



Document Object Model

Events Handlers

Sunt lucruri prin care JavaScript observa cand se intampla ceva anume in browser (redimensionarea ferestrei, click sau rollover pe un element etc.)

Pe local - `Test`

Pe JS Extern - `object.addEventListener("click", exemplu());`

Direct pe un obiect in JS - `object.onclick = function(){script};`

onload - Cand ceva a fost incarcat in pagina.

onclick - Cand dam click cu mouse-ul pe un element din html.

onmouseover - Cand ducem mouse-ul pe un element din html.

onmouseout - Cand luam mouse-ul de pe un element din html.

onkeydown - Cand incepem sa scriem ceva

onload Event

Mai jos avem un exemplu prin care ii spun sa imi afiseze un mesaj pe consola de test atunci cand intreg website-ul, imagini, html, css, js s-a incarcat.

```
window.onload = function(){  
  alert('Pagina s-a incarcat');  
}
```

The DOM

Vine de la Document Object Model. Defineste structura logica a unui document si modul in care este accesat si manipulat. Folosind DOM putem crea documente noi, naviga in structura lor, modifica sau sterge elemente de continut sau de stilizare. Cele mai folosite sunt:

1. `getElementById()`
2. `getElementsByTagName()`
3. `innerHTML()`
4. `setAttribute()`

getElementById()

Prin aceasta metoda putem accesa un element care are un id predefinit. Este folosit in marea majoritate a cazurilor cand vrem sa controlam sau sa luam o informatie dintr-un element al unui html.

```
document.getElementById('nume id')
```

Practica

- Sa cream un buton in HTML caruia sa ii atribuim id-ul 'link'.
- In JS vreau ca aceasta functie sa ruleze cand site-ul nostru s-a incarcat.
- In interiorul functiei de startup de mai sus adaugam `document.getElementById('nume id')` in care specificam id-ul nostru, il legam de un event `.onclick` apoi toata chestia o definim ca o functie anonima in care plasam o alerta cu mesajul Treaba buna!.

Codul

Sa cream in HTML un <button> cu id-ul link si textul Play.

Ex.<button id='link'>Play</button>

In JS sa cream functia:

```
window.onload = function() {  
    document.getElementById('link').onclick = function() {  
        alert('Treaba buna!');  
    }  
}
```


Practica

- Sa cream in html 2 input-uri cu type='number' cu 2 id-uri definite number1 si number2.
- Dupa ele vom avea un tag de button cu textul 'Care este mai mare?'.
- In JS specificam urmatoarele: dupa ce s-a incarcat documentul, definim o functie anonima in interior cu numele check
- In cadrul functiei legam cele 2 elemente html de cate o variabila
- Apoi printr-o declaratie de tip if/else va face comparatia dintre cele 2 numere si afisa mesajul pentru cele 2 situatii, "Primul numar este mai mare" sau "Al doilea numar este mai mare"

Practica Varianta 1

```
document.getElementById('exemplu').onclick = function() {  
    var number1 = document.getElementById('number1').value;  
    var number2 = document.getElementById('number2').value;  
    if(number1 > number2){  
        console.log('Primul numar este mai mare')  
    } else {  
        console.log('Al doilea numar este mai mare')  
    }  
}
```

Practica Varianta 2

```
var check = function() {  
    var number1 = document.getElementById('number1').value;  
    var number2 = document.getElementById('number2').value;  
    if(number1 > number2){  
        console.log('Primul numar este mai mare')  
    } else {  
        console.log('Al doilea numar este mai mare')  
    }  
}  
document.getElementById('exemplu').addEventListener("click", check());
```

Practica Varianta 3

In HTML: <button id='exemplu' onclick="check()">test</button>

```
var check = function() {  
    var number1 = document.getElementById('number1').value;  
    var number2 = document.getElementById('number2').value;  
    if(number1 > number2){  
        console.log('Primul numar este mai mare')  
    } else {  
        console.log('Al doilea numar este mai mare')  
    }  
}
```

getElementsByTagName()

Prin aceasta metoda putem accesa toate elementele dintr-un document cu un tag specific .

```
var test = document.getElementsByTagName('h1');
```

La ce ar putea fi folositoare? Daca vrem sa facem o schimbare generala sau sa numaram cate elemente de acel tip exista in document.

Practica - Cate elemente de acelasi gen avem?

Sa adaugam in html un h2 si o lista neordonata cu 3 elemente.

In JS sa cream functia de startup apoi in interior definim variabila `elements` cu valoarea: `document.getElementsByTagName('tag-ul')` in care punem tag-ul unui element de lista.

Printam pe consola in cadrul acelei functii denumirea variabilei cu `.length` astfel incat sa ne numere cate elemente li avem in document.

Codul

```
window.onload = function() {  
  var elements = document.getElementsByTagName('li');  
  console.log(elements.length);  
}
```

innerHTML()

Prin aceasta metoda putem modifica continutul unui element din HTML. Putem stabili specific unde va aparea continutul introdus de noi.

```
document.getElementById('id').innerHTML = continut nou aici;
```

Practica

- In HTML vreau sa adaugam un H3 cu id header, un input type text cu id-ul inputfield si un buton cu id-ul headerbtn care va avea textul Set Header.
- In JS definim functia de startup "cand se incarca site-ul" sa ruleze acel script.
- In cadrul ei adaugam un event handler "cand dam click pe acel buton" in care vom defini o functie anonima cu urmatoarele lucruri:
 - Definim variabile si le legam de elementele h3, input si button;
 - Setam ca ceea ce este introdus ca valoare in input sa devina continut html pentru header:
`header.innerHTML = input.value;`

Codul

```
window.onload = function(){  
    document.getElementById('headerbtn').onclick = function(){  
        var input = document.getElementById('inputfield');  
        var header = document.getElementById('header');  
        header.innerHTML = input.value;  
    }  
}
```

Just a header

Input

<input id="inputfield">

<button id="headerbtn">

setAttribute()

Prin aceasta metoda putem seta atributul unui element HTML selectat. Putem stabili specific unde va aparea continutul introdus de noi.

`<tag attribute="value">`

Orice lucru facut corect in HTML in stilul de mai sus este accesibil cu `setAttribute`. Daca are deja un atribut setat il poate schimba.

Vom folosi un lucru nou in urmatoarea practica: `this`. Cand folosim o functie anonima, 'this' face referire la obiectul atasat din acea functie.

In exemplul urmator vom folosi event-ul `onclick` si o sa ne referim la obiectul pe care am dat click.

Practica

In HTML vreau sa adaugam un H3 cu id header2.

In JS sa cream functia:

```
window.onload = function(){  
    document.getElementById('header2').onclick = function(){  
        this.setAttribute('style','background-color:red');  
    }  
}
```

Sa testam HTML-ul in browser si sa dam click pe header-ul nostru.