TEHNOLOGII WEB - Îndrumar de laborator

LUCRAREA NR. 8 :   
Limbajul JAVASCRIPT şi modelul DOM – Crearea paginilor dinamice utilizând limbajul JavaScript şi DOM

1. **Tematica lucrării**
2. Modelul DOM (Document Object Model)
3. Formatarea elementelor HTML utilizând limbajul JavaScript şi modelul DOM
4. **Teoria lucrării de laborator**
5. Modelul DOM (Document Object Model)

Conform World Wide Web Consortium (W3C), Document Object Model reprezintă un model orientat obiect, cross-platfrom şi independent de limbaj, o interfaţă gen API care permite scripturilor să realizeze în mod dinamic accesarea şi actualizarea conţinutului, structurii şi stilului documentelor.

Astfel, rezultatul unui document procesat cu modelul DOM poate fi incorporat înapoi în pagina curentă.

Utilizând modelul DOM, JavaScript poate accesa toate elementele dintr-un document HTML.

Atunci când o pagină web este încărcată, browser-ul crează o structură DOM a paginii respective. Structura DOM este construită ca un arbore de obiecte, ca în figura 8.1:

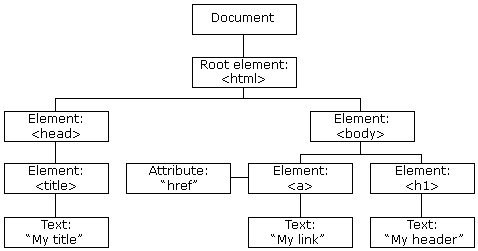


Figura 8.1. Arborele DOM al unui document HTML

Având acest model obiect programabil, JavaScript are tot ce-i trebuie pentru a crea pagini HTML dinamice:

* JavaScript poate modifica toate elementele HTML dintr-o pagină
* JavaScript poate modifica toate atributele HTML dintr-o pagină
* JavaScript poate modifica toate stilurile CSS dintr-o pagină
* JavaScript poate reacţiona la toate evenimentele din pagina web.

Accesarea elementelor HTML

Se poate realiza în trei moduri:

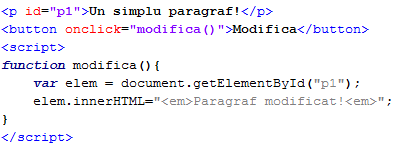
* după ID-ul unui element
* după numele etichetei unui element
* după numele clasei unui element (fără efect în IE 5, 6, 7 şi 8).

1. Formatarea elementelor HTML utilizând limbajul JavaScript şi modelul DOM
2. Modificarea conţinutului unui element HTML

Sintaxa:



Exemplu:



Rezultatul execuţiei în browser arată astfel:

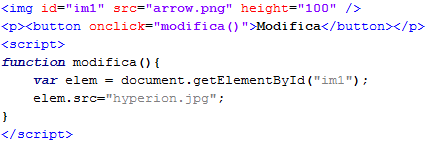
|  |  |
| --- | --- |
| *Prima afişare* | *A doua afişare* |
|  |  |

1. Modificarea atributelor unui element HTML

Sintaxa:



Exemplu, în care se modifică sursa unei imagini:



Iar execuţia codului este următoarea:

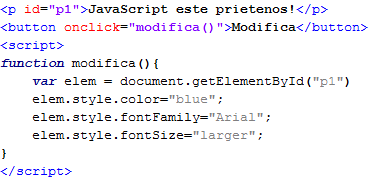
|  |  |
| --- | --- |
| *Prima afişare* | *A doua afişare* |
|  |  |

1. Modificarea stilului CSS al unui element HTML

Sintaxa:



În exemplul următor se vor schimba proprietăţile CSS ale unui paragraf pentru culoarea, familia şi dimensiunea fontului:



Iar rezultatul afişat în browser-ul web este:

|  |  |
| --- | --- |
| *Prima afişare* | *A doua afişare* |
|  |  |

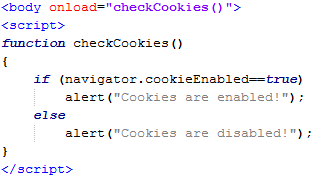
1. Evenimente HTML DOM - JavaScript

HTML DOM permite limbajului JavaScript să reacţioneze la evenimentele HTML. Astfel, la declanşarea unui eveniment, cum ar fi un clic de mouse, se poate executa un cod JavaScript.

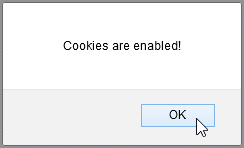
Example de evenimente HTML:

* Când un utilizator apasă un buton al mouse-ului
* Când s-a încărcat pagina web
* Când s-a încărcat o imagine
* Când mouse-ul glisează deasupra unui element
* Când se modifică un câmp de introducere date
* Când este trimis un formular spre procesare
* Când utilizatorul apasă o tastă de la tastatură

Spre exemplificare prezentăm un script prin care se verifică dacă cookie-urile navigatorului sunt activate sau nu, la încărcarea paginii, prin tratarea evenimentului *onload*:



Afişarea constă într-un mesaj alertă, ca cel de mai jos:



1. **Modul de lucru**
2. Să se creeze un document HTML cu elementele: paragraf, imagine, tabel, listă.
3. Să se modifice utilizând limbajul JavaScript şi modelul DOM atributele elementelor din pagina creată la punctul 1.
4. Să se creeze un script prin care, atunci când se glisează mouse-ul deasupra unei celule din tabel, culoarea de fundal a acesteia să se modifice automat.
5. Să se creeze un script care să trateze cât mai multe evenimente HTML.

TEHNOLOGII WEB - Îndrumar de laborator

LUCRAREA NR. 8 :   
Limbajul JAVASCRIPT şi modelul DOM – Crearea paginilor dinamice utilizând limbajul JavaScript şi DOM

1. **Tematica lucrării**
2. Modelul DOM (Document Object Model)
3. Formatarea elementelor HTML utilizând limbajul JavaScript şi modelul DOM
4. **Teoria lucrării de laborator**
5. Modelul DOM (Document Object Model)

Conform World Wide Web Consortium (W3C), Document Object Model reprezintă un model orientat obiect, cross-platfrom şi independent de limbaj, o interfaţă gen API care permite scripturilor să realizeze în mod dinamic accesarea şi actualizarea conţinutului, structurii şi stilului documentelor.

Astfel, rezultatul unui document procesat cu modelul DOM poate fi incorporat înapoi în pagina curentă.

Utilizând modelul DOM, JavaScript poate accesa toate elementele dintr-un document HTML.

Atunci când o pagină web este încărcată, browser-ul crează o structură DOM a paginii respective. Structura DOM este construită ca un arbore de obiecte, ca în figura 8.1:

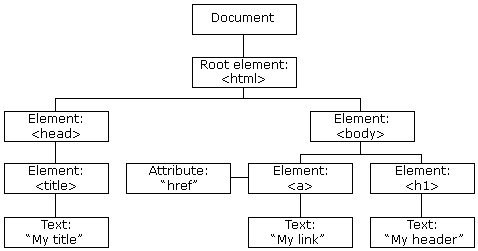


Figura 8.1. Arborele DOM al unui document HTML

Având acest model obiect programabil, JavaScript are tot ce-i trebuie pentru a crea pagini HTML dinamice:

* JavaScript poate modifica toate elementele HTML dintr-o pagină
* JavaScript poate modifica toate atributele HTML dintr-o pagină
* JavaScript poate modifica toate stilurile CSS dintr-o pagină
* JavaScript poate reacţiona la toate evenimentele din pagina web.

Accesarea elementelor HTML

Se poate realiza în trei moduri:

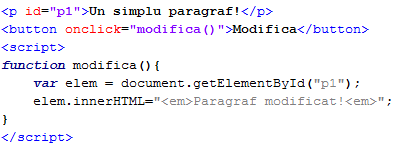
* după ID-ul unui element
* după numele etichetei unui element
* după numele clasei unui element (fără efect în IE 5, 6, 7 şi 8).

1. Formatarea elementelor HTML utilizând limbajul JavaScript şi modelul DOM
2. Modificarea conţinutului unui element HTML

Sintaxa:



Exemplu:



Rezultatul execuţiei în browser arată astfel:

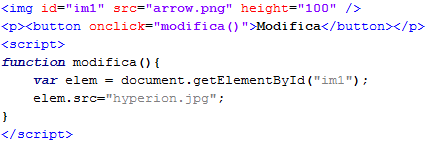
|  |  |
| --- | --- |
| *Prima afişare* | *A doua afişare* |
|  |  |

1. Modificarea atributelor unui element HTML

Sintaxa:



Exemplu, în care se modifică sursa unei imagini:



Iar execuţia codului este următoarea:

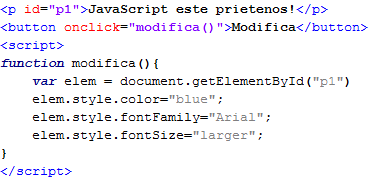
|  |  |
| --- | --- |
| *Prima afişare* | *A doua afişare* |
|  |  |

1. Modificarea stilului CSS al unui element HTML

Sintaxa:



În exemplul următor se vor schimba proprietăţile CSS ale unui paragraf pentru culoarea, familia şi dimensiunea fontului:



Iar rezultatul afişat în browser-ul web este:

|  |  |
| --- | --- |
| *Prima afişare* | *A doua afişare* |
|  |  |

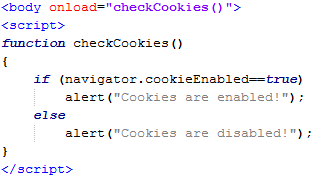
1. Evenimente HTML DOM - JavaScript

HTML DOM permite limbajului JavaScript să reacţioneze la evenimentele HTML. Astfel, la declanşarea unui eveniment, cum ar fi un clic de mouse, se poate executa un cod JavaScript.

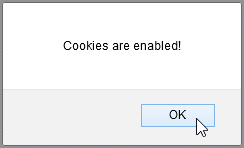
Example de evenimente HTML:

* Când un utilizator apasă un buton al mouse-ului
* Când s-a încărcat pagina web
* Când s-a încărcat o imagine
* Când mouse-ul glisează deasupra unui element
* Când se modifică un câmp de introducere date
* Când este trimis un formular spre procesare
* Când utilizatorul apasă o tastă de la tastatură

Spre exemplificare prezentăm un script prin care se verifică dacă cookie-urile navigatorului sunt activate sau nu, la încărcarea paginii, prin tratarea evenimentului *onload*:



Afişarea constă într-un mesaj alertă, ca cel de mai jos:



1. **Modul de lucru**
2. Să se creeze un document HTML cu elementele: paragraf, imagine, tabel, listă.
3. Să se modifice utilizând limbajul JavaScript şi modelul DOM atributele elementelor din pagina creată la punctul 1.
4. Să se creeze un script prin care, atunci când se glisează mouse-ul deasupra unei celule din tabel, culoarea de fundal a acesteia să se modifice automat.
5. Să se creeze un script care să trateze cât mai multe evenimente HTML.