# Занятие #10. Псевдоклассы и псевдоэлементы.

### Псевдоклассы

Часто мы хотим обратиться в css к первому или последнему элементу в разметке. Например, когда мы делаем 4 колонки и задаем им отступ справа, то последней колонке мы хотим этот отступ обнулить. В таком случае последняя, 4-я колонка прижмется к правой части родителя вплотную. Мы можем сделать это с помощью обычных классов -- добавляем дополнительный класс последней колонке:

<div class="clearfix">

<div class="col"></div>

<div class="col"></div>

<div class="col"></div>

<div class="col col-last"></div>

</div>

В стилях зададим ширину колонкам и отменим последний отступ:

.col {

float: left;

width: 22%;

margin-right: 4%;

}

.col-last {

margin-right: 0;

}

Это очень частый случай в css, и новый стандарт css позволяет нам использовать “псевдокласс” - то есть css сам умеет находить последний элемент с помощью псевдокласса. Мы уже видели псевдоклассы :hover, :active, :focus. Псевдокласс пишется через двоеточие сразу после селектора, например для “последнего элемента в классом col” запись css будет выглядеть так:

.col:last-child {

margin-right: 0;

}

Преимущество использования псевдокласса в том, что в html нам не нужно добавлять никаких дополнительных классов колонкам:

<div class="clearfix">

<div class="col"></div>

<div class="col"></div>

<div class="col"></div>

<div class="col"></div>

</div>

**Внимание!** :last-child находит элемент, если он последний в родителе! Если после колонок в блок .clearfix добавить еще какой-то элемент (например, a.btn), то четвертая колонка перестает быть последним элементом в родителе, и к ней перестает применяться псевдокласс last-child.

Иногда нам нужно найти первого ребенка в родителе - для этого можно использовать **:first-child**

Также мы можно использовать псевдокласс :nth-child(), где в круглых скобочках вы указываете номер элемента, которому хотите задать стили.

.col:nth-child(8) {

/\* стили для восьмой колонки \*/

}

Кроме того, для энных детей можно писать формулы, используя переменную n. N в этом случае будет принимать значение от 0 до бесконечности (0, 1, 2, 3 и так далее).

.menu-link:nth-child(2n) {

/\* стили для четных ссылок в меню \*/

}

.menu-link:nth-child(even) {

/\* стили для четных ссылок в меню \*/

}

.menu-link:nth-child(2n+1) {

/\* стили для нечетных ссылок в меню \*/

}

.menu-link:nth-child(odd) {

/\* стили для нечетных ссылок в меню \*/

}

.menu-link:nth-child(3n) {

/\* стили для каждой третьей ссылки в меню \*/

}

Псевдоклассов у css много, можете посмотреть особо интересные в справочнике <https://webref.ru/css#pseudo-class>

### Псевдоэлементы

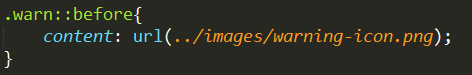
Мы с вами уже хорошо знакомы с элементами разметки html - тэгами, знаем, как они работают и как нам их применять. Сегодня мы узнаем про псевдоэлементы.

Псевдоэлементы ::before и ::after используются для того, чтобы добавить какое-то содержимое перед элементом или после него.

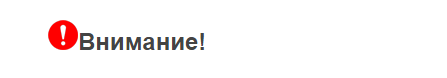
Например, если мы захотим вставить иконку warning-icon.png перед важным заголовком, мы можем в разметке html создать только заголовок, а все остальное - добавить в css



И добавить к нему в стилях вот такой код: мы обращаемся к заголовку по классу .warn, и перед ним создаем элемент(::before), в который с помощью атрибута content помещаем картинку.



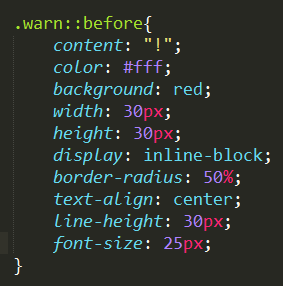
В браузере мы увидим, что картинка расположилась прямо перед текстом так, будто мы вставили ее тэгом img:



Раньше мы вставляли дополнительные тэги <i>, <b> или <span> в разметку html, только для того, чтобы добавить, например, иконочку для кнопки. Теперь мы можем воспользоваться для этого красивым способом, без усложнения разметки. Этот способ удобен тем, что он разгружает html, т.е. нам не нужны дополнительные тэги в разметке.

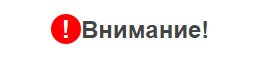
Обратите внимание на синтакис - в нашем css псевдоэлемент ::before написан с двумя двоеточиями. Два двоеточия были введены чтобы различать псевдоэлементы (например, before, after) от псевдоклассов (например, hover, first-child).

Для новых, введённых в CSS3, псевдоэлементов в спецификации вообще запрещают использовать одно двоеточие, допустимо только два. Для псевдоэлементов из CSS2 (before, after) можно по прежнему использовать и одно двоеточие, но чтобы отличать их от псевдоклассов, их часто пишут с двумя двоеточиями, по самым последним стандартам. Имейте ввиду, что IE8 поддерживает только старый синтаксис для старых псевдоэлементов, т.е. с одним двоеточием :before и :after. Но мы будем пользоваться новыми правилами, через два двоеточия.



Обратите внимание, что содержимое псевдоэлементу добавляется особым образом - из css атрибутом content. В него можно вложить не только картинку, но и любой текст или символ. Давайте попробуем вложить в псевдоэлемент не готовую картинку, а символ, и стилизовать его:

В браузере получим почти то же самое:



Еще стоит помнить, что

* псевдоэлементы создаются отдельно от разметки, они на особом счету
* им обязательно нужно добавить содержимое (content: “”;)
* к ним нельзя обратиться средствами javascript (запомните на будущее)
* в них нельзя вложить никакие другие элементы (они не могут быть родителями)
* их нельзя создать внутри элемента, который не принимает дочерних тэгов. Т.е. если элемент сам не может быть родителем, то у него не может быть ::before или ::after (таким образом псевдоэлементы неприменимы к input и img)
* изначально ::before и ::after - строчные элементы
* в инспекторе псевдоэлементы тоже выглядят по особому, их можно просмотреть как отдельный элемент, или как псевдоэлемент родителя.

