Új Magyarország FEJLESZTÉSI TERV

NONPROFIT ÉRDEKVÉDELMI SZERVEZETEK FEJLESZTÉSE

(PL.3346)

Java-script nyelv

programozás alapjai

Haramia László

A projekt az Európai Unió támogatásával,
az Európai Szociális Alap társfinanszírozásával valósul meg.

CIMET
- a civil világ fűszere

TÁMOP 5.5.3-09/1-2009-0013

JavaScript szerepe

- Netscape fejlesztette ki – LiveScript
- Interaktív weboldalak dinamikus kezelhetősége érdekében fejlesztették ki
- kommunikáció a felhasználóval
- legnépszerűbb szkript-nyelv
- C-típusú nyelv, nem módosított Java
- JavaScript futtatásához csak egy böngésző kell
- objektum alapú nyelv

Megj: a böngészők általában ismerik a JavaScriptet és képesek azt futtatni

JavaScript nyújtotta lehetőségek

- kész kódrészletek állnak rendelkezésre
- dinamikusan felépíthetők a weboldal egyes részei
- HTML/XHTML objektumokhoz kapcsolható eseményekre reagálni tudunk
- HTML/XHTML DOM elemei olvashatók/írhatók
- űrlap mezőit elküldés előtt ellenőrizhetjük

- szabályok

  pontosvessző az utasítások végén - nem kötelező

  kommentjelek (//, illetve /*.....*/)

  Megkülönbözteti a kis/nagybetűket

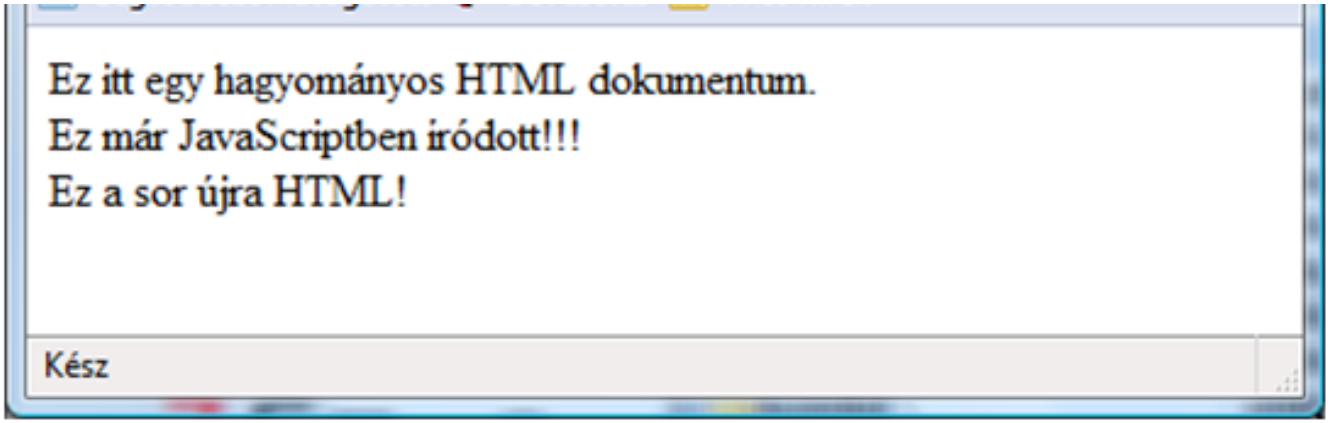
JavaScript a weboldalon

- Html oldalra a <script> taggal
- <head> elemen belül
  - függvények, változók deklarációja
  - látható kimenete itt nem lehet
  - az itt elhelyezkedő kódot kell meghívni
- <body> elemen belül
  - végrehajtás a feldolgozás során
  - függvényeket itt is meg kell hívni
- JavaScript kód külső fájlban

```
<script type="text/javascript" src=„js_minta.js">
```

JavaScript beágyazása HTML dokumentumba

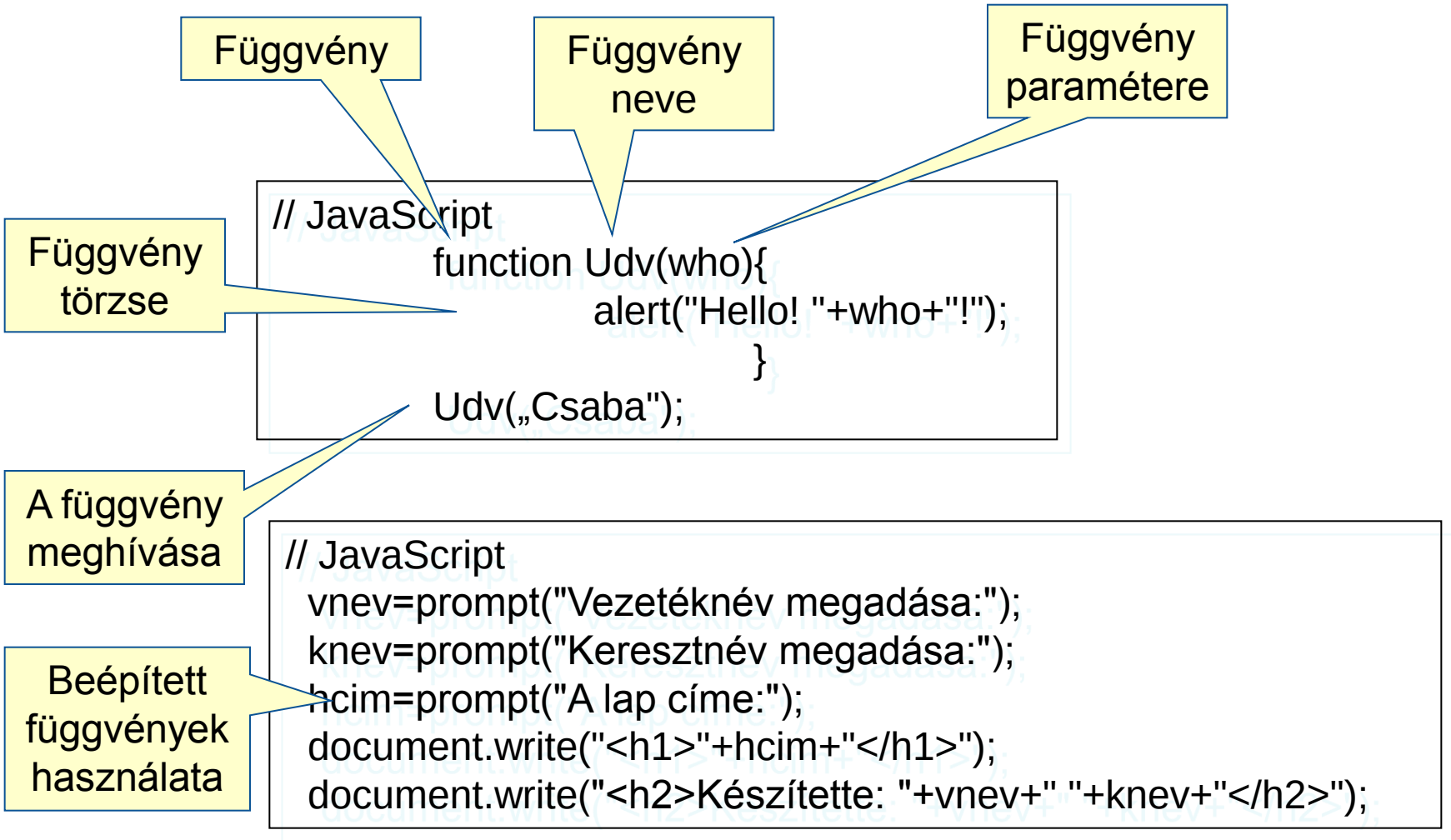
```
<html>
   <body>
   Ez itt egy hagyományos HTML dokumentum.<br>
   <script language="JavaScript">
   document.write("Ez már JavaScriptben íródott!<br>")
   </script>
   Ez a sor újra HTML!
   <br>
   </body>
</html>
```



JavaScript - függvények

- A ***function*** kulcsszót közvetlenül a függvény neve követi, majd zárójelekben a paramétereket adjuk meg
- A függvény törzse kapcsos zárójelek között van, és ide helyezzük el a függvényhez tartozó utasításokat
- function FuggvenyNeve(parameter1, parameter2, …)
  { …utasítások… }
- A nyelv számos függvényt tartalmaz, pl. *alert()* függvény
- Mi is írhatunk függvényeket, hogy nagyobb, összetett feladatokat kisebb, jobban kezelhetőbb részekre bontsuk
- A függvényeknek adhatunk át paramétereket, amelyekkel dolgozni fognak, értéket is visszaadhatnak, csakúgy, mint más programozási nyelvekben

JavaScript - függvények



Nyelvi elemek

- változók
  - nem típusos nyelv
  - érvényességi köre a deklaráció helyétől függ
- eljárások, függvények
  - alert("Figyelmeztetés");
  - confirm("Valóban?"); → OK: true, Mégsem: false
  - prompt("valtozo_nev", "kezdeti_ertek");
- vezérlőszerkezetek
  - feltételes szerkezetek (if…else, switch)
  - iterációk (for, while, do…while)

Eseménykezelés

- eseménykezelők segítségével dinamikus weboldalak hozhatók létre
- tipikus események
  - egérkattintás
  - egér mozgatása valamely pont fölött
  - űrlapbeviteli mező kiválasztása vagy elhagyása
  - űrlap elküldése vagy alaphelyzetbe állítása
  - weblap vagy kép betöltődése

JavaScript - változók - vezérlőszerkezetek

- JavaScript-ben a változók neve
  - betűvel, vagy aláhúzással (_) kezdődhet
  - különbség van kis- és nagybetűk között
- Változót a ***var*** kulcsszóval, majd a változó nevének megadásával deklarálhatunk
- Deklaráláskor az értékadó operátor (=) segítségével kezdőértéket is rendelhetünk a változókhoz

```
// JavaScript
        text=prompt("Írjunk be valamit!");
        document.write(text);
```

Vezérlő szerkezetek

- iterációk (for, while, do...while)
- feltételes szerkezetek (if...else, switch)

```
// JavaScript
 for (i=1;i<5;i++){
        alert("Még "+(5-i)+" alkalommal írom ki ezt az üzenetet!");
        if (i==4) alert(„Azt hiszed, vége? Az OK után nyomd meg az F5-öt!");
                }
```

Dokumentum Objektum Modell

- szabványos, platform és nyelv-független modell
- interfészeket definiál HTML dokumentumok és alkalmazások között
- objektumokat, metódusokat és tulajdonságokat definiál, melyek lekérdezhetők és módosíthatók

Fontosabb JavaScript Objektumok

- Array

  concat(), sort(), push(), pop(),
- String

  length(), toUpperCase(), match(), indexOf(), replace()
- Date

  getTime()

  getFullYear(), getMonth(), getDate(), getDay()
- Math

  random(), max(), min(), round()

JavaScript - tömbök

- nagy mennyiségű adat kényelmes tárolását és gyors hozzáférést biztosít
- tömb sok változóból felépülő összetett adattípus
- adatokhoz hozzáférés:
  egy név (a tömb neve) és egy szám
- definiálás:
  tombnev = new Array([a_tomb_hossza])

  tombnev = new Array([1.elem, 2.elem, 3.elem, … ,n.elem]) .
- A **tomb** nevű új tömbünkhöz érték rendelése

  tomb[0] = "JavaScript";

  tomb[1] = "2011";

  tomb[2] = „szövegesadat";

Array objektum – legfontosabb metódusok

Pl.

AutoTipusok =
new Array("Honda",„Skoda","BMW");

legfontosabb metódusai

join metódus összefűzi a tömb elemeit egyetlen sztringgé

reverse megfordítja (transzponálja) a tömb elemeit, az utolsóból lesz az első, az elsőből az utolsó

sort rendezi a tömb elemeit

Metódusok vizsgálata

AutoTipusok.join() a „ Honda,Skoda,BMW " sztringet adja vissza

AutoTipusok.reverse() megfordítja a sorrendet

Autotipusok.sort() rendezi a tömböt
AutoTipusok[0] BMW ,
AutoTipusok[1] Honda,
AutoTipusok[2] Skoda

String objektum

- Nincs sztring adattípus, de van a **String** objektum
- String objektum létrehozása

  String_objektum_neve = new String(sztring);

  A zárójelben tetszőleges sztring állhat, ez lesz a **String** objektum

  pl. szoveg = new String("JavaScript");
- Adatmezője:
  length(szoveg); - kiolvassa a tárolt sztring hosszát
- A **String** objektum metódusai két típusba sorolhatók.
  1) sztring egy módosított változatát adja vissza
  subString metódus a sztring egy részét adja vissza,
  tooLowerCase metódus kisbetűsre alakítja a stinget

  2)HTML formátumúra alakító metódusok
  bold függvény
  link függvény

String objektum

Metódus	Leírás
anchor	HTML hivatkozást készít a sztringből
big, blink, bold, fixed, italics, small, strike, sub, sup	HTML-ként formázott sztringet hoz létre
chartAt	a paraméterként átadott pozícióban lévő karakterrel tér vissza
indexOf, lastIndexOf	A paraméterben meghatározott részsztring első vagy utolsó pozíciójával tér vissza. Ha ilyen nem szerepel benne, akkor -1 a visszaadott érték
link	HTML linket készít a sztringből
split	felosztja a sztringet részsztringekre egy elválasztó karakter mentén, majd ezeket egy tömbbe teszi
substring	a sztring egy meghatározott részével tér vissza
toLowerCase, toUpperCase	csupa kisbetűssé ill. nagybetűssé alakítja a sztringet

Date objektum

- hasznos előre definiált objektum
- Idő- vagy dátumértékeket kezelő alkalmazások…

Pl. weboldalon pontos idő, dátum megjenítése

- **Date** objektum létrehozása

dátum_objektum_neve = new Date([paraméterek]);

- paraméterek

    ma = new Date(); nincs ->aktuális dátum, idő

  sztring "Hónap Nap, Év óra:perc:másodperc”;

    apríl1= new Date(„Április 1, 2011");

    apríl1= new Date(„2011, 04, 01");

- set metódus beállítja dátum, idő értékét (setYear, setMonth, setDate, setMinutes, setSeconds)
- get metódus kiolvassa dátum, idő értékét (getYear, getMonth, getDate, getMinutes, getSeconds)

Math objektum

Tartalmazza a legtöbb trigonometrikus, exponenciális és logaritmikus függvényt
Pl.l egy X szög szinusza Math.sin(X);

Függvény	Leírás
abs	abszolút érték
sin, cos, tan	trigonometrikus függvények; az argumentum radiánban
acos, asin, atan	az előbbi függvények inverze; argumentum radiánban
exp, log	exponenciális függvény, természetes alapú logaritmus
ceil	felső egészrész
floor	alső egészrész
min, max	az argumentumként megadott két érték közül a kisebbet, ill. a nagyobbat adják vissza
pow	exponenciális függvény; első argumentuma az alap, a második a kitevő
round	kerekítés a legközelebbi egészre
sqrt	négyzetgyök függvény

Math objektum

- **random**() metódus – véletlenszám generálás
- **Math.Random**() - 0 és 1 közé eső véletlenszámot ad

- **with (Math){ kor_terulet = PI * r * r; x = r * sin(beta); c = sqrt(a*a + b*b);}**

A with utasítás használható, ha a Math objektumra gyakran kell hivatkozni , mert így nem kell minden Math objektumbeli metódus és konstans elé odaírni a "Math" hivatkozást. Ezzel a kódunk is átláthatóbbá vált.

Eseménykezelés

- interaktívvá teszik a weboldalakat, eseményeket észlelhetnek, billentyű lenyomásra, egérmozgatásra és válaszolhatnak is azokra.
- minden esemény egy objektumhoz tartozik
- minden eseménynek van neve
  pl. onMouseOver (ehhez nem kell a <script> </script> elempárt használni)
- pl. Kattints ide!
- ha több utasításra van szükségünk, megadhatunk függvényt is az eseménykezelőnek, amit a fejlécben deklarálunk
- így is megadhatunk egy eseménykezelőt: document.onMouseDown = mousealert();

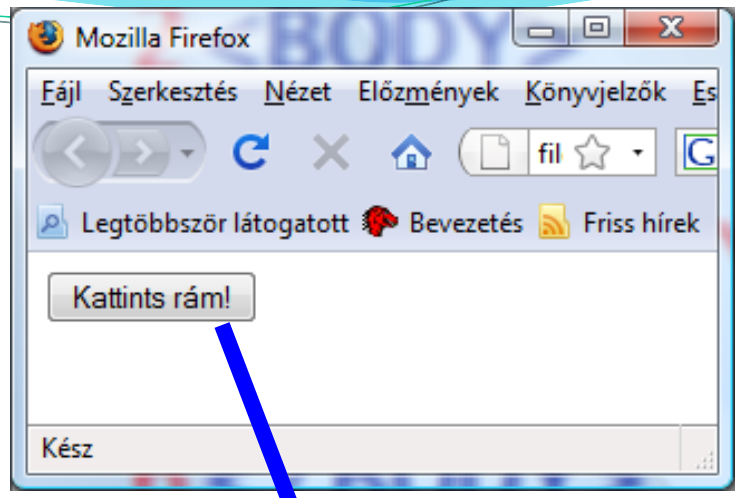
Gyakran használt eseménykezelők

- weboldal betöltésekor, elhagyásakor
  - onLoad, - böngésző típusának lekérdezése
  - onUnload, - kilépés kezelése
- űrlap kezelés
  - onFocus – belépés egy űrlap elembe
  - onBlur – űrlap elem elhagyása
  - onChange – űrlap elem tartalma megváltozik
  - onSubmit – űrlap elküldése
- egérmozgás
  - onMouseOver – egér fölé kerül egy elemnek
  - onMouseOut – egér elhagyja az elemet
- időzítő események
  - getHours(), getMinutes(), getSeconds()
  - setTimeout

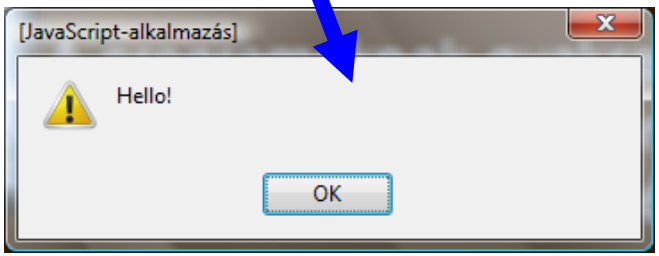
Események

Program:

reagálás az egér kattintására




```
<body>
        <form>
                <input type="button" VALUE="Kattints ide!"
        onClick="alert('Hello! Szöveg jelenik meg')">
        </form>
</body>
```



a felhasználó megnyomja a gombot,
Végrehajtódik az *alert('Hello!')* JavaScript kód
Ez a függvény létrehoz egy üzenetablakot, melyben a zárójelen belül idézőjelek közé írt szöveget jeleníti meg.
Idézőjelen belül aposztróf az idézőjel.

Események

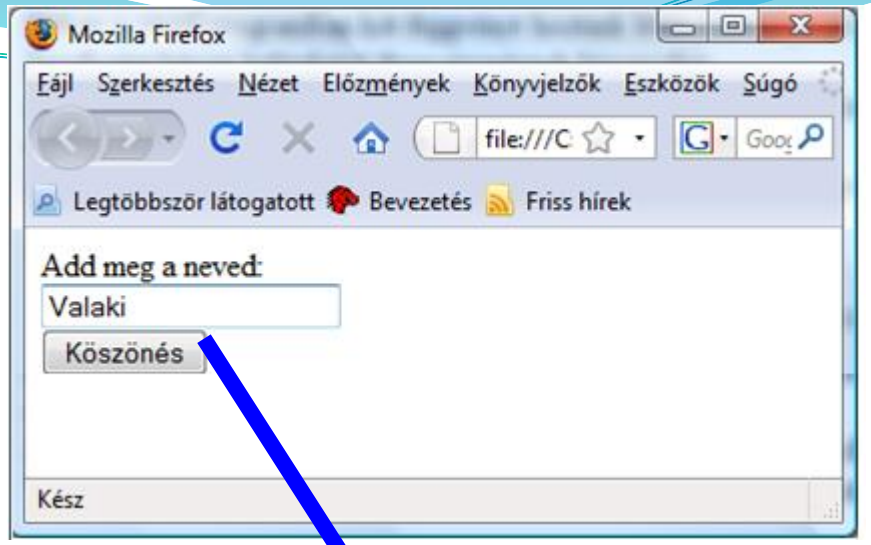
Események	Mire alkalmazható?	Mikor következik be?	Az eseménykezelő neve
abort	képek	A felhasználó megszakítja a kép betöltését	onAbort
blur	ablak, keret, és minden űrlapmező	Az egérrel az aktuális mezőn kívülre kattintunk	onBlur
change	szövegmező, listaablak	Megváltoztatunk egy űrlap beli értéket	onChange
click	gombok, rádió gomb, csatolások (linkek)	Az űrlap valamely elemére, vagy egy csatolásra (link) kattintunk	onClick
error	ablak, képek	Ha kép vagy dokumentum betöltésekor hiba lép fel	onError
focus	ablak, keret, minden űrlapmező	Kijelöljük az űrlap egy elemét	onFocus
load	dokumentum törzse (BODY)	A böngésző új lapot tölt be	onLoad

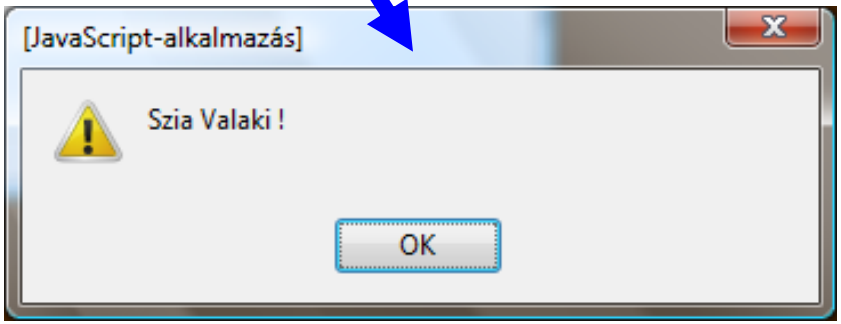
Események

Események	Mire alkalmazható?	Mikor következik be?	Az eseménykezelõ neve
mouseout	linkek	Az egérmutató elhagyja a linket	onMouseOut
mouseover	linkek	Az egérmutató valamely link felett tartózkodik	onMouseOver
reset	ûrlapokra	Ûrlap törlésekor	onReset
select	szövegmezõ	Új mezõt jelölünk ki	onSelect
submit	submit típusú nyomógomb	Ûrlap elküldésekor	onSubmit
unload	dokumentum törzse (BODY)	Ha másik HTML oldalra lépünk	onLoad

Függvények

Program: Név bekérés, alert ablakba írja ki: pl. Szia <NÉV>!






```
<html>
  <head>
    <script language="JavaScript">
        function koszon(){
          var x = document.urlap1.nev.value;
                    alert("Szia "+x+" !");
                    }
    </script>
   </head>

  <body>
    <form name="urlap1">
        add meg a neved:<br>
        <input type="text" name="nev"><br>
        <input type="button" value="köszönés"  onclick="koszon()">
    </form>
  </body>
</html>
```

HTML dokumentum

- A JavaScript egy HTML oldal minden elemét, és a böngésző ablakát is objektumként kezeli
- Rendelkezik tulajdonságokkal (adatmezőkkel) és megfelelő függvényekkel, amelyeken keresztül az oldal szinte minden tulajdonságát megváltoztathatjuk
- A böngésző ablaka a JavaScript szemszögéből egy ún. *window* objektum
- Az ablak belsejébe HTML oldalt tölthetünk, amit a JavaScript a *document* objektumon keresztül kezel

Böngésző objektumok

- window
  - a hierarchia csúcsán áll
  - böngészőablakot képvisel
- navigator
  - a böngészőről tárol információt
- screen
  - a kliens gép képernyőjéről rendelkezik információkkal
- history
  - korábban meglátogatott web oldalak címét tárolja
- location
  - az éppen aktuális oldal címét tartalmazza
  - reload() vagy replace() függvényei segítségével új oldal tölthető be

Window objektum

- Metódusok
  - open, showModelDialog, showModelessDialog, close, navigate
- Tulajdonságok
  - document, event, history, location, navigator
- Események
  - onload, onbeforeunload, onunload, onfocus, onblur
- Kollekció
  - frames

Dokument objektum

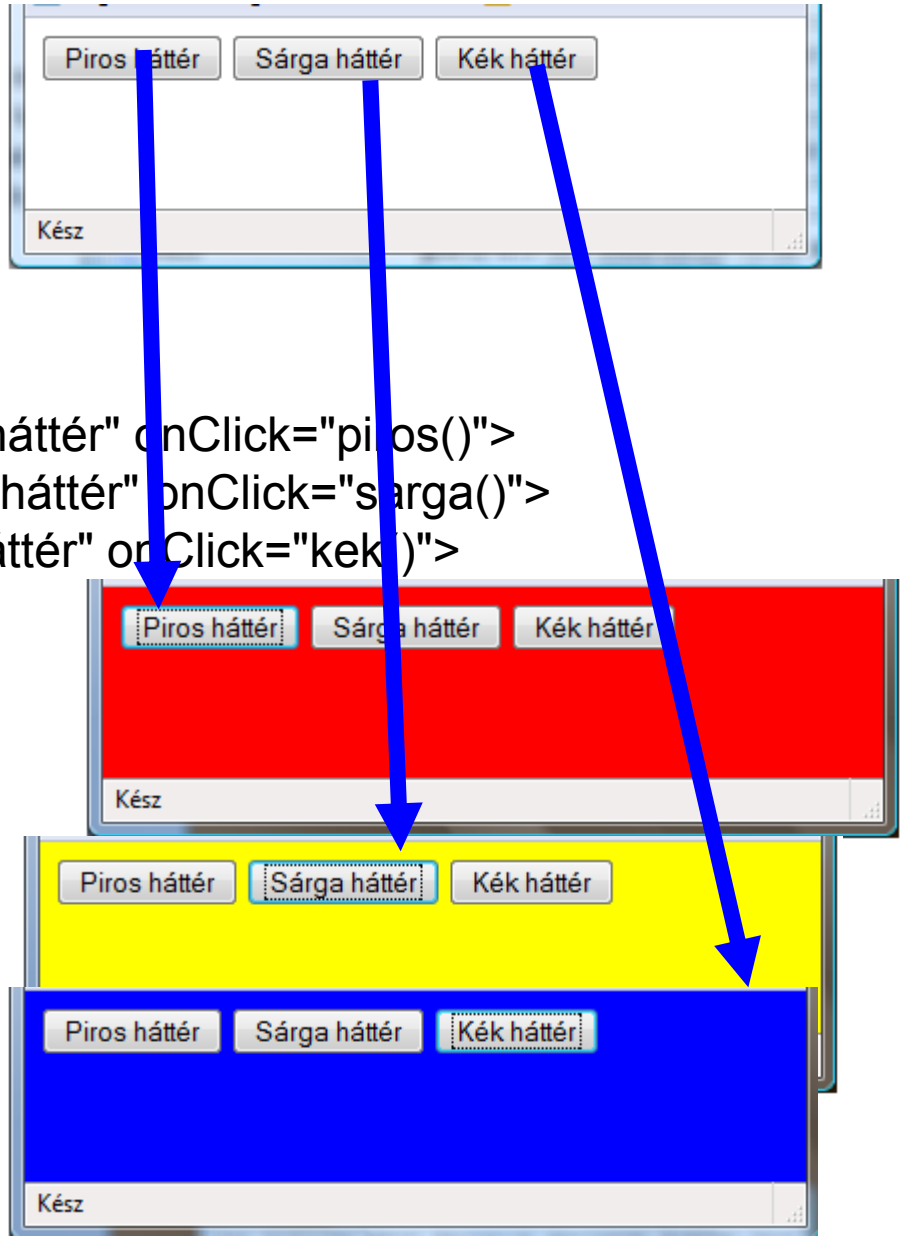
- Metódusok
  - open, write, close, createElement, insertAdjacentElement, insertBefore
- Tulajdonságok
  - body, cookie, title
- Események
  - onclick, onpropertychange, onmousexxx, onkeyxxx, ondragxxx
- Kollekciók
  - all, frames, forms

HTML dokumentum

Program: weblap háttérszínének változtatása nyomógombokkal

```
<FORM>
  <INPUT TYPE="button" VALUE="Piros háttér" onClick="piros()">
  <INPUT TYPE="button" VALUE="Sárga háttér" onClick="sarga()">
  <INPUT TYPE="button" VALUE="Kék háttér" onClick="kek()">
</FORM>
```

```
<SCRIPT LANGUAGE="JavaScript">
    function piros(){
      document.bgColor = "red";
    }
    function sarga(){
      document.bgColor = "yellow";
    }
    function kek(){
      document.bgColor = "blue";
    }
  </SCRIPT>
```



Ajánlott olvasmány:

http://zeus.nyf.hu/~akos/javascript/javascript.html
http://javascript.lap.hu/
http://weblabor.hu/cikkek/javascript-fuggvenyek

Köszönöm a figyelmet!