



# Каскадные таблицы стилей - основа профессионального сайта

День 4. Часть 6

# Что нового у CSS3?

**CSS3** - это новый стандарт оформления HTML документов значительно расширяющий возможности предыдущего стандарта CSS2.1.

Многие возможности, которые были труднодоступны в CSS2.1, то есть требовали использования дополнительных внешних программ (таких как Adobe Photoshop), скриптов (таких как JavaScript) или специальных "хитростей" могут легко достигаться в CSS3 за счет использования новых свойств оформления.

В **CSS3** Вы можете:

- Создавать элементы со сглаженными углами;
- Создавать линейные и сферические градиенты;
- Более гибко оформлять фоновую картинку элементов;
- Добавлять к элементам и тексту элементов тени;
- Использовать небезопасные шрифты (не боясь при этом, что они будут не поддерживаться браузером пользователя);
- Создавать анимацию и различные эффекты переходов;
- Задавать цвета несколькими новыми способами и многое другое.

# Что нового у CSS3?

Спецификация **CSS3** делится на "модули". Старые спецификации разбивались на более мелкие части, к которым также были добавлены новые части. К сожалению, на сегодняшний день еще нет ни одного веб-браузера, который бы поддерживал все возможности новых версий **HTML 5.0** и **CSS 3.0**. Однако с переходом к более новым версиям веб-браузеров наблюдается постоянное наращивание их возможностей в плане поддержки указанных спецификаций.

К числу наиболее важных модулей **CSS3** относятся следующие:

- Selectors
- BoxModel
- Backgrounds&Borders
- TextEffects
- 2D/3D Transformations
- Animations
- MultipleColumnLayout

# Новые селекторы CSS3

В таблице представлены новые селекторы CSS3.

Селектор	Пример	Описание
<code>:not(x)</code>	<code>:not(div)</code>	Выберет все элементы на странице кроме элементов <code>div</code>
<code>[атрибут^=значение]</code>	<code>[src^="http://"]</code>	Выберет все элементы имеющие атрибут <code>src</code> со значением начинающимся на <code>"http://"</code>
<code>[атрибут\$=значение]</code>	<code>[src\$=".gif"]</code>	Выберет все элементы имеющие атрибут <code>src</code> со значением заканчивающимся на <code>".gif"</code>
<code>[атрибут*=значение]</code>	<code>[src*="picture"]</code>	Позволяет выбрать все элементы имеющие атрибут <code>src</code> со значением содержащим подстроку <code>"picture"</code>
<code>:last-child</code>	<code>div:last-child</code>	Позволяет выбрать все элементы <code>div</code> являющиеся последними элементами потомками в родительском
<code>элемент1~элемент2</code>	<code>div~p</code>	Выберет все элементы <code>div</code> находящиеся перед элементом <code>p</code>
<code>:enabled</code>	<code>:enabled</code>	Позволяет выбрать все работоспособные элементы ввода
<code>:disabled</code>	<code>:disabled</code>	Позволяет выбрать все неработоспособные элементы ввода

# Новые селекторы CSS3

В таблице представлены новые селекторы CSS3:

Селектор	Пример	Описание
<b>:first-of-type</b>	<code>div:first-of-type</code>	Позволяет выбрать все элементы <code>div</code> являющиеся первыми в родительском
<b>:last-of-type</b>	<code>div:last-of-type</code>	Позволяет выбрать все элементы <code>div</code> являющиеся последними в родительском
<b>:only-of-type</b>	<code>div:only-of-type</code>	Позволяет выбрать все элементы <code>div</code> , которые являются уникальными родительском
<b>:nth-child(x)</b>	<code>div:nth-child(3)</code>	Позволяет выбрать все элементы <code>div</code> являющиеся третьими по счету в родительском
<b>:nth-last-child(x)</b>	<code>div:nth-last-child(2)</code>	Позволяет выбрать все элементы <code>div</code> являющиеся вторыми элементами потомками в родительском с конца
<b>::selection</b>	<code>::selection</code>	Оформляет выделенный пользователем текст
<b>:root</b>	<code>:root</code>	Позволяет выбрать корневой элемент документа
<b>:empty</b>	<code>p:empty</code>	Позволяет выбрать пустые абзацы



# Псевдокласс :not(x)

Селектор `:not()` позволяет выбрать все элементы в атрибутах которых не найдено совпадение с аргументом `X` передаваемым в качестве параметра селектору. В спецификации используется термин «простой» селектор. "Простые" селекторы применяются к *одиночным элементам, селекторам атрибутов, классам, уникальным идентификаторам или псевдо-классам*.

Примеры простых селекторов:

```
1. body
2. [value="foo"]
3. .foo
4. #foo
5. :hover
```

В рекомендуемых браузерах можно проверить применение данного селектора:

```
1. div:not(.home) { ... }
```

Здесь мы выбираем все блочные элементы `div`, не имеющие атрибут `class="home"`.

```
1. div:not(p) em { ... }
```

Выбираем все элементы `em`, находящиеся внутри элемента `div`, у которых нет родительского элемента `p`. Например, будет подходить `<div><strong><em>...</em></strong></div>` и `<div><em>...</em></div>`, но не `<div><p><em></em></p></div>`

```
1 <html>
2 <head>
3 <style type="text/css">
4 p{
5 color:aqua;
6 }
7 :not(p){
8 color: green;
9 }
10 </style>
11 </head>
12 <body>
13 <h1>Заголовок</h1>
14 <p>Абзац.</p>
15 <div>Текст внутри элемента div.</div>
16 <p>Абзац.</p>
17 </body>
18 </html>
```

**Заголовок**

Абзац.

Текст внутри элемента div.

Абзац.

# Селектор [атрибут^=значение]

Селектор	Пример	Описание
[атрибут^=значение]	[src^="http://"]	Выберет все элементы имеющие атрибут src со значением начинающимся на "http://"

```
1 <html>
2 <style type="text/css">
3 [href^="http://"] {
4 font-size:1.5em;
5 color:red;
6 }
7 </style>
8 <body>
9 <a href="wisdomweb.ru">Данный элемент не будет оформлен</a><br />
10 <a href="www.wisdomweb.ru/HTML/index.php">Данный элемент тоже не будет оформлен</a><br />
11 <a href="http://www.wisdomweb.ru">Данный элемент будет оформлен</a>
12 </body>
13 </html>
14
```

Данный элемент не будет оформлен

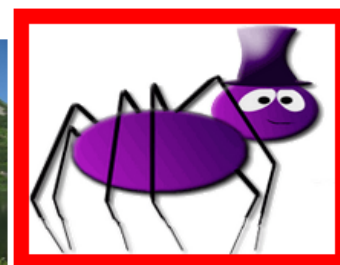
Данный элемент тоже не будет оформлен

Данный элемент будет оформлен

# Селектор [атрибут\$=значение]

Селектор	Пример	Описание
[атрибут\$=значение]	[src\$=".gif"]	Выберет все элементы имеющие атрибут src со значением заканчивающимся на ".gif"

```
1 <html>
2 <style type="text/css">
3 [src$=".gif"] {
4 border:10px #FF0000 solid;
5 }
6 </style>
7 <body>
8 
9 
10 
11 <p><b>Обратите внимание:
12 </b> будет оформлено только последнее изображение так как только оно имеет формат
13 .gif.</p>
14 </body>
15 </html>
16
```



**Обратите внимание:** будет оформлено только последнее изображение так как только оно имеет формат .gif.



# Селектор[атрибут=\*значение]

Селектор	Пример	Описание
[атрибут=*значение]	[alt*="Круглое"]	Позволяет выбрать все элементы имеющие атрибут alt со значением содержащим подстроку "Круглое".

```
1 <html>
2 <style type="text/css">
3 [alt*="Круглое"] {
4 border:10px #FF0000 solid;
5 }
6 </style>
7 <body>
8 
10 
12 
14 <p><b>Обратите внимание:</b>
15 будет оформлено только второе изображение так как только оно имеет в атрибуте alt
16 подстроку "Круглое".</p>
17 </body>
18 </html>
19
```



**Обратите внимание:** будет оформлено только второе изображение так как только оно имеет в атрибуте alt подстроку "Круглое".

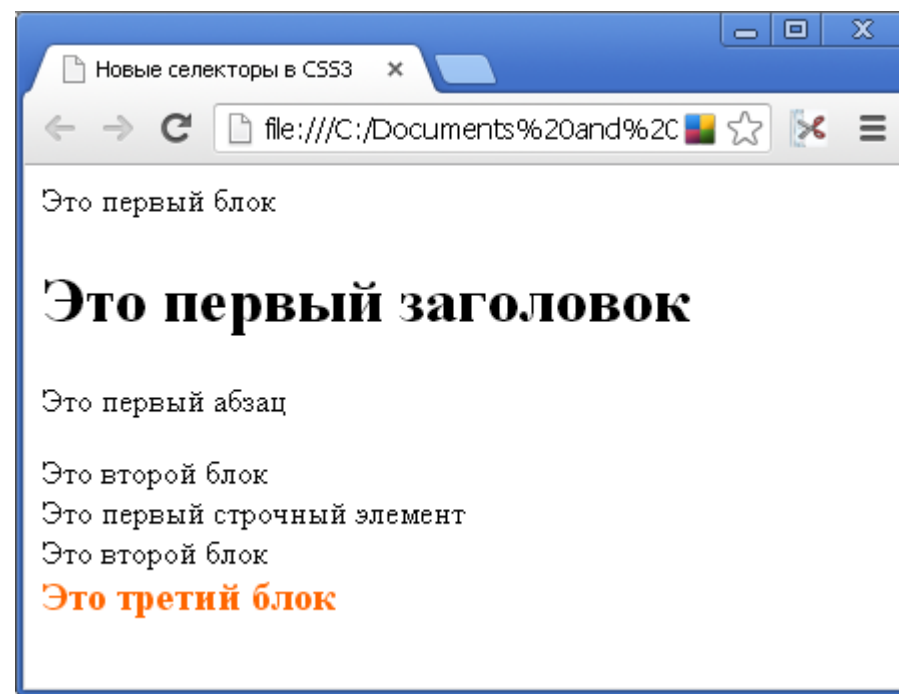
# Селектор :last-child

Селектор	Пример	Описание
<b>:last-child</b>	<code>div:last-child</code>	Позволяет выбрать все элементы <code>div</code> являющиеся последними элементами потомками в родительском

## html-код

```
<!DOCTYPE html>
<head>
<title>Новые селекторы в CSS3</title>
<style type="text/css">
div:last-child {
    color: #F60;
    font-size:20px;
    font-weight:bold;
}
</style>
</head>
<body>
<div>Это первый блок</div>
<h1>Это первый заголовок</h1>
<p>Это первый абзац</p>
<div>Это второй блок</div>
<span>Это первый строчный элемент</span>
<div>Это второй блок</div>
<div>Это третий блок</div>
</body>
</html>
```

## Отображение в браузере



# Селектор элемент1~элемент2

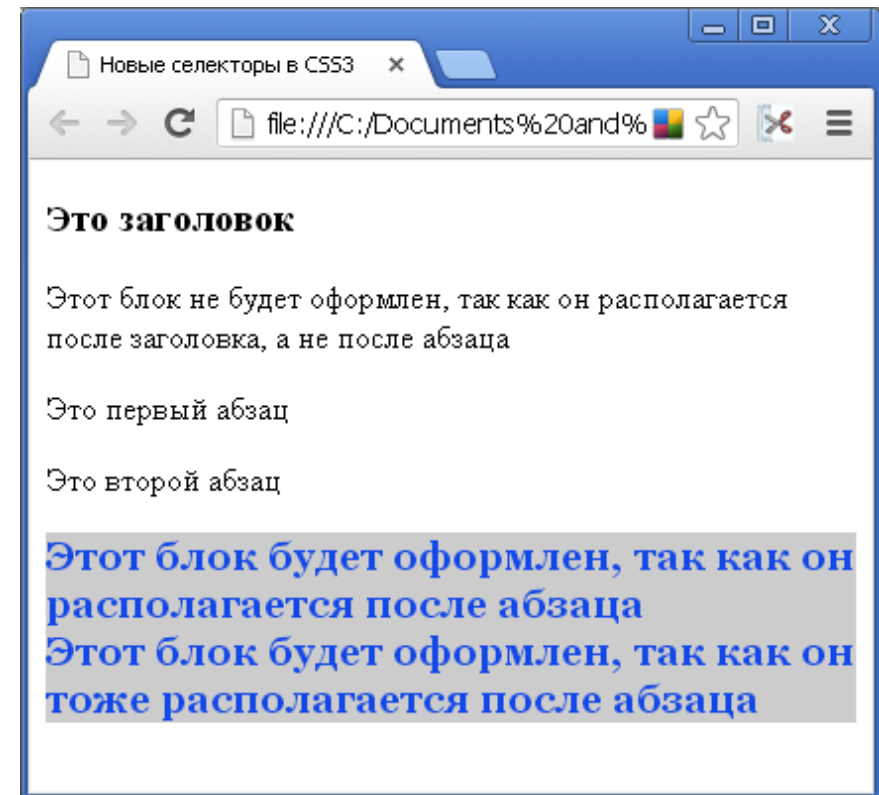
Селектор	Пример	Описание
элемент1~элемент2	p~div	Для выбора элемента2 следующего за элементом1, так же заданный стиль будет применен ко всем последующим элементам2, которые расположены далее. Выберет все элементы div находящиеся после элемента p

## html-код

```
<!DOCTYPE html>
<head>
<title>Новые селекторы в CSS3</title>
<style type="text/css">
p~div{
    color: #1348E8;
    font-size:20px;
    font-weight:bold;
    font-family:Georgia;
    background-color:#ccc;
}
</style></head>
<body>
<h3>Это заголовок</h3>
<div>Этот блок не будет оформлен, так как он располагается
после заголовка, а не после абзаца</div>
<p>Это первый абзац</p>
<p>Это второй абзац</p>
<div>Этот блок будет оформлен, так как он располагается
после абзаца</div>

<div>Этот блок будет оформлен, так как он тоже
располагается после абзаца</div>
</body></html>
```

## Отображение в браузере



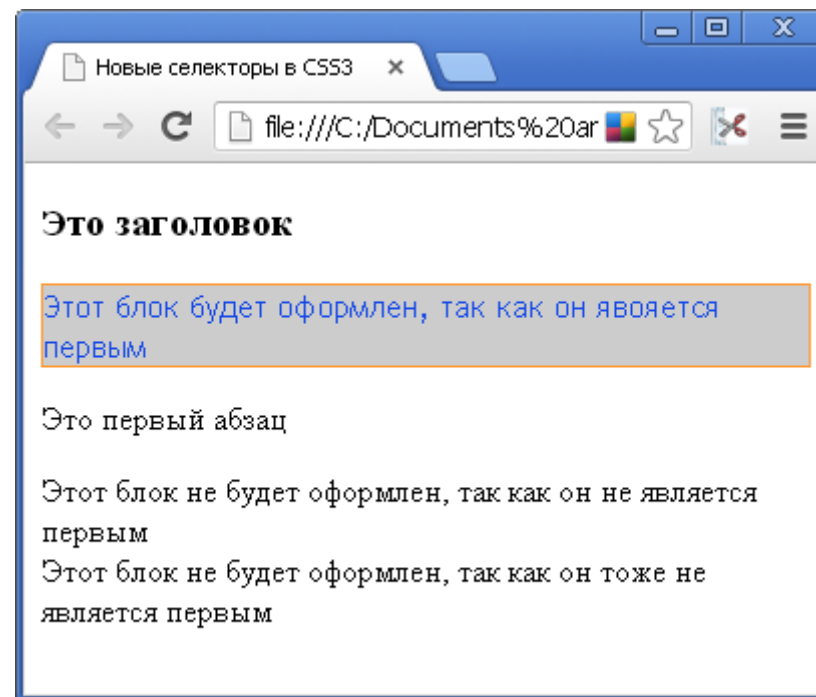
# Селектор :first-of-type

Селектор	Пример	Описание
<code>:first-of-type</code>	<code>div:first-of-type</code>	Позволяет выбрать все элементы <code>div</code> являющиеся первыми в родительском.

## html-код

```
<!DOCTYPE html>
<head>
<title>Новые селекторы в CSS3</title>
<style type="text/css">
div:first-of-type{
    color: #1348E8;
    font-size:15px;
    border: 1px #F93 solid;
    font-family: "Trebuchet MS";
    background-color:#CCC;
}
</style>
</head>
<body>
<h3>Это заголовок</h3>
<div>Этот блок будет оформлен, так как он является
первым</div>
<p>Это первый абзац</p>
<div>Этот блок не будет оформлен, так как он не
является первым</div>
<div>Этот блок не будет оформлен, так как он тоже не
является первым</div>
</body>
</html>
```

## Отображение в браузере



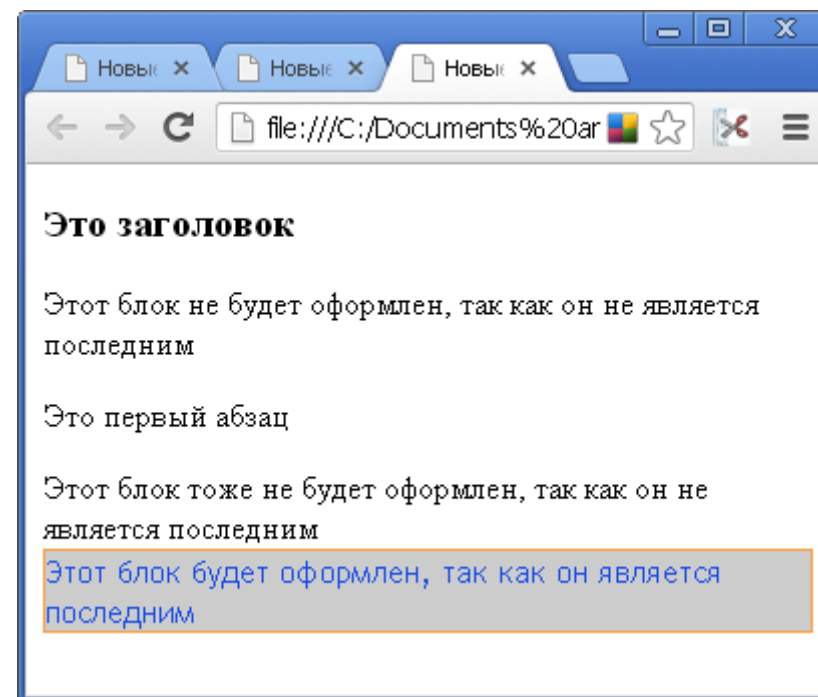
# Селектор :last-of-type

Селектор	Пример	Описание
<b>:last-of-type</b>	div:last-of-type	Позволяет выбрать все элементы div являющиеся последними в родительском.

## html-код

```
<!DOCTYPE html>
<head>
<title>Новые селекторы в CSS3</title>
<style type="text/css">
div:last-of-type{
    color: #1348E8;
    font-size:15px;
    border: 1px #F93 solid;
    font-family: "Trebuchet MS";
    background-color:#CCC;
}
</style>
</head>
<body>
<h3>Это заголовок</h3>
<div>Этот блок не будет оформлен, так как он не
является последним</div>
<p>Это первый абзац</p>
<div>Этот блок тоже не будет оформлен, так как он не
является последним</div>
<div>Этот блок будет оформлен, так как он является
последним</div>
</body>
</html>
```

## Отображение в браузере



# Селектор :only-of-type

Селектор	Пример	Описание
<code>:only-of-type</code>	<code>div:only-of-type</code>	Псевдокласс применяет заданный стиль к элементу, который является единственным дочерним элементом своего типа, в родителе. Позволяет выбрать все элементы <code>div</code> , которые являются уникальными в родительском.

```
1 <html>
2 <style type="text/css">
3   div p:only-of-type {
4     color:#FF0000;
5   }
6   div {
7     border:1px #000 solid;
8   }
9 </style>
10 <body>
11 <div>
12 <p>Данный абзац не будет выделен так как он не является единственным абзацем
13 в родительском.</p>
14 <p>Еще один абзац.</p>
15 </div>
16 <br />
17 <div>
18 <p>Данный абзац будет выделен так как он является единственным абзацем
19 родительском.</p>
20 <span>Я элемент span</span><br />
21 <b>Я элемент b</b>
22 </div>
23 </body>
24 </html>
25
```

Данный абзац не будет выделен так как он не является единственным абзацем в родительском.

Еще один абзац.

Данный абзац будет выделен так как он является единственным абзацем в родительском.

Я элемент span

Я элемент b

# Селектор :nth-child(x)

Селектор	Пример	Описание
<b>:nth-child(x)</b>	<code>div:nth-child(3)</code>	Позволяет выбрать все элементы <code>div</code> , являющиеся третьими по счету в родительском.

Псевдокласс **:nth-child(x)** применяет заданный стиль к элементу, который является "x" дочерним элементом, заданного типа, своего родителя. x - может быть числом, ключевым словом или формулой. Пример формулы ( $ax + b$ ):

```
1 <html>
2 <head>
3 <title></title>
4 <style type="text/css">
5   li:nth-child(3n-1){
6     list-style:none;
7   }
8 </style>
9 </head>
10 <body>
11 <p>Убираем отступ у второго, пятого, восьмого и т.д. элементов списка</p>
12 <ul>
13 <li>Первый</li>
14 <li>Второй</li>
15 <li>Третий</li>
16 <li>Четвертый</li>
17 <li>Пятый</li>
18 <li>Шестой</li>
19 <li>Седьмой</li>
20 <li>Восьмой</li>
21 </ul>
22 </body>
23 </html>
```

Убираем отступ у второго, пятого, восьмого и т.д. элементов списка

- Первый
- Второй
- Третий
- Четвертый
- Пятый
- Шестой
- Седьмой
- Восьмой



# Селектор :nth-last-child(x)

Селектор	Пример	Описание
<code>:nth-last-child(x)</code>	<code>p:nth-last-child(2)</code>	Псевдокласс <code>:nth-last-child</code> используется для добавления стиля к элементам на основе нумерации в дереве элементов. В отличие от псевдокласса <code>:nth-child</code> отсчёт ведётся не от первого элемента, а от последнего. Позволяет выбрать все элементы, являющиеся вторыми элементами-потомками в родительском с конца.

```
1 <html>
2 <head>
3 <style type="text/css">
4 p:nth-last-child(2){
5 color: red;
6 background-color: yellow;
7 }
8 </style>
9 </head>
10 <body>
11 <p>Первый абзац.</p>
12 <p>Второй абзац.</p>
13 <p>Третий абзац.</p>
14 <div><p>Первый абзац.</p><p>Второй абзац.</p></div>
15 </body>
16 </html>
```

Первый абзац.

Второй абзац.

Третий абзац.

Первый абзац.

Второй абзац.



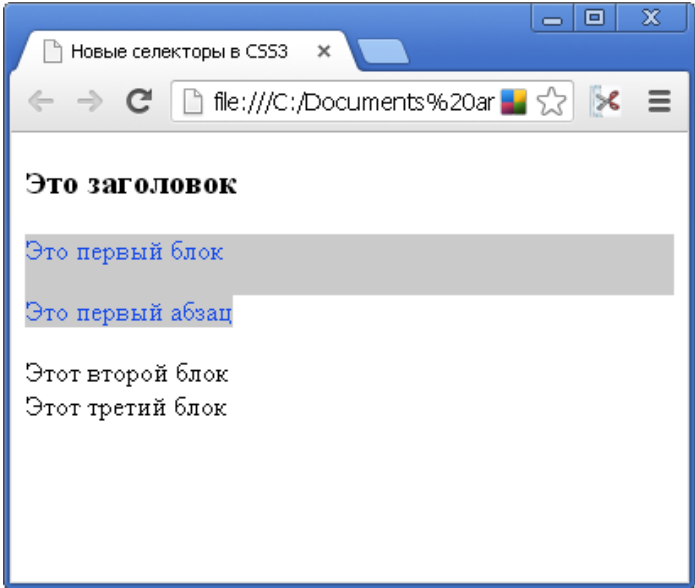
# Селектор ::selection

Селектор	Пример	Описание
<code>::selection</code>	<code>::selection</code>	Псевдоэлемент <code>::selection</code> применяет стиль к выделенному пользователем тексту. В правилах стилей допускается использовать следующие свойства: <code>color</code> , <code>background</code> и <code>background-color</code> . Оформляет выделенный пользователем текст

## html-код

```
<!DOCTYPE html>
<head>
<title>Новые селекторы в CSS3</title>
<style type="text/css">
::selection{
color: #1348E8;
background-color:#CCC;
}
</style>
</head>
<body>
<h3>Это заголовок</h3>
<div>Это первый блок </div>
<p>Это первый абзац</p>
<div>Этот второй блок </div>
<div>Этот третий блок </div>
</body>
</html>
```

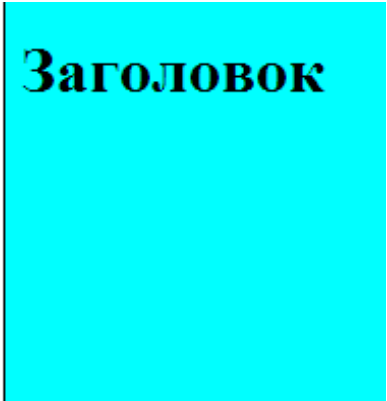
## Отображение в браузере








# Селектор :root

Селектор	Пример	Описание
<code>:root</code>	<code>:root</code>	Позволяет выбрать корневой элемент документа. В HTML этот селектор всегда соответствует элементу <code>&lt;html&gt;</code> .

```
1 <html>
2 <head>
3 <style type="text/css">
4 :root{
5 background-color: aqua;
6 }
7 </style>
8 </head>
9 <body>
10 <h1>Заголовок</h1>
11 </body>
12 </html>
```



# Браузеры и префиксы CSS

 WebKit	-webkit-
 Mozilla	-moz-
 Opera	-o-
 Konqueror	-khtml-
 Microsoft	-ms-
 Chrome	-chrome-

# Как работают браузерные префиксы?

CSS работает на практике с браузерными префиксами следующим образом. Возьмем свойство `border-radius` в качестве примера. Положим, мы хотим скруглить углы элемента с радиусом 10 пикселей; вот как это делается:

```
.foo {  
-webkit-border-radius: 10px;  
-moz-border-radius: 10px;  
border-radius: 10px;  
}
```

**Webkit** (движок, используемый в браузерах Chrome, Safari, и в Safari для мобильных устройств) и **Gecko** (движок браузера Firefox) поддерживают свойство `border-radius` посредством собственных префиксных свойств; Opera и IE10 поддерживают это свойство без префикса.

Префикс	Механизм рендеринга	Популярные браузеры на базе этого механизма рендеринга
-khtml-	KHTML	Konqueror
-ms-	Trident	Internet Explorer
-moz-	Mozilla	Firefox, Camino, Flock
-o-*	Presto	Opera, Opera Mobile, Opera Mini, браузер Nintendo Wii
-webkit-	Webkit	Safari, Safari для iOS, Chrome, браузер Android