

Wprowadzenie do JavaScript

Model DOM W3C (Document Object Model) służy do przedstawienia dokumentów HTML i XML (XHTML). CSS i JavaScript używają go do operowania na elementach dokumentu. Wg W3C DOM: dokument to drzewiasty zbiór węzłów. Węzeł może być **elementem**, ciągiem tekstu, komentarzem, itp. Każdy węzeł może mieć jednego rodzica, dowolną ilość braci (lub sąsiadów) i, jeśli jest **elementem**, dowolną ilość dzieci. Uproszczona struktura DOM przedstawia rys.1. JavaScript generalnie jest językiem obiektowym – prawie wszystkie funkcje, z których korzystamy są wywoływane z jakiegoś obiektu. Przeglądarka udostępnia nam obiekt **window**, którego bezpośrednim potomkiem jest obiekt **document** reprezentujący np. dokument HTML (szerzej na ten temat znajdziesz w materiałach wykładowych). Z obiektu **document** możemy wywoływać różne przydatne funkcje (metody), umożliwiające dostęp do dowolnego elementu na stronie a także jego modyfikacje.

W JS odwołanie się do danego elementu (metoda zwraca nam element HTML np. body, div, form, input), który posiada unikatowy atrybut **id** odbywa się za pomocą metody:

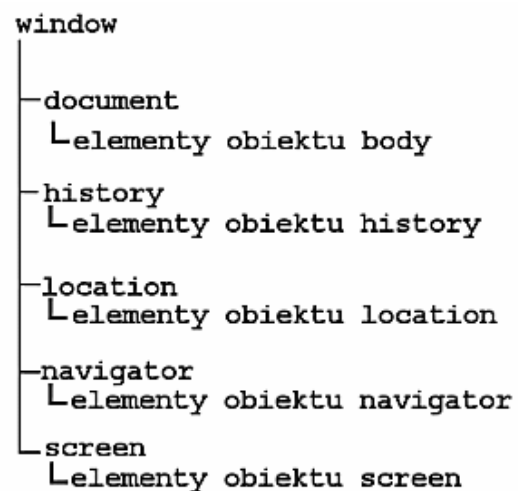
element=getElementById('id')

Do pobrania lub ustalenia zawartości tak pobranego elementu służy metoda:

element.value

Obsługa interakcji z użytkownikiem jest realizowana za pomocą obsługi zdarzeń, które mogą zachodzić na elementach dokumentu (np. zdarzenia związane z kliknięciem przycisku, przemieszczaniem się myszy nad elementem, wysłanie formularza, aktywacja /dezaktywacja danego elementu itp.). Najczęściej zdarzenia takie są odpowiednio obsługiwane za pomocą wywołania funkcji JS np.:

onclick="funkcja_js()".



Rys.1. Uproszczona struktura dom

UWAGA!

Pamiętaj, że JavaScript rozróżnia duże i małe litery, dlatego uważnie pisz nazwy metod. W JavaScript obowiązuje następująca konwencja nazw: wszystkie metody zaczynają się małą literą, ale poszczególne wyrazy w nazwie metody pisane są już z wielkiej litery np.

getElementById. Ponieważ jest to metoda obiektu **document** dlatego sposób jej wywołania jest postaci np. **document.getElementById('id');**

Ćw.1. Prosty formularz obliczeniowy

Poniższa strona ze skryptem JS generuje obraz przedstawiony na rys.2 – za pomocą funkcji **oblicz()** realizujemy **dostęp do elementów HTML za pomocą metody getElementById**. Należy:

- uruchomić i przeanalizować działanie skryptu osadzonego wewnątrz dokumentu *js1.html* (w części nagłówkowej **<head>**):

```

<!DOCTYPE ...>
<head>
  <meta ... />
  <title> Dostęp do pól formularza</title>
  <script type="text/javascript">

```

```

function oblicz()
{ //Słowo var jest opcjonalne - oznacza zmienną; typ tej zmiennej
  //będzie zależał od jej wartości;
  //Pobierz element o id=l1 do zmiennej l1:
  var l1=document.getElementById('l1');
  //pobierz wartość (łańcuch tekstowy) wpisaną w polu formularza o id='l1':
  l1=l1.value;
  //Przekonwertuj (jeśli się uda) l1 do wartości typu int:
  l1=parseInt(l1);
  //Jeśli udało się próba konwersji to l1 zawiera wartość całkowitą
  //Analogicznie realizujemy pobranie wartości z drugiego pola tekstowego,
  //ale tym razem wszystko w jednej instrukcji:
  var l2=parseInt(document.getElementById('l2').value);
  //Teraz już możemy obliczyć sumę i ustawić wartość pola tekstowego
  //o id='suma':
  var s=document.getElementById('suma');
  s.value=l1+l2;
}
</script>
</head>
<body> <div><h3>Dodawanie</h3>
<!-- Elementy formularza (pola tekstowe i przycisk) zostały umieszczone
      w polach tabeli; każde pole formularza posiada unikatowe id, dzięki
      któremu możemy w JavaScript zastosować metodę getElementById -->
<table><tbody>
  <tr><td>Liczba 1:</td><td><input id="l1" /></td> </tr>
  <tr><td>Liczba 2:</td><td><input id="l2" /></td> </tr>
  <tr><td><button onclick="oblicz()">Oblicz sumę : </button></td>
  <!-- Pole wynikowe suma posiada dodatkowy atrybut disabled, dzięki
        czemu tylko metoda obliczeniowa w JavaScript ma możliwość ustawiania
        wyniku, dla użytkownika strony pole jest niedostępne -->
  <td><input id="suma" disabled="disabled" value=""/></td></tr>
</tbody></table></div></
</body>
</html>

```

Zauważ, że w HTML pola formularza nie są ujęte w blok `<form>` - są one wykorzystywane tylko do pobrania i ustawienia na nich wartości po stronie klienta (przeglądarki), nie są wysyłane na serwer, wobec czego nie ma potrzeby stosowania znacznika `<form>`

- b) zmodyfikować stronę tak aby korzystała ze skryptu umieszczonego w zewnętrznym pliku - w tym celu należy cały kod funkcji:

```
function oblicz() { ... }
```

zapisać w pliku np. *funkcje.js* a następnie w dokumencie HTML wymienić fragment:

```
<script type="text/javascript">
```

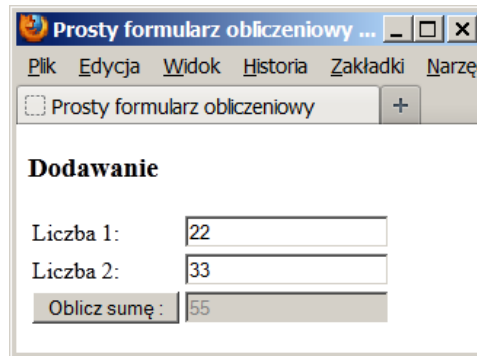
```
  function oblicz(){ ... }
```

```
</script>
```

na:

```
<script type="text/javascript" src="funkcje.js"></script>
```

- c) sprawdzić czy wszystko działa poprawnie.



Rys.2.

Naszym zadaniem będzie teraz dodanie do projektu z poprzednich laboratoriów kolejnych stron wykorzystujących JavaScript.

Dla przypomnienia struktura projektu, uzupełniona o folder na pliki *js* powinna wyglądać następująco:

- WWW – *index.html*
 - *tabele.html*
 - *formularze.html*
- CSS – *style.css*
 - *tabele.css*
 - *formularze.css*
- JS – *funkcje.js*

Ćw. 2. Formularz obliczeniowy w projekcie

Na podstawie ćwiczenia 1 opracować skrypt JS do obliczenia rat przy spłacie pożyczki

w równych ratach miesięcznych (rys.3) korzystając z wzoru:
$$rata = \frac{K * pr_mc}{1 - \frac{1}{(1 + pr_mc)^n}}$$

gdzie: *K* - kwota pożyczki

pr - oprocentowanie roczne

n - ilość rat

pr_mc=*pr*/12 – oprocentowanie w skali miesiąca

Zawartość formularza umieścić na stronie *obliczenia.html*, której struktura powinna bazować na szablonie z pliku *index.html*. Zwróć uwagę na rozwijaną opcję menu *Obliczenia w formularzu* z linkiem do nowej strony *obliczenia.html*, który pojawił się w menu (Rys.3). Pamiętaj o dodaniu w części nagłówkowej dokumentu znacznika:

```
<script type="text/javascript" src="JS/funkcje.js"></script>
```



Rys.3.