

Atrybuty elementu

Po co szukamy elementów na stronie?

Kiedy już znajdziemy element na stronie możemy z nim zrobić wiele rzeczy np. pobrać listę jego klas; zmodyfikować nazwę klas; zmienić tekst, który wyświetla; zmienić style; pobrać zbiór dataset itd.

Obok kilka przykładów atrybutów, które możemy pobierać lub modyfikować.

Przykładowe atrybuty

- `classList` – zwraca pseudotablicę klas HTML,
- `className` – zwraca lub ustawia nazwy klas HTML jako ciąg znaków,

- `id` – zwraca lub ustawia Id HTML jako ciąg znaków,
- `innerText` – zwraca/ustawia tekst znajdujący się w tagu (bez zagnieżdżonych tagów),
- `tagName` – zwraca nazwę tagu,
- `dataset` – zwraca obiekt dataset.

Atrybuty elementu - przykład

Kod HTML

```
<a href="www.google.com" class="bar" id="glink" data-foo="1">
  <h1>Google</h1>
</a>
```

Kod JavaScript

```
const link = document.querySelector('#glink');
link.className; /* "bar" */
link.id;        /* "glink" */
link.innerText; /* "Google" */
link.tagName;   /* "A" - nazwa zwracana jest wielkimi literami */
link.dataset;   /* { foo: "1" } */
```

<https://developer.mozilla.org/en-US/docs/Web/API/Element>

Atrybut dataset

Dane powiązane z tagiem

Możemy przetrzymać pewne dane powiązane z tagiem HTML, które mogą nam się później przydać, na przykład:

- tagi do zdjęcia,
- tooltip,
- ID obiektu.

Takie dane powinniśmy trzymać w specjalnym atrybucie zaczynającym się od `data-`

Więcej:

https://developer.mozilla.org/en-US/docs/Web/Guide/HTML/Using_data_attributes

- W JavaScript mamy dostęp do specjalnego obiektu `dataset`, należącego do elementu.
- Dzięki niemu możemy nastawiać lub wczytywać informacje z `dataset`.

dataset

Wszystkie dane poniższego elementu możemy z łatwością wypisać z obiektu **dataset**:

```
<div id="user" data-id="424" data-logged-in>John Doe</div>
```

```
const myUser = document.querySelector("#user");  
console.log(myUser.dataset);  
/* {id: "424", loggedIn: ""} */
```

dataset

Dostęp do pól obiektu

```
<div id="user" data-id="424" data-logged-in>John Doe</div>
```

```
const myUser = document.querySelector("#user");
```

```
console.log(myUser.dataset);  
/* {id: "424", loggedIn: ""} */  
console.log(myUser.dataset.id);  
/* 424 */  
console.log(myUser.dataset.loggedIn);  
/*ten element jest pusty */
```

dataset

Zmiana wartości w dataset

Do istniejącego obiektu `dataset` możemy przypisywać nowe wartości.

```
<div id="user" data-id="424" data-logged-in>John Doe</div>
```

```
const myUser = document.querySelector("#user");
```

```
console.log(myUser.dataset.id);
```

```
myUser.dataset.id = "345";
```

```
console.log(myUser.dataset.id);
```

dataset

Zmiana wartości w dataset

Do istniejącego obiektu `dataset` możemy przypisywać nowe wartości.

```
<div id="user" data-id="424" data-logged-in>John Doe</div>
```

```
const myUser = document.querySelector("#user");
```

```
console.log(myUser.dataset.id);
```

424 - stara wartość.

```
myUser.dataset.id = "345";
```

```
console.log(myUser.dataset.id);
```

dataset

Zmiana wartości w dataset

Do istniejącego obiektu `dataset` możemy przypisywać nowe wartości.

```
<div id="user" data-id="424" data-logged-in>John Doe</div>
```

```
const myUser = document.querySelector("#user");
```

```
console.log(myUser.dataset.id);
```

424 - stara wartość.

```
myUser.dataset.id = "345";
```

Ustawienie nowej wartości dla data-id.

```
console.log(myUser.dataset.id);
```


dataset

Zmiana wartości w dataset

Do istniejącego obiektu `dataset` możemy przypisywać nowe wartości.

```
<div id="user" data-id="424" data-logged-in>John Doe</div>
```

```
const myUser = document.querySelector("#user");
```

```
console.log(myUser.dataset.id);
```

424 - stara wartość.

```
myUser.dataset.id = "345";
```

Ustawienie nowej wartości dla data-id.

```
console.log(myUser.dataset.id);
```

345 - nowa wartość.

dataset

Nowe pole w dataset

Możemy dodawać nowy, nieistniejący wcześniej dataset.

```
<div id="user" data-id="424" data-logged-in>John Doe</div>
```

```
const myUser = document.querySelector("#user");
```

```
console.log(myUser.dataset.something);
```

```
myUser.dataset.something = "new value";
```

```
console.log(myUser.dataset.something);
```

dataset

Nowe pole w dataset

Możemy dodawać nowy, nieistniejący wcześniej dataset.

```
<div id="user" data-id="424" data-logged-in>John Doe</div>
```

```
const myUser = document.querySelector("#user");
```

```
console.log(myUser.dataset.something);
```

undefined

```
myUser.dataset.something = "new value";
```

```
console.log(myUser.dataset.something);
```

dataset

Nowe pole w dataset

Możemy dodawać nowy, nieistniejący wcześniej dataset.

```
<div id="user" data-id="424" data-logged-in>John Doe</div>
```

```
const myUser = document.querySelector("#user");
```

```
console.log(myUser.dataset.something);
```

undefined

```
myUser.dataset.something = "new value";
```

Wstawienie nowej wartości.

```
console.log(myUser.dataset.something);
```

dataset

Nowe pole w dataset

Możemy dodawać nowy, nieistniejący wcześniej dataset.

```
<div id="user" data-id="424" data-logged-in>John Doe</div>
```

```
const myUser = document.querySelector("#user");
```

```
console.log(myUser.dataset.something);
```

undefined

```
myUser.dataset.something = "new value";
```

Wstawienie nowej wartości.

```
console.log(myUser.dataset.something);
```

"new value"