Псевдоклассы

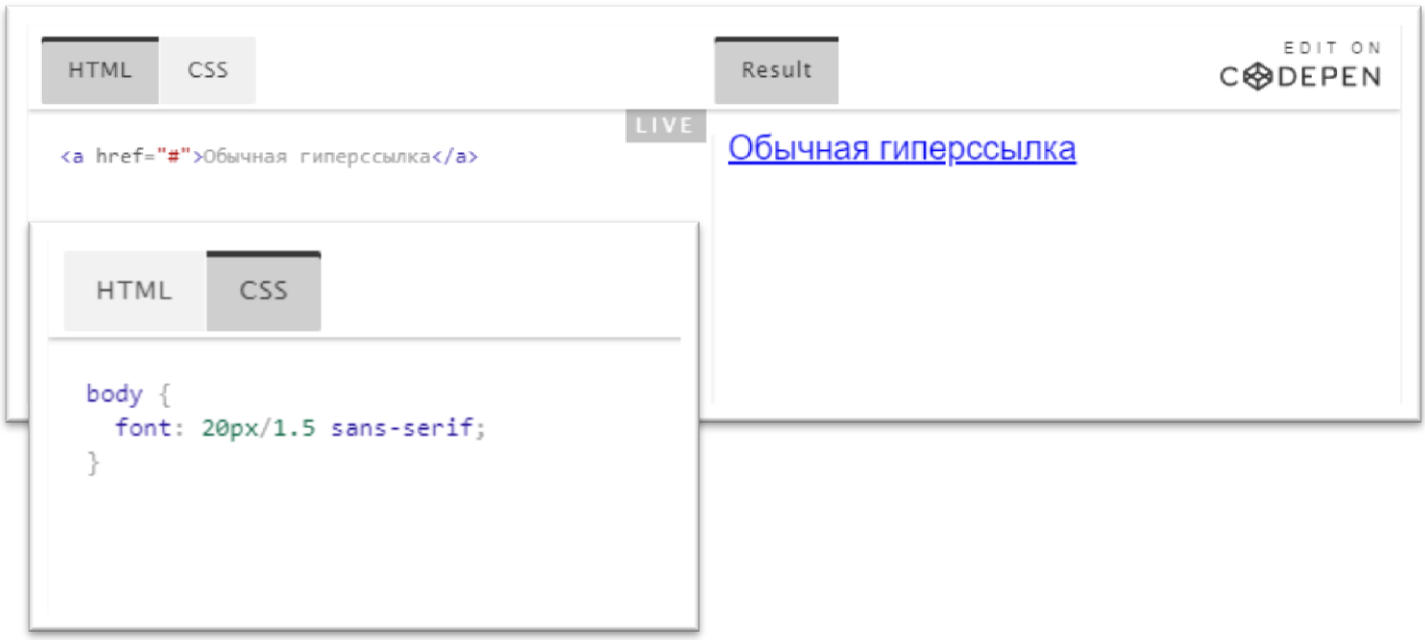
Псевдоклассы — краеугольный камень использования селекторов. Для начала определим, почему же такие селекторы имеют название *псевдо*. Псевдоклассами не выбирается элемент напрямую. Они указывают на какое-либо состояние элемента. Достаточно просто показать это на примере ссылок.

```
<a href="#">Обычная гиперссылка</a>
```

Что мы можем сказать об этом элементе? Это элемент HTML со стандартными стилями и возможностью переадресовать пользователя. Если попробовать перейти по ссылке, то случится целых 3 события!

1. Наведение на ссылку.
2. Момент клика по ссылке. Этот момент наступает при нажатии основной кнопки мыши, но до её отпускания.
3. Браузер автоматически помечает ссылку, по которой мы уже переходили.

Есть ещё одно событие, которое обрабатывает ссылка — событие фокуса. Оно возникает при переходе на ссылку с помощью клавиатуры.



Добейтесь воспроизведения всех событий и посмотрите, как будет меняться взаимодействие:

1. При наведении на ссылку изменится тип курсора мыши.
2. При нажатии основной клавиши мыши ссылка поменяет свой цвет.
3. При отпускании основной клавиши мыши ссылка ещё раз поменяет цвет, указывая на то, что пользователь уже совершал переход по ней.
4. Нажмите клавишу Tab. Так вы увидите состояние фокуса, при котором вокруг ссылки появятся границы.

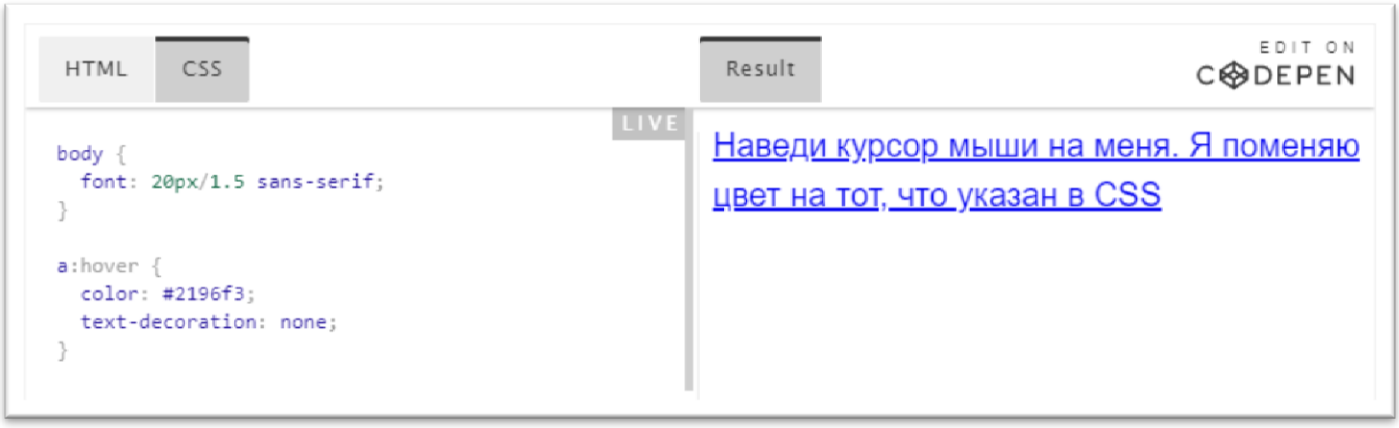
Можем ли мы, как разработчики, контролировать это поведение? Конечно да! В этом и кроется сила псевдоклассов — они позволяют задавать стили не напрямую для элементов, а для их состояний или для некоторых других условий.

Псевдоклассы имеют специальный синтаксис, который позволяет легко отличить их от других селекторов. Записываются они так: *селектор:псевдокласс*. Разберёмся на примере.

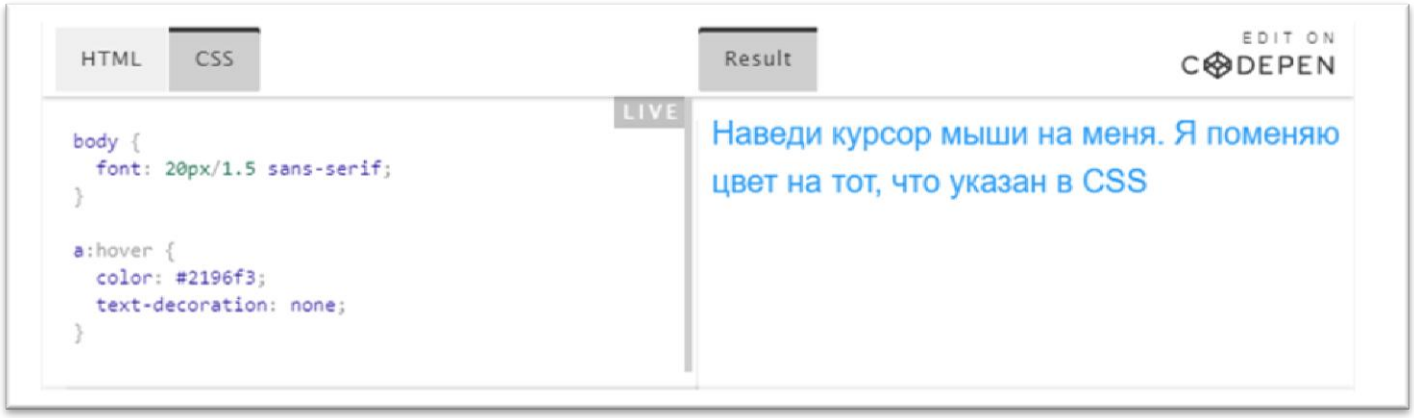
Для стилизации элемента при наведении используется псевдокласс `:hover`. Стили, указанные в таком селекторе, будут применяться только при наведении на элемент и удаляться при снятии состояния. Попробуем стилизовать стили ссылки при наведении.

```
a:hover {
  color: #2196f3;
  text-decoration: none;
}
```

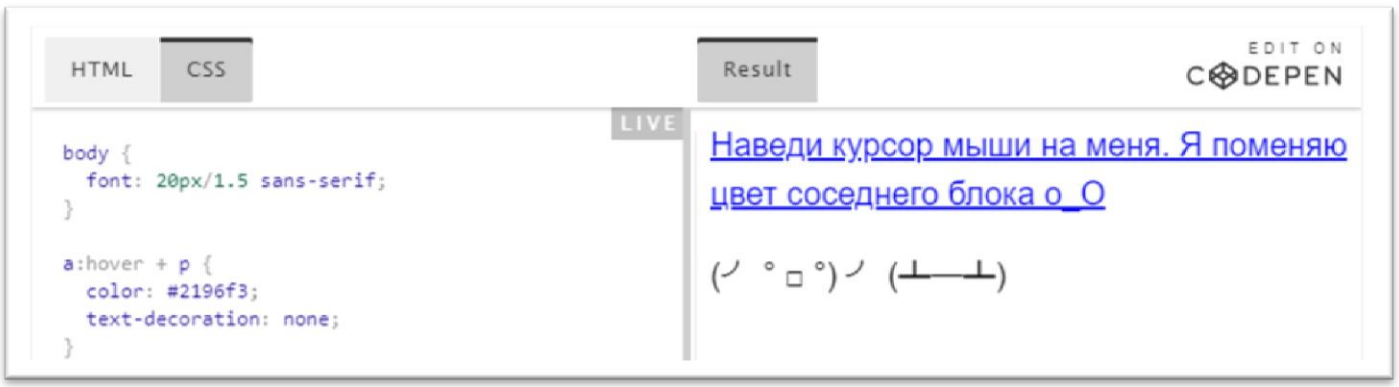
До наведения курсора мыши:

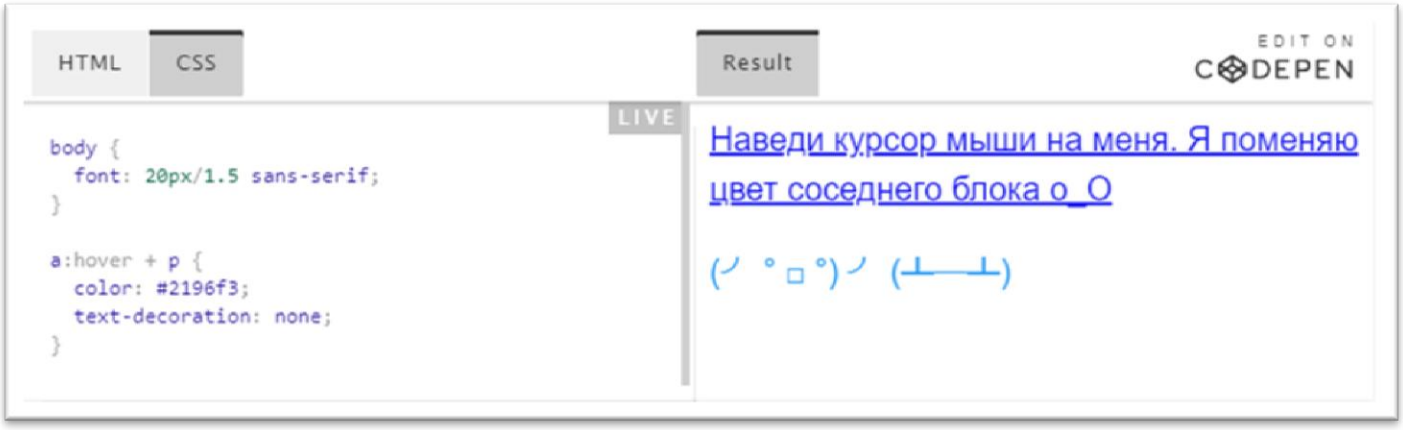


После наведения курсора мыши на гиперссылку:



Самое невероятное, что можно комбинировать уже изученные селекторы. Представьте, что при наведении на один элемент будет меняться совершенно другой! Помните про смежные и родственные селекторы? Никто не запрещает комбинировать их и получать интересные стили.





Для других состояний так же существуют свои псевдоклассы.

- `:active` — Стиль при нажатии на ссылку, но до перехода по ней.
- `:visited` — Стиль уже посещённой ссылки.
- `:focus` — Стиль при событии фокуса на элементе.

Обратите внимание, что все эти псевдоклассы могут работать не только для ссылок. Не бойтесь экспериментировать :)

Попробуйте создать HTML и CSS файлы и прописать эти параметры. Посмотрите на результат работы кода.

HTML

```html
<a href="#" class="link hover">Наведи курсор мыши на меня</a>

<a href="#" class="link active">Нажми на меня и не отпускай</a>

<a href="#" class="link visited">Посети меня</a>

<a href="#" class="link focus">Сфокусируйся на мне с помощью клавиши Tab</a>

<div class="hover square">А я блочный элемент. На меня тоже можно навести курсор мыши</div>
```

CSS

```css
body {
  font: 20px/1.5 sans-serif;
}

.link {
  display: block;
  margin-bottom: 20px;

  color: #333;
  text-decoration: none;
}

.hover:hover {
  color: #2196f3;
}

.active:active {
  color: #f44336;
}

.visited:visited {
  color: #673ab7;
}

.focus:focus {
  color: #4caf50;
}

.square {
  width: 150px;
  height: 150px;
  padding: 50px;

  font-size: 15px;

  background: #ccc;
}
```

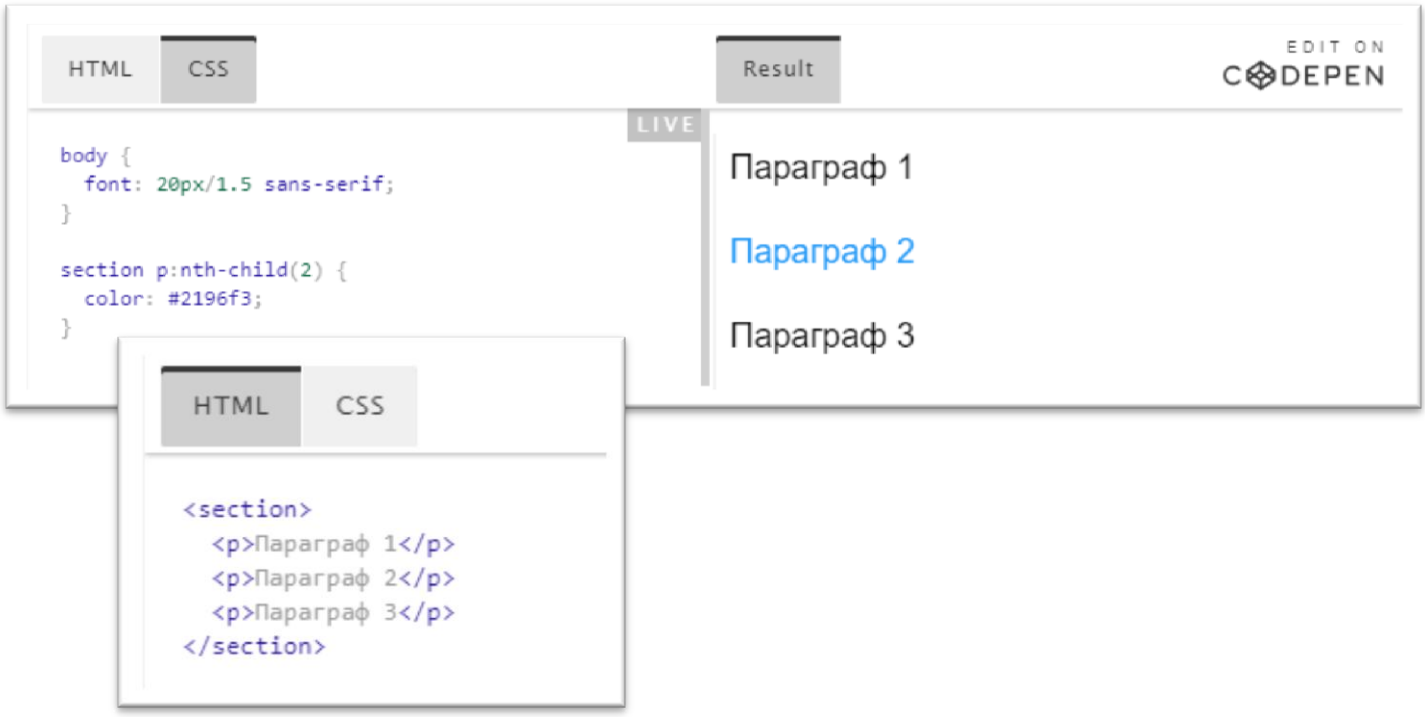
Структурные псевдоклассы

К структурным псевдоклассам можно отнести те, которые добавляют стили к элементу в зависимости от его месторасположения внутри HTML. Это мощное средство, позволяющее добиться сложных стилей без использования большого количества классов.

Основным структурным псевдоклассом является `:nth-child(условие)`. Вы можете увидеть, что это целая функция, которая принимает условие, по которому будет выбран элемент или элементы. Разберёмся, какие значения она может принимать и какие элементы будут выбраны.

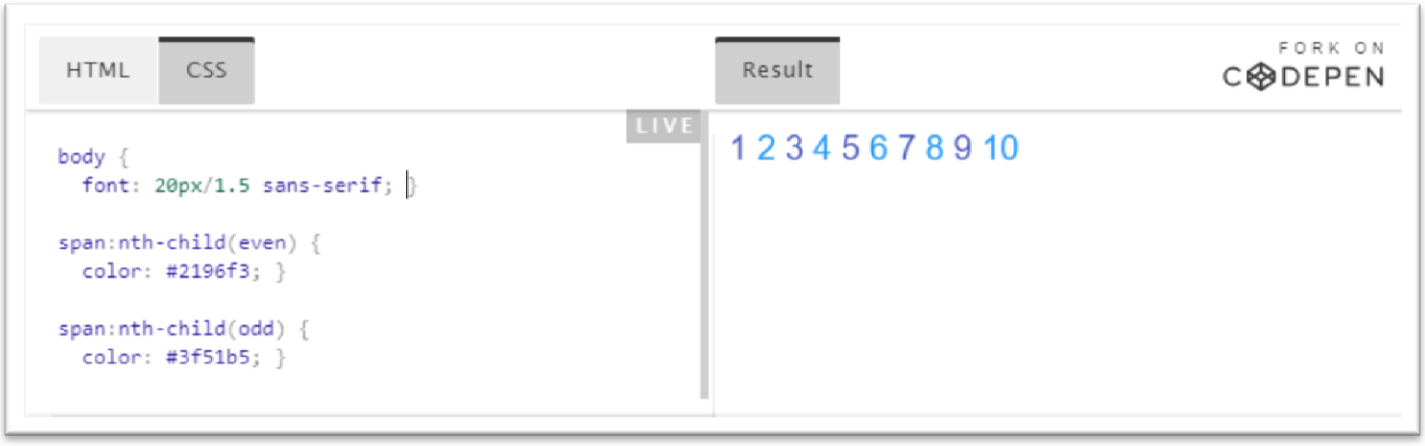
Самое простое — указать конкретный элемент, который нужен. Для этого достаточно указать порядковый номер элемента. Обратите внимание, что элементы должны быть потомками одного родителя и выбираться по одному селектору.

```
<section>
  <p>Параграф 1</p>
  <p>Параграф 2</p>
  <p>Параграф 3</p>
</section>
section p:nth-child(2) {
  color: #2196f3;
}
```

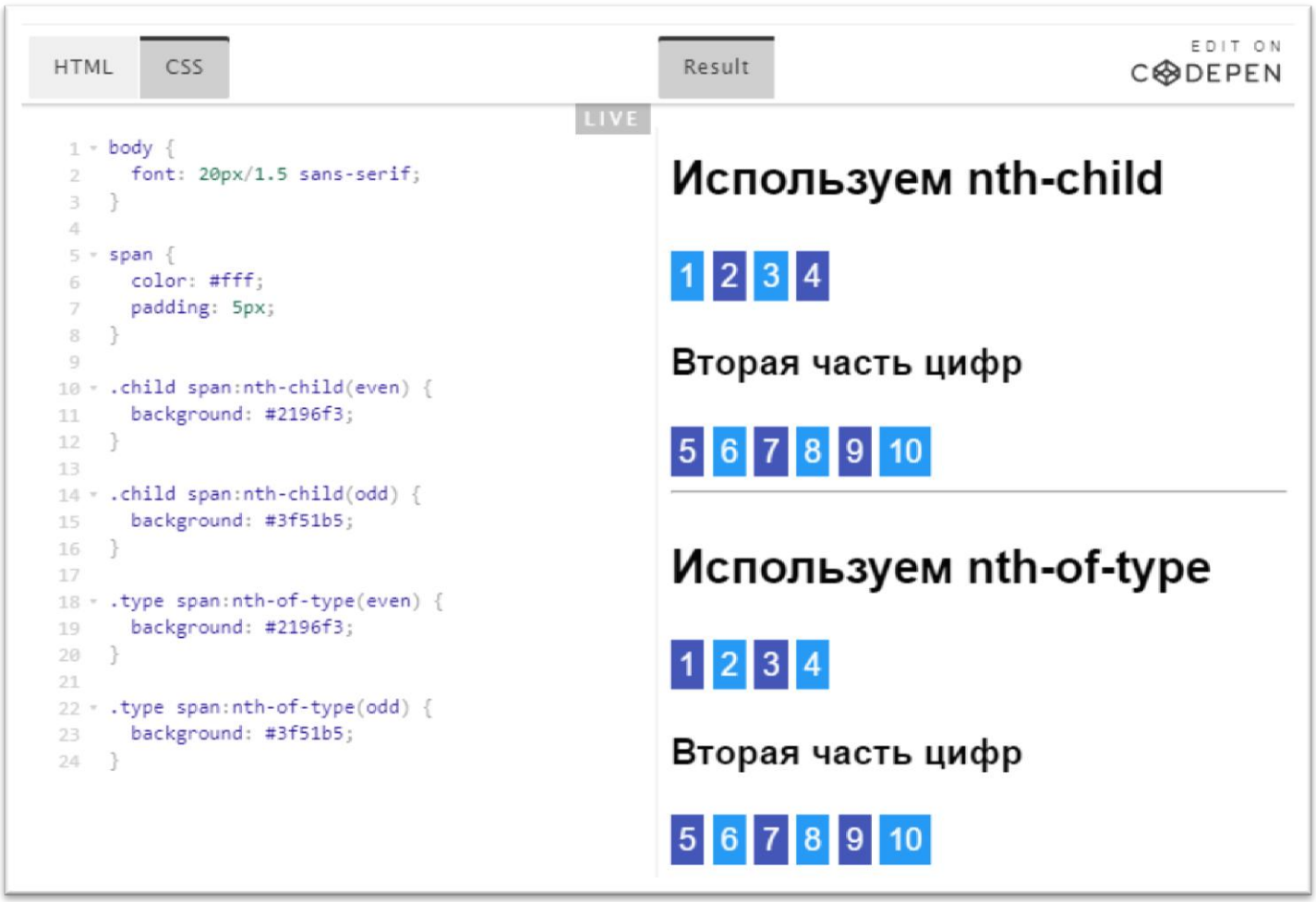


Помимо прямого выбора элемента, можно передать специальные последовательности, которые смогут выбрать не один, а сразу несколько элементов.

- `:nth-child(2n)` — выбрать каждый второй элемент. 2, 4, 6, 8... Число может стоять любое. Если поставить `3n`, то будет выбран каждый третий элемент и так далее. Это касается всех последовательностей.
- `:nth-child(2n + 1)` — выбрать каждый второй элемент, *начиная с первого*. 1, 3, 5, 7, 9...
- `:nth-child(even)` — выбрать все чётные элементы. То же самое, что и `:nth-child(2n)`.
- `:nth-child(odd)` — выбрать все нечётные элементы. То же самое, что и `:nth-child(2n + 1)`.



Похожим псевдоклассом является `:nth-of-type(условие)`. Попробуйте в примере выше заменить `nth-child` и результат останется тем же. Но зачем нужен ещё один псевдокласс с тем же функционалом?



Приглядитесь внимательно. В случае с `nth-child` отсчёт элементов `<span>` начался с чётного элемента. Но ведь 1 — это нечётный элемент. Вы будете правы с точки зрения математики, но не логики работы `nth-child`. Он выбрал все элементы `<span>` и при этом учитывал, на какой позиции они находятся относительно других элементов в блоке. По этой причине элементы 4 и 5 являются нечётными, хотя и идут подряд. Логика работы следующая:

1. Первый элемент внутри блока — h2. Он стоит на нечётной позиции относительно всех элементов внутри родителя.

2. Элементы 1 и 3 являются чётными, так как внутри родителя являются вторым и четвёртым элементом соответственно.
3. Элементы 2 и 4 являются нечётными, так как внутри родителя являются третьим и пятым элементом.
4. Заголовок «Вторая часть цифр» — чётный элемент внутри родителя.
5. Элемент 5 теперь тоже нечётный, так как идёт после чётного заголовка.

Псевдокласс `nth-of-type` распознаёт не только позицию элемента, но и его тип. В нашем случае для этого селектора не существует заголовков. Выборка идёт только по элементам `<span>` вне зависимости от того, какие ещё элементы находятся внутри родителя.

Не всегда есть потребность использовать такие сложные псевдоклассы. Для некоторых стандартных ситуаций существуют специальные псевдоклассы:

- `:first-child` — выбирает первый элемент внутри родителя.
- `:last-child` — выбирает последний элемент внутри родителя.
- `:first-of-type` — выбирает первый элемент внутри родителя учитывая тип элемента.
- `:last-of-type` — выбирает последний элемент внутри родителя учитывая тип элемента.
- `:only-child` — выбирает элемент, если он единственный внутри родителя.

Дополнительное задание

Возьмите любой из предыдущих уроков и, используя псевдоклассы, дополните стили. Попробуйте использовать разные выражения внутри структурных псевдоклассов.