Więcej o elemencie

classList

Właściwość classList elementu zwraca listę wszystkich klas tego elementu.

Możemy łatwo z nią pracować dzięki następującym metodom:

el. classList.add(className) – dodaje podaną klasę,

el. classList.remove(className) – usuwa podaną klasę,

el. classList.toggle(className) – przełącza podaną klasę (czyli usuwa ją jeżeli jest, jeżeli jej nie ma, to dodaje).

classList

Mamy taki element:

```
HTML:
<div id="myDiv" class="class1 class2"></div>

JavaScript:
var myDiv = document.getElementById("myDiv");
```

Możemy łatwo wczytać jego wszystkie klasy:

```
console.log(myDiv.classList);
/* ["class1", "class2"] - pseudotablica */
console.log(myDiv.className);
/* "class1 class2" - ciąg znaków */
```

classList

Możemy dodać nową klasę:

```
myDiv.classList.add("nowaKlasa");
console.log(myDiv.classList);
```

Możemy usunąć jedną z jego klas:

```
myDiv.classList.remove("class1");
console.log(myDiv.classList);
```

Możemy przełączać daną klasę:

```
myDiv.classList.toggle("toggleClass1");

myDiv.classList.toggle("nowaKlasa");
```

dataset

Wiesz już czym jest dataset i jak wyświetlić jego wartości. Sprawdźmy teraz jak możemy je ustawiać i zmieniać.

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

var 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

Do istniejącego obiektu dataset możemy przypisywać nowe wartości.

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

var myUser = document.querySelector("#user");

console.log(myUser.dataset.id);

myUser.dataset.id = 4444;

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

dataset – nowa wartość

Do istniejącego obiektu dataset możemy przypisywać nowe wartości.

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

var myUser = document.querySelector("#user");

console.log(myUser.dataset.something);

myUser.dataset.something = "new value";

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

atrybuty alementów

Z poziomu JavaScript możemy edytować wszystkie atrybuty danego elementu.

Służą do tego metody przedstawione na kolejnych slajdach.

```
HTML:
<a href="www.google.com" id="glink">Hello Google!</a>

JavaScript:
var link = document.querySelector("#glink");
```

atrybuty alementów

el.hasAttribute(attrName) –sprawdza, czy element ma podany atrybut. W odpowiedzi dostajemy wartość logiczną.

```
link.hasAttribute("href");
/* true */
```

el.getAttribute(attrName) – zwraca wartość podanego atrybutu.

```
link.getAttribute("href");
/* "www.google.com" */
```

atrybuty alementów

el.removeAttribute(attrName) – usuwa podany atrybut.

```
link.removeAttribute("href");
link.hasAttribute("href");
/* false */
```

el.setAttribute(attrName, attrValue) – nastawia wartość podanego atrybutu.

```
link.setAttribute("href", "www.something.com");
link.hasAttribute("href");
/* true */
link.getAttribute("href");
/* "www.something.com" */
```

Pobieranie i modyfikacja styli CSS

- Obiekt style przechowuje wszystkie wartości jako ciągi znaków.
- Tak samo będą one nam zwracane i tak powinniśmy je nastawiać.
- Obiekt style "widzi" tylko style ustawione za pomocą JavaScript, nie widzi styli CSS.
- Ustawiając właściwości CSS należy stosować zapis camelCase.

Aktualną wartość stylu możemy wczytać:

```
element.style.backgroundColor;
```

Albo nastawić nową wartość:

```
element.style.backgroundColor = "blue";
```

Czas na zadania