

CSS - псевдо что-то

Псевдо-классы и Псевдо-элементы;
overflow & text-overflow
ссылки-якори

Псевдоклассы

Псевдоклассы в CSS - это ключевые слова, которые добавляют специфическое поведение к выбранным элементам в различных состояниях. Они позволяют стилизовать элементы не только на основе их самих, но и на основе их состояний или позиций в документе

Псевдоклассы

- `:hover` - Применяет стили к элементу, когда курсор мыши наведен на него.
- `:active` - Применяет стили к элементу во время активации, например, когда нажата кнопка мыши на ссылке.
- `:focus` - Применяет стили к элементу, который находится в фокусе, например, элемент формы, когда на него переключаются клавишами Tab.
- `:first-child` - Применяет стили к элементу, который является первым дочерним элементом своего родителя.
- `:last-child` - Применяет стили к элементу, который является последним дочерним элементом своего родителя.
- `:nth-child(n)` - Применяет стили к элементу, который является n-м дочерним элементом своего родителя.
- `:nth-of-type(n)` - Применяет стили к элементу, который является n-м элементом указанного типа своего родителя.
- `:nth-last-child(n)` - Применяет стили к элементу, который является n-м дочерним элементом своего родителя, начиная с конца.
- `:nth-last-of-type(n)` - Применяет стили к элементу, который является n-м элементом указанного типа своего родителя, начиная с конца.
- `:not(selector)` - Применяет стили к элементу, если он не соответствует указанному селектору.
- `:checked` - Применяет стили к элементу input, который имеет состояние "выбран".
- `:disabled` - Применяет стили к элементу, который является отключенным.
- `:enabled` - Применяет стили к элементу, который является включенным.
- `:empty` - Применяет стили к элементу, который не содержит дочерних элементов или текста.
- `:visited` - Применяет стили к элементу ссылки, который был уже посещен пользователем.

Псевдоэлементы

Псевдоэлементы в CSS позволяют создавать виртуальные элементы, которые не являются частью HTML-структуры, но которые можно стилизовать так же, как и обычные элементы. Они добавляют дополнительные декоративные элементы или изменяют отображение существующих элементов.

Псевдоэлементы

- `::before` - создает виртуальный элемент, который располагается перед содержимым выбранного элемента.
- `::after` - создает виртуальный элемент, который располагается после содержимого выбранного элемента.
- `::first-line` - применяет стили к первой строке текста внутри выбранного элемента.
- `::first-letter` - применяет стили к первой букве текста внутри выбранного элемента.
- `::selection` - применяет стили к тексту, выделенному пользователем.

Псевдоэлементы для формы в HTML

В HTML формы часто содержат элементы ввода (input), кнопки (button), выпадающие списки (select), текстовые области (textarea) и т. д. Псевдоэлементы могут использоваться для стилизации этих элементов.

Примеры псевдоэлементов для формы в HTML

::placeholder - применяет стили к тексту-плейсхолдеру внутри элемента ввода. Например:

CSS

Copy code

```
input::placeholder {  
  color: #999;  
}
```

::selection - применяет стили к тексту, выделенному пользователем в элементе формы.

Например:

CSS

Copy code

```
input::selection {  
  background-color: #ffd700; /* желтый */  
  color: #333;  
}
```

::first-line - применяет стили к первой строке текста внутри элемента формы. Например:

CSS

Copy code

```
textarea::first-line {  
  font-weight: bold;  
}
```

::first-letter - применяет стили к первой букве текста внутри элемента формы. Например:

CSS

Copy code

```
input::first-letter {  
  color: red;  
}
```

Overflow

CSS `overflow` - это свойство, которое определяет, что происходит, когда содержимое элемента превышает размеры его контейнера. Это может быть полезно для управления тем, как браузер отображает контент, который не помещается в пределах определенной области.

Значения:

<code>visible</code>	Значение по умолчанию. Содержимое не обрезается, а отображается поверх границ блока-контейнера. Возможно перекрытие соседних блоков.
<code>hidden</code>	Содержимое блока обрезается без добавления какого-либо интерфейса прокрутки для просмотра содержимого вне области обрезки.
<code>scroll</code>	Содержимое обрезается до области полей, при этом блок становится прокручиваемым контейнером. Если браузер использует механизм прокрутки, который виден на экране, например, полосу прокрутки, этот механизм отображается независимо от того, обрезано ли какое-либо его содержимое. Это позволяет избежать проблем с появлением и исчезновением полос прокрутки в динамической среде. Размеры контейнера при этом не меняются, а полоса прокрутки вставляется между внутренним краем границы и внешним краем поля элемента.
<code>auto</code>	Браузер использует механизм прокрутки только при переполнении блока.
<code>initial</code>	Устанавливает значение свойства в значение по умолчанию.
<code>inherit</code>	Наследует значение свойства от родительского элемента.

```
overflow-x: visible;  
overflow-x: hidden;  
overflow-x: scroll;  
overflow-y: auto;  
overflow-x: inherit;  
overflow-x: initial;
```


Стилизация scrollbar-a

Для создания кастомного скроллбара в основных браузерах (Google Chrome, Firefox, Safari) вы можете использовать различные методы, включая вендорные префиксы и некоторые специфические свойства.

Помните, что стилизация скроллбара с использованием CSS ограничена и может не работать во всех браузерах, особенно в старых версиях. Кроме того, помните о возможных негативных последствиях для пользовательского опыта при изменении стандартного поведения скроллбара.

Safari:

Стандартные CSS-свойства для стилизации скроллбара не поддерживаются в Safari. Однако вы можете использовать JavaScript и библиотеки, такие как `simplebar`, чтобы создать кастомный скроллбар в Safari.

Google Chrome (WebKit):

Firefox

```
/* Скрыть полосу прокрутки */
* {
    scrollbar-width: none;
}

/* Установить ширину полосы прокрутки */
* {
    scrollbar-width: thin;
}

/* Установить цвет полосы прокрутки и ее фона */
* {
    scrollbar-color: #888 #f1f1f1;
}
```

Google-Chrome (opera)

```
/* Цвет фона */
::-webkit-scrollbar {
    background-color: #f1f1f1;
}

/* Цвет полосы прокрутки */
::-webkit-scrollbar-thumb {
    background-color: #888;
}

/* Ширина полосы прокрутки */
::-webkit-scrollbar-thumb {
    width: 8px;
}
```

Internet Explorer и Edge

```
/* Цвет фона */
::-ms-scrollbar {
    background-color: #f1f1f1;
}

/* Цвет полосы прокрутки */
::-ms-thumb {
    background-color: #888;
}

/* Ширина полосы прокрутки */
::-ms-thumb {
    width: 8px;
}
```

Вендорные префиксы

- Цель: Вендорные префиксы используются для экспериментальной или временной поддержки новых или недостаточно стандартизированных функций CSS. Это позволяет разработчикам использовать эти функции, несмотря на их неполное внедрение или поддержку в различных браузерах.
- Префиксы для разных браузеров: Различные браузеры имеют разные префиксы. Например, для WebKit (Chrome, Safari) используется -webkit-, для Mozilla (Firefox) - -moz-, для Opera - -o-, а для Microsoft (Internet Explorer, Edge) - -ms-.
- Использование: Вендорные префиксы добавляются перед стандартным именем свойства CSS. Например, если вы хотите использовать экспериментальную функцию, вы можете написать что-то вроде -webkit-... для WebKit браузеров.

text-overflow

Свойство позволяет обрезать строчное содержимое в случае, когда оно не помещается в блок-контейнер, визуалью обрезаю его или отображая многоточием. Текст может переполниться, например, когда ему запрещается перенос, например, из-за white-space: nowrap или отдельное слово слишком длинное, чтобы уместиться.

Значения:

<code>clip</code>	Значение по умолчанию. Текст обрезается в пределе области содержимого, при этом может отобразиться лишь часть символа.
<code>ellipsis</code>	Замещает текст, не уместившийся в блок, с помощью многоточия.
<code>initial</code>	Устанавливает значение свойства в значение по умолчанию.
<code>inherit</code>	Наследует значение свойства от родительского элемента.

```
text-overflow: clip;  
text-overflow: ellipsis;  
text-overflow: initial;  
text-overflow: inherit;
```

Ссылки + якорь

Ссылки с якорями в HTML используются для создания ссылок на конкретные разделы страницы. При клике на такую ссылку браузер автоматически прокручивает страницу к указанному элементу на странице. Для создания ссылки с якорем в HTML используется атрибут href с указанием якоря после символа решетки (#)

Обратите внимание, что элемент, на который ссылается якорь, должен иметь уникальный идентификатор (id), который указывается в атрибуте href.

Ссылка на элемент на той же странице:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Ссылка с якорем</title>
</head>
<body>

<a href="#section1">Перейти к разделу 1</a>

<!-- Раздел 1 -->
<h2 id="section1">Раздел 1</h2>
<p>Содержание раздела 1...</p>

</body>
</html>
```

В этом примере при клике на ссылку "Перейти к разделу 1" страница будет прокручиваться к элементу <h2> с идентификатором "section1".

Ссылка на элемент на другой странице:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Ссылка с якорем</title>
</head>
<body>

<a href="page2.html#section2">Перейти к разделу 2 на другой странице</a>

</body>
</html>
```

В этом примере ссылка "Перейти к разделу 2 на другой странице" ссылается на элемент с идентификатором "section2" на странице page2.html.

Ресурсы

stack-blitz - исходный код с урока - [ТЫК](#)

overflow и text-overflow - [ТЫК](#)

псевдоклассы - [ТЫК](#)

псевдоклассы документация - [ТЫК](#)

псевдоэлементы документация - [ТЫК](#)

псевдоклассы и псевдоэлементы для формы - [ТЫК](#)