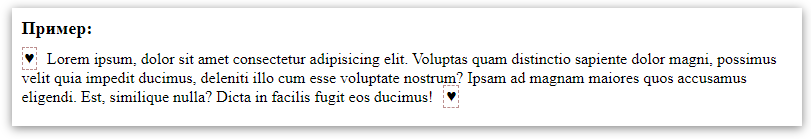
**Использование псевдоэлементов и псевдоклассов**

**Псевдоэлементы**

**Псевдоэлементы ::after и ::before**

Элементы :before и :after не будут сгенерированы, т.е. не будут видны в коде страницы, поэтому они и называются псевдоэлементами. ::after создаёт псевдоэлемент после содержимого элемента. Этот псевдоэлемент работает только совместно со стилевым свойством content, которое определяет содержимое для вставки. По своему действию ::before аналогичен ::after, но вставляет псевдоэлемент до содержимого элемента.

**Пример:**

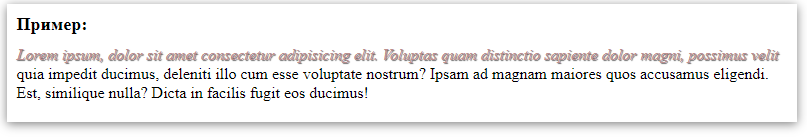


**Псевдоэлемент ::first-line**

По названию этого псевдоэлемента можно догадаться, что предназначен он для форматирования первой строки блочных элементов. Для оформления первой строки можно использовать не все css-свойства, а только те, что перечислены в списке ниже:

1. Свойства группы ***font (font-family, font-size, fontstyle, font-weight, font-variant и др.)***.
2. ***сolor.***
3. Свойства группы ***background***.
4. ***word-spacing.***
5. ***letter-spacing.***
6. ***text-decoration, text-decoration-color, text-decoration-line, text-decoration-style.***
7. ***vertical-align.***
8. ***text-shadow.***
9. ***text-transform.***
10. ***line-height.***
11. ***clear.***

**Пример:**

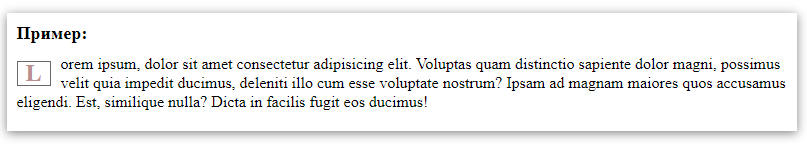


**Псевдоэлемент ::first-letter**

По названию этого псевдоэлемента также можно понять, что его предназначение — оформить первую букву блочного элемента. Для него тоже есть ограничения по используемым свойствам.

1. Свойства группы ***margin***.
2. Свойства группы ***padding***.
3. Свойства группы ***border***.
4. ***box-shadow***.
5. ***float***.
6. ***vertical-align*** (только если float равен none).

**Пример:**

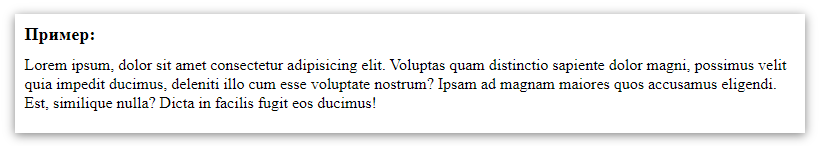


**Псевдоэлемент ::selection**

Псевдоэлемент ***::selection*** предназначен для изменения вида текста при выделении. Его можно назначить как для всего документа, так и для определенных элементов. Следует помнить, что с этим псевдоэлементом можно использовать небольшое количество css-свойств, а именно:

1. ***color***.
2. ***background-color***.
3. ***cursor***.
4. ***outline*** и его варианты.
5. Группа свойств ***text-decoration***.
6. ***text-shadow***.

**Пример:**



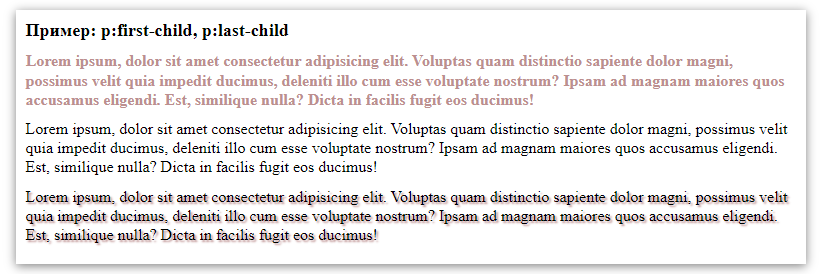
**Псевдоклассы**

Помимо псевдоэлементов в CSS существуют псевдоклассы. Они всегда записываются с одним двоеточием после селектора. Псевдоклассы определяют различные состояния элементов, например, их вид при наведении курсора мыши или при клике на элементе.

**Псевдоклассы first-child и last-child**

Например, псевдокласс ***:first-child*** отвечает за форматирование первого элемента, а псевдокласс ***:last-child*** отвечает за форматирование последнего элемента среди подобных.

**Пример: p:first-child, p:last-child**



**Псевдокласс nth-child**

Чтобы отформатировать какой-либо определенный элемент, необходимо использовать псевдокласс ***:nth-child(n)***. Вместо ***n*** нужно поставить какую-либо цифру.

**Пример: :nth-child(3), :nth-child(6)**

Если необходимо задать форматирование для каждого 3-го абзаца, стоит использовать псевдокласс ***:nth-child(3n)***. Вместо цифры 3 может быть любая цифра или выражение, например, ***4n-1, 3n+1*** и т.д. Достаточно часто бывает нужно отформатировать четные или нечетные элементы. Для этой цели можно использовать, например, псевдокласс ***:nth-child(2n)*** или ***:nth-child(2n-1)***. Кроме того, существуют специальные ключевые слова для четных элементов ***(even)*** и нечетных ***(odd)***.

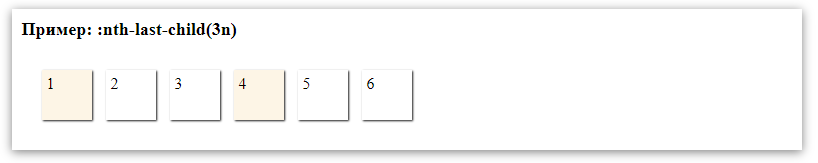
**Пример: :nth-child(2n-1)**

**Псевдокласс :nth-last-child(n)**

Псевдокласс ***:nth-last-child(n)*** предназначен для форматирования элементов, отсчет которых ведется не с начала их появления в коде, т.е. с первого элемента, а на оборот — с конца, т.е. с последнего элемента.

В скобках вместо n можно записывать число, выражение ***(2n, 3n-1, 4n+1 и т.п.)***, а также ключевые слова ***odd*** (нечетные числа) или ***even*** (четные числа)

**Пример: :nth-last-child(3n)**

**Псевдокласс :only-child**

Псевдокласс ***:only-child***предназначен для форматирования элементов, которые являются единственными дочерними для своего родительского тега. Т.е. он будет применен в случае, если в родительском элементе для указанного селектора отсутствуют другие элементы такого же типа.

**Псевдоклассы типа :nth-of-type**

Псевдокласс ***:nth-of-type(n)*** используется для форматирования элементов одного типа внутри одного родительского элемента. Этот псевдокласс может быть записан с цифрой, выражением или ключевыми словами ***odd*** или ***even*** в скобках

Кроме того, вы можете использовать псевдоклассы ***:first-of-type*** или ***:last-of-type*** для первого и последнего элемента одного типа внутри родительского контейнера.

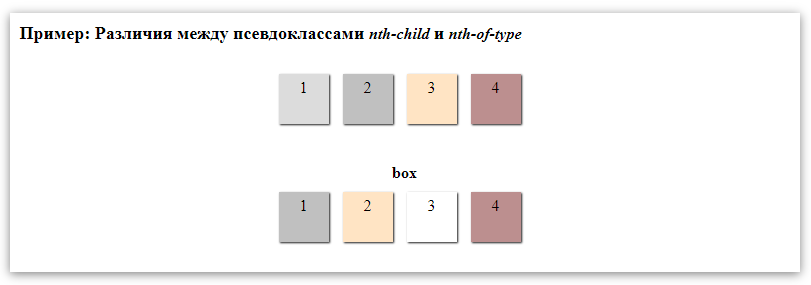
Также существует псевдокласс ***:nth-last-of-type(n)***, который позволяет задавать форматирование для однотипных элементов, отсчитывая их снизу разметки, т.е. от последнего элемента внутри одного контейнера.

Возможно, вы зададитесь вопросом — зачем нужны псевдоклассы, которые делают практически одно и то же? Почему нельзя ограничиться только лишь псевдоклассами типа ***:nth-child?***

Дело в том, что псевдоклассы типа ***:nth-child(n)*** не всегда ведут себя так, как вы этого ожидаете, особенно при изменении html-разметки.

Допустим, нам необходимо добавить к ним заголовок (довольно распространенная ситуация), который также будет размещен внутри общего контейнера. Внешний вид блоков в браузере изменится.

**Пример: Различия между псевдоклассами *nth-child* и *nth-of-type***

Можно заметить, что произошло смещение на один блок влево. Т.е. элемент, который подчинялся правилам для ***:first-child*** теперь подчиняется правилам для элемента с псевдоклассом ***:nth-child(2)***, а элемент, для которого свойства описывал псевдокласс ***:nth-child(2)*** теперь имеет свойства элемента с псевдоклассом ***:nth-child(3)***. Третий блок вообще потерял цвет фона. Неизменным осталось только форматирование для ***:last-child***.

Почему поменялись «дети» в родительском элементе в общем контейнере? Ответ прост — именно потому, что «детей» стало больше. При добавлении заголовка увеличилось количество дочерних, или вложенных, элементов, причем заголовок стал в иерархии дочерних элементов первым, т.е. псевдокласс ***:first-child*** теперь должен относиться именно к нему. Однако css-правила записаны для ***.box-item:first-child***, поэтому к заголовку не применяются. Если бы правила были написаны для ***.box-items:first-child***, тогда бы мы увидели серебристый фон у заголовка.

Ситуацию поможет изменить замена форматирования, заданного псевдоклассами :nth-child и подобных ему, на ***:nth-of-type***.

**Пример: *nth-of-type***

