

Атрибуты и свойства

Когда браузер загружает страницу, он «читает» (также говорят: «парсит») HTML и генерирует из него DOM-объекты. Для узлов-элементов большинство стандартных HTML-атрибутов автоматически становятся свойствами DOM-объектов. Например, для такого тега `<body id="page">` у DOM-объекта будет такое свойство `body.id="page"`. Но преобразование атрибута в свойство происходит не один-в-один! Когда у элемента есть `id` или другой стандартный атрибут, создаётся соответствующее свойство. Но этого не происходит, если атрибут нестандартный. Нужно учитывать, что стандартный атрибут для одного тега может быть нестандартным для другого.

Когда стандартный атрибут изменяется, соответствующее свойство автоматически обновляется. Это работает и в обратную сторону (за некоторыми исключениями), например, `input.value` синхронизируется только в одну сторону – атрибут → значение, но не в обратную:

DOM-узлы – это обычные объекты JavaScript. Мы можем их изменять, добавлять свойства и методы.

HTML-атрибуты

Все атрибуты (стандартные и нестандартные) доступны с помощью следующих методов:

`elem.hasAttribute(name)` – проверяет наличие атрибута.

`elem.getAttribute(name)` – получает значение атрибута.

`elem.setAttribute(name, value)` – устанавливает значение атрибута. Можем передать существующий и переустановить его.

`elem.removeAttribute(name)` – удаляет атрибут.

Кроме этого, получить все атрибуты элемента можно с помощью свойства **`elem.attributes`**: коллекция объектов, которая принадлежит ко встроенному классу **`Attr`** со свойствами **`name`** и **`value`**.

У **HTML-атрибутов** есть следующие особенности:

- Их имена *регистронезависимы* (`id` то же самое, что и `ID`).
- Их значения всегда являются *строками*.

DOM-свойства

У атрибутов, которые предназначены данному тегу HTML, как правило, есть соответствующие свойства.

DOM-свойства не всегда являются строками. Например, свойство **`input.checked`** (для чекбоксов) имеет логический тип. DOM-свойство **`href`** всегда содержит полный URL, даже если атрибут содержит относительный URL или просто `#hash`.

В большинстве ситуаций предпочтительнее использовать *DOM-свойства*. Нужно использовать *атрибуты* только тогда, когда DOM-свойства не подходят, когда нужны именно атрибуты, например:

- Нужен нестандартный атрибут. Но если он начинается с `data-`, тогда нужно использовать `dataset`.
- Мы хотим получить именно то значение, которое написано в HTML. Значение DOM-свойства может быть другим, например, свойство `href` – всегда полный URL, а нам может понадобиться получить «оригинальное» значение.

Синхронизация между атрибутами и свойствами

Когда **стандартный** атрибут изменяется, соответствующее свойство автоматически обновляется. Это работает и в обратную сторону. Но есть и исключения, например, **`input.value`** синхронизируется только в одну сторону – атрибут → значение, но не в обратную:

```
<input>
<script>
  let input = document.querySelector('input');
  // атрибут => значение
  input.setAttribute('value', 'text');
  alert(input.value); // text
```

```
// свойство => атрибут
input.value = 'newValue';
alert(input.getAttribute('value')); // text (не обновилось!)
</script>
```

В примере выше: Изменение атрибута value обновило свойство. Но изменение свойства не повлияло на атрибут. Иногда эта «особенность» может пригодиться, потому что действия пользователя могут приводить к изменениям value, и, если после этого мы захотим восстановить «оригинальное» значение из HTML, оно будет в атрибуте.

Нестандартные атрибуты, dataset

Используются для передачи пользовательских данных из HTML в JavaScript, или чтобы «помечать» HTML-элементы для JavaScript или для стилизации.

Чтобы избежать конфликтов, существуют атрибуты вида **data-***. Все атрибуты, начинающиеся с префикса «data-», зарезервированы для использования программистами. Они доступны в свойстве **dataset**. Например:

```
<body data-about="Elephants">
<script>
  alert(document.body.dataset.about); // Elephants
</script>
```

Атрибуты, состоящие из нескольких слов, к примеру **data-order-state**, становятся свойствами, записанными с помощью верблюжьей нотации: **dataset.orderState**.

Можно также **добавив свойство с dataset** получить **новый data-атрибут**

```
elem.dataset.size = "middle"; //<h1 data-size="middle">Изучаем JS</h1>
```