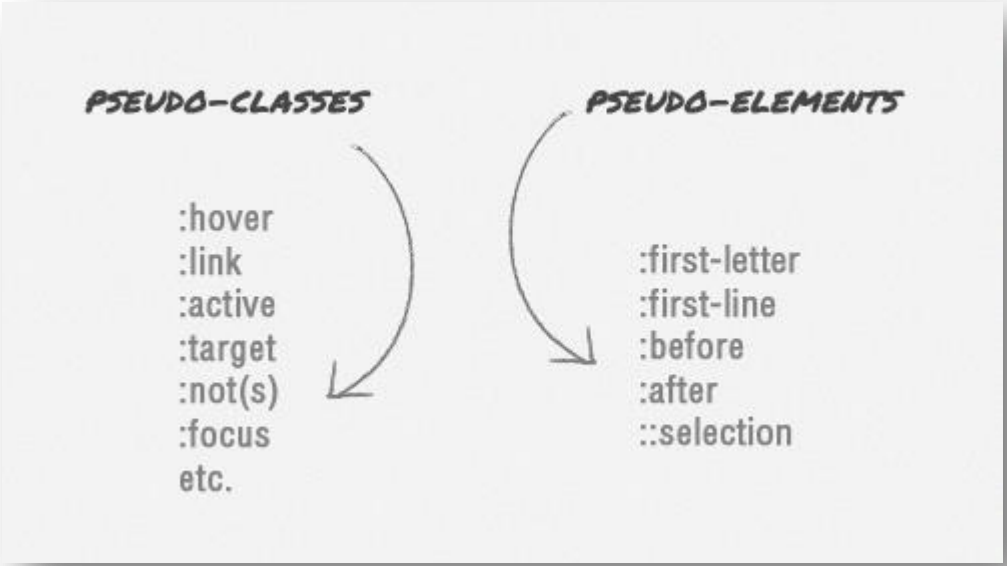
ЗАНЯТИЕ 5. ПСЕВДОКЛАССЫ, ПСЕВДОЭЛЕМЕНТЫ

Поскольку объекты страницы могут принимать разные состояния, нам нужны конструкции, которые бы позволяли с этими состояниями работать, а также нам потребуется работать не с целыми элементами, а с их частями. Для этого используются псевдоклассы и псевдоэлементы, о которых сегодня пойдет речь.



5.1 Псевдоэлементы

Псевдоэлементы позволяют работать с частями элементов, а также генерировать в css-файле элементы, которых нет в html-структуре.

Существует порядка 20 различных псевдоэлементов, однако не все они поддерживаются браузерами. Поэтому мы рассмотрим 4 основных, которые чаще всего используются и поддерживаются всеми браузерами:

:after

:before

:first-letter

:first-line

Общая структура синтаксиса псевдоэлементов:

```
selector:pseudo element { property: value; }
```

Вначале следует имя селектора, затем пишется двоеточие, после которого идёт имя псевдоэлемента. Каждый псевдоэлемент может применяться только к одному селектору, если требуется установить сразу несколько псевдоэлементов для одного селектора, правила стиля должны добавляться к ним по отдельности:

```
selector:before { property: value; }
selector:after { property: value; }
```

Рассмотрим детальнее каждый из элементов в списке.

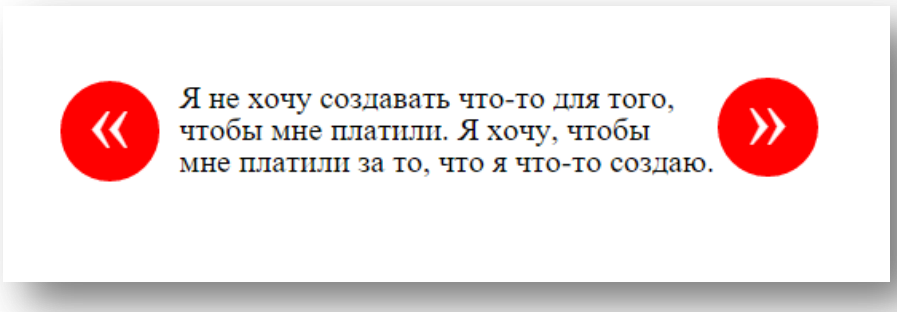
:before

Применяется для вставки определенного контента перед назначенным элементом, с помощью свойства content, которое и определяет какой именно контент будет вставлен.

:after

Аналогично :before, с разницей лишь в том, что контент вставляется после назначенного элемента.

Например, нам нужно добавить кавычки возле цитаты:



```css
.block1
{
    width: 400px;
    height: 100px;
    background: white;
    color: black;
}

.block1:before
{
    content: open-quote;
    background: red;
    display: block;
    float: left;
    width: 50px;
    height: 50px;
    border-radius:25px;
    color: white;
    font-size: 30pt;
    text-align: center;
    margin-left: 30px;
    margin-right: 10px;
}

.block1:after
{
    content: close-quote;
    background: red;
    display: block;
    float: right;
    width: 50px;
    height: 50px;
    border-radius:25px;
    color: white;
    font-size: 30pt;
    text-align: center;
    margin-top: -50px;
    margin-right: -10px;
}
```

Теперь разберемся в коде:

Псевдоэлементы :after и :before будут создаваться для класса block1. То есть мы выделим с двух сторон цитату, написанную в данном блоке.

Ставим тип контента для каждого псевдоэлемента: открытие и закрытие кавычек. Типом контента может быть также простой текст, либо картинка. Но в случае картинки мы оставляем свойство content пустым (content: " ";) и вставляем картинку с помощью background.

Если не прописать контент – содержимое псевдоэлемента будет отображаться некорректно, либо вообще не будет показано.

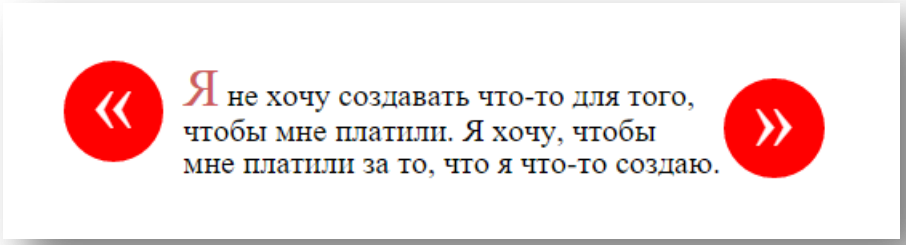
Если для псевдоэлемента нужно указать размеры, то следует явно задать display:block, т.к. по умолчанию псевдоэлемент имеет display:inline;

К псевдоэлементам можно применять такие же стили, как и к «реальным»: изменение цвета, добавление фона, регулировка размера шрифта, выравнивание текста и т.д.

:first-letter

Определяет стиль первого символа в тексте элемента, к которому добавляется. Это позволяет создавать в тексте буквицу и выступающий инициал.

Например, изменим цвет первой буквы в нашей цитате и увеличим ее размер:



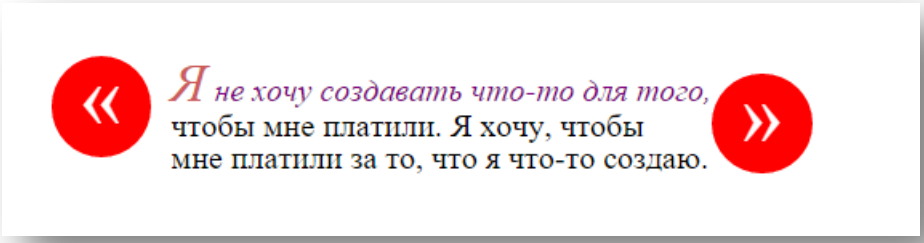

```
.block1 p
{
    color: black;
}

.block1 p:first-letter
{
    color: indianred;
    font-size: 20pt;
}
```

:first-line

Определяет стиль первой строки блочного текста. Длина этой строки зависит от многих факторов, таких как используемый шрифт, размер окна браузера, ширина блока, языка и т.д.

К псевдоэлементу :first-line могут применяться не все стилевые свойства. Допустимо использовать свойства, относящиеся к шрифту, изменению цвет текста и фона, а также: clear, line-height, letter-spacing, text-decoration, text-transform, vertical-align и word-spacing.

Изменим цвет для первой строки, а также стиль написания:



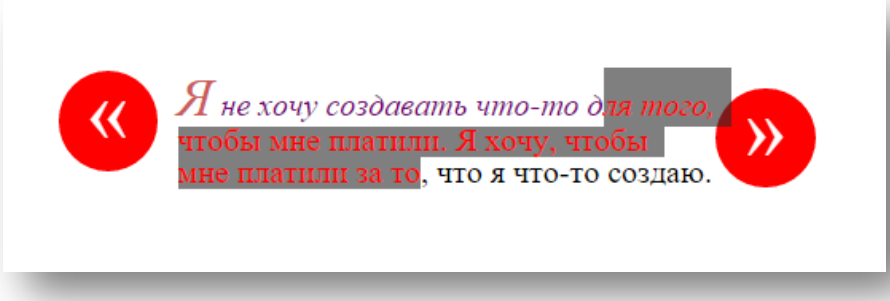

```css
.block1 p
{
    color: black;
}

.block1 p:first-letter
{
    color: indianred;
    font-size: 20pt;
}

.block1 p:first-line
{
    color: purple;
    font-style: italic;
}
```

Как видим, стиль применился для всех символов первой строки, исключая первый, т.к. ранее для него уже был определен стиль псевдоэлементом :first-letter.

Ну и в качестве плюшки рассмотрим ::selection

Не жизненно необходимый псевдоэлемент, но он позволяет красиво стилизовать выделенный текст. В правилах стилей допускается использовать следующие свойства: color, background и background-color.

Собственно, пример:



```
.block1 p:first-line
{
    color: purple;
    font-style: italic;
}

.block1 p::selection
{
    color: red;
    background: gray;
    font-style: bold;
}
```

Как видим, свойство font-style не применилось. Также мы не сможем выделить псевдоэлемент :first-letter.

Также существует еще множество менее используемых псевдоэлементов:

Псевдоэлемент ::-ms-value	Позволяет изменять стиль элементов формы, сделанных с помощью тега **<input>** или **<select>**, в браузере Internet Explorer.
Псевдоэлемент ::-ms-reveal	Задаёт стиль кнопки для просмотра пароля в поле input type="password". Кнопка исходно не видна и появляется при вводе пароля в правой части поля.
Псевдоэлемент ::-ms-fill	Задаёт стиль индикатора элемента **<progress>** в браузере Internet Explorer. Само значение индикатора и его положение меняется динамически посредством скриптов.
Псевдоэлемент ::-ms-expand	Задаёт стиль кнопки раскрытия списка, созданного с помощью тега **<select>** в браузере Internet Explorer.

Псевдоэлемент ::-ms-clear	Задаёт стиль кнопки для очистки текстового поля. Исходно эта кнопка не видна, она появляется в правой части поля только при вводе текста.
Псевдоэлемент ::-ms-check	Задаёт стиль переключателей (input type="radio") и флажков (input type="checkbox").
Псевдоэлемент ::-ms-browse	Позволяет задать стиль кнопки «Обзор» при загрузке файлов через <input type="file"> в Internet Explorer.
Псевдоэлемент ::-moz-selection	Применяется к выделенному пользователем фрагменту документа. Поддерживается только браузером Firefox.
Псевдоэлемент ::selection	Применяет стиль к выделенному пользователем фрагменту текста.

5.2 Псевдоклассы

Наверняка часто приходилось видеть как при наведении на пункт меню меняется цвет элемента, либо как ссылка, на которую вы перешли, подсвечивается другим цветом. Такие эффекты создаются с помощью псевдоклассов.

Псевдоклассы – это атрибуты, назначаемые строго к селекторам с намерением определить реакцию или состояние на какое-либо событие для данного селектора. То есть, это особые свойства, которые позволяют менять стиль элемента в зависимости от действий пользователя, а так же положения

этого элемента (тега) в общем потоке документа, что позволяет разбавить дизайн страницы некой динамикой и логикой.

Общая структура синтаксиса псевдоклассов:

```
selector:pseudo class { property: value; }
```

т.е. вам лишь надо поместить двоеточие между селектором и псевдоклассом.

Условно все псевдоклассы делятся на три группы:

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

Существуют следующие псевдоклассы:

Псевдокласс :invalid	Применяется к полям формы, содержимое которых не соответствует указанному типу.
Псевдокласс :read-only	Применяется к полям формы, у которых задан атрибут readonly.
Псевдокласс ::-moz-placeholder	Псевдокласс, с помощью которого задаётся стилевое оформление подсказывающего текста в Firefox.
Псевдокласс ::-webkit-input-placeholder	Псевдокласс, с помощью которого задаётся стилевое оформление подсказывающего текста в Chrome.
Псевдокласс :active	Определяет стиль активной ссылки.
Псевдокласс :checked	Применяется к элементам интерфейса, таким как

	переключатели (checkbox) и флажки (radio), когда они находятся в положение «включено»
Псевдокласс :default	Применяет стиль к элементам форм, которые установлены по умолчанию в группе похожих элементов.
Псевдокласс :disabled	Применяет стиль к заблокированным элементам форм.
Псевдокласс :empty	Представляет пустые элементы, т.е. те, которые не содержат дочерних элементов, текста или пробелов.
Псевдокласс :enabled	Используется для применения стиля к доступным (не заблокированным) элементам форм.
Псевдокласс :first-child	Применяет стилевое оформление к первому дочернему элементу своего родителя.
Псевдокласс :first-of-type	Задает правила стилей для первого элемента в списке дочерних элементов своего родителя.
Псевдокласс :focus	Определяет стиль для элемента получающего фокус.
Псевдокласс :hover	Определяет стиль элемента при наведении на него курсора мыши, но при этом элемент еще не активирован.
Псевдокласс :lang	Определяет язык, который используется в документе или его фрагменте.

Псевдокласс :last-child	Задает стилевое оформление последнего элемента своего родителя.
Псевдокласс :last-of-type	Задает правила стилей для последнего элемента в списке дочерних элементов своего родителя.
Псевдокласс :link	Применяется к ссылкам, которые еще не посещались пользователем.
Псевдокласс :not	Задает правила стилей для элементов, которые не содержат указанный селектор.
Псевдокласс :nth-child	Используется для добавления стиля к элементам на основе нумерации в дереве элементов
Псевдокласс :nth-of-type	Используется для добавления стиля к элементам указанного типа на основе нумерации в дереве элементов.
Псевдокласс :visited	Применяется к ссылкам, уже посещённым пользователем, и задает для них стилевое оформление.

Начнем рассмотрение с псевдоклассов, определяющих состояние элементов.

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

Рассмотрим наиболее используемые:

:active

Происходит при активации пользователем элемента. Например, ссылка становится активной, если по ней перейти. Несмотря на то, что активным может стать практически любой элемент веб-страницы, псевдокласс :active используется преимущественно для ссылок.

:hover

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

:link

Применяется к непосещенным ссылкам. Браузер некоторое время сохраняет историю посещений, поэтому ссылка может быть помечена как посещенная хотя бы потому, что по ней был зафиксирован переход ранее.

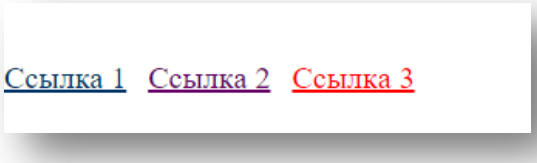
:visited

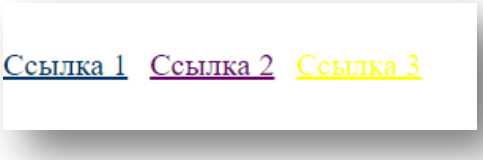
Данный псевдокласс применяется к посещённым ссылкам. Обычно такая ссылка меняет свой цвет по умолчанию на фиолетовый, но с помощью стилей цвет и другие параметры можно задать самостоятельно.

:focus

Применяется к элементу при получении им фокуса. Например, для текстового поля формы получение фокуса означает, что курсор установлен в поле, и с помощью клавиатуры можно вводить в него текст.

Первые четыре даже рассмотрим в одном примере: разберем использование псевдоклассов совместно со ссылками.





```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 */
}
```

Стоит учесть, что имеет значение порядок следования псевдоклассов. Вначале указывается :visited, а затем идёт :hover, если нарушить порядок, то посещённые ссылки не будут изменять свой цвет при наведении на них курсора.

Селекторы могут содержать более одного псевдокласса, которые перечисляются подряд через двоеточие, но только в том случае, если их действия не противоречат друг другу. Так, запись a:visited:hover является корректной, а запись a:link:visited — нет.