

Цели занятия

- подробно разобрать что такое псевдоклассы в CSS и для чего они используются;
- разобрать основные псевдоэлементы и сферу их применения;

Полезные материалы

- [Полная документация по псевдоклассам](#)
- [Полная документация по псевдоэлементам](#)

Что такое псевдокласс?

Псевдокласс в CSS — это ключевое слово, добавленное к селектору, которое определяет его особое состояние. Например, `:hover` может быть использован для изменения цвета кнопки при наведении курсора на неё.

Псевдоклассы описывают характеристики элементов, такие как динамическое состояние — нажатая ссылка, язык кодировки — абзац на французском языке и т.д. Они не отображаются в исходном документе и не принадлежат дереву документа DOM (дереву документа принадлежат только сами html-элементы).

Типы псевдоклассов

1. Динамические псевдоклассы

- `:link` — не посещенная ссылка;
- `:visited` — посещенная ссылка;
- `:focus` — ссылки, а также элементы форм, активированные посредством курсора мыши или на которые перешли с

помощью клавиатуры (кнопка TAB);

- `:hover` — ссылки, а также другие элементы, стили, которые применяются при наведении курсора на элемент;
- `:active` — выбирает элемент, активированный пользователем с помощью клика мышки. Обычно применяется для ссылок или кнопок;

2. Псевдоклассы пользовательского интерфейса

Относятся к элементам форм, таким как кнопки, радио-кнопки, поля ввода и так далее.

- `:disabled` — используется для отбора и стилизации заблокированных для выбора и изменения элементов форм;
- `:enabled` — отбирает не заблокированные для выбора и изменения элементы форм;
- `:checked` — применяется для выбора и стилизации элементов `<input type="radio">`, `<input type="checkbox">`, а также элементов `<option></option>`, находящихся внутри элемента `<select></select>`;
- `:indeterminate` — элементы радио и чекбокс могут быть включены или выключены пользователем. Некоторые из них могут находиться в неопределенном состоянии, к которым и применяется данный псевдокласс.

3. Структурные псевдоклассы

Представляют концепцию структурных псевдоклассов, которая позволяет отбирать элементы на основании информации, отраженной в дереве документа и не может быть получена при помощи простых селекторов или их комбинации. Отсчёт начинается с элемента с индексом 1.

- `:root` — элемент, являющийся корневым в документе;
- `:first-child` — элемент, который является первым дочерним элементом в контейнере;
- `:last-child` — последний дочерний элемент элемента-контейнера;
- `:first-of-type` — первый элемент данного типа среди дочерних элементов родительского элемента;
- `:last-of-type` — последний элемент данного типа среди дочерних элементов родительского элемента;
- `:nth-child()` — элементы на основе их индекса (в порядке очереди) внутри родительского контейнера

Варианты:

- `li:nth-child(even)` — каждый чётный элемент списка, то есть второй, четвертый, шестой и т.д.
- `li:nth-child(odd)` — каждый нечетный элемент списка, то есть первый, третий, пятый и т.д.
- `li:nth-child(3)` — только третий элемент в списке

Псевдоэлементы

Псевдоэлементы — это селекторы CSS, использующиеся для вставки содержимого, которого нет в исходном коде; их используют для стилизации определённой части выбранного элемента.

Псевдоэлементы начинаются с двойного двоеточия, например, `::before`, `::after`, `::placeholder`, `::first-letter`.

Псевдоэлементы `::before` и `::after`

Большая часть псевдоэлементов позволяет выбирать контент, который уже присутствует в исходниках документа, но не задан с

помощью языка (другими словами, ваш HTML). Однако `::before` и `::after` работают по-другому. Эти псевдоэлементы добавляют генерируемый контент в дерево документа. Созданный контент не существует в HTML исходниках, но он отображается. У данных псевдоэлементов обязательно должно быть свойство `content`, иначе браузер не будет их отображать, причём допустимо указывать в качестве значения пустую строку, например, `content: ""`;

Зачем использовать генерируемый контент?

Например, можно помечать обязательные поля формы, добавляя контент после лейблов:

```
/* Применяется к лейблу, ассоциирующемуся с требуемым полем */
.required::after {
    content: " (Required) ";
    color: #c00;
    font-size: 0.8em;
}
```

В нужном поле формы необходимо использовать HTML свойство `required`. Так как данная информация уже доступна в DOM, то `::before` и `::after` выступают в роли помощников. Это не критический контент, поэтому его можно и не вносить в исходники.

Другие псевдоэлементы

- `::placeholder` — позволяет задавать стили для подсказок внутри полей ввода, таких как `<input>` или `<textarea>`
- `::first-letter` — позволяет применять стили для первой буквы текста внутри элемента, чтобы, к примеру, создать [эффект буквицы](#)
- `::first-line` — применяет указанные CSS-свойства к первой строке текста внутри элемента (на практике используется

довольно редко)

- `::selection` — позволяет задавать свой стиль для выделения текста на сайте. Большинство браузеров разрешают изменять в данном псевдоэлементе только свойства `color` и `background-color`