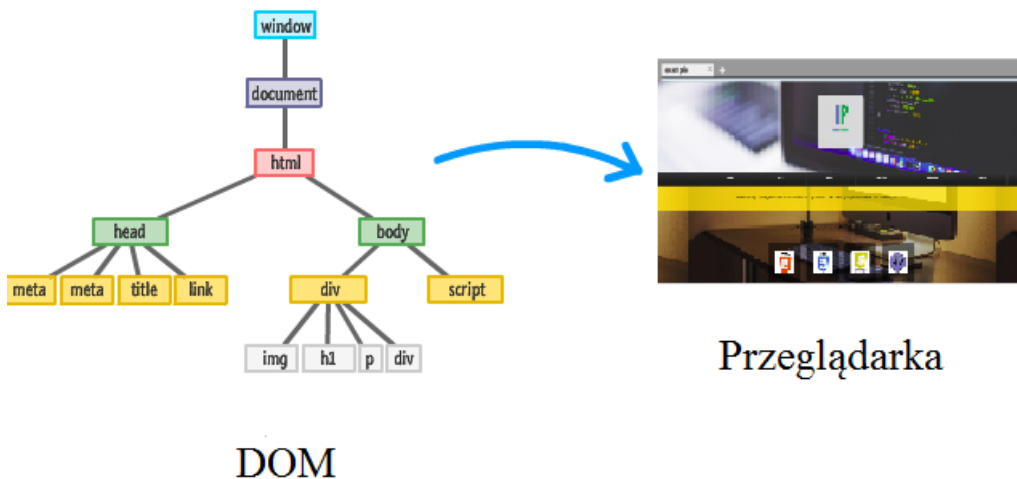


DOM – DOCUMENT OBJECT MODEL

DOM to sposób reprezentacji złożonych dokumentów XML i HTML w postaci modelu obiektowego. Elementy struktury kodu HTML danej witryny zamieniane są przez przeglądarkę na stronę internetową. Informacje na temat interpretacji kodu HTML zapisane są w elementach będących obiektami. Obiekty te tworzą obiektowy model dokumentu.

DOM opisuje hierarchię obiektów na stronie oraz udostępnia metody i właściwości, które umożliwiają manipulowanie nimi. Głównym obiektem jest obiekt `window` (okno przeglądarki) – zawiera on wszystkie inne obiekty, funkcje i właściwości. W oknie znajduje się obiekt `document`, czyli otwarta strona internetowa. W obiekcie `document` znajdują się obiekty strony.

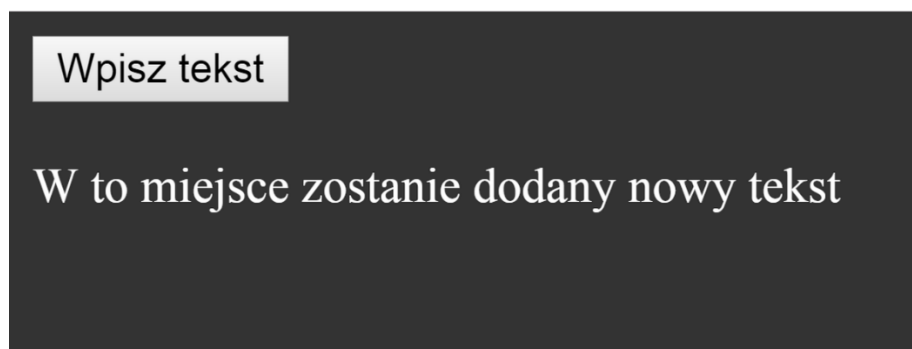


Dzięki DOM język JavaScript staje się narzędziem tworzenia dynamicznych stron internetowych.

PODSTAWOWE METODY I WŁAŚCIWOŚCI DO MODYFIKACJI ELEMENTÓW DOM

1. Metoda `getElementById()` to metoda pozwalająca na utworzenie tzw. uchwytu dla danego elementu wskazanego poprzez jego `id`
2. Właściwość `innerHTML` pozwala na wstawienie treści do wskazanego przez metodę `getElementById()` elementu.

Przykład:



Kod html:

```
<body>
<button onclick="wpisz()">Wpisz tekst</button>
<br /><br />
<div id="b1">W to miejsce zostanie dodany nowy tekst</div>
</body>
```

Kod JavaScript:

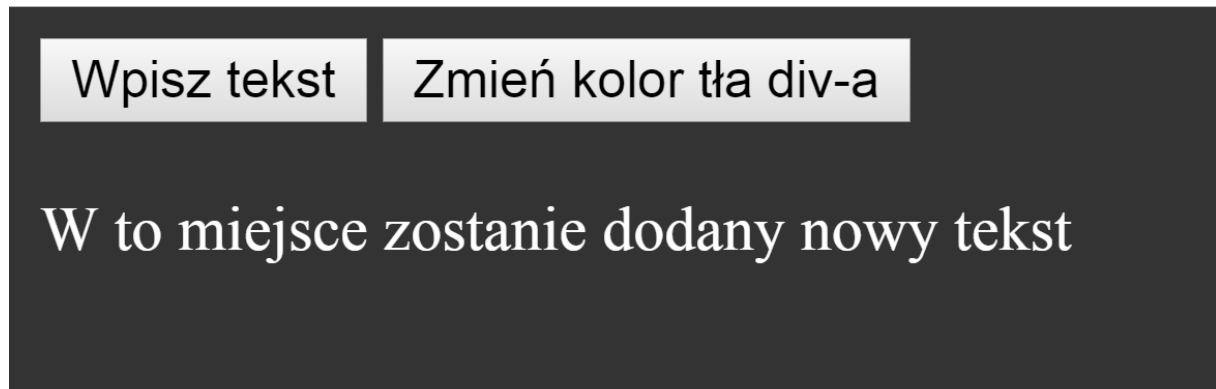
```
<script>
function wpisz () {
document.getElementById ("b1").innerHTML = "<b>Wpisany
tekst</b>";
}
</script>
```



Kliknięcie na buton „Wpisz tekst” spowoduje uruchomienie funkcji `wpisz ()`, w której najpierw tworzony jest uchwyt za pomocą metody `getElementById ("b1")` dla elementu o `id="b1"` i wyczytanie za pomocą właściwości `innerHTML` treści: „wpisany tekst”.

3. Metoda `getElementsByTagName ()` to metoda pozwalająca na utworzenie tzw. uchwytu (odwołania się) dla danego elementu wskazanego poprzez nazwę tagu HTML.

Przykład:



Kod html:

```
<body>

<button onclick="wpisz()">Wpisz tekst</button>
<button onclick="zmien()">Zmień kolor tła div-a</button>

<br /><br />

<div id="b1">W to miejsce zostanie dodany nowy tekst</div>

</body>
```

Kod JavaScript:

```
<script>

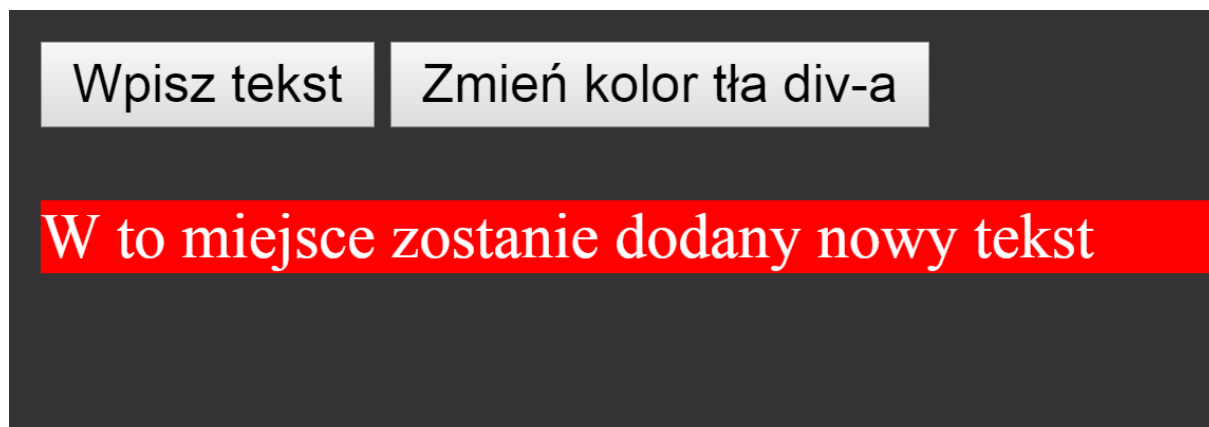
function zmien() {

var element = document.getElementsByTagName("div")[0];
element.style.backgroundColor = "red";

}

</script>
```

Efekt działania - kolor tła tekstu zmieniony na czerwony:



Używając metody `document.getElementsByTagName("div")[0]` musimy doprecyzować, by oprócz wskazania konkretnego typu znacznika HTML (tagu) określić o który konkretnie nam chodzi – w tym celu ustawiamy wartość indeksu naszego elementu `[0]` licząc od 0 jak w tablicach. Przykładowo, jeżeli w kodzie będziemy mieli więcej elementów typu `<div>`, to indeksujemy je tak:

```
<div></div>      <!-- indeks [0] -->
<div></div>      <!-- indeks [1] -->
<div></div>      <!-- indeks [2] -->
<div></div>      <!-- indeks [3] -->
```

Jeśli więc zapiszę `document.getElementsByTagName("div")[3]` to utworzę uchwyt dla czwartego elementu `<div>`.

W naszym przykładzie wybraliśmy element o indeksie `[0]`

```
var element = document.getElementsByTagName("div")[0];
element.style.backgroundColor = "red";
```

Język JavaScript jest zorientowany obiektowo, element który wskazaliśmy jest obiektem któremu możemy zmieniać pewne właściwości, jak np. styl i tak mamy taki zapis:

Obiekt.właściwość = 'wartość';



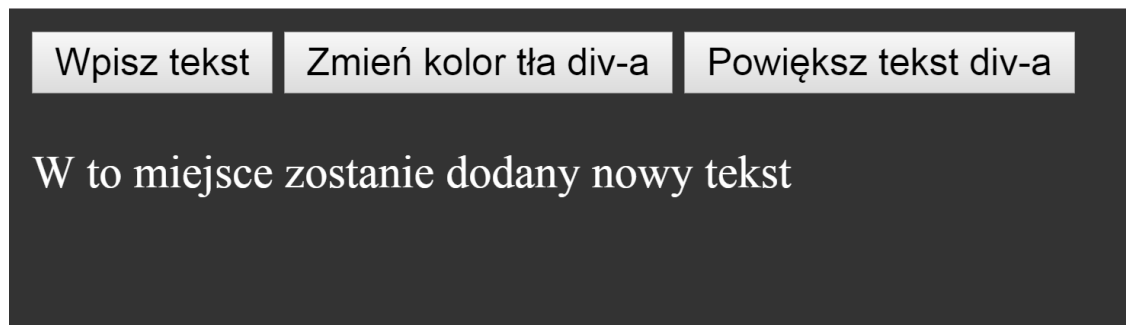
Zatem możemy z poziomu JavaScript-u dynamicznie zmieniać i tworzyć nowy styl, ustawiając nowe atrybuty:

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

W naszym przykładzie zmieniamy kolor tła: mamy obiekt `element`, któremu zmieniamy właściwość `backgroundColor` w `style`.

4. Metoda `getElementsByClassName()` to metoda pozwalająca na utworzenie tzw. uchwytu (odwołania się) dla danego elementu wskazanego poprzez nazwę klasy przypisanej do tego elementu.

Przykład:



Kod HTML:

```
<body>

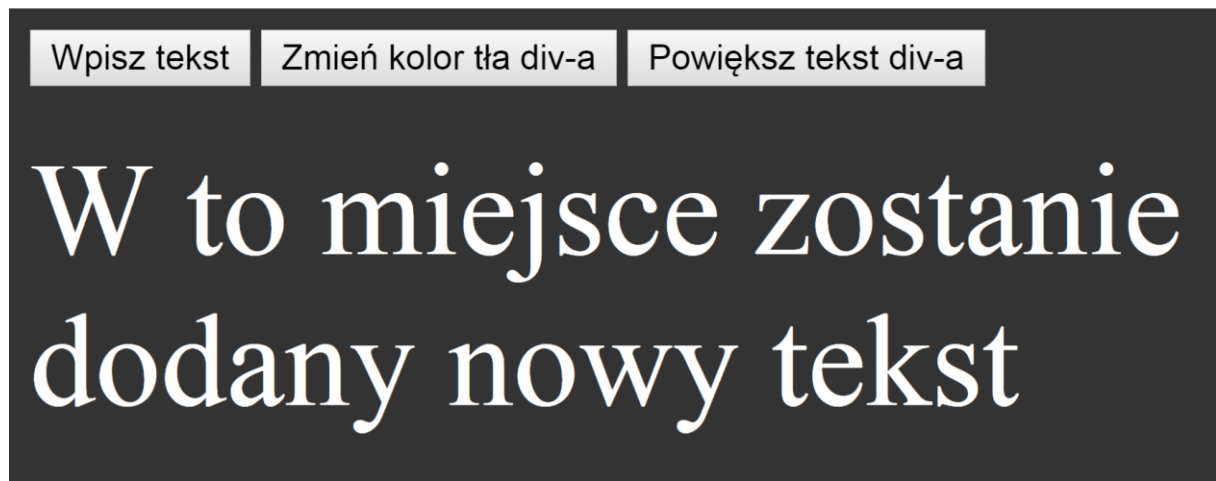
<button onclick="wpisz()">Wpisz tekst</button>
<button onclick="zmien()">Zmień kolor tła div-a</button>
<button onclick="powieksz()">Powiększ tekst div-a</button>
<br /><br />
<div id="b1" class="c1">W to miejsce zostanie dodany nowy
tekst</div>

</body>
```

Kod JavaScript:

```
function powieksz () {  
  
var element2 = document.getElementsByClassName ("c1") [0];  
element2.style.fontSize = "50px";  
  
}
```

Efekt działania – powiększenie tekstu do 50px:



Tworzenie uchwytu i zmiana właściwości obiektu wykonana jest w analogiczny sposób jak w poprzedniej metodzie.