

ZAJĘCIA SPECJALISTYCZNE

JavaScript



TEMAT 3-2: JavaScript DOM – obiekt „document”.

Autor dokumentu: Wojciech Galiński

wtorek, 5 lutego 2013 r.

312[01]/T,SP/MENiS/2004.06.14

ŹRÓDŁA WIEDZY: <http://pl.wikipedia.org/>, <http://webmaster.helion.pl/index.php/kurs-javascript>,
<http://krook.org/jsdom/>, <http://www.dynamicdrive.com/>, <http://www.w3schools.com>.

Zagadnienia obowiązkowe

1. **Ciasteczka (ang. cookies)** – „niewielkie informacje tekstowe, wysyłane przez serwer WWW i zapisywane po stronie użytkownika (zazwyczaj na twardym dysku). Domyślne parametry ciasteczek pozwalają na odczytanie informacji w nich zawartych jedynie serwerowi, który je utworzył”. Ciasteczka mają postać tekstu w formacie:

`nazwa_ciasteczka_1=wartosc_ciasteczka_1; nazwa_ciasteczka_2=wartosc_ciasteczka_2; ...`

2. **Obiekt „document”** – umożliwia dostęp do elementów dokumentu HTML oraz manipulowanie tymi elementami. Bardzo pomocna okazuje witryna zawierająca hierarchię dokumentu HTML w JavaScript DOM. Znajduje się ona m. in. pod adresem: <http://krook.org/jsdom/>.

Obiekt „document” posiada następujące składowe:

➔ właściwości:

- ✓ **domain** – zwraca nazwę domeny serwera, na którym został umieszczony bieżący dokument HTML (dotyczy tylko dokumentów umieszczonych na serwerze WWW);
- ✓ **location, URL** – zwraca URL bieżącego dokumentu.
- ✓ **title** – zwraca tytuł dokumentu HTML (tekst pomiędzy znacznikami „<title></title>”);
- ✓ **referrer** – zwraca URL dokumentu z którego wczytano bieżący dokument HTML (dotyczy tylko dokumentów umieszczonych na serwerze WWW);

PRZYKŁADY:

```
document.write('Domain: "<b>' + document.domain + '</b>"<br />' +  
  'Location: "<b>' + document.location + '</b>"<br />' +  
  'URL: "<b>' + document.URL + '</b>"<br />' +  
  'Title: "<b>' + document.title + '</b>"<br />' +  
  'Referrer: "<b>' + document.referrer + '</b>"');
```

- ✓ **linkColor** – domyślny kolor odsyłacza (brak wartości oznacza, że użyty zostanie domyślny kolor dla odsyłaczy zdefiniowany w przeglądarce);
- ✓ **body** – to obiekt reprezentujący ciało (treść) dokumentu – każdy znacznik przedstawiony jest jako obiekt składowy tej właściwości (obiekty mogą stać obok siebie albo zawierać się w sobie – tak powstaje hierarchia obiektów w dokumencie HTML);
- ✓ **characterSet** – zwraca informację o kodowaniu znaków w dokumencie (zwykle jest to: **UTF-8**);
- ✓ **compatMode** – zwraca informację o trybie kompatybilności dokumentu ze standardami (np. **CSS1Compat**). PRZYKŁAD: `document.write('document.compatMode');`
- ✓ **cookie** – zwraca ciąg znaków zawierający „cookie” danego dokumentu;
- ✓ **lastModified** – zwraca informację o czasie ostatniej modyfikacji dokumentu HTML;
- ✓ **anchors, applets, embeds, forms, images, links, plugins, styleSheets** – zawierają listy elementów: etykiet, apletów Java, obiektów zagnieżdżonych, formularzy, obrazów, odsyłaczy, dodatków, arkuszy stylów CSS, np.
`for (id in document.links) document.write(document.links[id]);`
- ✓ **fgColor, bgColor** – domyślny kolor tekstu oraz tła dokumentu HTML (brak wartości oznacza, że użyty zostanie domyślny **kolor / tła** zdefiniowany w przeglądarce).

➔ metody (wybrane):

- ✓ **getElementById(id)** – umożliwia odwoływanie się do elementu o „id” przekazanym w parametrze funkcji, np.
`// Zwraca obiekt elementu o id "moj_znacznik"`
`document.write(document.getElementById("moj_znacznik"));`
- ✓ **getElementsByName(nazwa)** – zwraca tablicę elementów o nazwie „name” i umożliwia odwoływanie się do jej elementów (wykorzystywane przeważnie do obsługi kontrolek formularza), np.
`// Zwraca tablicę wszystkich elementów o nazwie "przycisk"`
`document.write(document.getElementsByName("przycisk"));`
- ✓ **getElementsByTagName(tagname)** – zwraca tablicę znaczników HTML „tagname” i umożliwia odwoływanie się do jej elementów, np.
`<div>Tekst 1</div><div>Tekst 2</div><div>Tekst 3</div>`
`<script type="text/javascript">`
`nr = prompt('Podaj numer tekstu:') - 1;`
`if (nr!=0 && nr!=1 && nr!=2) alert('Błędny numer tekstu.');`
`else`
`{`
`tab = document.getElementsByTagName('div');`
`alert(tab[nr].innerHTML);`
`}`
`</script>`
- ✓ **write(tekst), writeln(tekst)** – wysyła tekst do dokumentu (może zawierać m. in. znaczniki HTML). Druga z funkcji dodaje na końcu znak końca wiersza widziany np. wewnątrz znaczników „**<pre>**”, np. `document.write("Suma: " + (5+5));`

PRZYKŁAD:

```
<form name="f1" action="#">
  <input type="text" name="x" value="0" /> +
  <input type="text" name="x" value="0" /> =
  <span id="wynik">0</span><br />
  <input type="button" name="przycisk" value="Wylicz"
onclick="wylicz_sume()" />
</form>

<script type="text/javascript">
  function wylicz_sume()
  {
    x = document.getElementsByName('x');
    suma = parseInt(x[0].value) + parseInt(x[1].value);
    span = document.getElementById('wynik');
    span.innerHTML = suma;
  }
</script>
```

Zadania

1. Dodaj do przykładu z informacją o domenie, lokalizacji, URL, tytule i przekierowaniu dokumentu 4 następujące elementy: informację o czasie ostatniej modyfikacji dokumentu, ciąg zawierający „ciasteczkowe” ustawienia dokumentu, informacje o kodowaniu znaków i trybie kompatybilności dokumentu HTML.
2. Za pomocą funkcji składowej „prompt” poproś o wpisanie tekstu i ustaw ten tekst jako tytuł dokumentu HTML.
3. Za pomocą kontrolek „input text” (odwołuj się do nich za pomocą nazwy znacznika) wczytaj 10 wyrazów i wyświetl je wewnątrz znacznika „div” w porządku alfabetycznym (w roli separatora użyj znak przecinka).