текста будет установлен для всех абзацев.

***h1 ~ p { color: red; }  
<h1>Заголовок</h1>  
<p>Абзац 1</p>  
<p>Абзац 2</p>***

Здесь красный цвет текста будет установлен для первого и третьего абзацев. Ко второму абзацу стиль не применяется, поскольку <h1> и <p> не имеют общего родителя.

***h1 ~ p { color: red; }  
<h1>Заголовок</h1>  
<p>Абзац 1</p>  
 <div><p>Абзац 2</p></div>  
<p>Абзац 3</p>***

*E ~ F { Описание правил стиля }*

Для управления стилем родственных элементов используется символ тильды (~), он добавляется между двумя селекторами E и F. Пробелы вокруг тильды не обязательны. Стиль при такой записи применяется к элементу F в том случае, если он имеет того же родителя, что и элемент E и следует после него.

***img { display: none; }  
#switch:checked ~ img { display: block; }***

Селекторы атрибутов Устанавливает стиль для элемента, если задан специфичный атрибут. Его значение в данном случае не важно.

*[<атрибут>] { Описание правил стиля }  
E[<атрибут>] { Описание правил стиля }*

Стиль применяется к тем элементам, внутри которых добавлен указанный атрибут. Пробел между именем селектора и квадратными скобками не допускается.

***q {  
 font-style: italic; /\* Курсивное начертание \*/  
 quotes: "\00AB" "\00BB"; /\* Меняем вид кавычек в цитате \*/ }  
q[title] { color: maroon; /\* Цвет текста \*/ }***

[атрибут="значение"] Устанавливает стиль для элемента в том случае, если задано определённое значение специфичного атрибута.

*[атрибут="<значение>"] { Описание правил стиля }  
E[атрибут="<значение>"] { Описание правил стиля }*

В первом случае стиль применяется ко всем тегам, которые содержат указанное значение атрибута. А во втором — только к определённым селекторам.

***a[target="\_blank"] {   
background: url(/example/image/blank.png) 0 6px no-repeat; /\* Фоновый рисунок \*/  
padding-left: 15px; /\* Смещаем текст вправо \*/ }***

[атрибут^="значение"] Устанавливает стиль для элемента в том случае, если значение атрибута начинается с указанного текста.

*[атрибут^="<значение>"] { Описание правил стиля }  
E[атрибут^="<значение>"] { Описание правил стиля }*

В первом случае стиль применяется ко всем элементам, у которых значение атрибута начинаются с указанного текста. А во втором — только к определенным селекторам. Использование кавычек не обязательно.

*a[href^="http://"] {  
 font-weight: bold; /\* Жирное начертание \*/}*

[атрибут$="значение"] Устанавливает стиль для элемента в том случае, если значение атрибута оканчивается указанным текстом.

*[<атрибут>$="<значение>"] { Описание правил стиля }  
E[<атрибут>$="<значение>"] { Описание правил стиля }*

В первом случае стиль применяется ко всем элементам у которых значение атрибута завершается заданным текстом. А во втором — только к определенным селекторам.

***a[href$=".ru"] { /\* Если ссылка заканчивается на .ru \*/  
background: url(/example/image/ru.png) no-repeat 0 50%; /\* Добавляем фоновый рисунок \*/   
padding-left: 18px; /\* Смещаем текст вправо \*/}   
a[href$=".com"] { /\* Если ссылка заканчивается на .com \*/  
background: url(/example/image/com.png) no-repeat 0 50%;   
padding-left: 18px; }***

[атрибут\*="значение"] Возможны варианты, когда стиль следует применить к тегу с определённым атрибутом, когда частью его значения является некоторый текст. При этом точно не известно, в каком месте значения включён данный текст — в начале, середине или конце. В подобном случае следует использовать конструкцию \*=. Она определяет, что значение атрибута содержит указанный текст.

*[<атрибут>\*="<значение>"] { Описание правил стиля }  
E[<атрибут>\*="<значение>"] { Описание правил стиля }*

***[href\*="htmlbook"] {   
background: yellow; /\* Жёлтый цвет фона \*/}  
<p><a href="http://www.htmlbook.ru/html/">Теги HTML</a> |   
<a href="http://stepbystep.htmlbook.ru">Шаг за шагом</a> |   
<a href="http://webimg.ru">Графика для Веб</a></p>***

[атрибут~="значение"] Некоторые значения атрибутов могут перечисляться через пробел, например имена классов. Чтобы задать стиль при наличии в списке требуемого значения требуется использовать конструкцию ~=.

*[<атрибут>~="<значение>"] { Описание правил стиля }  
E[<атрибут>~="<значение>"] { Описание правил стиля }*

***[class~="block"] h3 { color: green; }   
<div class="block tag"> <h3>Заголовок</h3></div>***

[атрибут|="значение"] В именах идентификаторов и классов разрешено использовать символ дефиса (-), что позволяет создавать значащие значения атрибутов id и class.

*[<атрибут>|="<значение>"] { Описание правил стиля }  
E[<атрибут>|="<значение>"] { Описание правил стиля }*

Стиль применяется к элементам, у которых атрибут точно равен указанному значению или начинается с указанного значения, после которого идёт дефис.

***div[class|="block"] {  
background: #306589; /\* Цвет фона \*/ color: #acdb4c; /\* Цвет текста \*/  
padding: 5px; /\* Поля \*/}  
div[class|="block"] a {color: #fff; /\* Цвет ссылок \*/}***