Каскадные таблицы стилей CSS

Стиль – набор параметров, задающий внешнее представление объекта.

Таблица стилей – это инструмент языка html, предоставляющий возможности по улучшению внешнего отображения страницы средствами структурного программирования.

Таблица стилей содержит набор правил (стилей), описывающих оформление самой Web-страницы и отдельных ее фрагментов.

Каждый элемент на странице может иметь свой стиль (параграфы, заголовки, линии, текст...).

Набор стилей всех элементов называют **таблицей стилей**.

Преимущества использования CSS:

1. Разграничение кода и оформления.
2. Разное оформление для разных устройств.
3. Расширенные по сравнению с HTML способы оформления элементов.
4. Уменьшение размеров страниц.
5. Ускорение загрузки сайта.
6. Единое стилевое оформление множества документов.
7. Централизованное хранение

Уровни CSS:

Уровень 1 (CSS1) – принята 17.12.96,
Уровень 2 (CSS2) – принята 12.05.98
Уровень 2 (CSS2.1) – принята 07.06.11
Уровень 3 (CSS3) – разрабатывается

Правило таблиц CSS состоит из двух частей:

селектор {свойство1: значение; свойство2: значение}

1. **Селектор** – любой тэг html, для которого определение задает каким образом необходимо его форматировать.
2. **Определение** – состоит из двух частей:

✔свойства

✔значения

Пример

P { color: #0000FF }

P - это селектор. Он представляет собой имя тега <P>.

color - это свойство (атрибут) стиля. Он задает цвет текста.

#0000FF - это значение атрибута стиля color. Оно представляет код синего цвета, записанный в формате RGB

Правила CSS

1. Несколько параметров можно перечислять через двоеточие, либо задавать отдельно каждый параметр.
2. Если для одного селектора определяются одинаковые атрибуты, но с разными значениями, то использоваться будет тот, что указан в коде последним.

```
P { color: green }
P { color: red }
```

3. Свойства и их значения в CSS не чувствительны к регистру, поэтому их можно набирать как заглавными, так и строчными символами.

Способы связывания документа и таблиц стилей:

1. Связывание – позволяет использовать одну таблицу стилей для форматирования многих страниц html. Для этого таблица стилей хранится в отдельном файле с расширением CSS. Присоединяется к документам с помощью тэга <**LINK**>, задаваемого в разделе <**HEAD**>.

Пример:

```
<HEAD>
<LINK rel="stylesheet" type="text/css“
href="mystyles.css">
</HEAD>
```

В связываемом файле содержатся наборы правил CSS, определяющих форматирование документа.

Содержимое файла mystyles.css

```
BODY {background-color: #000000; color: #FFFFFF}
P {color: #0000FF}
EM {color: #00FF00; font-weight: bold}
.attention {color: #FF0000; font-style: italic}
.bigtext {font-size: large}
```

2. Внедрение – позволяет задавать все правила таблицы стилей непосредственно в самом документе в стилевом блоке, ограниченном тэгами **STYLE**:

Пример:

```
<HEAD>
<STYLE type="text/css">
    B {text-transform: uppercase}
    P {background-color: lightgray; text-align: center}
</STYLE>
</HEAD>
```

```
<html>
 <head><title>Заголовки</title>
  <style type="text/css">
   h1 { color: #a6780a; font-weight: normal; }
   h2 {
    color: olive;
    border-bottom: 2px solid black;
   }
  </style>
 </head>
 <body>
    <h1>Заголовок 1</h1>
  <h2>Заголовок 2</h2>
   </body>
</html>
```

3. Импортирование - позволяет встраивать в документ таблицу стилей, расположенную на сервере. Выполняется это с помощью свойства @import:url("mystyles.css").

4. Встраивание в тэги документа – позволяет изменить форматирование конкретных элементов страницы.

Пример:

<H1 style="color:red"> Заголовок 1. Отображается красным цветом

</H1>

Приоритеты CSS (от низшего к высшему), используемые при форматировании:

1. Связанная таблица стилей (по LINK)
2. Импортируемая таблица стилей (@import)
3. Правила с элементом html в качестве селектора
4. Правило с параметром CLASS в качестве селектора
5. Правило с параметром ID в качестве селектора
6. Встроенное в тэг html правило

Связанные, внедренные и импортированные таблицы стилей влияют на форматирование всех элементов документа.

Встраивание таблицы стилей в конкретный тэг влияет только на отображение его элементов.

Все способы встраивания таблиц стилей свободно могут сочетаться в одном документе.

Вложенный элемент наследует правила форматирования элемента-родителя

Группирование

1.Группирование селекторов

H1 {font-family: Verdana}
H2 {font-family: Verdana}

→ H1, H2 {font-family: Verdana}

2. Группирование определений

H2 {font-weight: bold}
H2 {font-size: 14pt}
H2 {font-family: Verdana}

↓

H2 (font-weight: bold; font-size: 14pt; font-family: Verdana}

3. Группирование свойств

H2 {font: bold 14pt Verdana}

При задании таблицы стилей можно свободно комбинировать все три правила группирования для уменьшения её размеров.

Наследование

Наследованием называется перенос правил форматирования для элементов, находящихся внутри других.

Такие элементы являются дочерними, и они наследуют некоторые стилевые свойства своих родителей, внутри которых располагаются.

Идентификаторы

Идентификатор элемента задается при помощи параметра **id**, в качестве значения которого указывается уникальное имя.

На странице можно использовать только один идентификатор с определенным именем, но несколько идентификаторов с разными именами.

Т.е. идентификатору соответствует только один элемент на странице.

Параметр ID можно применять к любому элементу документа.

#Имя идентификатора { свойство1: значение; свойство2: значение; ... }

Идентификаторы можно применять к конкретному тегу.

Тег#Имя идентификатора { свойство1: значение; свойство2: значение; ... }

Пример

```
<style type="text/css">
p{ color: blue}
p#green {color: green}
</style>
</head>
<body>
<p>Обычный абзац</p>
<p id=“green">Текст параграфа с идентификатором
</p>
</body>
```

Классы

Класс позволяет задать разные правила форматирования для одного элемента определённого типа или всех элементов документа. Имя класса указывается в селекторе правила после имени тега и отделяется от него точкой. Можно определить несколько правил форматирования для одного элемента и с помощью параметра **class** соответствующего тега применять разные правила форматирования.

Тег.Имя_класса { свойство1: значение; свойство2: значение; ... }

Пример

```
<STYLE type="text/css">
    H1.red {color: red}
    H1.blue {color:red; background-color: blue}
</STYLE>
<BODY>
<H1 class="red">Красный шрифт</H1>
<H1 class="blue">Красный шрифт на синем фоне</H1>
</BODY>
```

Если класс должен применяться ко всем элементам документа, то в селекторе задаётся имя класса с лидирующей точкой без указания конкретного элемента.

.Имя класса { свойство1: значение; свойство2: значение; ... }

Пример

```
<STYLE type="text/css">
    .red {color: red}
    .blue {color: red; background-color: blue}
</STYLE>
<BODY>
<P class="red">Красный шрифт</P>
<P class="blue">Красный шрифт на синем фоне</P>
<H1 class="blue">Заголовок  красного цвета на синем фоне</H1>
<H2 class="red">Заголовок красного цвета</H2>
</BODY>
```

Универсальный селектор

Используется, если требуется установить одновременно один стиль для всех элементов веб-страницы, например, задать шрифт или начертание текста.

```
* { Описание правил стиля }
```

Пример

```
<head>
<style type="text/css">
   * {
    font-family: Arial, Verdana, sans-serif; /* Рубленый шрифт для текста */
    font-size: 96%; /* Размер текста */
   }
  </style>
 </head>
 <body>
  <p>А здесь какой-то текст</p> </body>
```

Контекстные селекторы

Контекстный селектор состоит из простых селекторов разделенных пробелом.

Тег1 Тег2 { ... }

В этом случае стиль будет применяться к Тегу2 когда он размещается внутри Тега1.

```
<Тег1>
 <Тег2> ... </Тег2>
</Тег1>
```

Пример

```
<style type="text/css">
 P B {
  font-family: Times, serif; /* Семейство шрифта */
  font-weight: bold; /* Жирное начертание */
  color: navy; /* Синий цвет текста */
 }
</style>
```

Селекторы атрибутов

Простой селектор атрибута

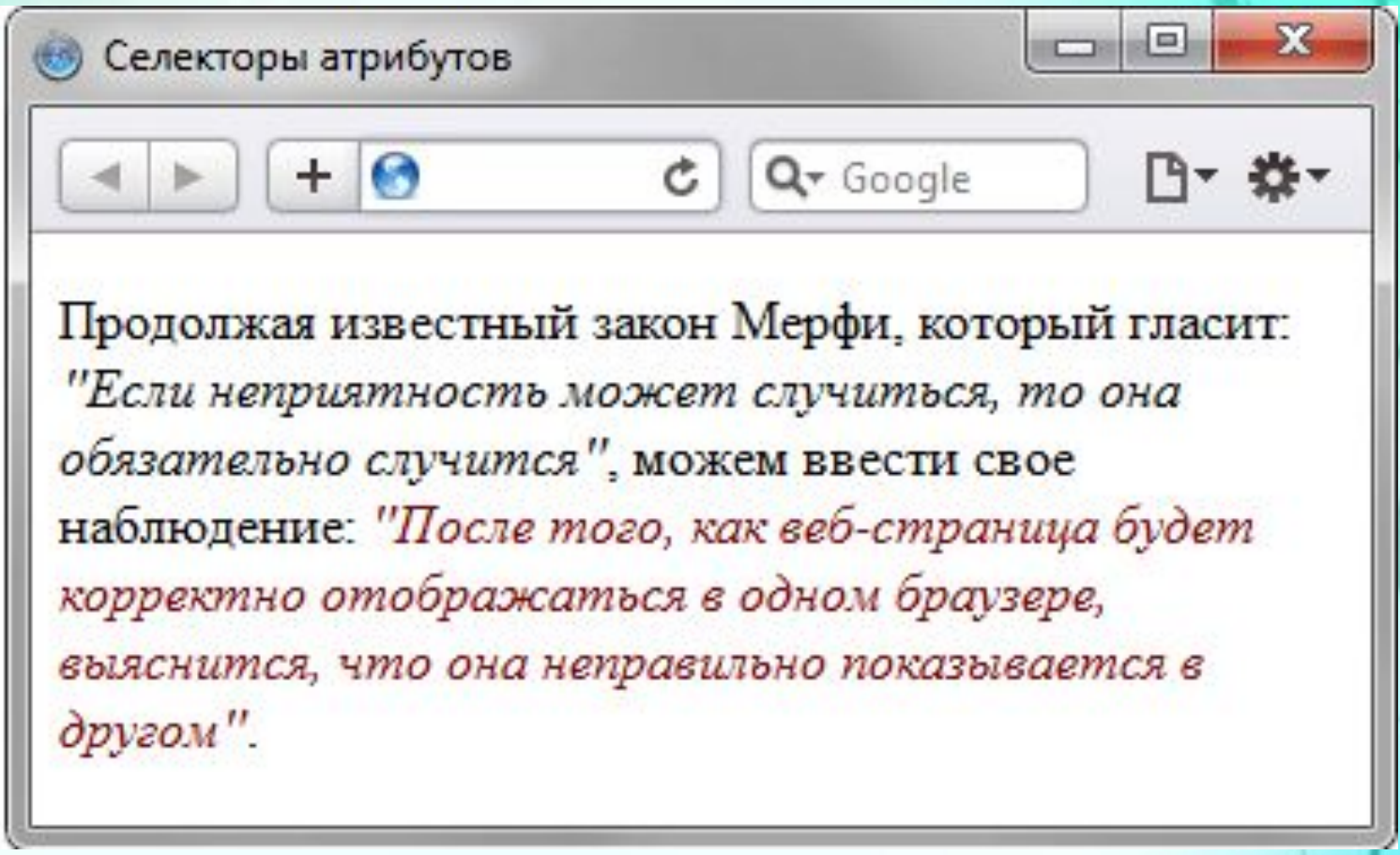
Устанавливает стиль для элемента, если задан специфичный атрибут тега. Его значение в данном случае не важно.

Селектор[атрибут] { Описание правил стиля }

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

Пример

```
<html> <head>
<style type="text/css">
  Q {font-style: italic}
  Q[title] {color: maroon}
 </style> </head>
<body>
 <p>Продолжая известный закон Мерфи, который гласит: <q>Если неприятность
  может случиться, то она обязательно случится</q>, можем ввести свое наблюдение:
 <q title="Из законов Фергюссона-Мержевича">После того, как веб-страница
  будет корректно отображаться в одном браузере, выяснится,
  что она неправильно показывается в другом</q>.</p>
</body></html>
```


Селекторы атрибутов
Google
Продолжая известный закон Мерфи, который гласит: "Если неприятность может случиться, то она обязательно случится", можем ввести свое наблюдение: "После того, как веб-страница будет корректно отображаться в одном браузере, выяснится, что она неправильно показывается в другом".

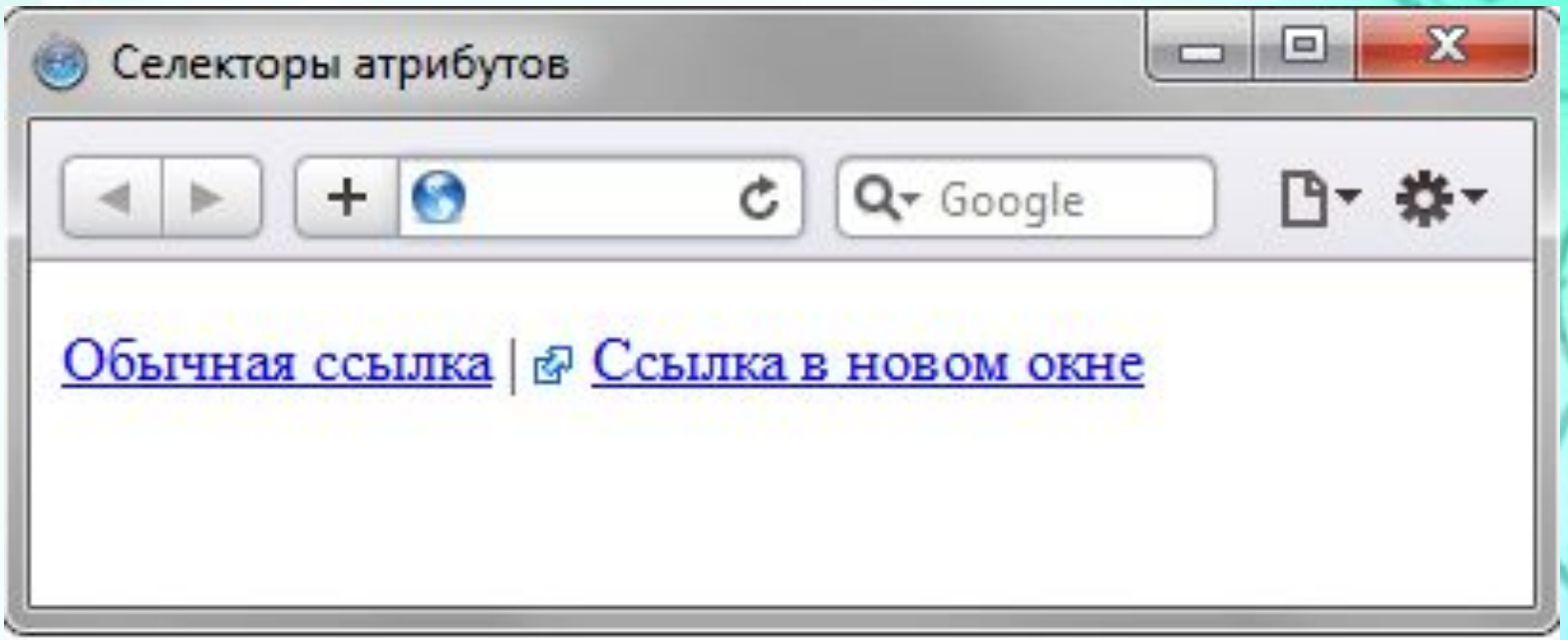
Атрибут со значением

Устанавливает стиль для элемента в том случае, если задано определенное значение специфичного атрибута.

Селектор[атрибут="значение"] { Описание правил стиля }

Пример

```
<html>
 <head>
 <style type="text/css">
   A[target="_blank"] {
    background: url(images/blank.png) 0 6px no-repeat; /* Параметры фонового рисунка */
    padding-left: 15px; /* Смещаем текст вправо */
   }
  </style>
 </head>
 <body>
   <p><a href="1.html">Обычная ссылка</a> |
   <a href="link2" target="_blank">Ссылка в новом окне</a></p>
 </body>
</html>
```


Селекторы атрибутов
Google
Обычная ссылка | Ссылка в новом окне

Значение атрибута начинается с определенного текста

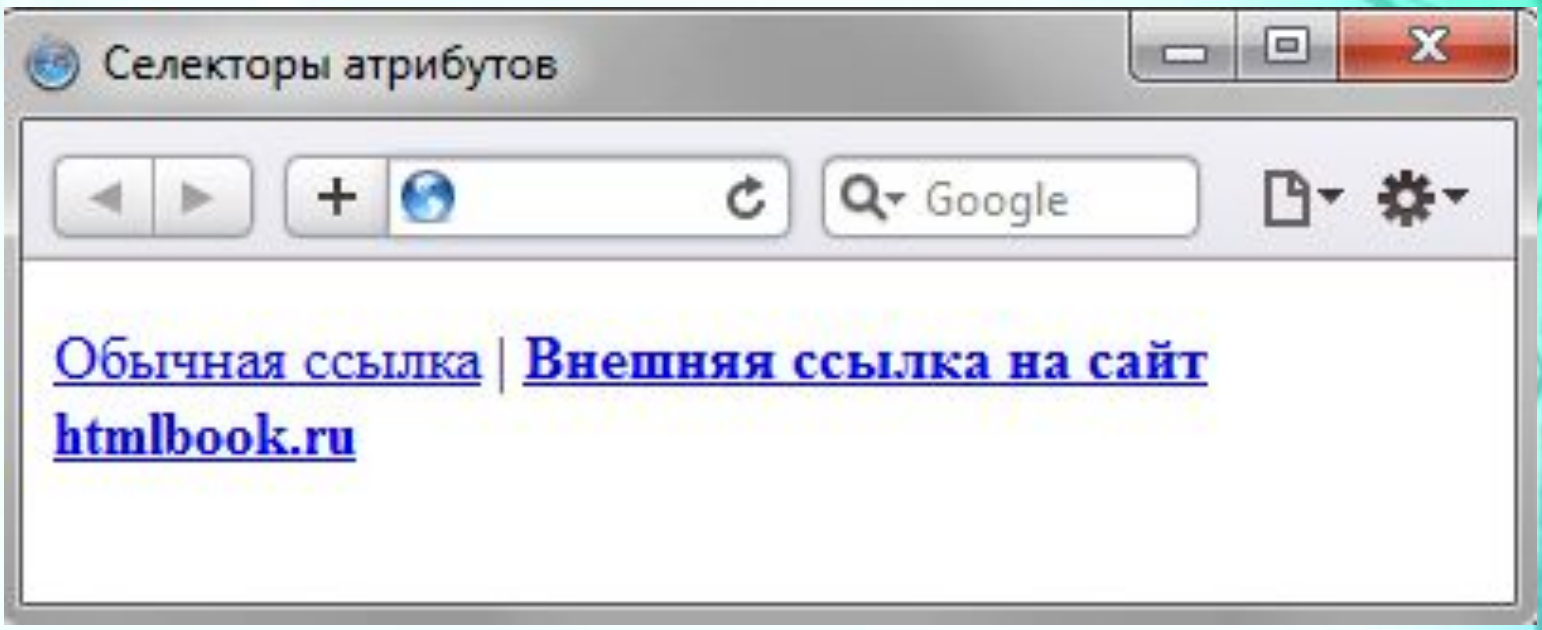
Устанавливает стиль для элемента в том случае, если значение атрибута тега начинается с указанного текста.

[атрибут^="значение"] { Описание правил стиля }

Селектор[атрибут^="значение"] { Описание правил стиля }

Пример

```
<html>
 <head>
<style type="text/css">
   A[href^="http://"] {
    font-weight: bold /* Жирное начертание */
   }
  </style>
 </head>
 <body>
 <p><a href="1.html">Обычная ссылка</a> |
  <a href="http://htmlbook.ru" target="_blank">Внешняя
  ссылка на сайт htmlbook.ru</a></p>
   </body>
</html>
```


Селекторы атрибутов
Google
Обычная ссылка | Внешняя ссылка на сайт htmlbook.ru

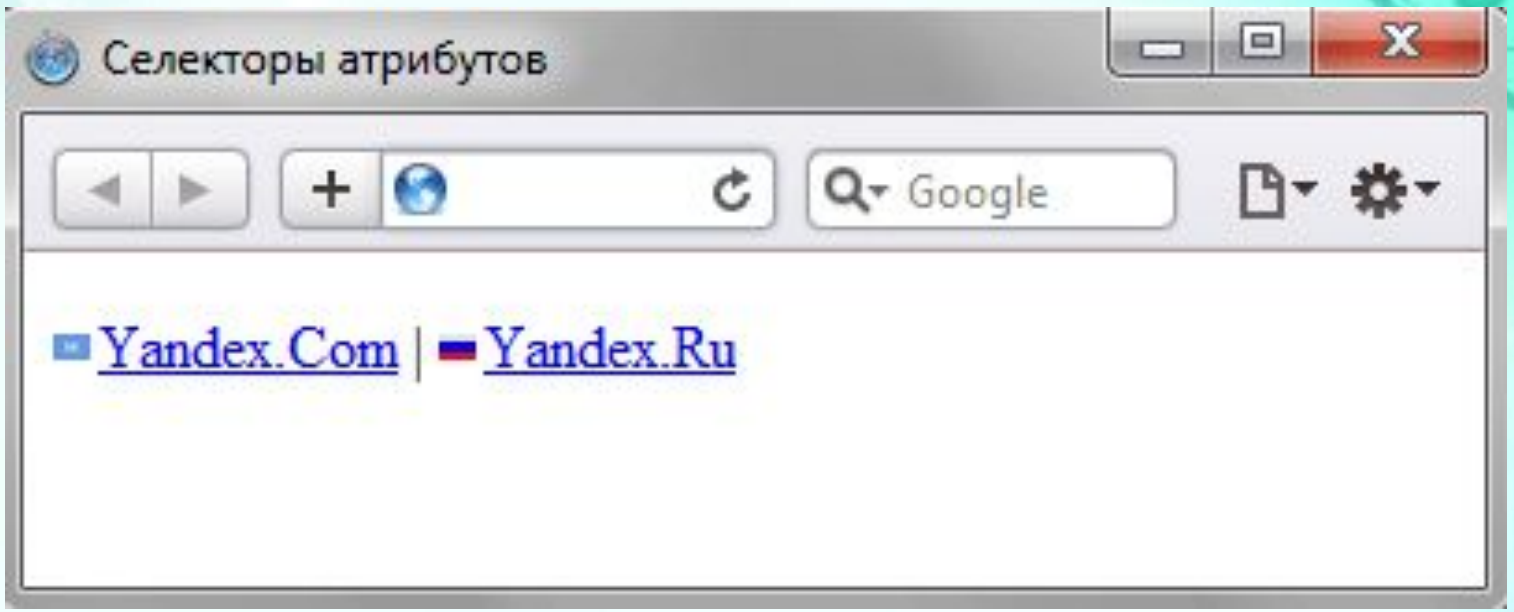
Значение атрибута оканчивается определенным текстом

Устанавливает стиль для элемента в том случае, если значение атрибута оканчивается указанным текстом.

Селектор[атрибут$="значение"] { Описание правил стиля }

Пример

```
<html> <head>
<style type="text/css">
  A[href$=".ru"] { /* Если ссылка заканчивается на .ru */
   background: url(images/ru.png) no-repeat 0 6px; /* Добавляем фоновый рисунок */
   padding-left: 12px; /* Смещаем текст вправо */ }
  A[href$=".com"] { /* Если ссылка заканчивается на .com */
   background: url(images/com.png) no-repeat 0 6px;
   padding-left: 12px;}
 </style>
</head>
<body>
   <p><a href="http://www.yandex.com">Yandex.Com</a> |
    <a href="http://www.yandex.ru">Yandex.Ru</a></p>
  </body></html>
```


Селекторы атрибутов
Google
Yandex.Com | Yandex.Ru

Псевдоклассы

Псевдоклассы определяют динамическое состояние элементов, которое изменяется со временем или с помощью действий пользователя, а также положение в дереве документа. Примером такого состояния служит текстовая ссылка, которая меняет свой цвет при наведении на нее курсора мыши.

Селектор:Псевдокласс { Описание правил стиля }

Допускается применять псевдоклассы к именам идентификаторов или классов (A.menu:hover {color: green}) и к контекстным селекторам (.menu A:hover {background: #fc0}). Если псевдокласс указывается без селектора впереди (:hover), то он будет применяться ко всем элементам документа.

Допускается применять псевдоклассы

✔к именам идентификаторов или классов

A.menu:hover {color: green};

✔к контекстным селекторам

.menu A:hover {background: #fc0}

Если псевдокласс указывается без селектора впереди (:hover), то он будет применяться ко всем элементам документа.

Условно все псевдоклассы делятся на три группы:

- ✔определяющие состояние элементов;
- ✔имеющие отношение к дереву элементов;
- ✔указывающие язык текста.

Псевдоклассы, определяющие состояние элементов

К этой группе относятся псевдоклассы, которые распознают текущее состояние элемента и применяют стиль только для этого состояния.

1. :active

Происходит при активации пользователем элемента.

2. :link

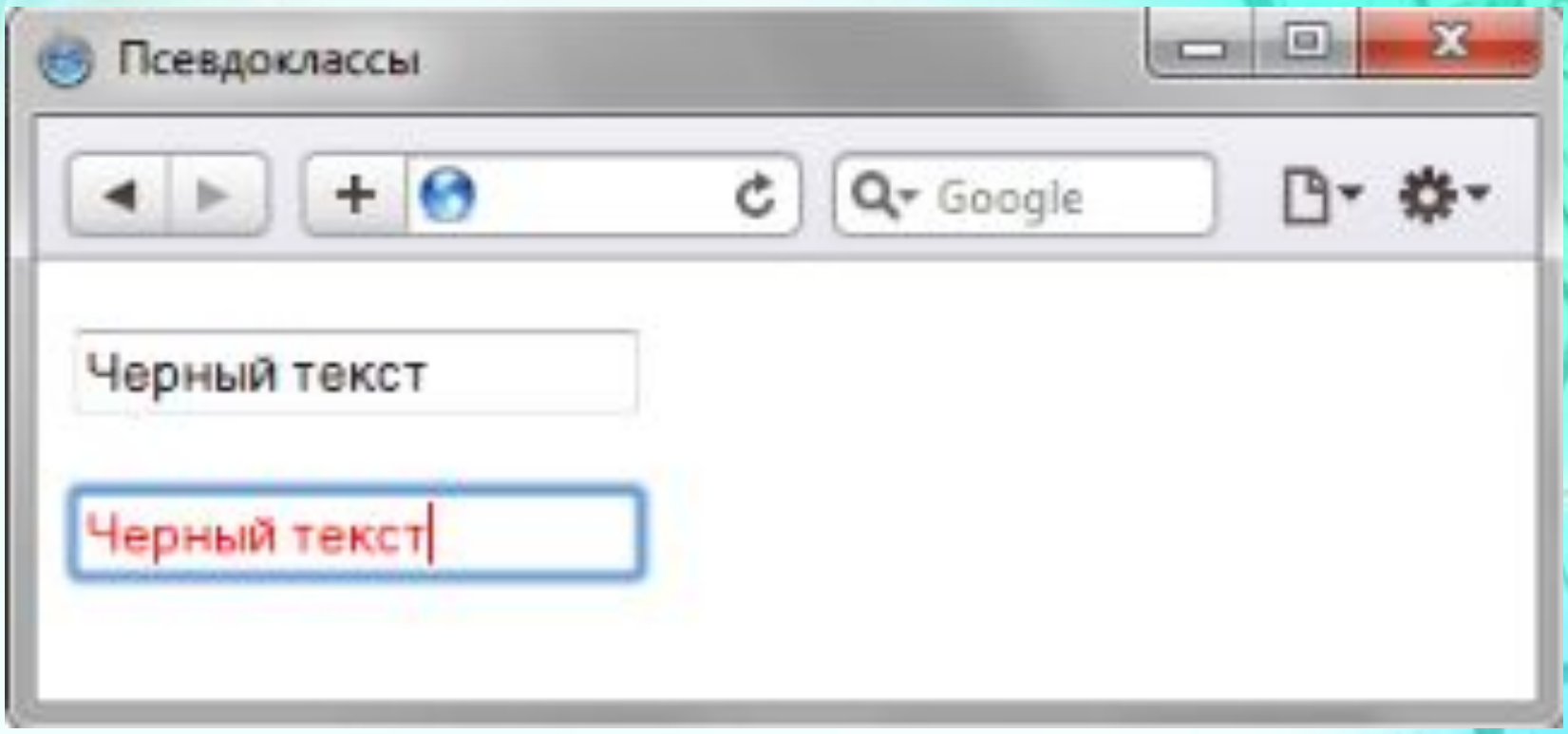
Применяется к непосещенным ссылкам.

3. :focus

Применяется к элементу при получении им фокуса.

Пример

```
<html>
 <head>
<style type="text/css">
   INPUT:focus {
    color: red; /* Красный цвет текста */
   }
  </style>
 </head>
 <body>
  <form action="">
   <p><input type="text" value="Черный текст"></p>
   <p><input type="text" value="Черный текст"></p>
  </form>
 </body>
</html>
```


Псевдоклассы
Google
Черный текст
Черный текст

4. :hover

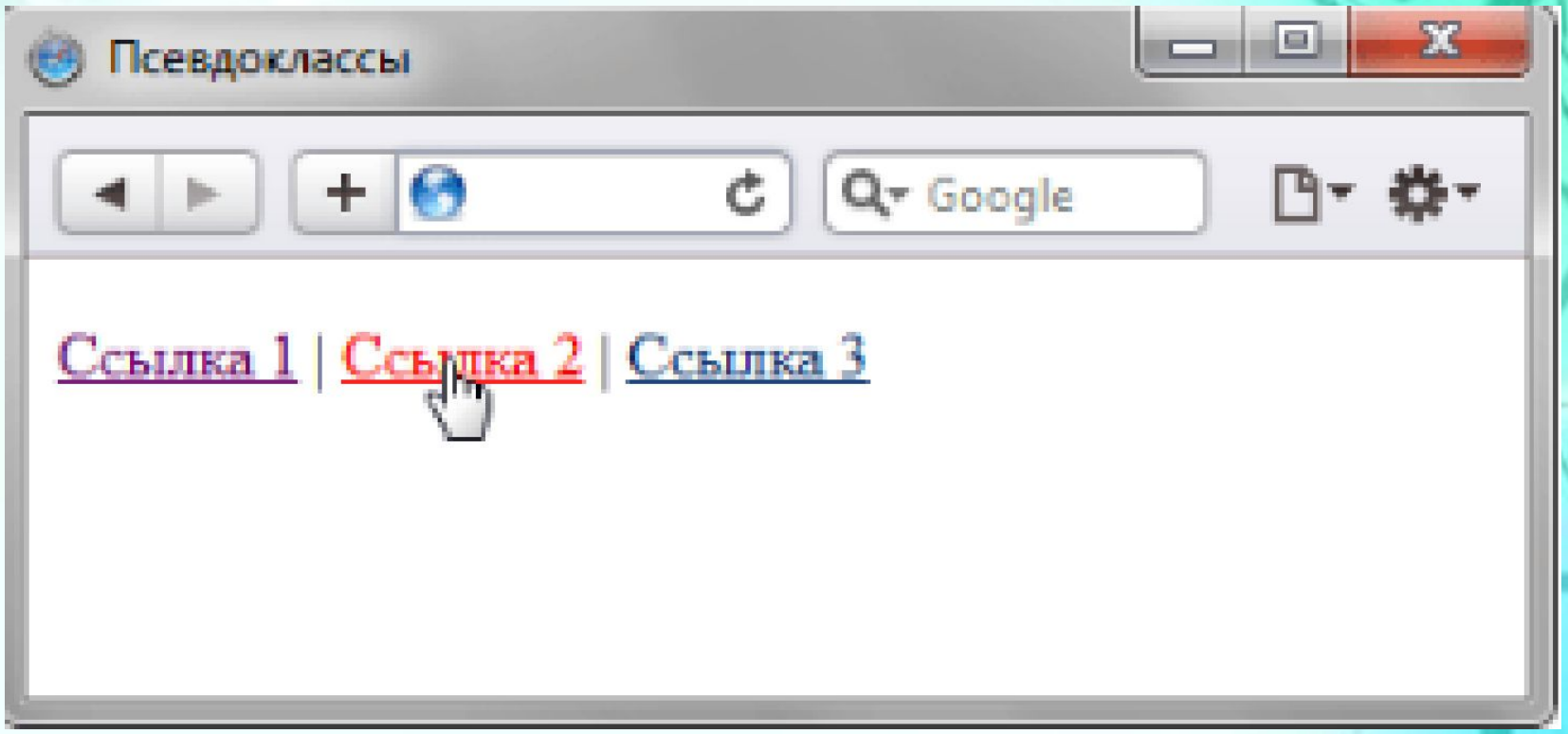
Псевдокласс :hover активизируется, когда курсор мыши находится в пределах элемента, но щелчка по нему не происходит.

5. :visited

Данный псевдокласс применяется к посещенным ссылкам.

Пример

```
<html> <head>
<style type="text/css">
  A:link {color: #036; /* Цвет непосещенных ссылок */ }
  A:visited {color: #606; /* Цвет посещенных ссылок */}
  A:hover {color: #f00; /* Цвет ссылок при наведении на них курсора мыши */}
  A:active {color: #ff0; /* Цвет активных ссылок */}
 </style> </head>
 <body>
  <p>
   <a href="1.html">Ссылка 1</a> |
   <a href="2.html">Ссылка 2</a> |
   <a href="3.html">Ссылка 3</a></p>
 </body></html>
```


Псевдоклассы
Google
Ссылка 1 | Ссылка 2 | Ссылка 3

Псевдокласс :hover не обязательно должен применяться к ссылкам, его можно добавлять и к другим элементам документа (например, к таблице, строки которой меняют свой цвет при наведении на них курсора мыши).

```
<style type="text/css">
  TR:hover {
   background: #fc0; /* Меняем цвет фона строки таблицы */
  }
</style>
```

	Пики	Трефы	Бубны	Червы
Чебурашка	5	2	4	2
Крокодил Гена	2	7	1	3
Шапокляк	5	4	3	1
Крыса Лариса	1	0	5	7