

## 2. Website Building

Adrian Adiaconitei

***LINKAcademy***

# Objective

- ✓ DOM și BOM
  - ✓ Gestionarea obiectelor HTML
  - ✓ Utilizarea lui JS pentru stilizarea și gestionarea atributelor
  - ✓ Evenimentele JS și DOM
- ✓ Bootstrap Js

# Manipularea lui BOM

## Browser Object Model.

- ✓ Un obiect Window este un obiect global al browserului web, ceea ce înseamnă că proprietățile și metodele sale pot fi accesate fără a specifica o variabilă window.
- ✓ În interiorul obiectului Window se află obiectul Screen, care oferă acces la informațiile de bază despre ecranul dispozitivului utilizatorului.
- ✓ Obiectul Location, care este o parte integrantă a obiectului global Window, este utilizat pentru obținerea informațiilor despre adresa URL în care se află documentul HTML.

# Manipularea lui BOM

## Browser Object Model.

✓ Obiectul Window are, de asemenea, trei metode speciale care vă permit să creați ferestre modale cu scopuri diferite, acestea sunt: **alert()**, **confirm()** și **prompt()**.

✓ **window.history**

**Ap1.html**

✓ **window.location**

✓ **window.screen**

**Ap2.html**

✓ **window.devicePixelRatio**

[https://www.w3schools.com/js/js\\_ex\\_browser.asp](https://www.w3schools.com/js/js_ex_browser.asp)

# Manipularea lui DOM

## Ce este DOM?

Document Object Model.

Document XML(Extensible Markup Language) transformat în model obiectual.

Se activează după parsarea paginii.

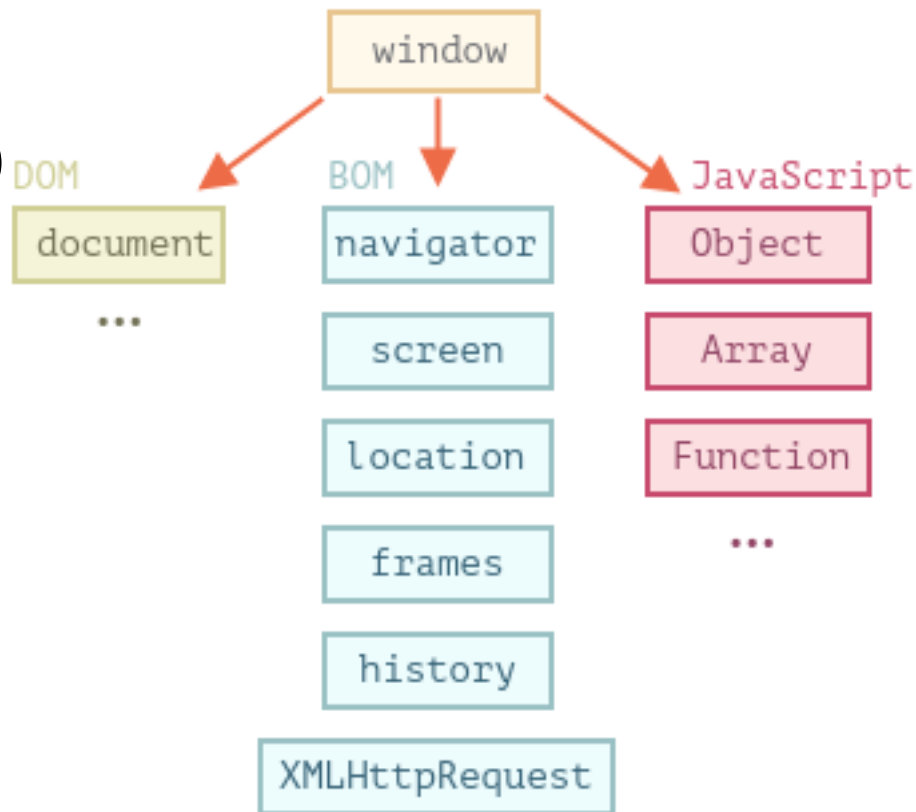
În JavaScript, întreaga structură este conținută în obiectul **document**.

[https://www.w3schools.com/jsref/dom\\_obj\\_document.asp](https://www.w3schools.com/jsref/dom_obj_document.asp)

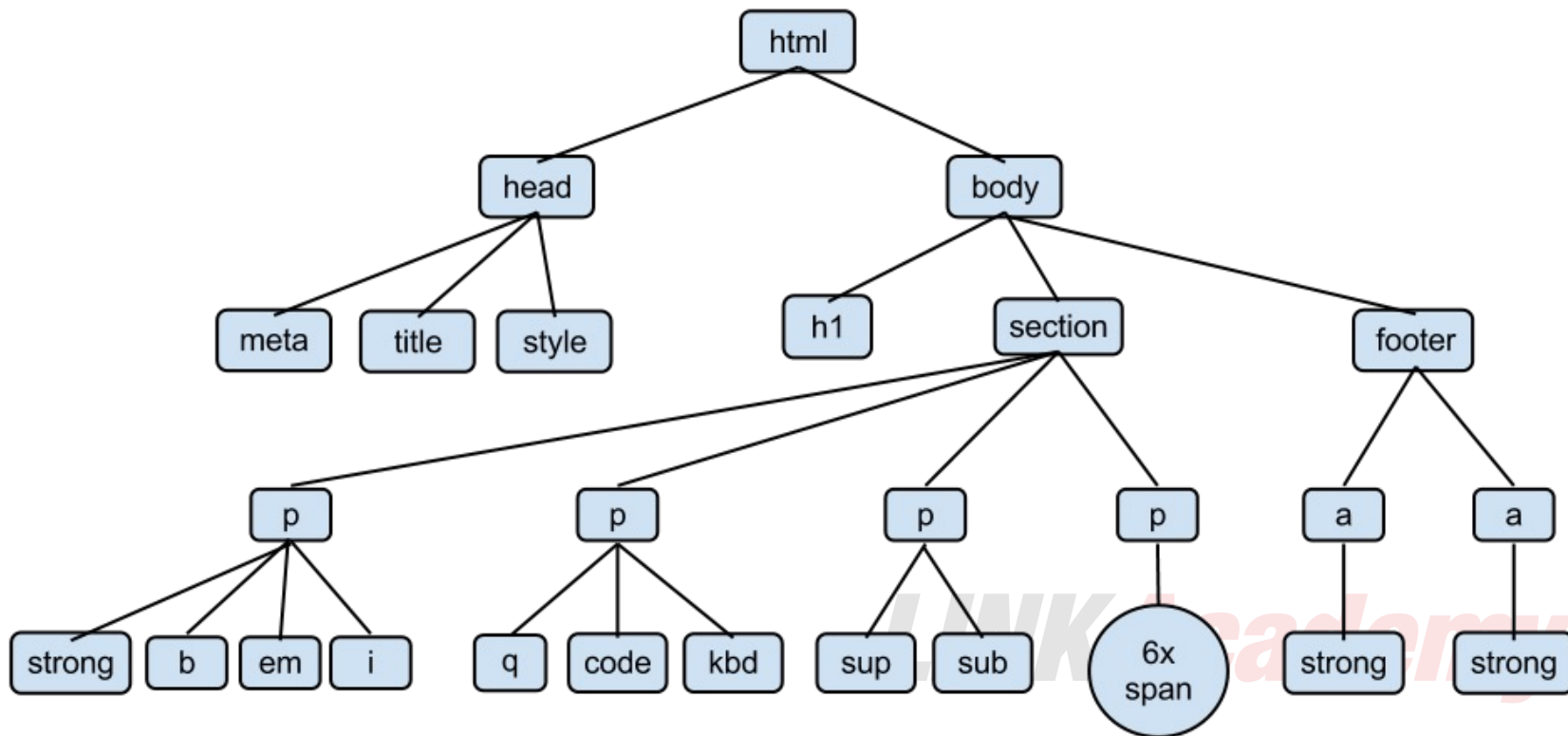
Conține noduri, subnoduri (nodurile părinte/copil) și attribute.

# Structura DOM-ului

- ✓ Browser Object Model (BOM)
- ✓ Document Object Model (DOM)



# Structura DOM-ului



# Metodele DOM

✓ Metodele DOM sunt comenzi (metode) prin care putem manipula diferiți membri ai obiectului DOM.

- *createElement('element\_name')*
- *createTextNode('text content')*
- *appendChild(element)*
- *setAttribute()*



# Navigarea în cadrul documentului

## ✓ Căutarea în funcție de id

**document.getElementById("myid");**

- ✓ Metoda getElementById se folosește atunci când vrem să găsim un element al cărui id îl știm.
- ✓ Id-ul pe care îl căutăm trebuie să fie unic pentru pagina respectivă.
- ✓ Oricare dintre elemente se poate identifica prin metoda **getElementById**
- ✓ Când elementul este identificat și plasat într-o variabilă acesta se poate manipula prin variabila respectivă

# Navigarea în cadrul documentului

- ✓ Căutarea în funcție de numele tagului

`document.getElementsByTagName("div");`

- ✓ Această căutare are sens doar atunci când nu avem alți identificatori (de exemplu id) sau când vrem să preluăm mai multe elemente simultan.
- ✓ Spre deosebire de căutarea de id, această metoda returnează un șir de elemente (chiar dacă găsește unul singur).

# Navigarea în cadrul documentului

✓ Navigarea după valoarea atributului **name**

**document.getElementsByName("author");**

✓ Navigarea după valoarea atributului **class**

**document.getElementsByClassName("myclass");**

Ap3.html

Ap4.html

# Navigarea în cadrul documentului

`document.getElementById("myid");`

`document.getElementsByTagName("div");`

`document.getElementsByName`

`document.getElementsByClassName`

Returnează HTMLCollection live [NodeList](#).

# Navigarea în cadrul documentului

**document.querySelector()** returnează primul din document sau null

**document.querySelectorAll()** returnează o listă de elemente ale documentului

Returnează static [NodeList](#).

Ap5.html

Ap6.html

# Evenimente:

- ✓ Evenimentele sunt acțiuni declanșate de utilizatori și pot fi detectate de JavaScript.
- ✓ Putem configura execuția de acțiuni la detectarea de evenimente
- ✓ Exemple de evenimente: S-a efectuat click pe un buton; S-a terminat de încărcat pagina

# Evenimente:

- ✓ Cel mai des, evenimentele sunt declanșate de către un formular
- ✓ Evenimentele JavaScript nu trebuie scrise neaparat între etichetele `<script>` și `</script>`, deoarece fac parte din HTML.
- ✓ În HTML NU contează dacă se scrie cu litere mari (ONCLICK), cu litere mici (onclick) , în javascript DA
- ✓ Funcțiile se scriu de obicei înaintea apelarii lor de către evenimente, dar nu vor fi executate înaintea tratării evenimentelor respective.
- ✓ **Evenimentele** sunt: **Click** (click de mouse), **MouseOver** (cursorul mouse-ului trece peste un anumit element);
- ✓ **Tratarea evenimentelor** reprezintă acțiunea ce urmează la acționarea evenimentului, cum ar fi: `onClick`, `onMouseOver`, `onKeyDown`.

# Evenimente:

- ✓ [addEventListener\(\)](#) - Ne permite să interacționăm cu pagina.
- ✓ [removeEventListener\(\)](#) – oprește interacțiunea cu pagina
- ✓ **Tratarea evenimentelor de către un formular**

**onBlur** acțiune ce va fi declanșată când se pierde focalizarea de pe un element.

**onChange** acțiune ce va fi declanșată la schimbarea unui element.

**onFocus** acțiune ce va fi declanșată de focalizarea asupra unui element.

**onReset** acțiune ce va fi declanșată când va fi acționat butonul Reset al unui formular.

**onSelect** acțiune ce va fi declanșată atunci când se selectează un element.

**onSubmit** acțiune ce va fi declanșată când va fi acționat butonul Submit al unui formular.



# Evenimente:

## ✓ Tratarea evenimentelor de catre mouse

**onClick** acțiune ce va fi declanșată când se da click stânga pe mouse.

**onDblClick** acțiune ce va fi declanșată când se dă dublu click stânga pe mouse.

**onMouseMove** acțiune ce va fi declanșată când se mișca cursorul mouse-ului.

**onMouseOut** acțiune ce va fi declanșată când cursorul mouse-ului se deplasează în afara elementului sau a suprafeței delimitate.

**onMouseOver** acțiune ce va fi declanșată când cursorul mouse-ului se deplasează peste un element sau peste suprafața delimitată.

**onMouseUp** acțiune ce va fi declanșată când butonul mouse-ului este eliberat.

**onMouseDown** acțiune ce va fi declanșată când un buton al mouse-ului este apăsat.

# Evenimente:

## ✓ Tratarea evenimentelor de către sistem

**onLoad** va acționa în urma încărcării unui document

**onbeforeunload** va acționa în urma închiderii tab-ului sau browserului

## ✓ Tratarea evenimentelor de către tastatura

**onKeyDown** acțiune ce va fi declanșată când se apasă o tastă.

**onKeyPress** acțiune ce va fi declanșată când se apasă și se eliberează o tastă.

**onKeyUp** acțiune ce va fi declanșată când se eliberează o tastă.

# Formulare:

Toate formularele pot fi accesate cu `document.forms` prin indexul sau numele acestuia:

- `document.forms[0];`
- `document.forms.myForm`
- `document.forms.form1.action="my.php"`
- `document.forms.myLa.value = "Text modificat"`

```
<form id="form1" name="myform" action="test.php" method="post">
```

```
  <input type="text" name="myLa">
```

```
</form>
```

# Formulare:

- Preluarea valorii alese

*this.value*

- Preluarea indexului ales

*this.selectedIndex*

- Preluarea denumirii alese

*this.options[this.options.selectedIndex].text);*

- Postarea membrului ales

*sel.options[2].selected=true;*

*sel.options[2].selected="selected";*

**ap7.html:** Atenție când completați formularele pe anumite site-uri!!!

# Validarea Formulelor:

[https://www.w3schools.com/js/js\\_validation.asp](https://www.w3schools.com/js/js_validation.asp)

```
<script type="text/javascript">
  function limitarelungime(obj, length){
    var lungime=length
    if (obj.value.length>lungime)
      obj.value=obj.value.substring(0, lungime)
  }
</script>
<form>
  <textarea onkeyup="return limitarelungime(this, 20)"></textarea>
</form>
```

# Cookies:

Un cookie este o variabilă păstrată în calculatorul vizitatorului. Cu JavaScript, puteți crea și extrage cookies.

## Exemple de cookies:

- 1. Numele utilizatorului** – Prima dată când utilizatorul va vizitează pagina trebuie să-și completeze numele. Numele este stocat într-un cookie. Numele este recuperat dintr-un cookie.
- 2. Parola** – Parola este memorată într-un cookie. Data viitoare când vizitatorul ajunge în pagina, parola poate fi recuperata dintr-un cookie.
- 3. Data calendaristică** – Prima dată când utilizatorul va viziteaza pagina, data curenta este memorată într-un cookie

# Cookies:

Crearea și memorarea unui cookie

-memorează numele vizitatorului, apoi vom folosi numele memorat în variabilă cookie pentru a afișa un mesaj de bun venit. Prima dată vom crea o funcție care memorează numele vizitatorului într-o variabilă cookie:

```
function setCookie(c_name,value,expiredays) {  
var exdate=new Date();  
exdate.setDate(exdate.getDate()+expiredays);  
document.cookie=c_name+ "=" +escape(value)+ ((expiredays==null) ? "" :  
";expires="+exdate.toGMTString());  
}
```

Parametrii funcției reprezintă numele și valoarea cookie-ului și numărul de zile până când acesta expiră. Datele nume, cookie-ului și data expirării sunt memorate într-un obiect **document.cookie**.

# Cookies:

```
function getCookie(c_name) {  
    if (document.cookie.length>0) {  
        c_start=document.cookie.indexOf(c_name + "=");  
        if (c_start!=-1) {  
            c_start=c_start + c_name.length+1;  
            c_end=document.cookie.indexOf(";",c_start);  
            if (c_end==-1) c_end=document.cookie.length;  
            return  
                unescape(document.cookie.substring(c_start,c_end));  
        }  
    }  
    return "";  
}  
}  
// funcția verifică dacă un cookie este setat
```



# Cookies:

Funcția care afișează un mesaj de bun venit dacă cookie-ul este setat și o casetă prompt care cere numele vizitatorului, în caz contrar:

```
function checkCookie() {  
    username=getCookie('username');  
    if (username!=null && username!=""){alert('Welcome again '+username+'!');  
    }  
    else {  
        username=prompt('Please enter your name:', '');  
        if (username!=null && username!="") {  
            setCookie('username',username,365);  
        }  
    }  
}
```

# Bootstrap

## Aplicația: Shop:

- ✓ Optimizare carousel – imagini încărcate în funcție de dispozitiv
- ✓ Adăugare funcționalitate: Go to top
- ✓ Pe baza unui cookie să afișăm un mesaj de salut: Bine ai veni sau Bine ai revenit
- ✓ Dark mode
- ✓ Adăugare funcționalitate: Add to cart
- ✓ Google captcha

<https://developers.google.com/recaptcha/docs/v3>

# Resurse

[https://www.w3schools.com/js/exercise.asp?filename=exercise\\_events1](https://www.w3schools.com/js/exercise.asp?filename=exercise_events1)

[https://www.w3schools.com/js/exercise.asp?filename=exercise\\_events2](https://www.w3schools.com/js/exercise.asp?filename=exercise_events2)

[https://www.w3schools.com/js/exercise.asp?filename=exercise\\_events3](https://www.w3schools.com/js/exercise.asp?filename=exercise_events3)

[https://www.w3schools.com/js/exercise.asp?filename=exercise\\_arraysmet1](https://www.w3schools.com/js/exercise.asp?filename=exercise_arraysmet1)

[https://www.w3schools.com/js/exercise.asp?filename=exercise\\_arraysmet2](https://www.w3schools.com/js/exercise.asp?filename=exercise_arraysmet2)

[https://www.w3schools.com/js/exercise.asp?filename=exercise\\_arraysmet4](https://www.w3schools.com/js/exercise.asp?filename=exercise_arraysmet4)

<https://www.w3resource.com/javascript-exercises/javascript-dom-exercises.php>

<https://developer.mozilla.org/en-US/docs/Web/Events>