

# Programozás JavaScript nyelven

# Főbb témakörök

- A JavaScript nyelv szerepe
- A JavaScript nyújtotta lehetőségek
- A JavaScript beillesztése a weboldalakra

# A JavaScript szerepe

- weblapok dinamikus kezelhetősége érdekében fejlesztették ki
- legnépszerűbb szkriptnyelv
- C-típusú nyelv, nem módosított Java
- kliens-oldali JavaScript:
  - böngésző kezelése
  - kommunikáció a felhasználóval
- objektum alapú nyelv

# A JavaScript nyújtotta lehetőségek

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

# JavaScript a weboldalon

- beillesztése a `<script>` tag segítségével
- `<head>` elemen belül
  - függvények, változók deklarációja
  - látható kimenete itt ne legyen
  - az itt lévő kódot meg kell hívni
- `<body>` elemen belül
  - végrehajtás a feldolgozás során
  - függvényeket itt is meg kell hívni
- JavaScript kód külső fájlban is elhelyezhető  
`<script type="text/javascript" src="szkript.js">`

# Kliensoldali szkript alkalmazása

- HTML komment technikát kell használni

```
<SCRIPT Language=JavaScript>
```

```
<!--
```

```
Function Valami()
```

```
.....
```

```
End
```

```
-->
```

```
</SCRIPT>
```

- Ha a böngésző nem ért JavaScript-ül, akkor, ignorálja, mert komment-nek tekinti.

# JavaScript szabályok

- pontosvessző az utasítások végén
  - Nem kötelező használni!
- kommentjelek ( `//`, illetve `/*.....*/` )
- változódeklaráció
  - `var` Valtozo
    - érzékeny a kisbetű-nagybetű különbségre
    - a deklaráció nem kötelező.

# Példa – HEAD rész

```
<HEAD>
```

```
  <SCRIPT Language="JavaScript">
```

```
    <!--
```

```
    function openWin(URL) {
```

```
        awindow=window.open(URL,"new_window",  
        "toolbar=no,width=350,height=400,status=no,  
        scrollbars=yes,resizable=no,menubar=no")}
```

```
    End
```

```
  //-->
```

```
  </SCRIPT>
```

```
</HEAD>
```



# Példa – BODY rész

```
<BODY>
```

```
<A HREF=  
"javascript:openWin('mozart.html');">Mozart</A>
```

```
<A HREF=  
"javascript:openWin('beethoven.html');">Beethoven</A>
```

```
<A HREF=  
"javascript:openWin('wagner.html');">Wagner</A>
```

```
</BODY>
```

# Nyelvi elemek

- változók
  - nem típusos nyelv
  - érvényességi köre a deklaráció helyétől függ
- vezérlőszerkezetek
  - feltételes szerkezetek (if...else, switch)
  - iterációk (for, while, do...while)
- eljárások, függvények
  - alert("Figyelmeztetés");
  - confirm("Valóban?"); -> OK: true, Mégsem: false
  - prompt("valtozo\_nev", "kezdeti\_ertek");

# Eseménykezelés

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

# 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

# Gyakran használt eseménykezelők

- egérmozgás
  - `onMouseOver` – egér felö kerül egy elemnek
  - `onMouseOut` – egér elhagyja az elemet
- időzítő események
  - `getHours()`, `getMinutes()`, `getSeconds()`
  - `setTimeout`

# Fontosabb JavaScript Objektumok

- String
  - length(), toUpperCase(), match(), indexOf(), replace()
- Date
  - getTime()
  - getFullYear(), getMonth(), getDate(), getDay()
- Array
  - concat(), sort(), push(), pop(),
- Math
  - random(), max(), min(), round()

# 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

# Dokumentum Objektum Modell

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



# DOM története

- 1990 közepétől a W3C fejleszti
- DOM 0 specifikáció nem létezett, mégis HTML 4 specifikáció része
- DOM 1 – 1998
  - HTML és XML dokumentummodellre irányul
- DOM 2 – 2000
  - stíluslap objektummodelt ad meg
  - stílusinformáció kezelése
  - átfogó eseménymodell
- DOM 3 – 2004
  - XML-hez tartozó tartalommodellek
  - dokumentumvalidálás
  - dokumentumnézetek és formázás

# Window objektum

- Metódusok
  - open, showModalDialog, showModallessDialog, close, navigate
- Tulajdonságok
  - document, event, history, location, navigator
- Események
  - onload, onbeforeunload, onunload, onfocus, onblur
- Kollekción
  - frames

# Dokument objektum

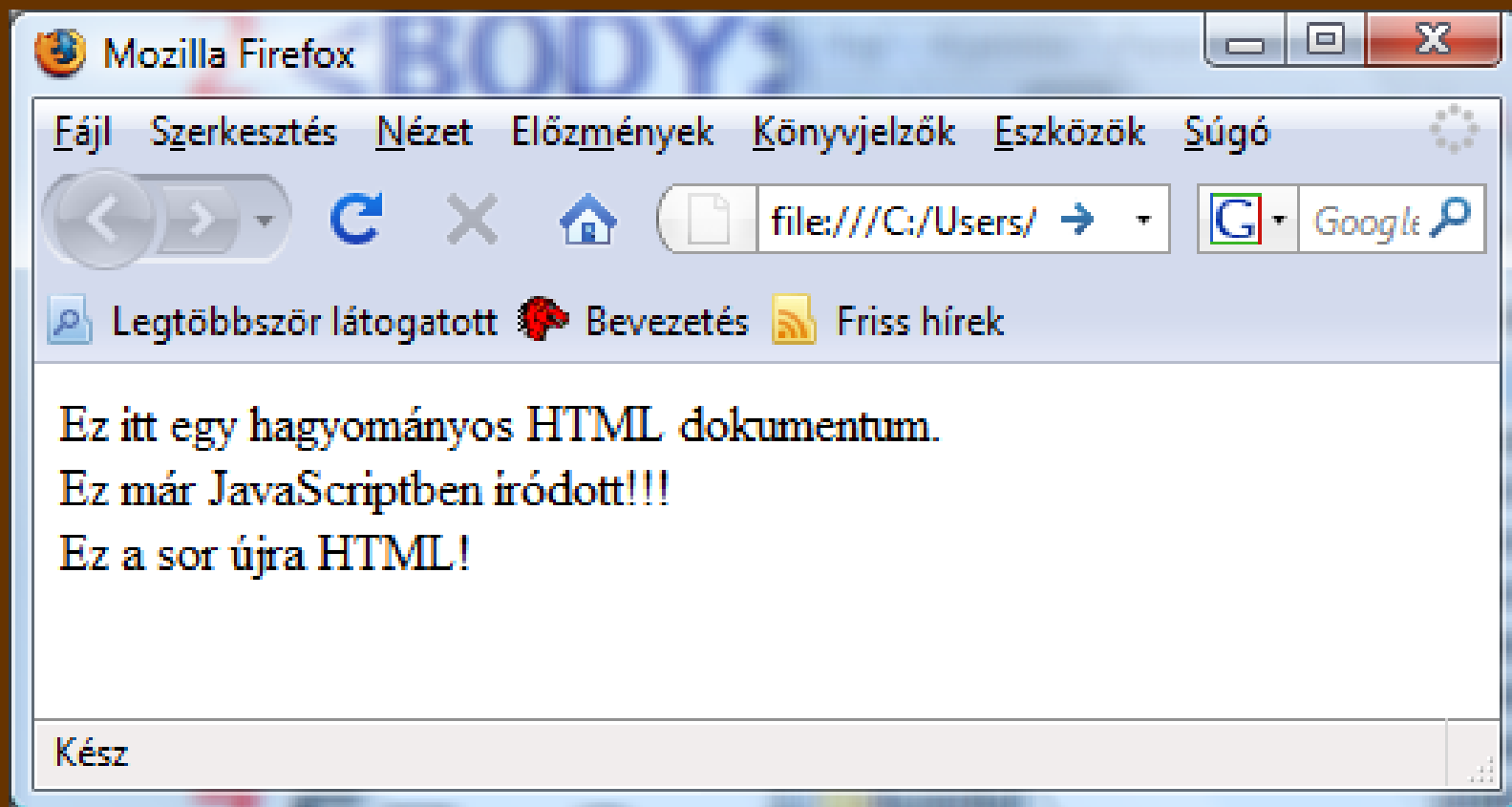
- Metódusok
  - open, write, close, createElement, insertAdjacentElement, insertBefore
- Tulajdonságok
  - body, cookie, title
- Események
  - onclick, onpropertychange, onmousexxx, onkeyxxx, ondragxxx
- Kollektciók
  - all, frames, forms

# A JavaScript nem Java

- Java önálló programozási nyelv, szükség van fordító programra, amivel a forráslistából bájtkódot készíthetünk
- A JavaScript pedig parancsnyelv, futtatásához nem kell más, mint egy böngésző, amely kezelni képes a JavaScriptet

# JavaScript beágyazása HTML dokumentumba

- Készítsük el a következő Weblapot!



<HTML>

<BODY>

Ez itt egy hagyományos HTML  
dokumentum.<BR>

<SCRIPT LANGUAGE="JavaScript">

**document.write("Ez már JavaScriptben  
íródott!!!<BR>")**

</SCRIPT>

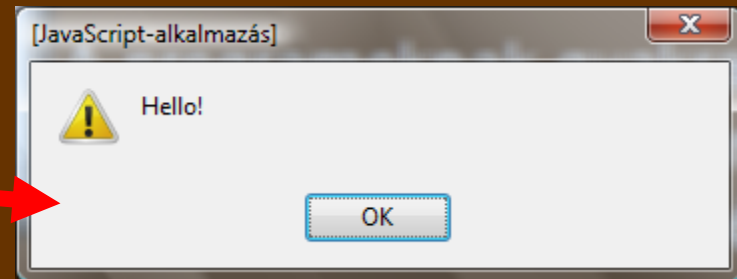
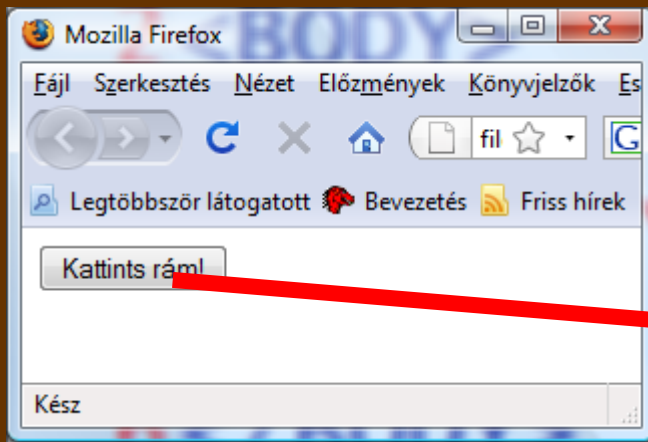
Ez a sor újra HTML!<BR>

</BODY>

</HTML>

# Események

- A programoknak gyakran kell reagálnia olyan eseményekre, mint billentyű lenyomása, vagy az egér mozgatása
- Készítsük el a következő weblapot!



```
<BODY>
```

```
<FORM>
```

```
<INPUT TYPE="button" VALUE="Kattints  
rám!" onClick="alert('Hello!')">
```

```
</FORM>
```

```
</BODY>
```



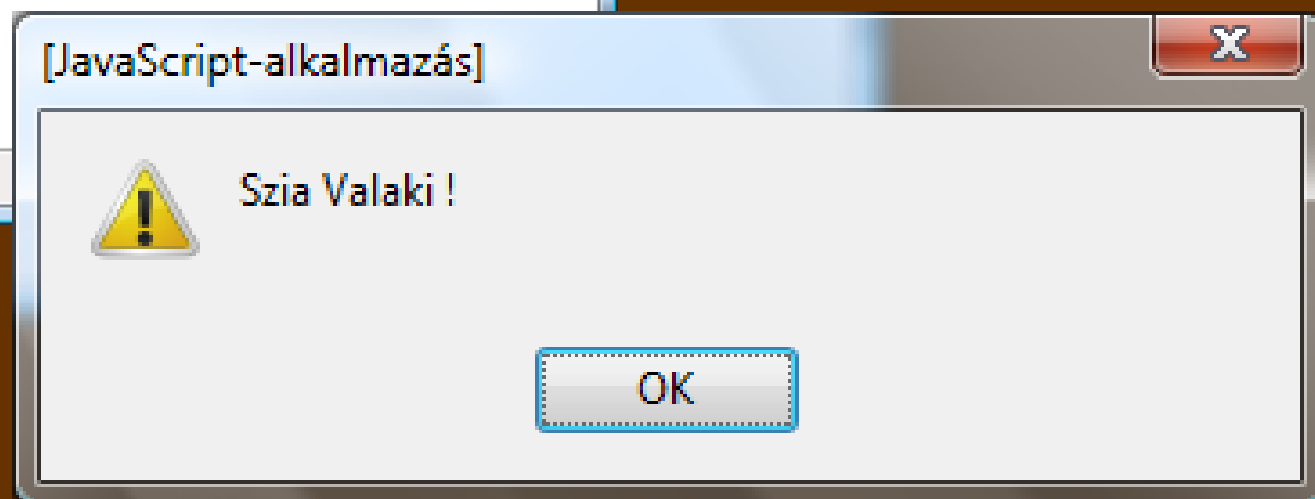
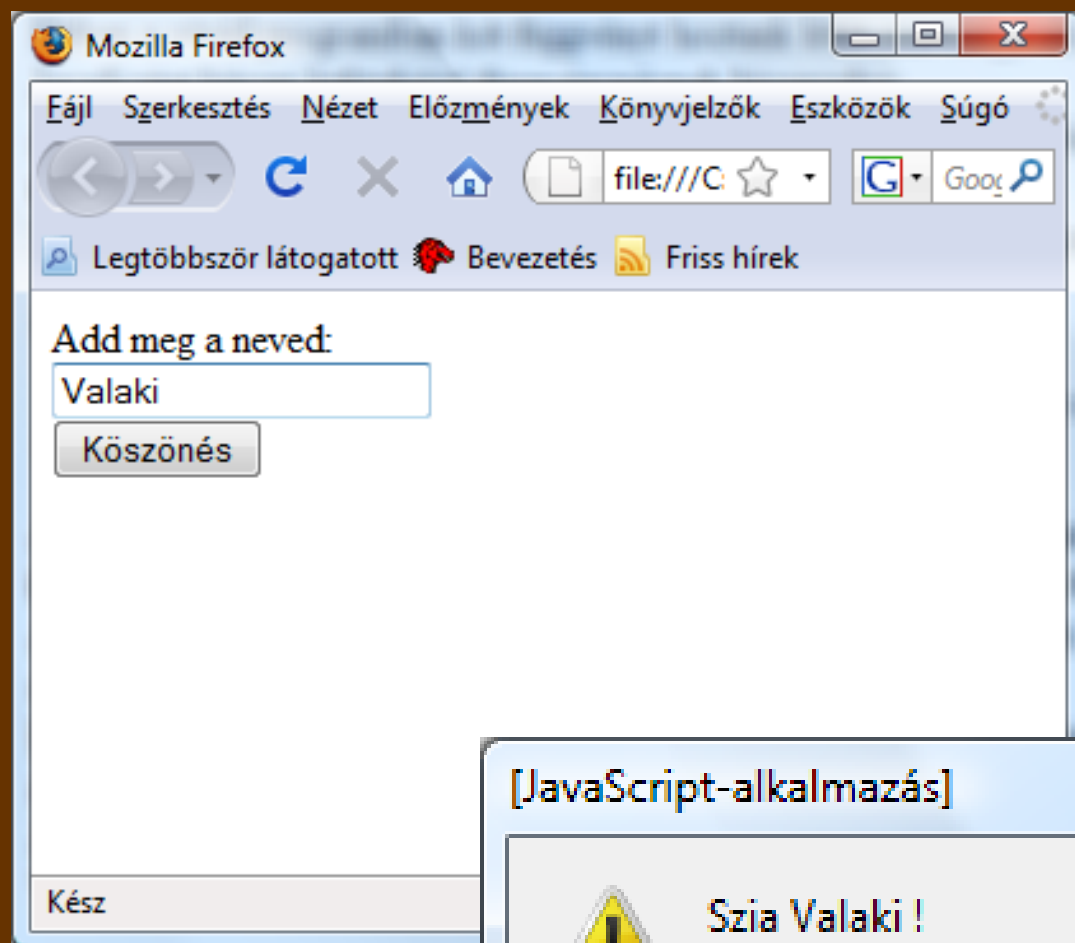
# onClick

- ha a felhasználó megnyomja a gombot
- végrehajtja az *alert('Hello!')* JavaScript kódot
- Ez egy függvény, amely létrehoz egy üzenetablakot, melyben a zárójelen belül idézőjelek közé írt szöveget jeleníti meg
- Figyeljük meg azt is, hogy lehet idézőjelen belül idézőjelet használni!

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	Megváltottatunk egy űrlap belső értékét	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
mouseout	linkek	Az egérmutató elhagyja a linket	onMouseOut
mouseover	linkek	Az egérmutató valamely link felett tartózkodik	onMouseOver
reset	űrlapokra	Űrlap törlések	onReset
select	szövegmező	Új mezőt jelölünk ki	onSelect
submit	submit típusú nyomógomb	Űrlap elküldések	onSubmit
unload	dokumentum törzse (BODY)	Ha másik HTML oldalra lépünk	onLoad

# Függvények

- a JavaScript-ben is rendkívül fontos szerepet kapnak a függvények
- Mi magunk is írhatunk függvényeket, hogy a nagyobb, összetettebb feladatokat kisebb, jobban kezelhetőbb részekre bontsuk
- Készítsünk olyan programot, amely bekér egy nevet, majd egy alert ablakban kiírja: Szia <NÉV>!



# BODY rész

- <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>

# HEAD rész

- <HTML>
- <HEAD>
- <SCRIPT LANGUAGE="JavaScript">
- function **koszon()**{
- var x = **document.urlap1.nev.value**;
- alert("Szia "+x+" !");
- }
- </SCRIPT>
- </HEAD>

# function

- 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
- A JavaScript programok változókbán tárolják az információkat, adatokat

# Változók

- a JavaScript-ben nem kell definiálnunk a változók típusát
  - Ebből lesz is majd néha bonyodalom
- azt a programot végrehajtó böngésző állapítja meg a változók használata alapján
- a változók neve betűvel, vagy aláhúzással (\_\_) kezdődhet
- különbség van kis- és nagybetűk között



# Változódeklaráció

- Változót a **var** kulcsszóval, majd a változó nevének megadásával deklarálhatunk
- Deklaráláskor az értékadó operátor (=) segítségével kezdőértéket is rendelhetünk a változókhoz
- A függvényeket a nevükkel hívjuk meg, és híváskor adjuk át a paraméterek aktuális értékét

# Feladat

- Készítsünk olyan programot, amely bekér két egész számot, és kiírja az összegüket!

```
<FORM NAME="urlap1">
```

Egyik szám:<BR>

```
<INPUT TYPE="text"  
  NAME="szam1"><BR>
```

Másik szám:<BR>

```
<INPUT TYPE="text"  
  NAME="szam2"><BR>
```

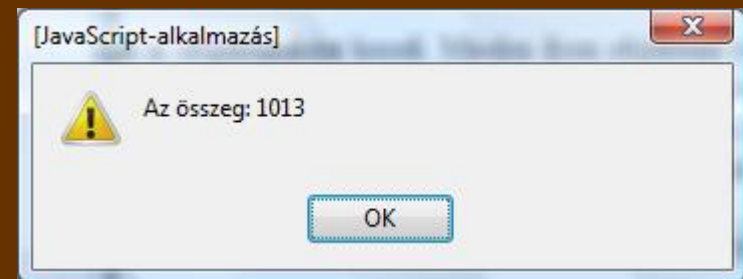
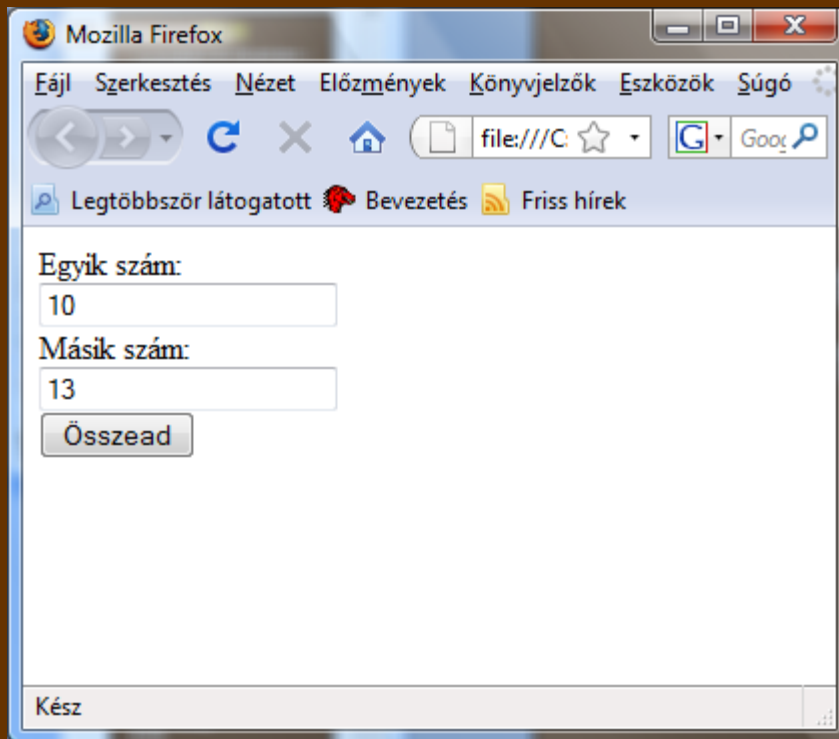
```
<INPUT TYPE="button" VALUE="Összead"  
  onClick="osszeg()">
```

```
</FORM>
```

- `<SCRIPT LANGUAGE="JavaScript">`
- `function osszeg(){`
- `var x = document.urlap1.szam1.value;`
- `var y = document.urlap1.szam2.value;`
- `var eredmeny = x + y;`
- `alert("Az összeg: "+eredmeny);`
- `}`
- `</SCRIPT>`

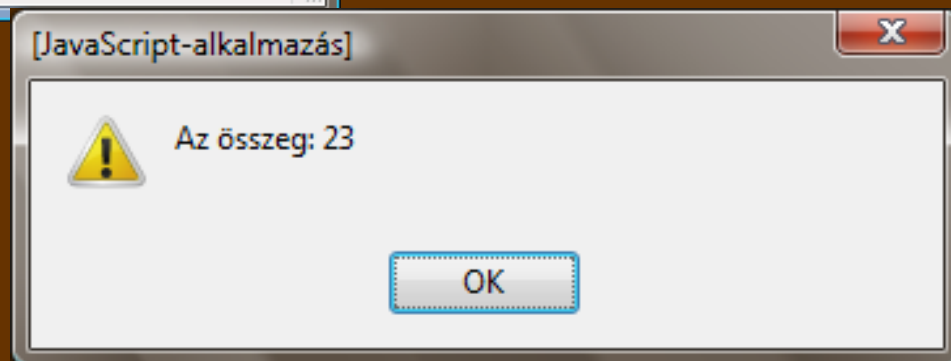
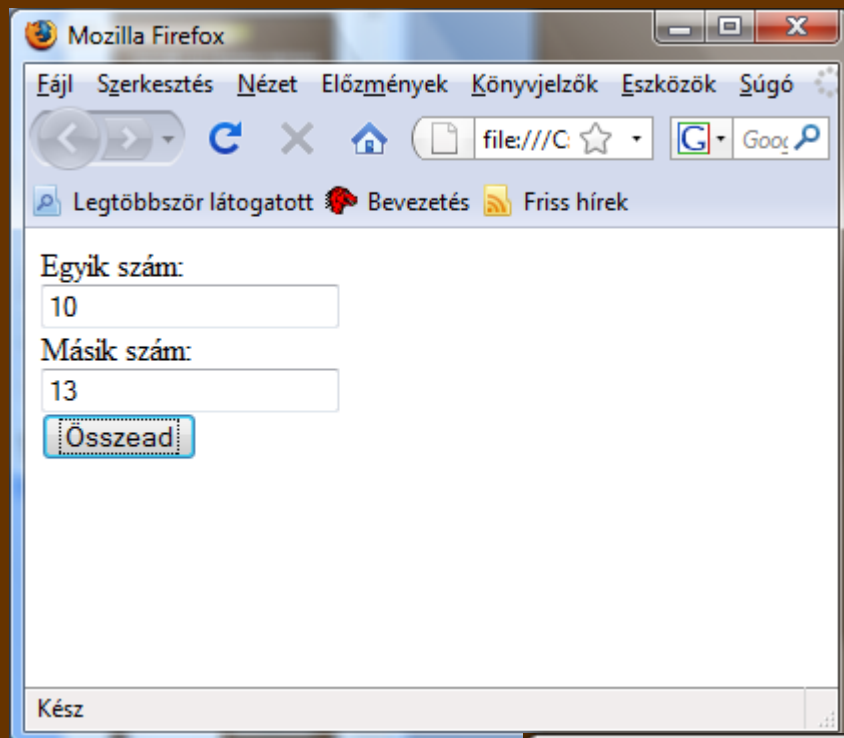
# Egy kis probléma

- Nem számként adja össze az értékeket, hanem szöveggként!



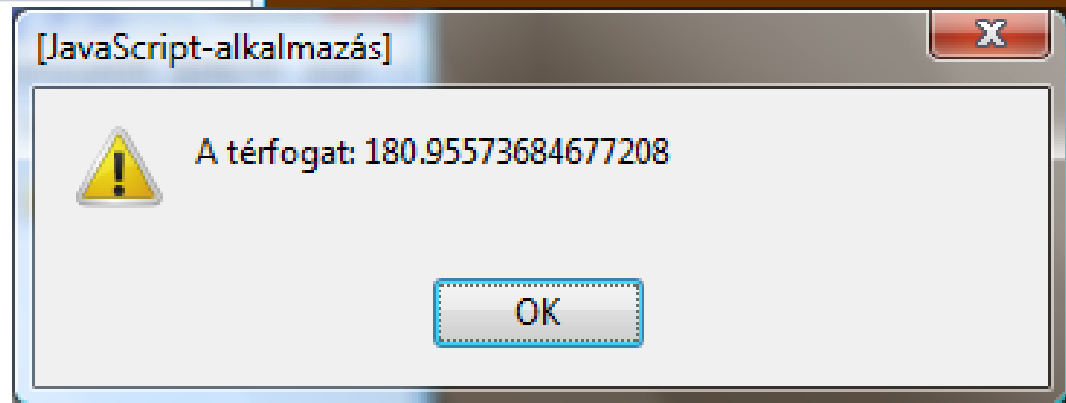
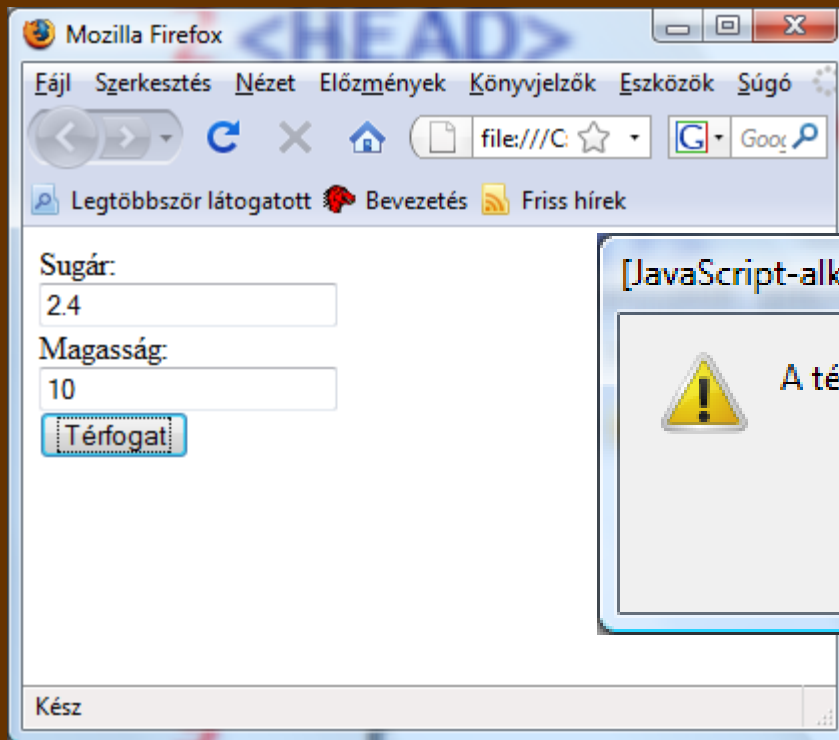
# Javítás

- Nem hivatalos megoldás:
  - `var x = document.urlap1.szam1.value*1;`
  - `var y = document.urlap1.szam2.value*1;`
- Hivatalos megoldás:
  - `var x =`  
`parseInt(document.urlap1.szam1.value);`
  - `var y =`  
`parseInt(document.urlap1.szam2.value);`



# Feladat

- Kérjük be egy henger sugarát és magasságát, majd írjuk ki a térfogatát!





```
<FORM NAME="urlap1">
```

```
Sugár:<BR>
```

```
<INPUT TYPE="text" NAME="sugar"><BR>
```

```
Magasság:<BR>
```

```
<INPUT TYPE="text"  
  NAME="magassag"><BR>
```

```
<INPUT TYPE="button" VALUE="Térfogat"  
  onClick="terfogat()">
```

```
</FORM>
```

```
<SCRIPT LANGUAGE="JavaScript">
function terfogat(){
var r = document.urlap1.sugar.value*1;
var m =
document.urlap1.magassag.value*1;
var eredmeny = r*r*Math.PI*m;
alert("A térfogat: "+eredmeny);
}
</SCRIPT>
```

Hivatalosan a parseFloat függvény alakít át szöveget  
valós számmá!!!

# Feladat

- Ha beteszünk a bankba egy adott összeget, adott éves kamatszázalékra, adott hónapra, akkor mennyi pénzt vehetünk majd fel az idő lejártakor?

Mozilla Firefox

Fájl Szerkesztés Nézet Előzmények Könyvjelzők Eszközök Súgó

file:///C: Goo

Legtöbbször látogatott Bevezetés Friss hírek

Összeg (Ft):  
10000


Kamat %:  
20

Hány hónapra köti le  
10

Végösszeg

Kész

[JavaScript-alkalmazás]

 A térfogat: 11640.841385349606

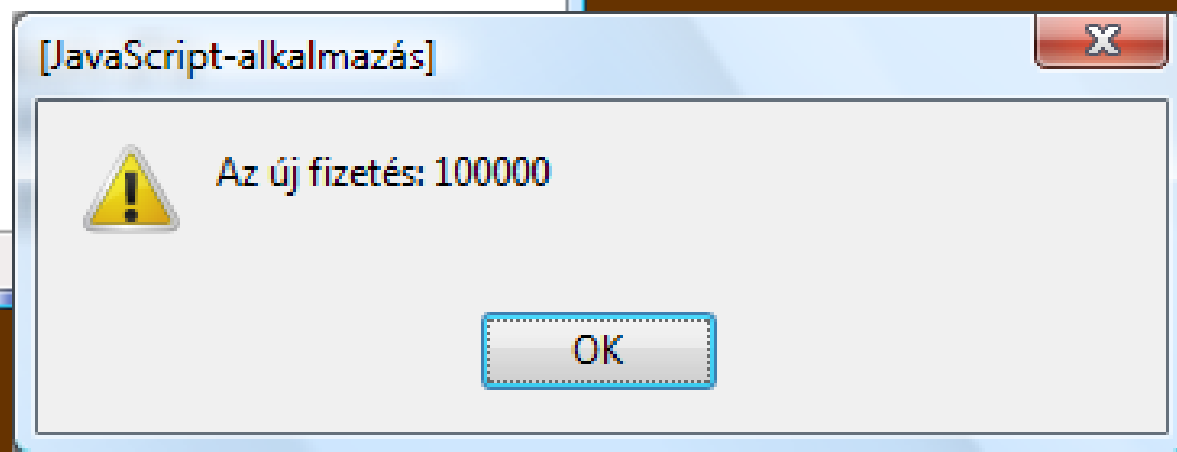
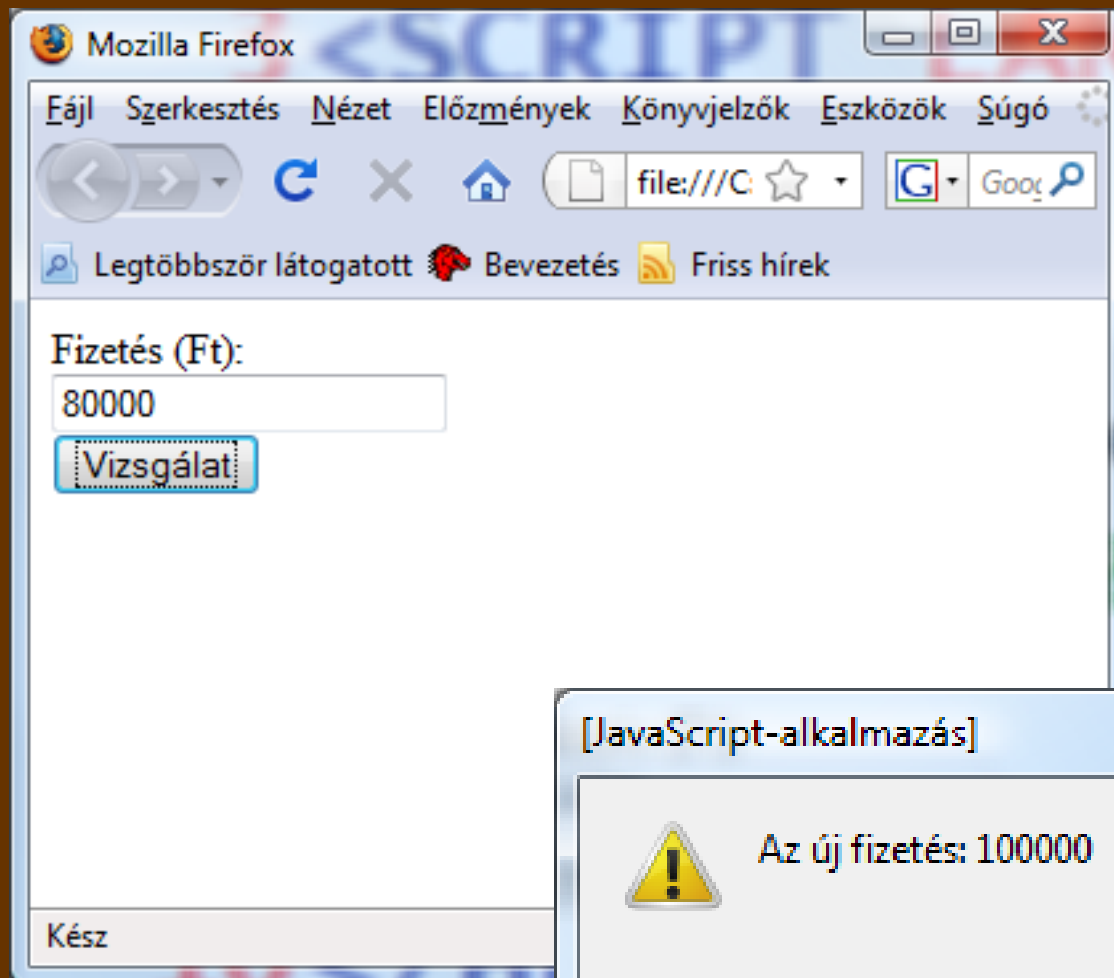
OK

- `<FORM NAME="urlap1">`
- `Összeg (Ft):<BR>`
- `<INPUT TYPE="text" NAME="osszeg"><BR>`
- `Kamat %:<BR>`
- `<INPUT TYPE="text" NAME="kamat"><BR>`
- `Hány hónapra köti le<BR>`
- `<INPUT TYPE="text" NAME="honap"><BR>`
- `<INPUT TYPE="button" VALUE="Végösszeg"`  
`onClick="szamit()">`
- `</FORM>`

- `<SCRIPT LANGUAGE="JavaScript">`
- `function szamit(){`
- `var ossz =`  
`document.urlap1.osszeg.value*1;`
- `var k = document.urlap1.kamat.value*1;`
- `var ho = document.urlap1.honap.value*1;`
- `var eredmeny =`  
`ossz*Math.pow(1+k/100,ho/12);`
- `alert("A térfogat: "+eredmeny);`
- `}`
- `</SCRIPT>`

# Feladat

- Kérjük be egy alkalmazott fizetését. Ha ez a fizetés 100000 Ft-nál nem nagyobb, akkor emeljük meg 25%-kal. Végül írjuk ki az alkalmazott fizetését!





- <FORM NAME="urlap1">
- Fizetés (Ft):<BR>
- <INPUT TYPE="text"  
NAME="fizetes"><BR>
- <INPUT TYPE="button"  
VALUE="Vizsgálat" onClick="emel()">
- </FORM>

- `<SCRIPT LANGUAGE="JavaScript">`
- `function emel(){`
- `var ossz =`  
`document.urlap1.fizetes.value*1;`
- `if (ossz<=100000) ossz = ossz * 1.25`
- `alert("Az új fizetés: "+ossz);`
- `}`
- `</SCRIPT>`

# Feladat

- Egy dolgozat ponthatárai a következők: 0 – 13 elégtelen, 14 – 17 elégsége, 18 – 23 közepes, 24 – 29 jó, 30 – 34 jeles. Írjunk olyan programot, amely bekéri a pontszámot, majd kiírja az osztályzatot.



Mozilla Firefox

Fájl Szerkesztés Nézet Előzmények Könyvjelzők Eszközök Súgó



file:///C:



Google



Legtöbbször látogatott



Bevezetés



Friss hírek

Pontszám:

24

Osztályzat

[JavaScript-alkalmazás]



jó

OK

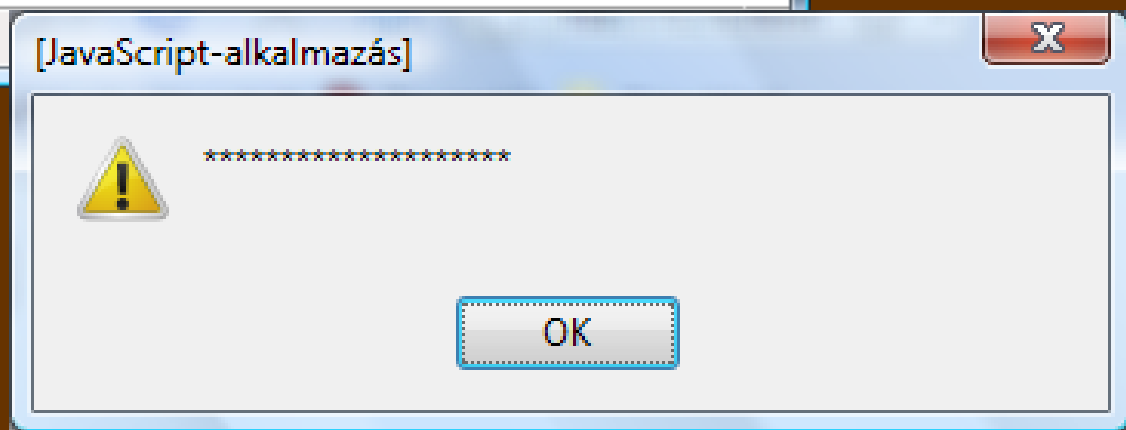
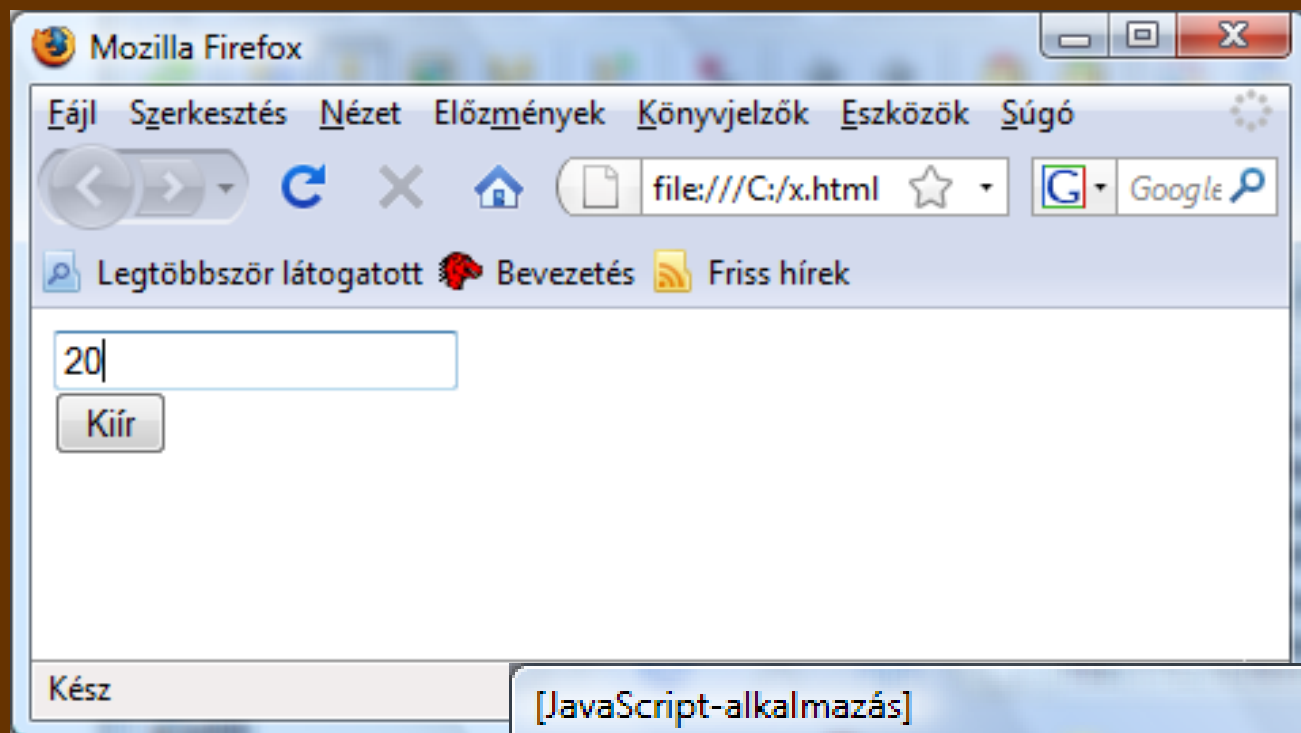
Kész

- <FORM NAME="urlap1">
- Pontszám:<BR>
- <INPUT TYPE="text"  
NAME="pontszam"><BR>
- <INPUT TYPE="button"  
VALUE="Osztályzat"  
onClick="osztalyzat()">
- </FORM>

- `<SCRIPT LANGUAGE="JavaScript">`
- `function osztalyzat(){`
- `var pont = document.urlap1.pontszam.value*1;`
- `if (pont<0) alert("Hibás adat!");`
- `else if (pont<=13) alert("elégtelen")`
- `else if (pont<=17) alert("elégséges")`
- `else if (pont<=23) alert("közepes")`
- `else if (pont<=30) alert("jó")`
- `else if (pont<=34) alert("jeles")`
- `else alert("Hibás adat");`
- `}`
- `</SCRIPT>`

# Feladat

- Készítsünk olyan weblapot, amely bekér egy egész számot, majd kiír annyi darab \*-ot!





```
<FORM NAME="urlap">
```

```
<INPUT TYPE="text" NAME="db"><BR>
```

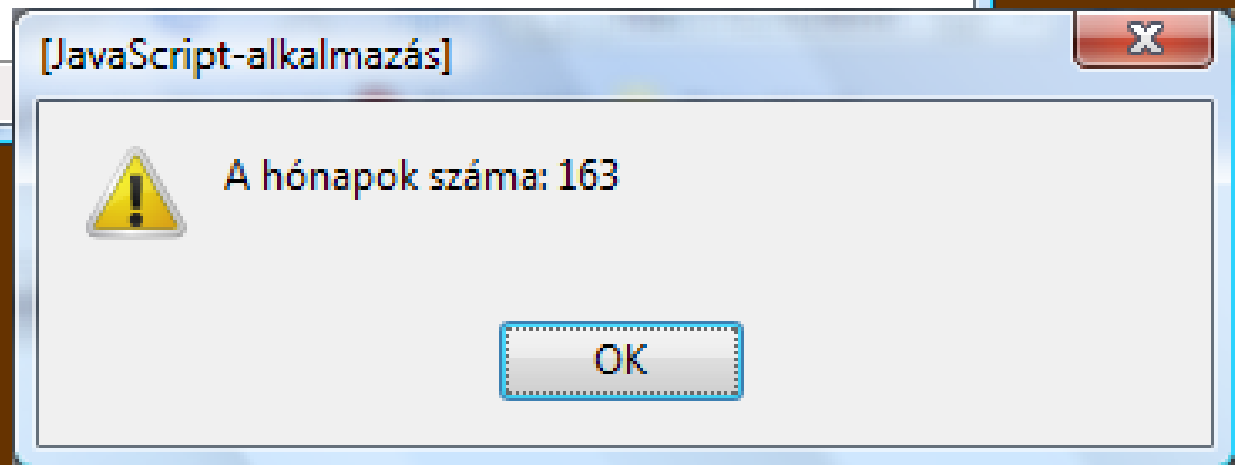
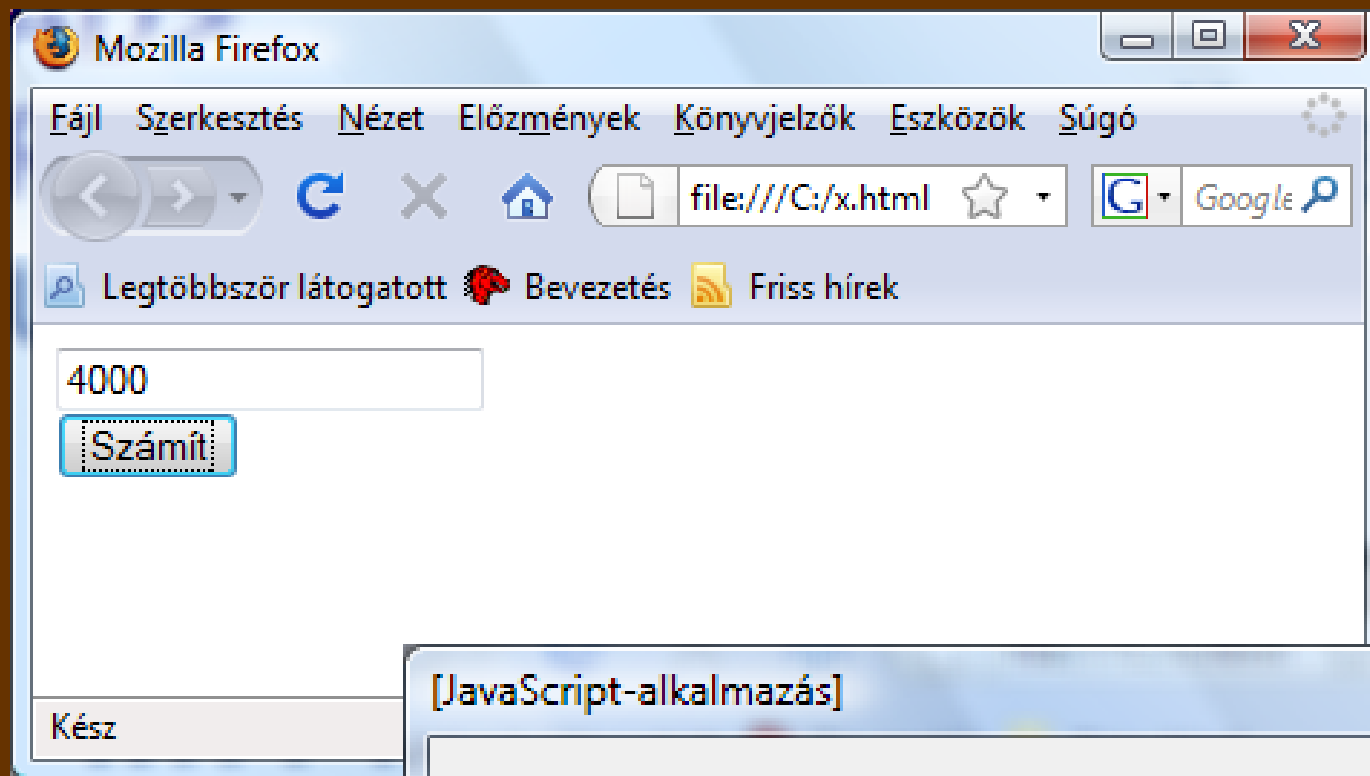
```
<INPUT TYPE="button" VALUE="Kiír"  
  onClick="csillag()">
```

```
</FORM>
```

```
<SCRIPT LANGUAGE="JavaScript">
function csillag(){
    szoveg = "";
    darab = document.urlap.db.value*1;
    for (var i=1;i<=darab;i++){
        szoveg = szoveg + "*";
    }
    alert(szoveg);
}
</SCRIPT>
```

# Feladat

- Van egy kis megtakarított pénzem. Arra vagyok kíváncsi, hány hónap múlva éri el az összeg a bankban a 100 000 Ft-ot, ha havi 2%-os kamattal számolhatok?



```
<FORM NAME="urlap">  
<INPUT TYPE="text" NAME="ft"><BR>  
<INPUT TYPE="button" VALUE="Számít"  
  onClick="kamat()">  
</FORM>
```

```
<SCRIPT LANGUAGE="JavaScript">
function kamat(){
    var honap = 0;
    var penz = document.urlap.ft.value*1;
    while (penz < 100000){
        honap++;
        penz = penz*1.02;
    }
    alert("A hónapok száma: "+honap);
}
</SCRIPT>
```

# Feladat

- Az előző feladatot oldjuk meg hátultesztelő ciklus segítségével is!

```
<SCRIPT LANGUAGE="JavaScript">
function kamat(){
    var honap = 0;
    var penz = document.urlap.ft.value*1;
    do {
        honap++;
        penz = penz*1.02;
    } while (penz < 100000)
    alert("A hónapok száma: "+honap);
}
</SCRIPT>
```



# A HTML dokumentum

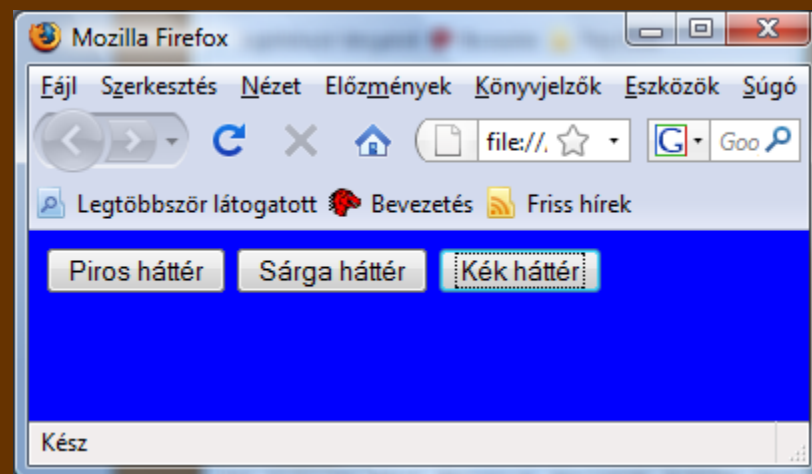
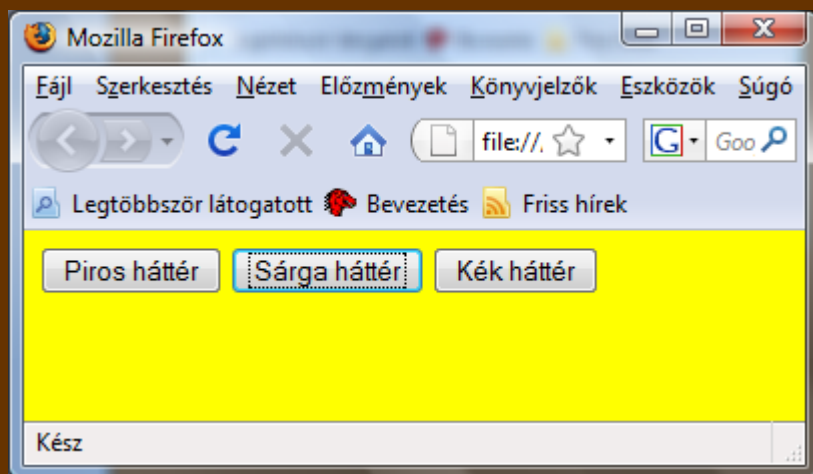
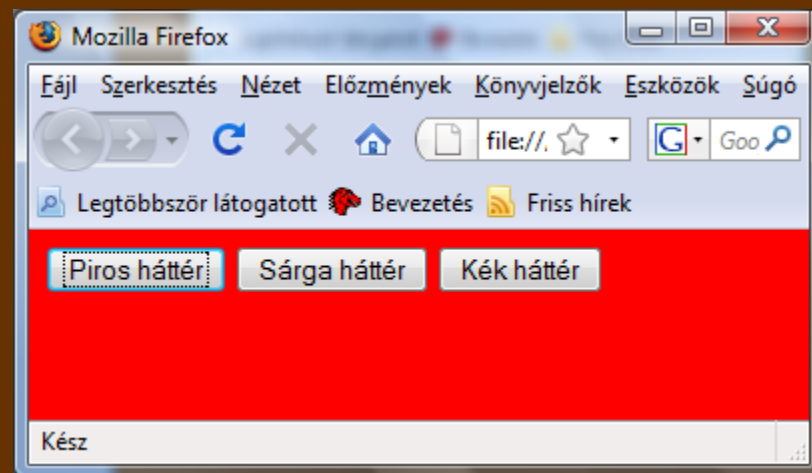
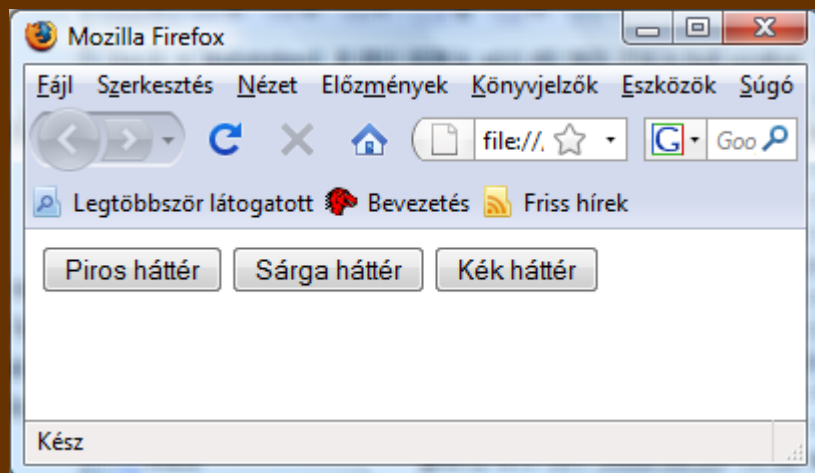
- A JavaScript egy HTML oldal minden elemét, és a böngésző ablakát is objektumként kezeli
- rendelkezik tulajdonságokkal (adatmezőkkel) és megfelelő függvényekkel, amelyeken keresztül az oldal szinte minden tulajdonságát megváltoztathatjuk

# A HTML dokumentum

- A böngésző ablaka a JavaScript szemszögéből egy ún. *window* objektum
- Az ablak belsejébe HTML oldalt tölthetünk, amit a JavaScript a *document* objektumon keresztül kezel

# Feladat

- Készítsünk olyan weblapot, melynek háttérszínét nyomógombokkal tudjuk változtatni!



<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>
```

# Egyéb színek

<code>alinkColor</code>	Az aktív linkek színe
<code>bgColor</code>	Háttérszín
<code>fgColor</code>	Az előtér színe
<code>linkColor</code>	A linkek színe
<code>vlinkColor</code>	Már meglátogatott linkek színe

# Feladat

- Készítsünk olyan programot, amely szövegbeviteli mezőkbe bekért két szám összegét egy harmadik szövegbeviteli mezőbe írja ki!





Mozilla Firefox



Fájl Szerkesztés Nézet Előzmények Könyvjelzők Eszközök Súgó



file://.



Goo



Legtöbbször látogatott



Bevezetés



Friss hírek

12

4

Összeadás

16

Kész

```
<FORM NAME="urlap">  
  <INPUT TYPE="text" NAME="a"><BR>  
  <INPUT TYPE="text" NAME="b"><BR>  
  <INPUT TYPE="button"  
  VALUE="Összeadás"  
  onClick="osszead()"><BR>  
  <INPUT TYPE="text"  
  NAME="ossz"><BR>  
</FORM>
```

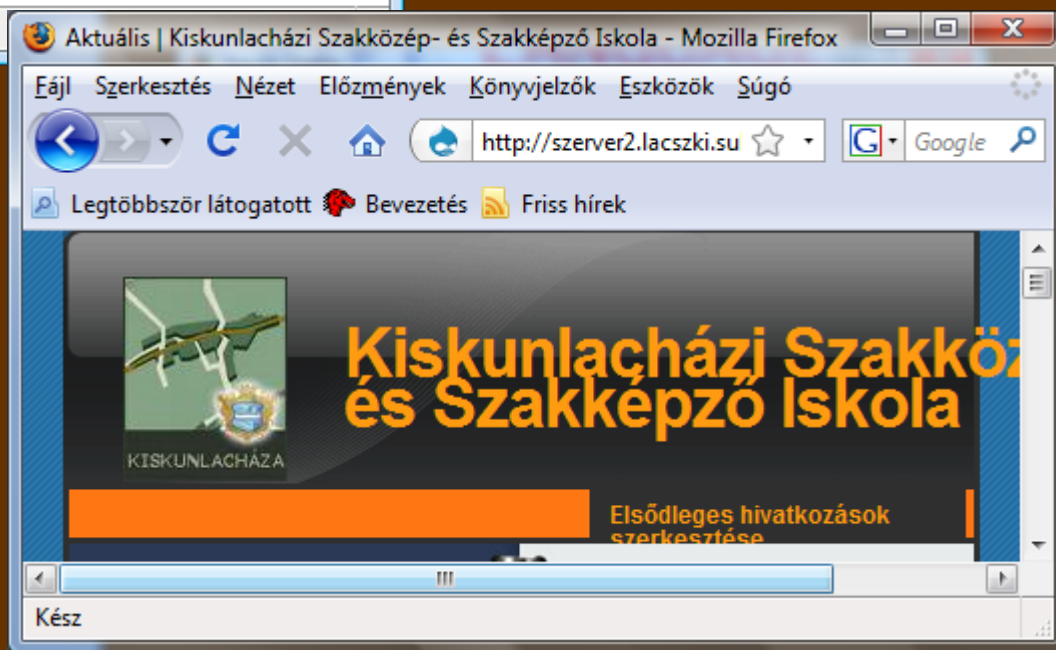
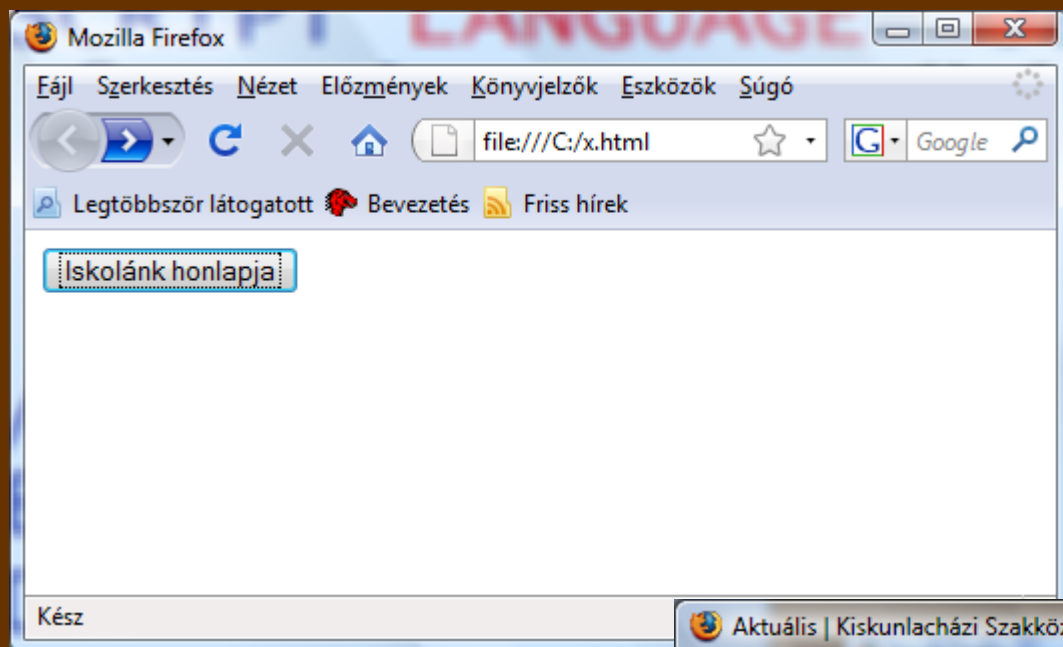
```
<SCRIPT LANGUAGE="JavaScript">  
  function osszead(){  
    var x = document.urlap.a.value*1;  
    var y = document.urlap.b.value*1;  
    var z = x + y;  
    document.urlap.ossz.value = z;  
  }  
</SCRIPT>
```

# LOCATION

- Ez az objektum reprezentálja a betöltött HTML dokumentum címét
- Ha például betöltöttünk egy oldalt, akkor a *location* objektum *href* mezője (*location.href*) ennek az oldalnak a címét fogja tárolni

# Feladat

- Készítsünk olyan weboldalt, amelyen egy gomb jelenti a hivatkozást az iskolánk honlapjára!



```
<FORM NAME="urlap">
```

```
  <INPUT TYPE="button" VALUE="Iskolánk  
honlapja" onClick="ugras()"><BR>
```

```
</FORM>
```

```
<SCRIPT LANGUAGE="JavaScript">  
  function ugras(){  
    location.href =  
    "http://szerver2.lacszki.sulinet.hu";  
  }  
</SCRIPT>
```



# Keretek kezelése JavaScript-ben

- A JavaScript a kereteket is egy logikus hierarchiába rendezi.
- A hierarchia legtetején maga a böngésző ablaka van
- Az ablak a két keret szülője, míg a keretek a gyermekek
- az ablakra *parent* néven, míg a keretekre azon a néven hivatkozhatunk, amit a `<FRAME>` HTML elem `NAME` attribútumában megadtunk

# Feladat

- Készítsünk olyan weblapot, amelyben az egyik keretben lévő nyomógombok meg tudják változtatni egy másik keret háttérszínét!



Mozilla Firefox



Fájl Szerkesztés Nézet Előzmények Könyvjelzők Eszközök Súgó



file:///C:/qqq.html



Google



Legtöbbször látogatott



Bevezetés



Friss hírek

Piros

Sárga

Kék

Ez a jobb oldali keret.

Kész

# Kezdőlap

```
<HTML>
```

```
<HEAD>
```

```
</HEAD>
```

```
<FRAMESET COLS="30%,*">
```

```
  <FRAME SRC="bal.html">
```

```
  <FRAME SRC="jobb.html">
```

```
</FRAMESET>
```

```
</HTML>
```

# Jobb oldali keret

<HTML>

<HEAD>

</HEAD>

<BODY>

Ez a jobb oldali keret.

</BODY>

</HTML>

# Bal oldali keret (FORM)

```
<FORM>
```

```
  <INPUT TYPE="button" VALUE="Piros"  
onClick="piros()"><BR>
```

```
  <INPUT TYPE="button" VALUE="Sárga"  
onClick="sarga()"><BR>
```

```
  <INPUT TYPE="button" VALUE="Kék"  
onClick="kek()"><BR>
```

```
</FORM>
```

# Bal oldali keret (SCRIPT)

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

# Feladat

- Készítsünk olyan weblapot, ahol az egyik keret szövegbeviteli mezőibe beírt két számot összeadjuk, és egy másik keret szövegbeviteli mezőjébe írjuk ki az összeget!





Mozilla Firefox



Fájl Szerkesztés Nézet Előzmények Könyvjelzők Eszközök Súgó



file:///C:/qqq.html



Google



Legtöbbször látogatott



Bevezetés



Friss hírek

3

12

Összeadás

15



Kész

# Jobb oldali keret

```
<HTML>
```

```
<HEAD>
```

```
</HEAD>
```

```
<BODY>
```

```
  <FORM NAME="urlap2">
```

```
    <INPUT TYPE="text" NAME="osszeg">
```

```
  </FORM>
```

```
</BODY>
```

```
</HTML>
```

# Bal oldali keret (FORM)

```
<FORM NAME="urlap">  
  <INPUT TYPE="text" NAME="a"><BR>  
  <INPUT TYPE="text" NAME="b"><BR>  
  <INPUT TYPE="button"  
  VALUE="Összeadás"  
  onClick="osszead()"><BR>  
</FORM>
```

# Bal oldali keret (SCRIPT)

```
<SCRIPT LANGUAGE="JavaScript">  
  function osszead(){  
    var x = document.urlap.a.value*1;  
    var y = document.urlap.b.value*1;  
    var z = x + y;  
    parent.jobb.document.urlap2.osszeg.value = z;  
  }  
</SCRIPT>
```

# Feladat

- Készítsünk olyan weblapot, amelyben az egyik keretben lévő nyomógomb megnyomásának hatására, a másik keretbe betöltődik az iskolánk weblapja!



file:///C:/qqq.html



Google



Legtöbbször látogatott

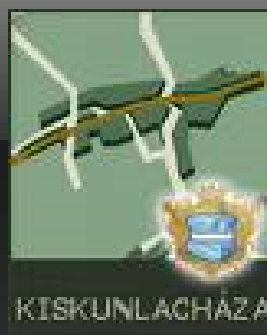


Bevezetés



Friss hírek

Iskola



KISKUNLACHÁZA

Kiskunlacháza  
Szakközép-  
Szakképző  
Iskola

Elsődleg  
hivatkoz  
kes

Kész

```
<FORM NAME="urlap">
```

```
  <INPUT TYPE="button" VALUE="Iskola"  
  onClick="betolt()"><BR>
```

```
</FORM>
```

```
<SCRIPT LANGUAGE="JavaScript">  
  function betolt(){  
    parent.jobb.location.href =  
    "http://szerver2.lacszki.sulinet.hu";  
  }  
</SCRIPT>
```

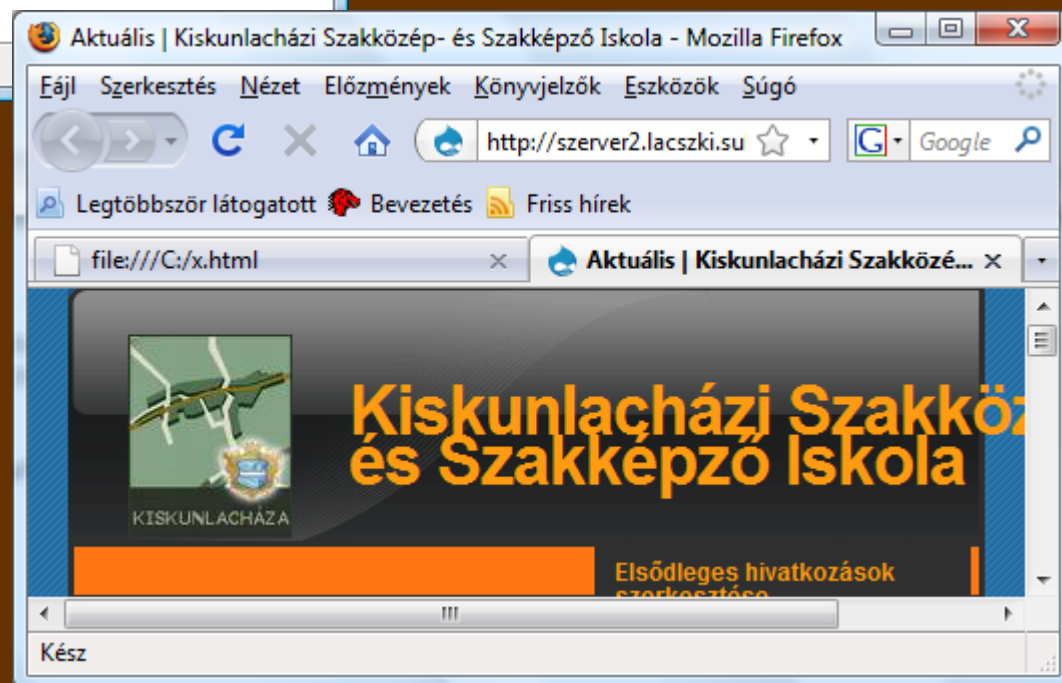
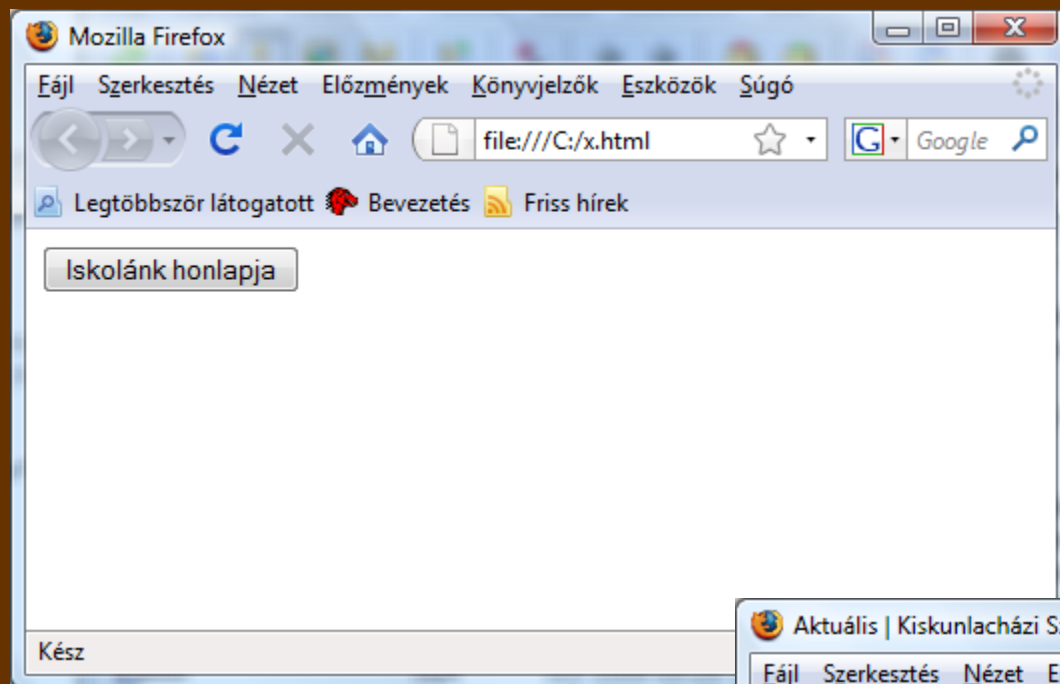


# Ablakok

- A JavaScript segítségével megnyithatunk új böngésző ablakokat, amelyekbe betölthetünk új HTML oldalakat.

# Feladat

- Nyissunk meg egy új ablakot, amelyben betöltődik iskolánk honlapja!



```
<FORM NAME="urlap">  
  <INPUT TYPE="button" VALUE="Iskolánk  
honlapja" onClick="ujablak()"><BR>  
</FORM>
```

```
<SCRIPT LANGUAGE="JavaScript">  
  function ujablak(){  
    var ablak =  
    open("http://szerver2.lacszki.sulinet.hu");  
  }  
</SCRIPT>
```

# Új ablak

- az ***open()*** függvénnnyel nyitottuk meg az új böngésző ablakot
- megjelenését is szabályozhatjuk
- a tulajdonságokat egyetlen, ***szóközöket nem tartalmazó*** sztringben adjuk meg
- `ablak = open("pelda.htm",  
"uj_ablak", "width=300,height=200,status=no,menubar=no");`

# Tulajdonságok

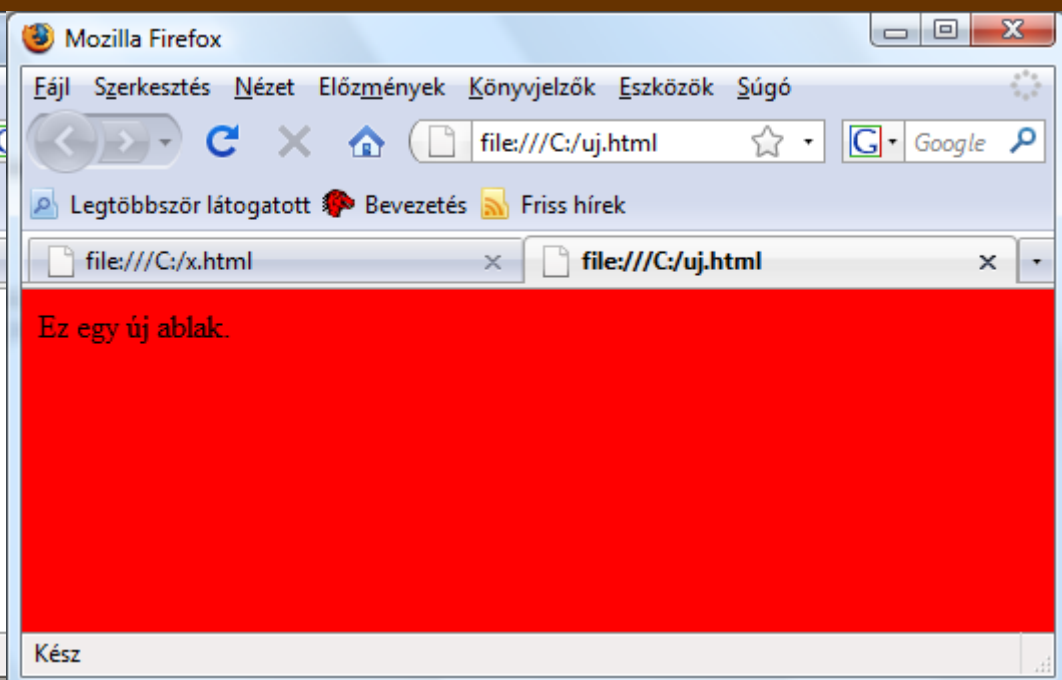
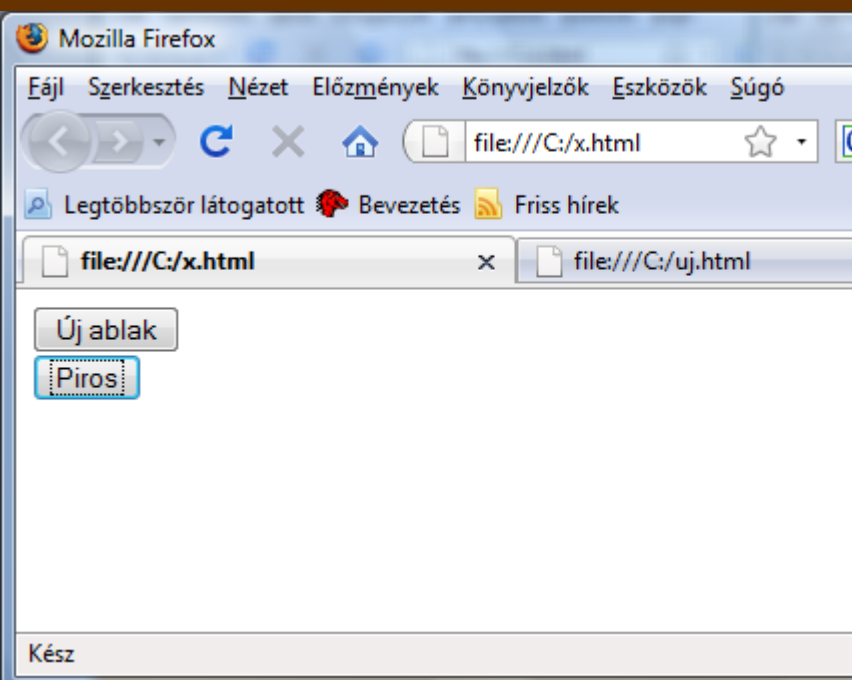
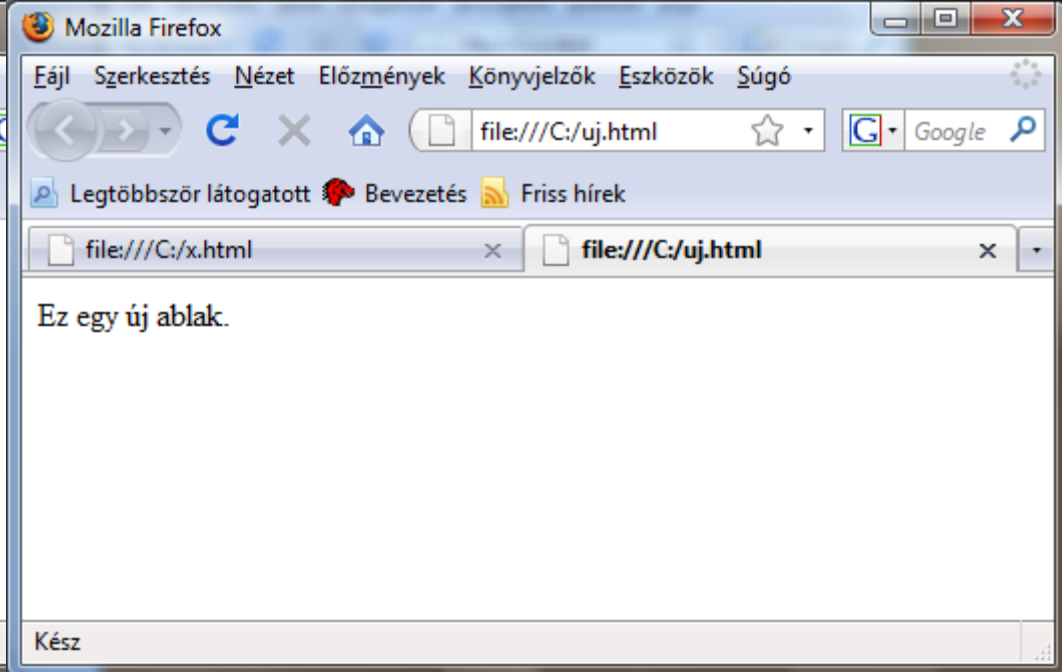
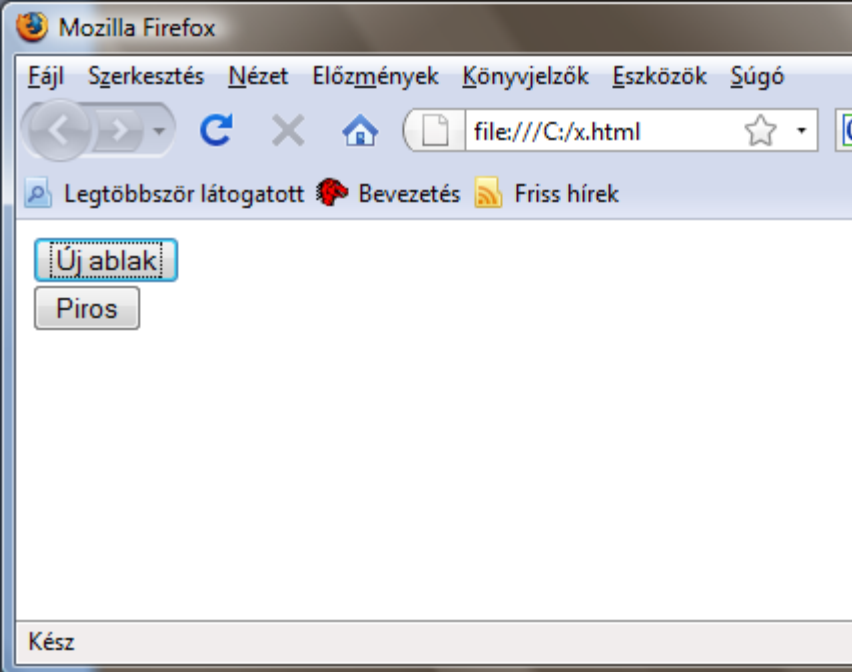
Tulajdonság	Beállítható érték	
directories	yes   no	katalógust eltünteteti ill. megjeleníti
height	a képpontok száma	az ablak magassága
location	yes   no	a címsort eltünteteti ill. megjeleníti
menubar	yes   no	eltünteteti ill. megjeleníti a menüsört
resizable	yes   no	átméretezhető legyen-e az ablak
scrollbars	yes   no	a gördítősávokat eltünteteti ill. megjeleníti
status	yes   no	a státuszsort eltünteteti ill. megjeleníti
toolbar	yes   no	eltünteteti ill. megjeleníti az eszközsört
width	a képpontok száma	az ablak szélessége

Tulajdonság	Beállítható érték	Mit állít?
alwaysLowered	yes   no	Az új ablak mindig a többi alatt helyezkedik el, ha értéke "yes"
alwaysRaised	yes   no	Az új ablak mindig a többi felett látszik, ha értéke "yes"
dependent	yes   no	Az új ablak a jelenlegi gyermekeként jön létre
hotkeys	yes   no	Ki- bekapcsolja a gyorsító billentyűk használatának lehetőségét
innerWidth	a képpontok száma	A régi width tulajdonságot váltja fel
innerHeight	a képpontok száma	A régi height tulajdonságot váltja fel
outerWidth	a képpontok száma	Az ablak külső határának szélességét határozza meg
outerHeight	a képpontok száma	Az ablak külső határának magasságát határozza meg
screenX	elhelyezkedés képpontokban mérve	Az ablak bal felső sarkának X koordinátája
screenY	elhelyezkedés képpontokban mérve	Az ablak bal felső sarkának Y koordinátája
titlebar	yes   no	Megjeleníti a címsort
z-lock	yes   no	Ha értéke "yes", akkor egy olyan ablakot kapunk, amely nem emelkedik a többi fölé, ha ő lesz az aktiv ablak



# Feladat

- Készítsünk olyan weblapot, amelyen egy nyomógomb megnyit egy másik ablakot, egy weblappal, és az első weblapon van olyan nyomógomb, amely utólag tudja módosítani a másik ablak háttérszínét!



# Uj.html

```
<HTML>
```

```
<HEAD>
```

```
</HEAD>
```

```
<BODY>
```

Ez egy új ablak.

```
</BODY>
```

```
</HTML>
```

```
<FORM NAME="urlap">
```

```
  <INPUT TYPE="button" VALUE="Új  
ablak" onClick="ujablak()"><BR>
```

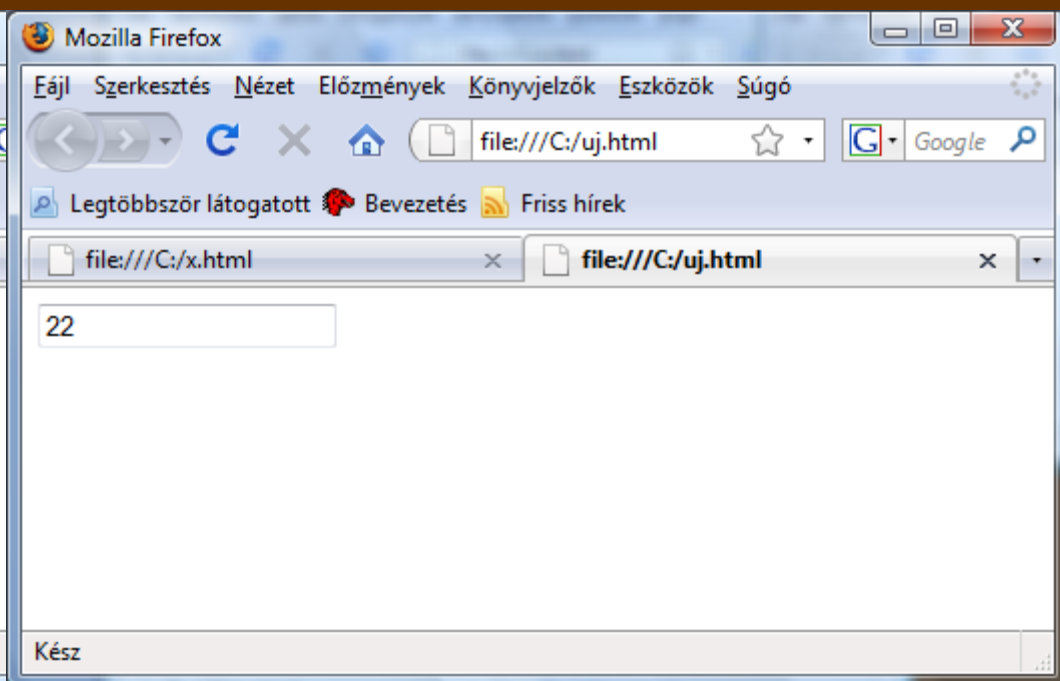
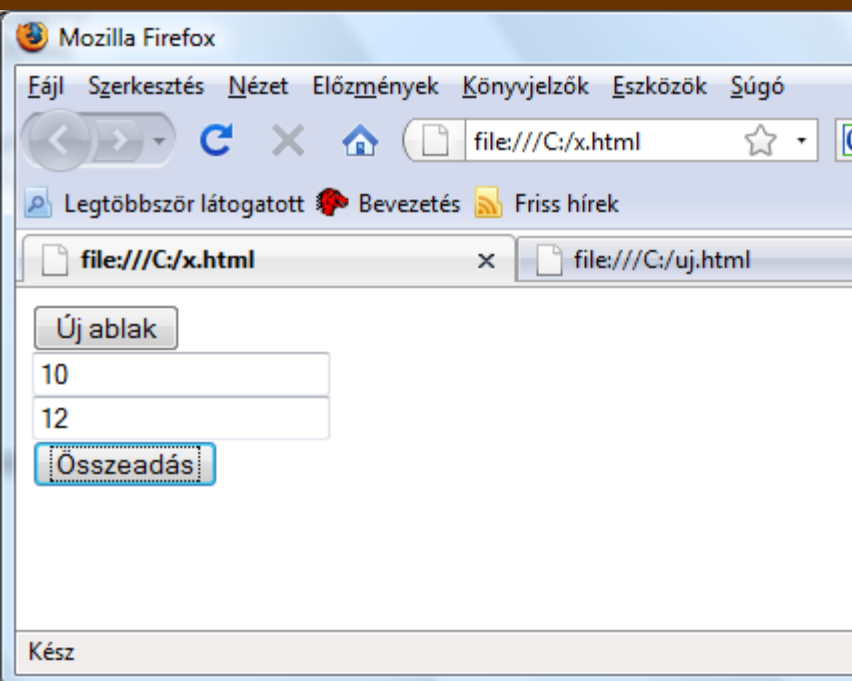
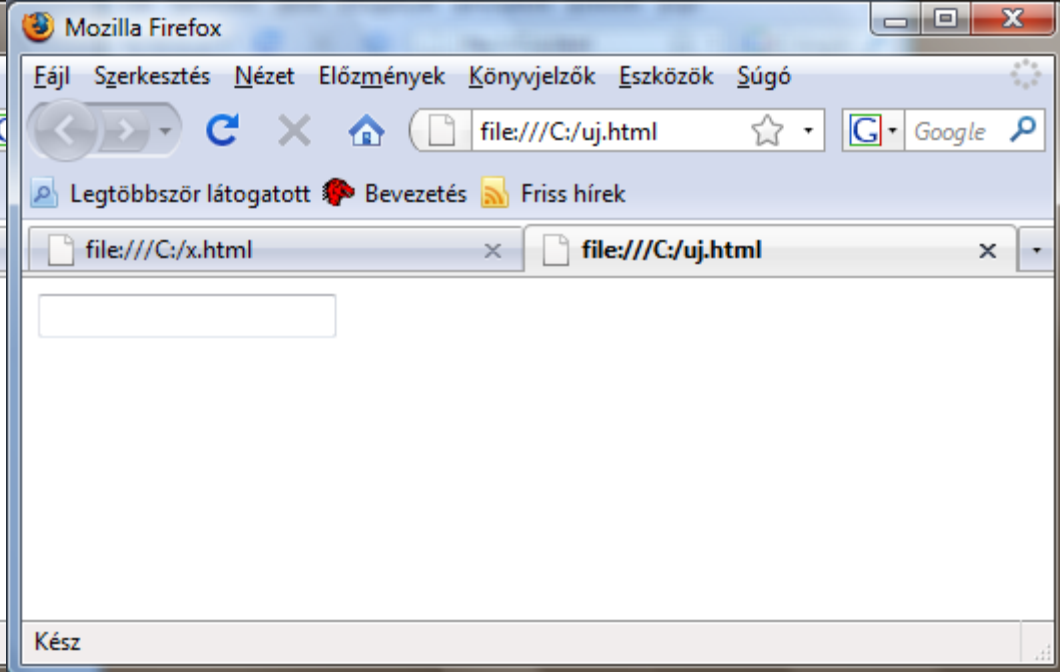
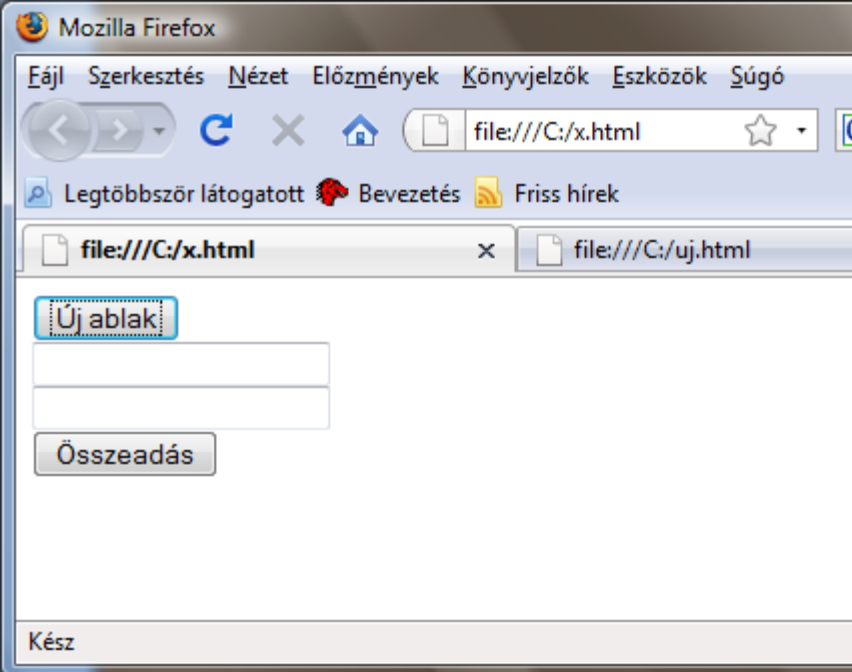
```
  <INPUT TYPE="button" VALUE="Piros"  
onClick="piros()"><BR>
```

```
</FORM>
```

```
<SCRIPT LANGUAGE="JavaScript">  
  var ablak;  
  function ujablak(){  
    ablak = open("uj.html");  
  }  
  function piros(){  
    ablak.document.bgColor = "red";  
  }  
</SCRIPT>
```

# Feladat

- Készítsünk olyan weblapot, amely bekér két számot egy-egy szövegbeviteli mezőben, majd az összegüket egy új ablakon írja ki!



# uj.html

```
<HTML>
```

```
<HEAD>
```

```
</HEAD>
```

```
<BODY>
```

```
<FORM NAME="urlapuj">
```

```
<INPUT TYPE="text" NAME="ossz">
```

```
</FORM>
```

```
</BODY>
```

```
</HTML>
```



```
<FORM NAME="urlap">  
  <INPUT TYPE="button" VALUE="Új  
ablak" onClick="ujablak()"><BR>  
  <INPUT TYPE="text" NAME="a"><BR>  
  <INPUT TYPE="text" NAME="b"><BR>  
  <INPUT TYPE="button"  
  VALUE="Összeadás"  
  onClick="osszead()"><BR>  
</FORM>
```

```
<SCRIPT LANGUAGE="JavaScript">
  var ablak;
  function ujablak(){
    ablak = open("uj.html");
  }
  function osszead(){
    var x = document.urlap.a.value*1;
    var y = document.urlap.b.value*1;
    z = x + y;
    ablak.document.urlapuj.ossz.value=z;
  }
</SCRIPT>
```

# Ablak bezárása

- Az új ablakot egészítsük ki egy bezárás gombbal, ami zárja is be az ablakot!

```
<FORM NAME="urlapuj">  
<INPUT TYPE="text" NAME="ossz"><BR>  
<INPUT TYPE="button" VALUE="Bezárás"  
  onClick="bezar()">  
</FORM>
```

- <SCRIPT LANGUAGE="JavaScript">
- function bezar(){
- **close();**
- }
- </SCRIPT>

# Feladat

- A kezdőlapot is egészítsük ki egy nyomógommbal, ami be tudja zárni az újonnan megnyitott ablakot!

```
<INPUT TYPE="button" VALUE="Új ablak  
bezárása" onClick="ujablakbezar()"><BR>
```

```
function ujablakbezar(){  
    ablak.close();  
}
```

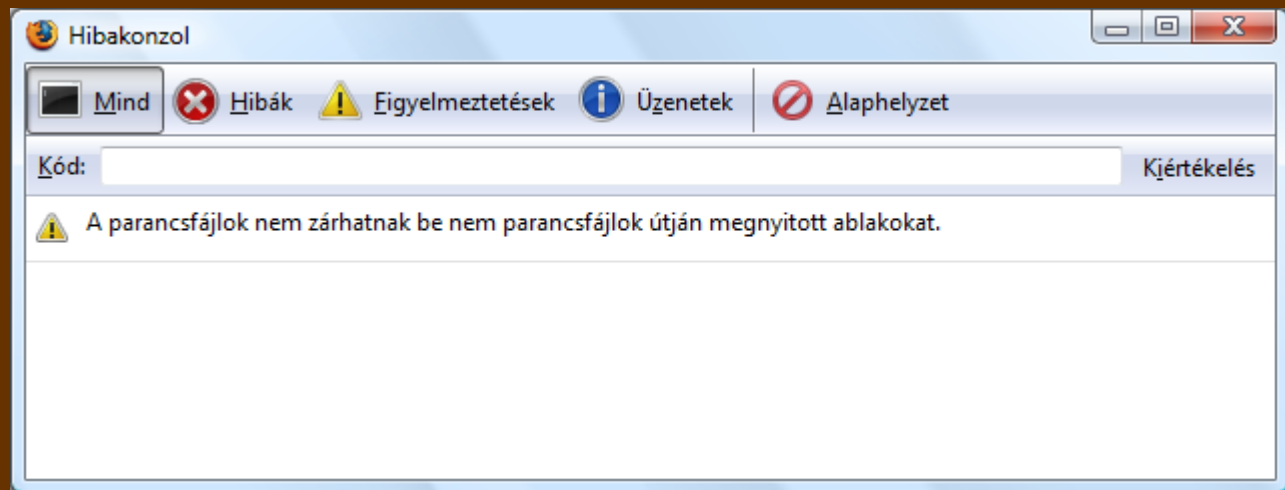
# Feladat !

- Készítsünk a kezdőlapon egy olyan gombot, ami bezárja a kezdőlapot!



```
<INPUT TYPE="button" VALUE="Bezárás"  
onClick="bezar()"><BR>
```

```
function bezar(){  
    close();  
}
```



# A Timeout-ok (késleltetés)

- Készítsünk olyan weblapot, amelynek háttérszínét egy nyomógomb 5 másodpercre képes lecserélni!

```
<FORM NAME="urlap">  
  <INPUT TYPE="button"  
  VALUE="Színváltás"  
  onClick="idozit()"><BR>  
</FORM>
```

```
<SCRIPT LANGUAGE="JavaScript">  
  function szin(){  
    document.bgColor = "white";  
  }  
  function idozit(){  
    document.bgColor = "blue";  
    setTimeout("szin()",5000);  
  }  
</SCRIPT>
```

# A JavaScript objektumai

- Array (tömb)
- Boolean
- Date (dátum)
- Function (függvény)
- Math (matematikai)
- Number (szám)
- String (karakterlánc)

# Array (tömb) objektum

- sok változóból felépülő összetett adattípus
- Az egyes adatokhoz egy név (a tömb neve) és egy szám segítségével férhetünk hozzá
- **tomb = new Array()**
  - tomb[0] = "JavaScript";
  - tomb[1] = "1998";
  - tomb[2] = "Aurum";

# Egyéb létrehozási módok

- `tombnev = new Array([a_tomb_hossza])`
- `tombnev = new Array([1.elem, 2.elem, 3.elem, ... ,n.elem])`

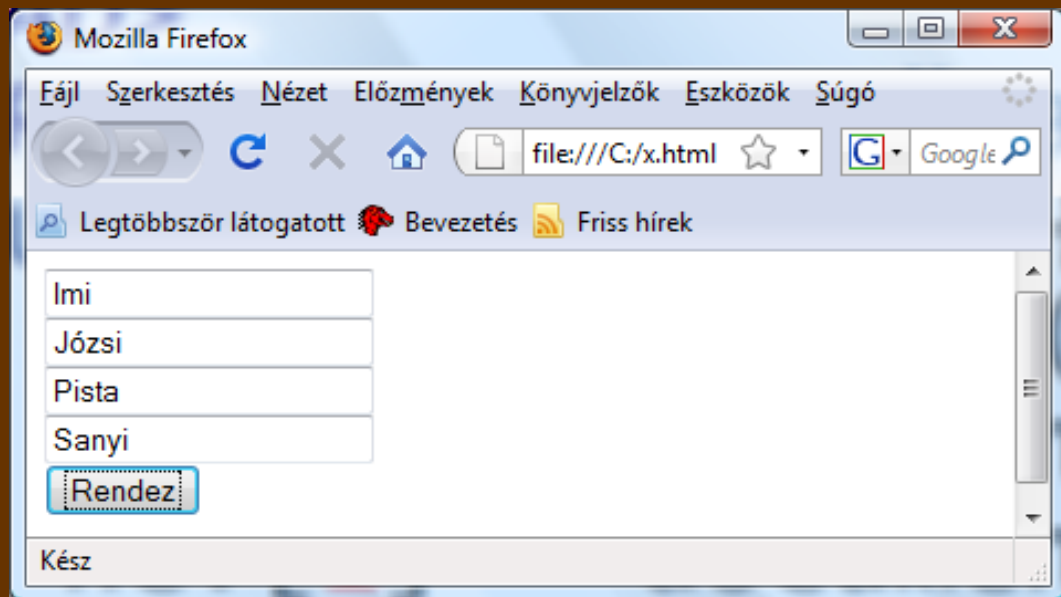
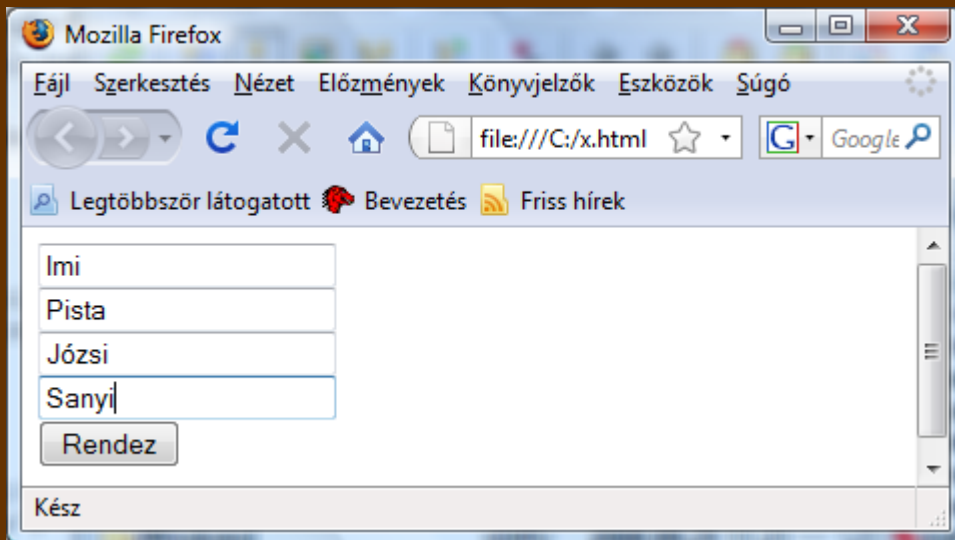
# Az Array objektum legfontosabb metódusai

- a **join** metódus összefűzi a tömb elemeit egyetlen sztringgá
- a **reverse** megfordítja (transzponálja) a tömb elemeit, az utolsóból lesz az első, az elsőből az utolsó
- a **sort** rendezi a tömb elemeit



# Feladat

- Készítsünk olyan weblapot, amely négy szövegbeviteli mezőbe bekér neveket, majd sorba rendezve írja őket ki!



```
<FORM NAME="urlap">  
  <INPUT TYPE="text" NAME="szoveg0"><BR>  
  <INPUT TYPE="text" NAME="szoveg1"><BR>  
  <INPUT TYPE="text" NAME="szoveg2"><BR>  
  <INPUT TYPE="text" NAME="szoveg3"><BR>  
  <INPUT TYPE="button" VALUE="Rendez"  
  onClick="rendez()"><BR>  
</FORM>
```

```
<SCRIPT LANGUAGE="JavaScript">
function rendez(){
    nev = new Array([4]);
    nev[0] = document.urlap.szoveg0.value;
    nev[1] = document.urlap.szoveg1.value;
    nev[2] = document.urlap.szoveg2.value;
    nev[3] = document.urlap.szoveg3.value;
    nev.sort();
    document.urlap.szoveg0.value = nev[0];
    document.urlap.szoveg1.value = nev[1];
    document.urlap.szoveg2.value = nev[2];
    document.urlap.szoveg3.value = nev[3];
}
</SCRIPT>
```

# A Date (dátum) objektum

- Segítségével olyan alkalmazásokat írhatunk, amelyek idő- vagy dátumértékeket kezelnek
- dátum\_objektum\_neve = new Date([paraméterek] );
- az aktuális dátum és idő értéke
- **ma = new Date();**

# Feladat

- Egészítsük ki az előző weblapot úgy, hogy megjelenítse az aktuális dátumot, és időt!

```
most = new Date();  
document.write("Dátum: "+most.getFullYear()+"-  
    "+(1+most.getMonth())+"-  
    "+most.getDate()+"<BR>");  
document.write("Idő:  
    "+most.getHours()+":"+most.getMinutes()+  
    ":"+most.getSeconds());
```

# Feladat

- Készítsünk olyan weblapot, amely két szövegbeviteli mezőben mutatja az aktuális dátumot és időt!



```
<BODY onLoad="ido()">
```

```
<FORM NAME="digiora">
```

```
Idő: <INPUT TYPE="text" NAME="ido"  
      SIZE="8" VALUE=""><BR>
```

```
Dátum: <INPUT TYPE="text"  
        NAME="datum" SIZE="10"  
        VALUE=""><BR>
```

```
</FORM>
```

```
</BODY>
```

Ha az onLoad eljárásba open metódust teszünk: popup ablak

```
<BODY onLoad="open('http://szerver2.lacszki.sulinet.hu')">
```

```
<SCRIPT LANGUAGE="JavaScript">
var idoStr, datumStr;
function ido(){
most = new Date();
ora  = most.getHours();
perc = most.getMinutes();
mp   = most.getSeconds();
ev   = most.getYear();
honap= most.getMonth()+1;
nap  = most.getDate();
idoStr = ora;
idoStr += ((perc < 10) ? ":0" : ":") + perc;
idoStr += ((mp < 10 ) ? ":0" : ":") + mp;
datumStr = 1900 + ev;
datumStr += ((honap < 10) ? "-0" : "-") + honap;
datumStr += ((nap < 10) ? "-0" : "-") + nap;
document.digiora.ido.value = idoStr;
document.digiora.datum.value = datumStr;
setTimeout("ido()",1000);  }
</SCRIPT>
```

# A Math (matematikai) objektum

- matematikai függvényeket és konstansokat tartalmaz
- `kor_kerulete = 2*sugar*Math.PI;`
- `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	alsó 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

# String objektum

- JavaScript-ben nem létezik sztring adattípus
- `String_objektum_neve = new String(sztring);`

Metódus	Leírás
anchor	HTML hivatkozást készít a sztringből
big, blink, bold, fixed, italics, small, strike, sub, sup	HTML-ként formázott sztringet hoz létre
charAt	a paraméterként átadott pozícióban lévő karakterrel tér vissza
indexOf, lastIndexOf	A paraméterben meghatározott részsstring 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 részsstringekre 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

# Példák

Az eredeti sztring: JavaScript

szoveg.length = 10

szoveg.substring(3,8) = aScri

szoveg.toUpperCase() = JAVASCRIPT

szoveg.toLowerCase() = javascript

szoveg.charAt(0) = J

szoveg.charAt(4) = S

szoveg.indexOf('S') = 4

szoveg.indexOf('s') = -1

szoveg.lastIndexOf('S') = 4

szoveg.lastIndexOf('s') = -1

# Példák

- szoveg.link('http://www.netscape.com') = JavaScript  
szoveg.big() = JavaScript  
szoveg.blink() = JavaScript (villog)  
szoveg.bold() = **JavaScript**  
szoveg.fixed() = JavaScript  
szoveg.italics() = *JavaScript*  
szoveg.small() = JavaScript  
szoveg.strike() = ~~JavaScript~~  
szoveg.sub() = <sub>JavaScript</sub>  
szoveg.sup() = <sup>JavaScript</sup>

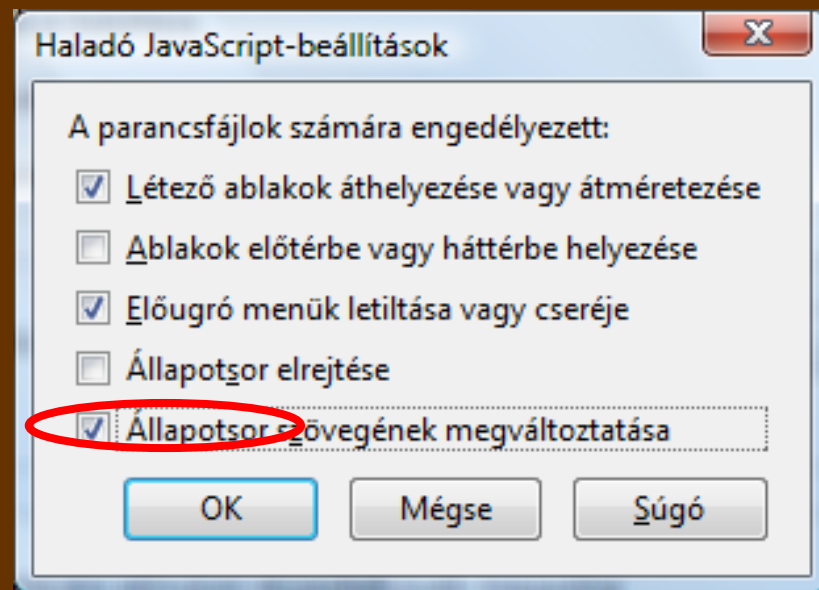
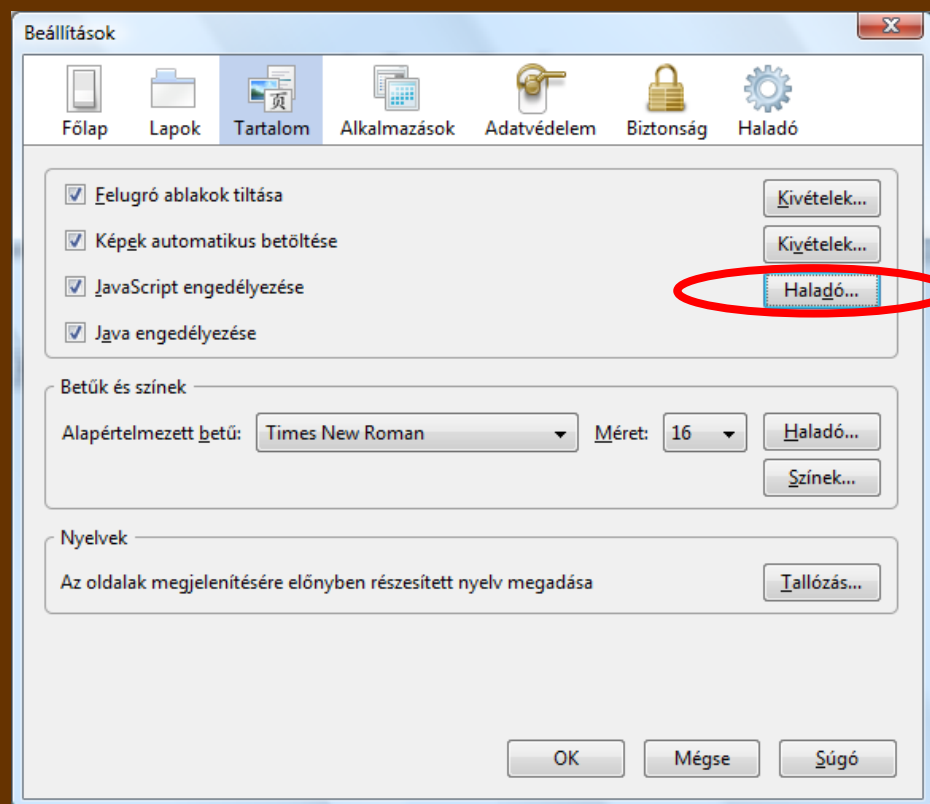


# Az állapotsor

- A JavaScript programok képesek írni az állapotsorra, amely a böngésző alján található
- Nem kell mást tenni, mint hozzárendelni egy sztringet a **window.status**-hoz (azaz a **window** objektum **status** mezőjéhez)
- Egyes böngészőkben ezt a lehetőséget engedélyezni kell.

# Engedélyezés

- Mozilla Firefox
- Eszközök > Beállítások > Tartalom



# Feladat

- Készítsünk olyan weblapot, amely két bekért szám összegét az állapotsorra írja ki!



Mozilla Firefox



Fájl Szerkesztés Nézet Előzmények Könyvjelzők Eszközök Súgó



file:///C:/x.html



Google



Legtöbbször látogatott



Bevezetés



Friss hírek

20

23

Összead

43

```
<FORM NAME="szamolo">  
  <INPUT TYPE="text" NAME="a"><BR>  
  <INPUT TYPE="text" NAME="b"><BR>  
  <INPUT TYPE="button" VALUE="Összead"  
    onClick="ossze()"><BR>  
</FORM>
```

- `<SCRIPT LANGUAGE="JavaScript">`
- `function ossze(){`
- `var x = document.szamolo.a.value*1;`
- `var y = document.szamolo.b.value*1;`
- `var z = x + y;`
- `window.status = z;`
- `}`
- `</SCRIPT>`

# Űrlapok elküldése levélben

- Valójában ez nem igényel egyetlen JavaScript utasítást sem
- tisztán megoldható HTML segítségével
- `<FORM METHOD="post" ACTION="mailto:e-mail@cim" ENCTYPE="text/plain">`
- ... Itt az űrlap mezőinek definiálása következik ...
- `</FORM>`

# Feladat

- Készítsünk olyan űrlapot, amelynek tartalma e-mailben elküldhető!



Mozilla Firefox

Fájl Szerkesztés Nézet Előzmények Könyvjelzők Eszközök Súgó

file:/// Goo

Legtöbbször látogatott Bevezetés Friss hírek

Neved?  
x

Hány éves vagy?  
10

Telefonszám:  
200

Az űrlap a következő e-mail címre lesz elküldve...  
laci@x.y.hu

Elküld

Kész

Úrlapküldés a Firefoxból

Fájl Szerkesztés Nézet Beszúrás Formátum Eszközök >>

Küldés

Címzett: laci@x.y.hu <laci@x.y.hu>;

Másolatot kap:

Tárgy: Úrlapküldés a Firefoxból

≡ B I U >>

nev=x  
ev=10  
telefon=200  
[email=laci@x.y.hu](#)  
submit=Elküld

```
<FORM NAME="form" METHOD="post"  
  ENCTYPE="text/plain" onSubmit="kuld()">
```

Neved?<BR>

```
<INPUT TYPE="text" NAME="nev" SIZE="20">
```

```
<P>
```

Hány éves vagy?<BR>

```
<INPUT TYPE="text" NAME="ev" SIZE="20">
```

```
<P>
```

Telefonszám:<BR>

```
<INPUT TYPE="text" NAME="telefon" SIZE="20">
```

```
<P>
```

Az űrlap a következő e-mail címre lesz elküldve...<BR>

```
<INPUT TYPE="text" NAME="email" SIZE="20">
```

```
<P>
```

```
<INPUT TYPE="submit" NAME="submit" VALUE="Elküld">
```

```
</FORM>
```

```
function kuld(){  
    document.form.action =  
        "mailto:"+document.form.email.value;  
}
```

# A képek kezelése

- A képeket a JavaScript **Image** (kép) objektumán keresztül kezelhetjük
- Az **Image** objektum segítségével megváltoztathatjuk az oldalunkon lévő képek paramétereit, illetve magát a képet is
- Minden képet az **Image** objektum egy példányának tekintünk

# A képek tulajdonságai

- **name** - a kép neve
- **src** - a kép helye, az a cím, ahonnan betöltjük
- **width** - szélessége (képpontokban)
- **height** - magassága (képpontokban)
- **border** - keret mérete képpontokban

# Feladat

- Készítsünk programot, amely egy weblapon lévő képet gombnyomásra másik képre cserél!



```
<BODY>
```

```
<IMG SRC="elso.jpg" NAME="elso">
```

```
<FORM>
```

```
<INPUT TYPE="button" VALUE="Kép  
cseréje" onClick="csere()">
```

```
</FORM>
```

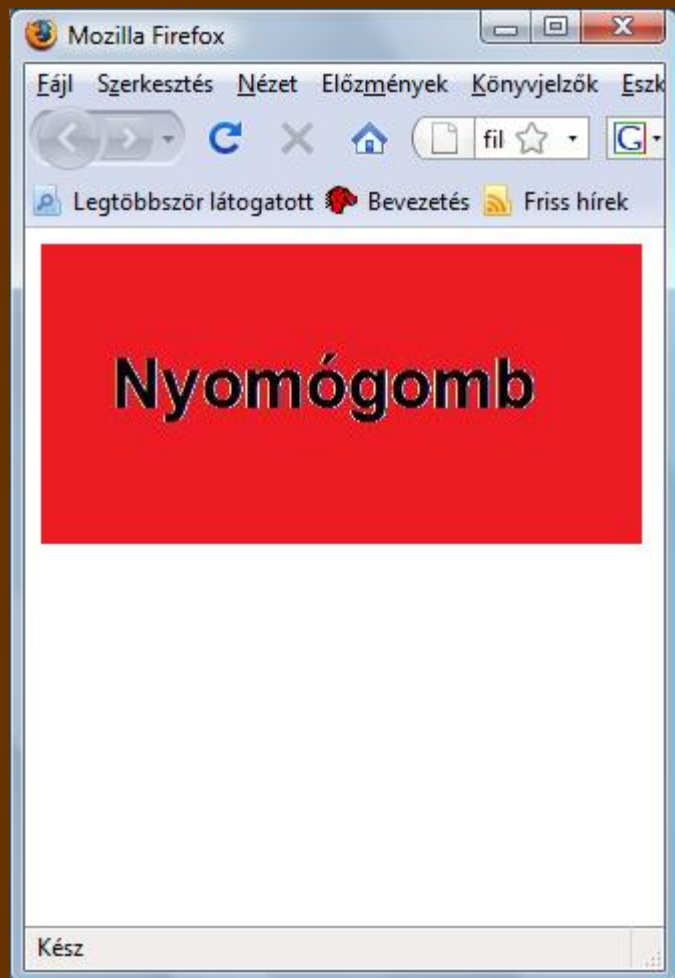
```
</BODY>
```



```
<SCRIPT LANGUAGE="JavaScript">  
  var i=1;  
  function csere(){  
    if (i == 1){  
      document.images.else.src = "masodik.jpg";  
      i=0;  
    } else {  
      document.images.else.src = "else.jpg";  
      i=1;  
    }  
  }  
</SCRIPT>
```

# Feladat

- Készítsünk olyan nyomógombot, amely abban az esetben, ha az egeret fölé visszük, megváltozik! A nyomógomb legyen olyan hivatkozás, amelyik az iskola honlapjára mutat!



```
<HTML>
```

```
<HEAD>
```

```
</HEAD>
```

```
<BODY>
```

```
<A HREF="http://szerver2.lacszki.sulinet.hu"  
  onMouseOver="document.kep.src='gomb2.jpg'"
```

```
  onMouseOut="document.kep.src='gomb1.jpg'">
```

```
<IMG SRC="gomb1.jpg" NAME="kep"  
  border=0></A>
```

```
</BODY>
```

```
</HTML>
```

# Sütik

- cookies
- a HyperText Transfer Protocol-t (HTTP) nem alkalmas arra, hogy kapcsolatot teremtsen két felhasználói (kliens) kérés között
- Tegyük fel például, hogy egy olyan oldalt szeretnénk készíteni, amely felismeri azt, hogy ha valaki már nem az első alkalommal látogatja meg, és ismerve azt, hogy előzőleg milyen témákat tartalmazó oldalakat járt be az illető, most máshová küldi

# Sütik

- A sütik segítségével adatokat tárolhatunk a felhasználó gépén
- **cookies.txt** állományban megjelenő szöveges bejegyzések
- Formátumuk a következő:
- *név=érték;expires=dátum;*

# Megszorítások

- legfeljebb 300 ilyen bejegyzés lehet a **cookies.txt** file-ban
- egy bejegyzés mérete legfeljebb 4Kb
- Szerverenként vagy domain-enként legfeljebb 20 "süti" lehet

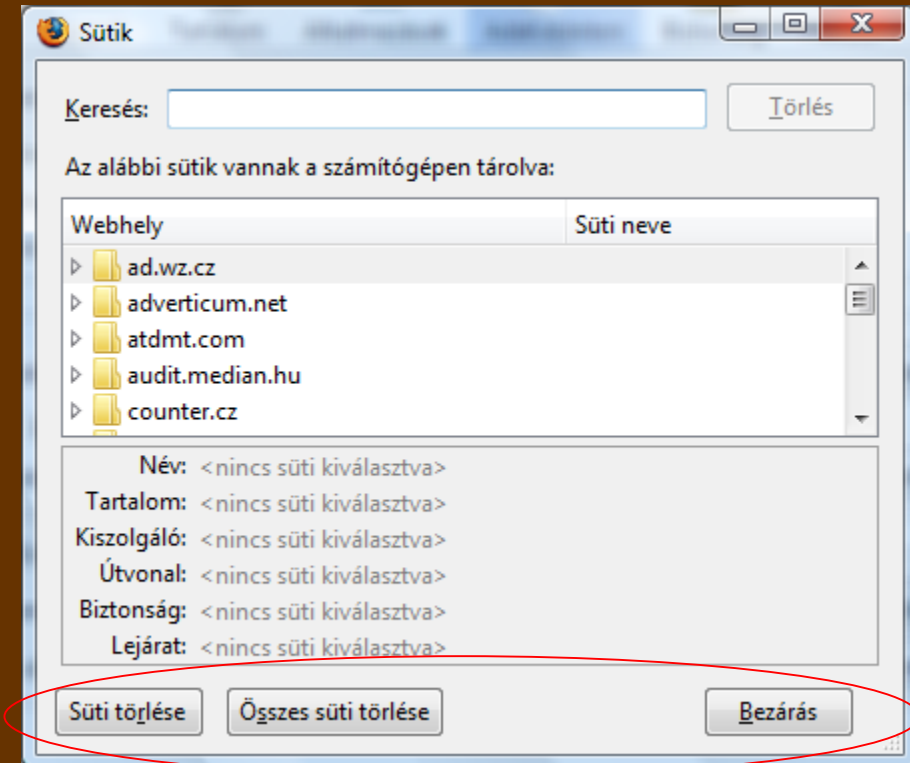
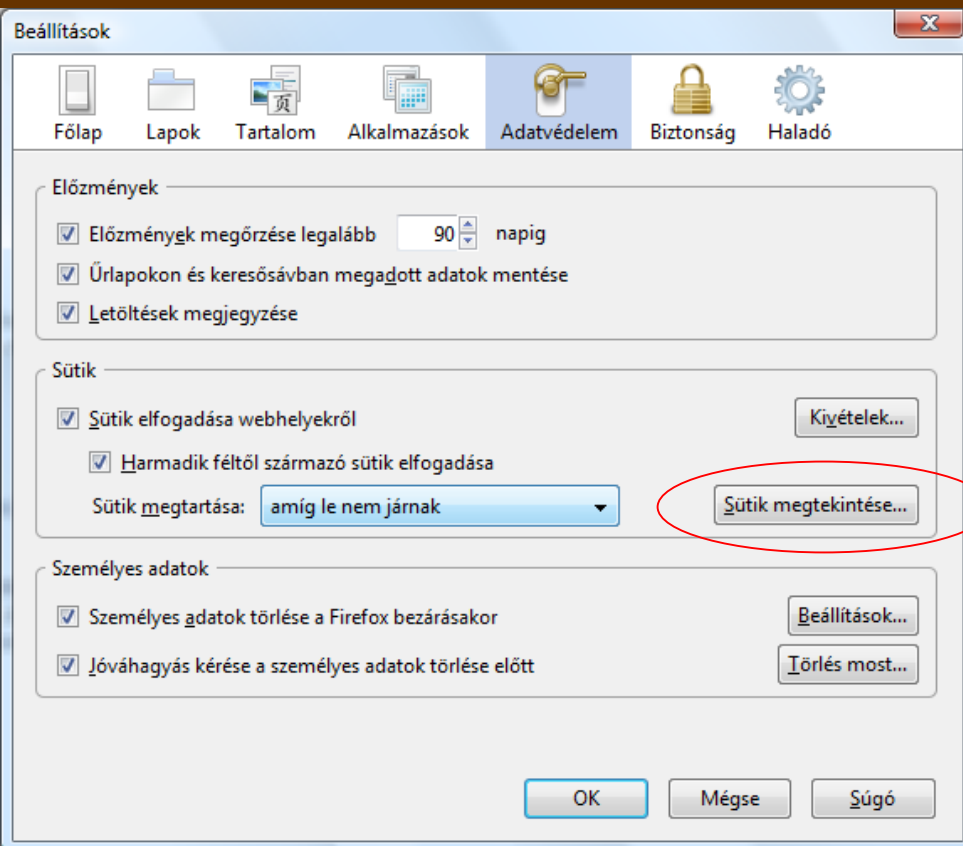
# Sütik

- A **dokument** objektum **cookies** mezője egy sztring
- A "sütikkel" a következőket tehetjük:
- beállíthatjuk az értékét illetve az érvényességi idejét
- megkaphatjuk az értékét a megadott nevű "sütinek"



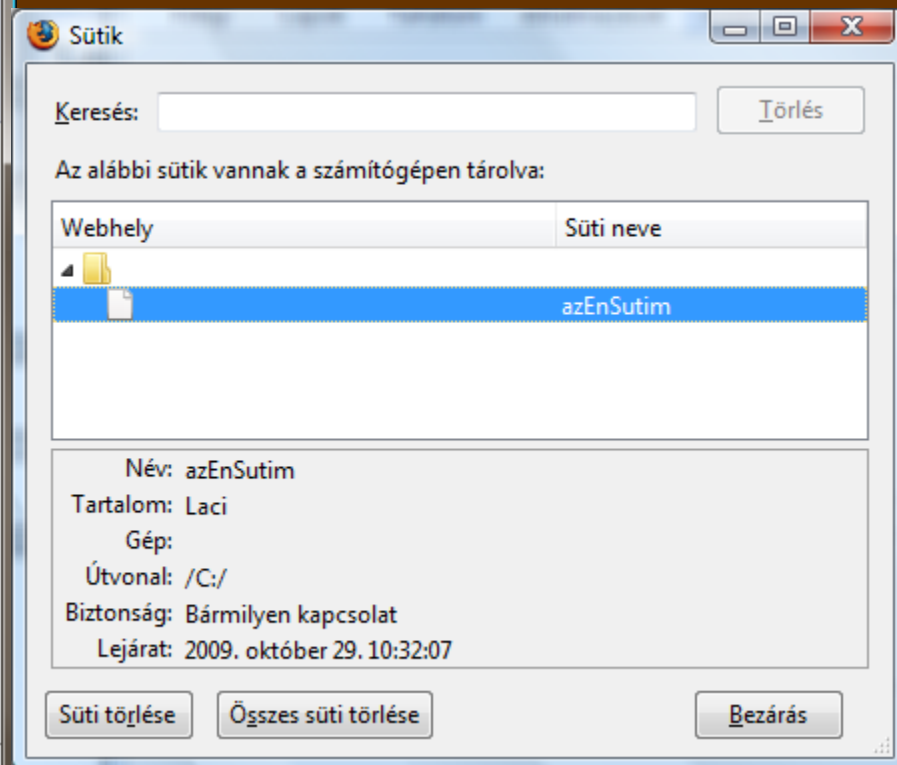
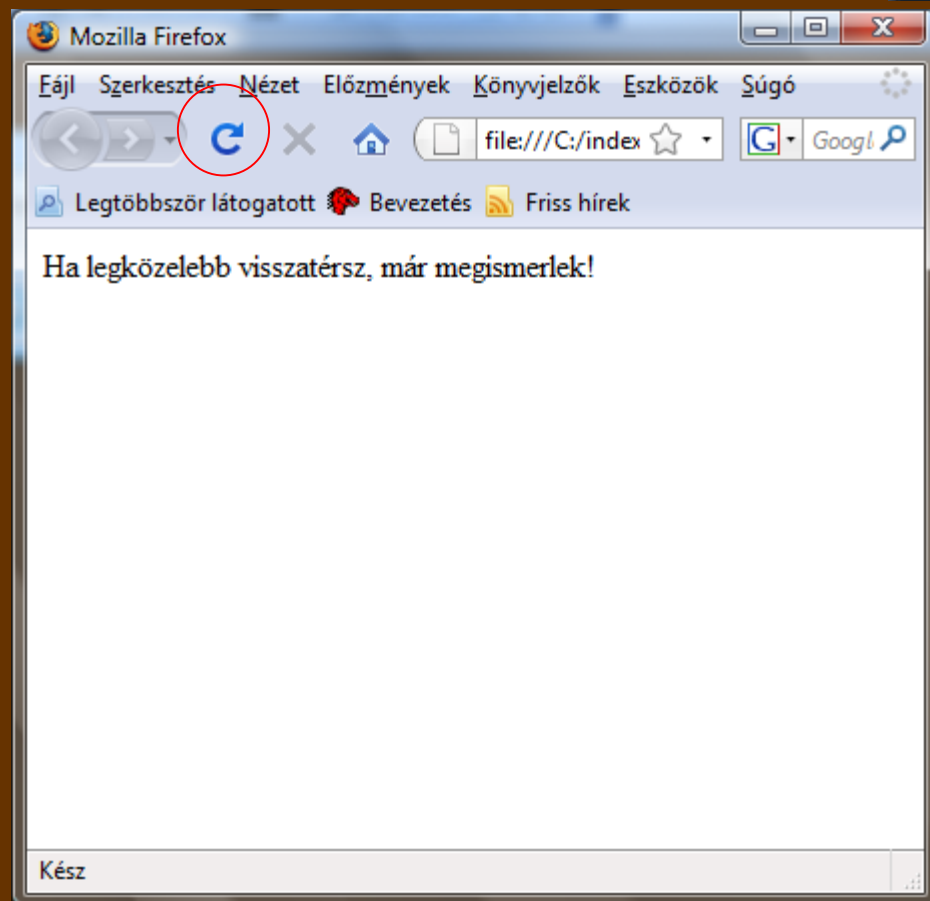
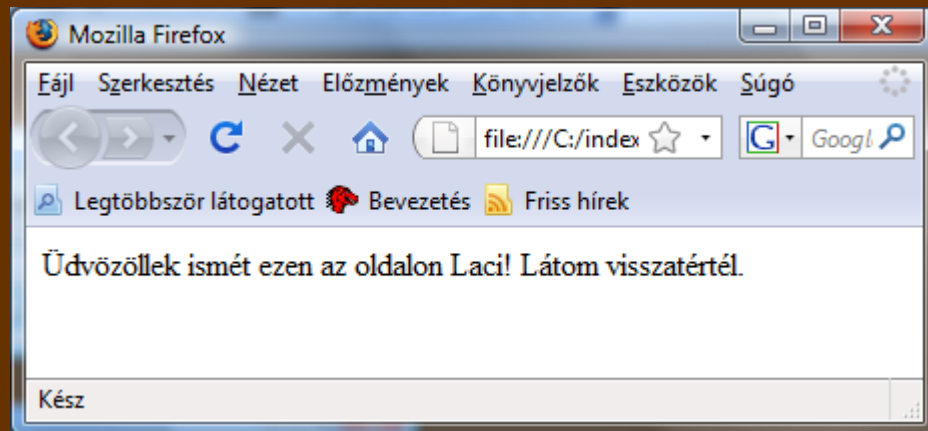
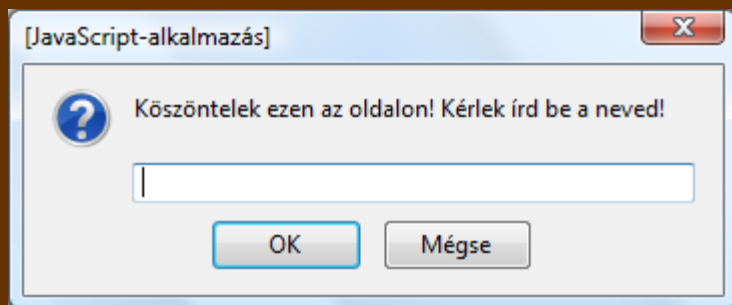
# A sütik megtekintése

- Mozilla Firefox



# Feladat

- Készítsünk olyan weblapot, amely bekér egy felhasználói nevet, és a következő megnyitáskor már ezen a néven üdvözlöl minket!



```
<HTML>
```

```
<HEAD>
```

```
<SCRIPT LANGUAGE="JavaScript">
```

```
function setCookie(name, value, expire) {  
    document.cookie = name + "=" + escape(value) + ((expire == null) ? "" : ("; expires=" +  
    + expire.toGMTString()));  
}
```

```
function getCookie(Name){  
    var search = Name + "=";  
    if (document.cookie.length > 0){  
        offset = document.cookie.indexOf(search);  
        if (offset != -1){  
            offset += search.length;  
            end = document.cookie.indexOf(";", offset);  
            if (end == -1) end = document.cookie.length;  
            return unescape(document.cookie.substring(offset, end));  
        }  
    }  
}
```

```
</SCRIPT>
```

```
</HEAD>
```

```
<BODY>
<SCRIPT LANGUAGE="JavaScript">
  var user = getCookie("azEnSutim");
  if (user == null){
    user = prompt("Köszöntelek ezen az oldalon! Kérlek írd be a neved!", "");
    if (user == null){
      alert("Nem írtad be a neved ezért a továbbiakban          \"Fantomnak\"
nevezlek!");
      user = "Fantom";
    }
    expDate = new Date();
    expDate.setTime(expDate.getTime() +      (365*24*60*60*1000));
    setCookie('azEnSutim',user,expDate);
    document.write("Ha legközelebb visszatérsz, már megismerlek!");
  } else{
    document.write("Üdvözöllek ismét ezen az oldalon "+user+"! Látom
visszatértél."); }
</SCRIPT>
</BODY>
</HTML>
```

# Befejezés

- Sajnos a JavaScript is küszködik egy betegséggel, azzal, hogy az egyes böngészők nem teljesen egyformán kezelik a nyelvet.
- több böngészőben is teszteljük le az oldalainkat és az azokba ágyazott programokat!
- Sajnos amíg a legelterjedtebb böngészők egymással csatáznak, addig csak újabb és újabb eltérések várhatók

VÉGE