

Использование атрибутов данных

HTML разработан с учетом расширяемости данных, которые должны быть связаны с определенным элементом, но не обязательно должны иметь какое-либо определенное значение. [data-* атрибуты](#) позволяют нам хранить дополнительную информацию о стандартных семантических элементах HTML без других хаков, таких как нестандартные атрибуты или дополнительные свойства в DOM.

Синтаксис HTML

Синтаксис прост. Любой атрибут любого элемента, имя атрибута которого начинается с `data-`, является атрибутом данных. Предположим, у вас есть статья и вы хотите сохранить дополнительную информацию, не имеющую никакого визуального представления. Просто используйте `data` для этого атрибуты:

HTML

```
<article
  id="electric-cars"
  data-columns="3"
  data-index-number="12314"
  data-parent="cars">
  ...
</article>
```

Доступ к JavaScript

Считать значения этих атрибутов в [JavaScript](#) также очень просто. Вы можете использовать `getAttribute()` их полное имя HTML, чтобы прочитать их, но стандарт определяет более простой способ: [DOMStringMap](#) вы можете прочитать их через `dataset` свойство.

Чтобы получить `data` атрибут через `dataset` объект, получите свойство по части имени атрибута после `data-` (обратите внимание, что дефисы преобразуются в [Camel case](#) ([верблюжий регистр](#))).

JavaScript

```
const article = document.querySelector("#electric-cars");
// The following would also work:
// const article = document.getElementById("electric-cars")

article.dataset.columns; // "3"
article.dataset.indexNumber; // "12314"
```

```
article.dataset.parent; // "cars"
```

Каждое свойство представляет собой строку и может быть прочитано и записано. В приведенном выше случае настройка `article.dataset.columns = 5` изменит этот атрибут на «5».

CSS-доступ

Обратите внимание: поскольку атрибуты данных являются простыми атрибутами HTML, вы можете получить к ним доступ даже из [CSS](#). Например, чтобы показать родительские данные в статье, вы можете использовать [сгенерированный контент](#) в CSS с помощью `attr()` функции:

CSS

```
article::before {  
    content: attr(data-parent);  
}
```

Вы также можете использовать [селекторы атрибутов](#) в CSS для изменения стилей в соответствии с данными:

CSS

```
article[data-columns="3"] {  
    width: 400px;  
}  
article[data-columns="4"] {  
    width: 600px;  
}
```

Вы можете увидеть, как все это работает вместе, в этом примере [JSBin](#).

Атрибуты данных также могут храниться для хранения постоянно меняющейся информации, например, результатов в игре. Используя здесь селекторы CSS и доступ к JavaScript, вы можете создавать отличные эффекты без необходимости писать собственные процедуры отображения. См. это [видео](#) с примером использования сгенерированного контента и переходов CSS ([пример JSBin](#)).

Значения данных представляют собой строки. Числовые значения должны быть заключены в селекторе, чтобы стиль вступил в силу.

Проблемы

Не храните контент, который должен быть видимым и доступным, в атрибутах данных, поскольку вспомогательные технологии могут не получить к ним доступ. Кроме того, поисковые роботы не могут индексировать значения атрибутов данных.

Ссылки и дополнения

- Эта статья адаптирована из статьи «[Использование атрибутов данных в JavaScript и CSS](#)» на сайте hacks.mozilla.org.
- Пользовательские атрибуты также поддерживаются в SVG 2; см. [HTMLElement.dataset](#) и [data-*](#) для получения дополнительной информации.
- [Как использовать атрибуты данных HTML](#) (Sitepoint)

From:
<https://wwoss.ru/> - worldwide open-source software

Permanent link:
https://wwoss.ru/doku.php?id=software:development:web:docs:learn:html:howto:use_data_attributes

Last update: **2025/12/06 19:58**

