

Limbajul JavaScript si HTML

Mihai Gabroveau

Document Object Model (DOM)

- n Pentru a schimba continutul paginii afisate in fereastra browserului este necesara cunoastrea modului de accesare a elementelor
- n Basic DOM este standard al W3C si este compatibil cu majoritatea browserelor
- n Elementul care se afla la baza ierahiei este **window**, si toate celelalte elemente sunt descendente ale acestuia:
 - .. Fiecare variabila JavaScript este un camp al acestui obiect
 - .. In DOM, toate variabilele se presupun ca incep cu "window."

Limbajul JavaScript

3

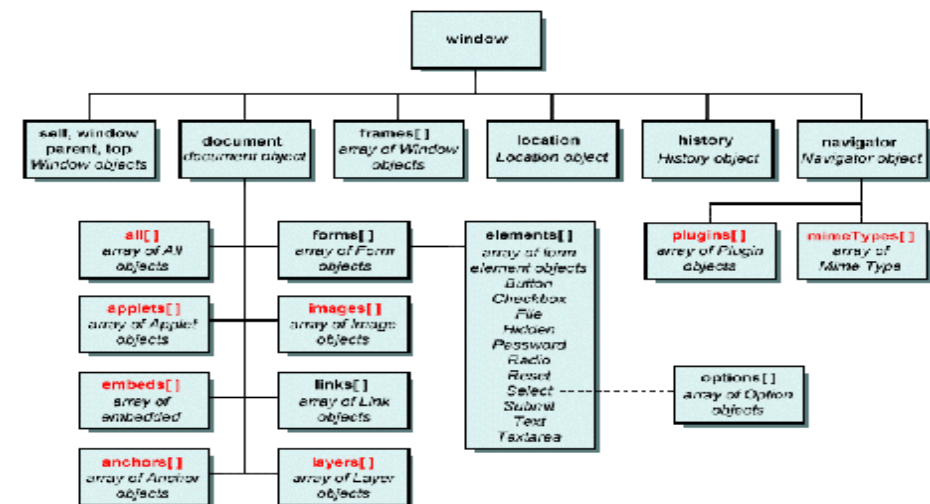
HTML DOM si JavaScript

- n JavaScript permite manipularea unei pagini HTML prin intermediul Document Object Model (DOM) ce descrie structura paginii HTML
 - .. HTML DOM $\neq$ XML DOM
- n Prin intermediul DOM-ului putem face o multime de operatii:
 - .. Putem accesa prin elementele din cadrul unei pagini HTML
 - .. Putem schimba continutul unei pagini HTML
 - .. Putem captura evenimente atasand handleri de evenimente

Limbajul JavaScript

2

Ierarhia de clase DOM

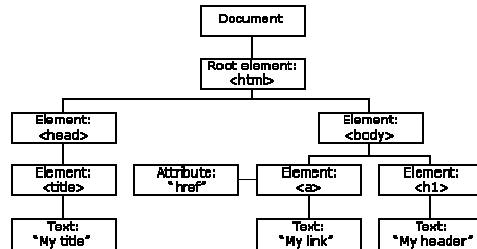


Limbajul JavaScript

4

HTML DOM - Exemplu

```
1. <html>
2. <head>
3.   <title>My Title</title>
4. </head>
5. <body>
6.   <a href="">My Link</a>
7.   <h1>My Header</h1>
8. </body>
9. </html>
```



DOM-ul asociat paginii HTML

Campurile obiectului **window**, I

- n **window**
 - .. Fereastra curenta (de regula nu se specifica).
- n **self**
 - .. Referinta la fereastra curenta. Identica cu window (window.window).
- n **parent**
 - .. Referinta la fereastra/cadrul parinte al ferestrei/cadrului curent.
- n **top**
 - .. Referinta catre fereastra navigatorului situata in prim plan.
- n **frames[]**
 - .. Setul de cadre (frames, in limba engleza) afisate. Frame-urile la randul lor sunt de tipul window.
- n **frames.length**
 - .. numarul de cadre din fereastra curenta.

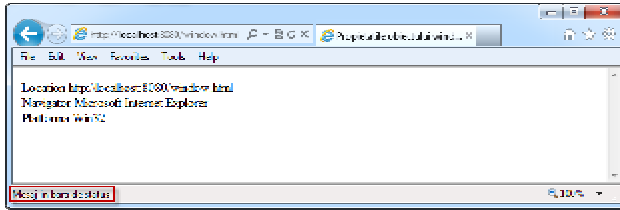
Document Object Model (DOM)

- n Fiecare document HTML incarcat intr-o fereastra (**window**) devine un obiect de tip **Document**.
- n Obiectul Document pune la dispozitie metode prin care putem accesa toate elementele HTML din cadrul paginii

Campurile obiectului **window**, II

- n **document**
 - .. Documentul HTML ce este afisat in fereastra curenta.
- n **location**
 - .. URL-ul documentului afisat in fereastra curenta. Daca aceasta proprietate primeste un nou URL atunci document referit de acest URL este incarcat. Daca dorim sa facem refresh la pagina curenta putem apela **location.reload()**.
- n **navigator**
 - .. Referinta la un obiect de tip Navigator ce reprezinta browser-ul ce a incarcat acea pagina.
 - n **navigator.appName** – numele browserului, ex: "Netscape"
 - n **navigator.platform** – numele sistemului de operare, ex: "Win32"
- n **status**
 - .. Sirul de caractere afisat in bara de status a browserului.
 - .. Putem schimba acest text printr-o simpla atribuire.

Exemplu



```
1. <html>
2.   <head>
3.     <title>Proprietatile obiectului window</title>
4.   </head>
5.   <body>
6.     <script type="text/javascript">
7.       document.write("Location " + window.location + "<br>");
8.       document.write("Navigator " + navigator.appName + "<br>");
9.       document.write("Platforma " + navigator.platform + "<br>");
10.      window.status = "Mesaj in bara de status";
11.    </script>
12.  </body>
13. </html>
```

Metode ale obiectului **window**, II

n **open(URL)**

- Deschide o fereastră nouă în care se încarcă documentul specificat prin URL.

n **close()**

- Închide fereastra (nu un frame).

Metode ale obiectului **window**, I

n **alert(string)**

- Afișează un casută de dialog ce conține stringul specificat și un buton cu eticheta OK.

n **confirm(string)**

- Afișează o fereastră de confirmare ce conține stringul specificat împreună cu butoanele Cancel și OK. Returnează true dacă este apăsător OK, false dacă este apăsător Cancel.

n **prompt(string)**

- Afișează o fereastră de confirmare ce conține stringul specificat, un input de tip text împreună cu butoanele Cancel și OK. Returnează stringul introdus de user dacă este apăsător OK, null dacă este apăsător Cancel.

Exemplu

```
1. <html>
2.   <head>
3.     <title>Proprietatile obiectului window</title>
4.   </head>
5.   <body>
6.     <script type="text/javascript">
7.       var url = prompt('Introduceti adresa unei pagini web:');
8.       if (url != null){
9.         var myWindow = open(url);
10.        if (confirm('Doriti inchiderea paginii ' + url + '?')){
11.          myWindow.close();
12.        }
13.      } else {
14.        alert('Nu ati introdus nimic');
15.      }
16.    </script>
17.  </body>
18. </html>
```

Proprietati ale obiectului **document**, I

- n Toate aceste proprietati trebuie prefixate de **document**.
- n **anchors[]**
 - .. Un tablou de obiecte de tip **Anchor** (ce contin tag-urile)
- n **applets[]**
 - .. Un tablou de obiecte de tip **Applet** reprezentand apletii din document
 - n Proprietatile acestor obiecte sunt campurile declarate public in aplet
 - n Metodele sunt metodele publice din aplet

Proprietati ale obiectului **document**, III

- n **bgColor**
 - .. Culoarea de background a paginii web
 - n Poate fi schimbata in orice moment
- n **title**
 - .. Proprietate read-only ce retine titlul paginii web
- n **URL**
 - .. Proprietate read-only ce retine URL-ul documentului curent

Proprietati ale obiectului **document**, II

- n **forms[]**
 - .. Un tablou de obiecte de tip **Form** ce reprezinta formele din document
 - n Daca documentul contine numai o forma, atunci acesta este **forms[0]**
- n **images[]**
 - .. Un tablou de obiecte de tip **Image** reprezentand imaginile din document
 - n Daca dorim sa schimbam o anumita imagine, atribuim noul URL proprietatii **src**
- n **links[]**
 - .. Un tablou de obiecte de tip **Link** objects reprezentand link-urile din pagina
 - n Proprietatea ce retine adresa paginii referite este **href**

Metode ale obiectului **document**

- n **getElementByID(string)**
 - .. Intoarce o referinta la primul element cu id-ul specificat
- n **getElementsByName(string)**
 - .. Intoarce un tablou ce contine toate obiectele cu un nume specificat
- n **getElementsByTagName(string)**
 - .. Intoarce un tablou ce contine toate obiectele cu tag-ul specificat

Proprietatile obiectului **Form**

n `elements[ ]`

- Tablou ce contine toate input-urile din forma

n `action`

- URL-ul specificat in atributul `action` catre care se vor trimite datele din forma

n `method`

- Metoda prin care se trimit datele din forma (POST,GET)

Evenimente

- n** Majoritatea elementelor dintr-o pagina HTML “raspund” la actiuni ale utilizatorului (apasarea tastelor, efectuarea de click-uri) prin generarea de evenimente
 - Fiecare tip de element produce evenimente specifice.
 - Ne vom concentra pe evenimentele recunoscute de elementele de tip **form** si cele comune
- n** Pentru a gestiona evenimentele trebuie sa scriem niste functii speciale numite *handlers* care le asociem diverselor tipuri de evenimente
 - La aparitia evenimentului respectiv, este apelat automat handlerul corespunzator

Metode obiectului **Form**

n `submit()`

- Trimite(submite) datele catre server

n `reset()`

- Reseteaza continutul formei

Exemplu

```
n <form method="post" action="">
    <input type="button"
        name= "testButton"
        value="Apasa-ma!"
        onclick="alert('Ai apasat butonul');">
</form>
```

- n** Apasarea butonului va declansa executia instructiunii JavaScript

alert('Ai apasat butonul');

care va afisa pe ecran o fereastra de dialog cu mesajul specificat.

Evenimente si handlere de evenimente, I

Event	Applies to	Occurs when	Handler
Load	Document body	User loads the page in a browser	onLoad
Unload	Document body	User exits the page	onUnload
Error	Images, window	Error on loading an image or a window	onError
Abort	Images	User aborts the loading of an image	onAbort

Evenimente si handlere de evenimente, III

Event	Applies to	Occurs when	Handler
MouseDown	Documents, buttons, links	User depresses a mouse button	onMouseDown
MouseUp	Documents, buttons, links	User releases a mouse button	onMouseUp
Click	Buttons, radio buttons, checkboxes, submit buttons, reset buttons, links	User clicks a form element or link	onClick

Evenimente si handlere de evenimente, II

Event	Applies to	Occurs when	Handler
KeyDown	Documents, images, links, text areas	User depresses a key	onKeyDown
KeyUp	Documents, images, links, text areas	User releases a key	onKeyUp
KeyPress	Documents, images, links, text areas	User presses or holds down a key	onKeyPress
Change	Text fields, text areas, select lists	User changes the value of an element	onChange

Evenimente si handlere de evenimente, IV

Event	Applies to	Occurs when	Handler
MouseOver	Links	User moves cursor over a link	onMouseOver
MouseOut	Areas, links	User moves cursor out of an image map or link	onMouseOut
Select	Text fields, text areas	User selects form element's input field	onSelect



Evenimente si handlere de evenimente, V

Event	Applies to	Occurs when	Handler
Move	Windows	User or script moves a window	onMove
Resize	Windows	User or script resizes a window	onResize
DragDrop	Windows	User drops an object onto the browser window	onDragDrop



Evenimente si handlere de evenimente, VI

Event	Applies to	Occurs when	Handler
Focus	Windows and all form elements	User gives element input focus	onFocus
Blur	Windows and all form elements	User moves focus to some other element	onBlur
Reset	Forms	User clicks a Reset button	onReset