

Занятие 4

Псевдоклассы и псевдоэлементы.
Позиционирование элементов.

CSS псевдоклассы

```
//:first-child - Первый дочерний элемент  
выборки
```

```
ul li:first-child {...}
```

```
//:last-child - Последний дочерний элемент  
выборки
```

```
.contact-page div:last-child {...}
```

```
//:first-of-type - Первый элемент выборки
```

```
.team-staff-item:first-of-type {...}
```

```
//:last-of-type - Последний элемент выборки
```

```
.command div.item:last-of-type {...}
```

```
//:active
```

```
//:focus
```

```
//:hover
```

```
//:link
```

```
//:nth-child(odd || even || n) - Выборка по  
нумерации элемента
```

```
table tbody tr:nth-child(odd) //все нечётные
```

```
table tbody tr:nth-child(even) //все чётные
```

```
ul li:nth-child(3n) // каждый третий элемент
```

```
//:not() - Исключение
```

```
a:not([href="#"]) {...}
```

```
.contact-form input:not([type="submit"]) {...}
```

```
// :not() может использоваться в любом из
```

```
псевдоселекторов, кроме еще одного :not
```

```
// // .box div:not(:not()) Не работает
```

```
// Все абзацы, кроме первого
```

```
.main .info p:not(:first-line) {...}
```

```
// Все элементы списка, не кратные 5
```

```
ul li:not(:nth-child(5n)) {...}
```

```
// Реакция на наведение для всех элементов,  
кроме активного
```

```
.menu .menu-item:not(.active):hover {...}
```

CSS псевдоэлементы

```
//::first-line - Стили для первой строки  
.text-block::first-line {...}  
  
//::first-letter - Стили для первого символа  
.review p::first-letter {...}  
  
//::placeholder - Стили для дефолтного текста  
// в input  
.input::placeholder {...}
```

CSS псевдоэлементы

```
<div class="element"><p>Hello world</p></div>  
:before _____ :after
```

// Для добавления контента на страницу без изменения HTML документа можно использовать псевдоэлементы `::before` и `::after`.

// Псевдоэлементы наследуют все свойства блока, к которому они относятся.

// Псевдоэлементы не будут работать без свойства "content"

// Свойство content позволяет вставлять генерируемый контент, которого изначально нет на странице

```
.element::before{content: '[';}
```

```
.element::after{content: ']';}
```

// Указанная конструкция выведет [Hello world]

```
// Также с помощью псевдоэлементов можно добавить кастомный блок и опустить content. Для этого надо в свойствах прописать content: ""; и указать нужные CSS свойства
```

```
// Все псевдоэлементы по умолчанию строчные, независимо от того, к какому элементу относятся, поэтому для кастомного блока необходимо указать display: inline-block; float; или position: absolute;
```

```
// Рассмотрим добавление кастомных маркеров для списка
```

```
ul.custom-list{  
  list-style-type: none;  
}  
ul.custom-list li:before {  
  content: '';  
  width: 5px;  
  height: 5px;  
  background: red;  
  border-radius: 50%;  
  display: inline-block;  
  vertical-align: middle;  
}
```

- Атрибут data-*
- Всплывающая подсказка на CSS
- Генератор шрифтов Fontello
- Линия отреза
- Начинаем работать с CSS-счётчиками
- Список с русскими буквами

CSS счетчик

```
// С помощью CSS также можно пользоваться счетчиками, как правило, это может понадобиться для
// кастомных нумерованных списков, тк выводить результат возможно только в свойстве content: "...";

// Для запуска счетчика необходимо использовать свойство counter-reset: counterName
// Для увеличения счетчика на единицу - counter-increment: counterName
// Для вывода - content: `Part counter(counterName)`
// Где counterName уникальное пользовательское имя для счётчика

ol{
  list-style-type: none;
  counter-reset: lorem-text;
}
ol li{
  counter-increment: lorem-text;
  font: 20px Arial, Helvetica, sans-serif;
}
ol li:before{
  content: "Секция " counter(lorem-text) ": ";
}
```

Секция 1: Lorem Ipsum - это текст-"рыба"
Секция 2: часто используемый в печати
Секция 3: Lorem Ipsum является стандартной
Секция 4: "рыбой" для текстов на латинице
Секция 5: с начала XVI века.

Задание

// Создаем три списка с кастомными буллитами:

// // С кастомными буллитами (прим. в виде круга красного цвета);

// // С буллитами из шрифтовых иконок или мнемоники;

// // С кастомными цифрами (текстом "N страница" или цифра счетчика в круге);

Позиционирование элементов

// Позиционирование позволяет установить положение элемента относительно окна браузера или других объектов на веб-странице. Назначается с помощью CSS свойства position

```
position: absolute | fixed | relative | static;
```

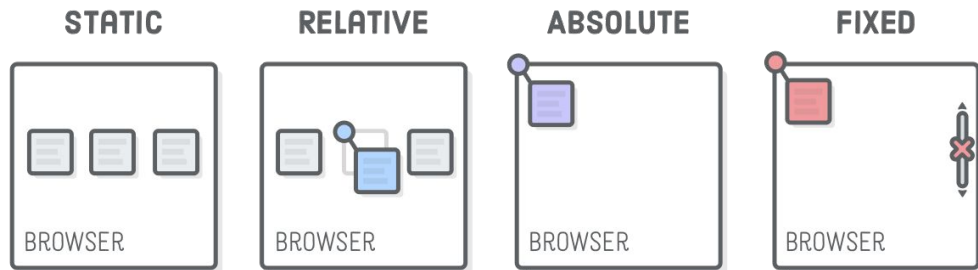
// Перечисленные значения устанавливают отображение относительно окна:

// // absolute - элемент позиционируется абсолютно, относительно браузера, остальные элементы отображаются так, словно абсолютно позиционированного элемента не существует

// // fixed - элемент не меняет своего положения при прокрутке страницы

// // relative - элемент остается на своем месте в потоке, при изменении положения будет изменяться его позиция отображения элемента, физически он все еще будет находиться на своем месте в потоке

// // static - отображение по-умолчанию, сменить позицию через top | right | bottom | top невозможно



Перемещение позиционированных элементов

```
// Перемещать позиционированные элементы можно с помощью свойств  
top; right; bottom; left;
```

```
(top | right | bottom | left) : value | % | auto;
```

```
// position: static; не перемещается таким образом!!!
```

```
// position: fixed; перемещается относительно соответствующих  
сторон экрана
```

```
// position: absolute; перемещается относительно ближайшего не  
дефолтно позиционированного родителя, если все родители  
позиционированы по-умолчанию, перемещается относительно body
```

```
// position: relative; перемещается относительно своего текущего  
местоположения
```

```
// z-index: любой позиционированный элемент может накладываться  
поверх другого, и находится как выше, так и ниже других элементов  
по z-оси. Для управления размещением используется z-index
```

