



G3.f: Bezeroaren aldeko teknologiak. (DOM). JavaScript

Rosa Arruabarrena, Xabier Arregi, Jose Ángel Vadillo
LSI, UPV/EHU

1



Motibazioa

- Aplikazio baten exekuzioa zerbitzariaren eta bezeroaren artean banatzeko ideia lantzea
- JavaScript aurkeztea WS irakasgaiko beharren baitan
- DOM ereduaren erabilera jorratzea
- Web aplikazioen garapenerako JavaScript lengoaiaren erabilera berezitu batzuk azaltzea eta lantzea

2

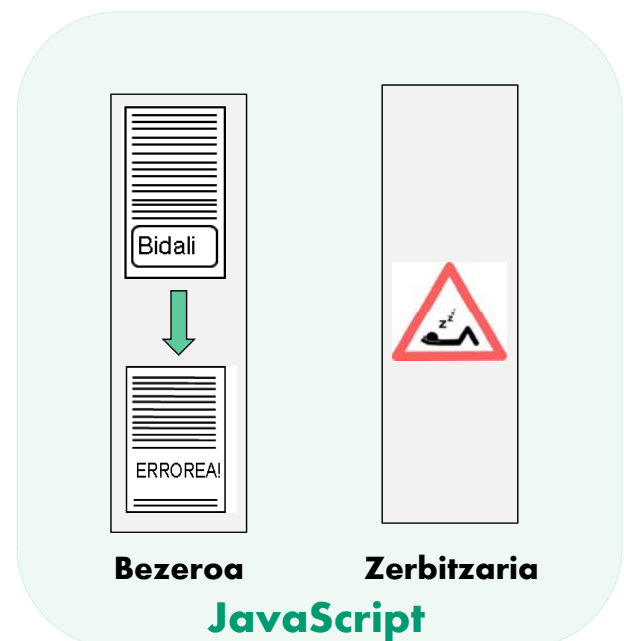
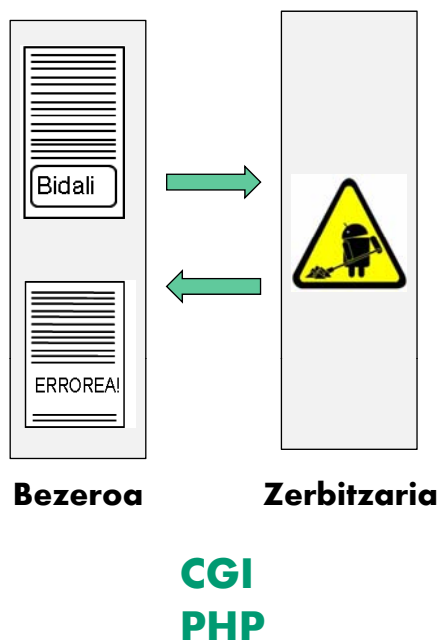


Bezeroan ala zerbitzarian

- Bezeroko script-ak
 - Nabigatzailean exekutatzen dira
 - Bezeroko baliabideak kudeatzen dira modu mugatuan
 - JavaScript, VisualBasicScript, java applets,...
- Zerbitzariko script-ak
 - Web zerbitzariak interpretatzen ditu
 - Bezerora dokumentu interpretatuak (HTML) bidaltzen dira
 - CGI, PHP, ASP, Python, Perl,...

3

Zerbitzariaren eta bezeroaren arteko lan-banaketa



4



Zer da JavaScript (JS)?

- Espektro zabaleko lengoaia da gaur egun, nahiz hasieran web orrien portaerarekin erlazionatua egon soilik; eta programatzailearen konpetentziaren arabera, programazioko kontzeptu eta teknika aurreratuenak ere pairatzen dituen.
- Programazio aurreratuen ezagutzari gabe erabiltzen hasi litekeen lengoaia da, baina aplikazio eta erabilpen ahaltsuenak programatzaile adituen gaitasunak behar ditu.
- Script lengoaia da; hots, euren programak interpretatuak direnak (ez konpilatuak).
- Master/Slave (Zerbitzari/bezero) web ereduari etekina ateratzen dio, web aplikazioaren exekuzioaren parte bat bezerora garraiatzea ahalbidetzen du (arakatzailera).
- JavaScript programak eraiki eta HTML orrietan barne daude.
- JavaScript-ik arakatzzailearen ezaugarriak atzi litezke bai eta HTML orrietan dauden objektuak ere DOM-aren API-a erabiliz.

5



JS. Zertarako?

- Web-orrien funtzionalitatea eta interakzio-gaitasuna areagotzeko. Adibidez:
 - Formularioetan erroreen detekzioari eta tratamenduari aurre egiteko
 - Egiaztapenen eta *Warning* mezuen bidezko elkarrekintza
- HTML orrietan objektu aktiboak sartzeko:
 - kalkulagailuak, kalkulu-orriak,...
- Orriaren aurkezpenean aldaketa dinamikoak egiteko:
 - menuak, dokumentuen eguneraketa zatika...
- URLak aztertu eta arakatutako URLen historia gordetzeko
- Leihoak maneiatzeko aukera
- Irudiak eta mapak maneiatzeko
- *Cookie*-ak erabiltzeko
-

6



JS. Historia

- NetScape-k web arakatzailleek applet-ak exekuta ahal izan ditzatela gauzatzen du.
- NetScape-k programazio-lengoaia bat garatzen hasten da: LiveScript. Honek, web orrietan programa txikiak sortzeko aukera ematen zuen.
- 1995en Sun garapen proiektura gehitzen denean, JavaScript izena hartzen du (merkatu estrategia izanik izen aldaketa)
 - Brendan Eich-ek diseinatu
- NetScape 2.0 JavaScript ulertzen zuen lehen nabigatzailea izan zen, eta ostean, atzetik, Microsoften beste nabigatzaile batzuk, 3.0 bertsiotik aurrera.

7



Java vs JavaScript

- Java eta JavaScript guztiz desberdinak diren lengoaiak dira. Besteak beste, JavaScript-ek
 - Garapena egiteko ez du KIT-ik behar
 - Ez da helburu orokorreko lengoaia
 - Motaketa ahula du
 - Erabat interpretatua da (Javan existitzen du itzulpena tarte koderak)



JavaScript. Ezaugarri orokorrak

- Script lengoaia interpretatua (ez konpilatua)
- Object-Based: objektuetan oinarritua
- Objektuen orientazioa: prototipatzea
- Motatze dinamikoa duena
- Lehen mailako funtzioak
- Multiparadigma: agintzailea, objektuen orientazioa, funtzionala
- Web-era orientatua,
 - (batez ere) Brendan Eich-ek bezzeroaren aldean
- Azken aldian lan-ingurune eta liburutegi asko garatu dira (AJAXek izan du eragina horretan), eta “Server-Side JavaScript” (SSJS) ere izugarri hedatu da

9



JavaScript. Ezaugarri orokorrak (II)

- JavaScript programak (=funtzioak)
 - web-orrietan txertatzen dira, edo
 - web-orriekin lotzen dira
- JavaScript lengoaiak erraztasunak ematen ditu arakatzailearen propietateak eta HTML dokumentuetan dauden objektuak atzitzeko, sortzeko eta manipulatzeko

10



Nola sartu JavaScript

- Non jar daitezke JavaScript script-ak HTML orrietan?
 - Nonahi...
 - Sarri `<head>` eta `</head>` etiketen artean

11



Nola sartu JavaScript (II)

- *Script* etiketa erabiliz
 - Zuzenean

```
<script language="JavaScript">  
    //hemen JavaScript-eko kodea  
</script>
```
 - Script-eko kodea aparteko fitxategian egon liteke eta hura erreferentziatuz

```
<script src="http:// .../izena.js"> </script>
```

12



Nola sartu JavaScript (III)

- *JavaScript* gidoiak HTML orrietan nola sartu?
 - Gertaera bati lotuta, objektu baten atributuari balioa emanez

```
<input type="password"
onclick="window.status='pasahitz berria'...>
```

- `<a>` etiketaren href atributuari balioa emanez:

```
<a href="javascript:window.history.back()">
```

13

JavaScript zatian gaude!

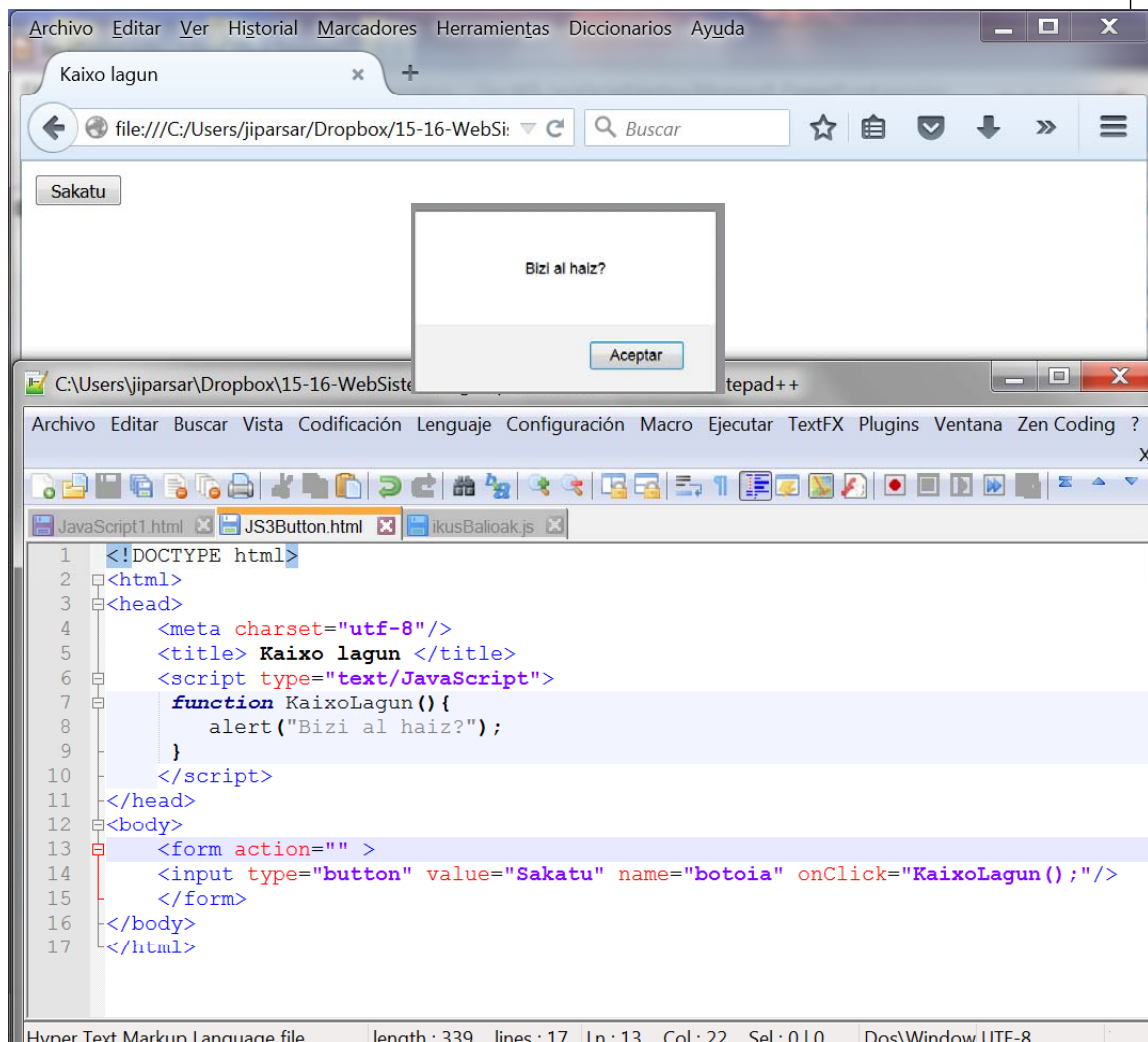
HTML orria + JS

scripting lengoaiekin probak egiten

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <meta charset="utf-8"/>
5   <title> JS probak </title>
6
7   <script type="text/javascript">
8     document.write("JavaScript zatian gaude!")
9   </script>
10
11 </head>
12 <body>
13   <h1> HTML orria + JS </h1>
14   scripting lengoaiekin probak egiten
15 </body>
16 </html>
```

length: 286 line: Ln: 2 Col: 7 Sel: 0 | 0 Dos\Window UTF-8

14



15

Script elementua

- Atributuak:
 - type:
 - beharrezkoa HTML 4n, ez ordea HTML5en.
 - Script-lengoaia adierazten du, MIME motaren bidez: text/javascript
 - src: script-a daukan fitxategiaren URLa adierazten du.
 - charset: Script-a kodetzeko erabiltzen den karaktere-multzoa.
 - defer: Boolearra da.
 - "defer" balioa hartzen du adierazteko script-a exekutatu dela orriaren analisia bukatutakoan.
 - async: Boolearra da.
 - **Berria HTML 5en.**
 - "async" balioa hartzen du adierazteko scripta modu asinkronoan exekutatu dela, eskuragarri dagoen bezain azkar.
 - Kanpoko scriptekin baino ezin daiteke erabili ("src" atributua jarri bada)

16



Script elementua. Oharrak

- `src` atributua jartzen bada, `<script>` elementuak hutsik egon behar du
- Kanpoko script-ak exekutatzeke moduak:
 - `async="async"` denean: script-a modu asinkronoan exekutatzen da gainontzeko orriarekiko
 - `async` jarri gabe eta `defer="defer"`: script-a exekutatzen da orriaren analisia bukatutakoan
 - `async` eta `defer` jarri gabe: script-a berehala exekutatzen da, arakatzailerak orriaren analisiarekin jarraitu baino lehen

17



JavaScript programen exekuzioaz

- Arakatzailerak interpretatzen dituzte.
- Besterik ezean `<script>` eta `</script>` etiketen arteko kodea:
 - Agertu ahala exekutatzen da
 - Exekuzioa HTML dokumentuaren analisiarekin batera egiten da, alegia, script-aren aurretik definitu ez diren elementuak ezin daitezke erreferentziatu.
 - Horregatik, kargatzean exekutatzen den kodeak ez luke denbora asko kontsumitu behar, script horien exekuzioa bukatu arte ezin baita dokumentua erakutsi.
 - Egokiagoa da, ahal denean, funtzioak definitzea eta funtzio horiei dei egitea gertaera-manegiatzailearen bitartez.
 - Etiketen artean dagoen kodea eta `src` atributuaren bitartez irakurtzen diren kanpo-fitxategiak, besterik ezean, berdin-berdin exekutatzen dira
 - Baina, kanpoko script-en exekuzio-modua eralda daiteke `async` eta `defer` atributuekin.

18



Egitura lexikala eta sintaxia

- Cren antzekoa da eta, beraz, Javaren antza ere badu
- Banatzaileak: ; (ez da derrigorrezkoa) eta { }
- Iruzkinak: // , /* */
- Maiuskulak/minuskulak bereizi egiten dira.
- Identifikadoreak:
- Lehenengo karaktereak letra bat, _, edo \$ izan behar du.
- Hortik aurrera digituak ere onartzen dira.
- Aldagaien asignazioa: = , -= , += , /= , *=
- Eragile erlazionalak: < , > , == , != , ...
- Eragile logikoak: && , || , !

19



Datu-motak

- Motatze-eredu ahula: aldagai berak mota desberdineko balioak har ditzake, eta lengoaiak berak egokitu egiten ditu balioak mota batetik bestera, komeni den bezala
- Zenbakizkoak: `var adina= 21, z=6.0;`
 - Osoak eta errealak berdin errepresentatzen dira
- Boolearrak: `erretzailea=true;`
- string edo karaktere-kateak: `izena="Pello"`
 - ' edo " karaktereak erabil daitezke string literalak mugatzeko
- Objektuak: Datu-mota generikoa. Aurrez definitutako edo erabiltzaileak sor ditzakeen datu-motak.
 - Objektuen orientazioa: eraikitzaileak, propietateak, metodoak.
 - Metodoak: `document.write("idatziko den testua");`
 - Propietateak: `image.width` edo `image["width"]`
 - Objektuen eta array asoziatiboan arteko parekotasuna

20



Datu-motak (II)

- Array-ak: elementuen bildumak.
 - Elementuak mota desberdinekoak izan daitezke
- Indizeak zerotik hasten dira
 - `document.images[1].width`
`// document objektuaren images`
`// propietatearen bigarren`
`// elementuaren width propietatea`
- Array-ak nola sortu: Array eraikitzailea
 - `var a = new Array();`
 - `var a = new Array(1.2, "aupa", true);`
 - `var a = new Array(3);`

21



Aldagaiak

- Aldagaiak erazagutu egiten dira (ez da derrigorrezkoa, baina)
 - `var` hitz erreserbatuaren bidez erazagutuz gero, aldagaiak iraunkorrak (permanent) *dira*.
 - Aldagai bat baino gehiago erazagut daitezke `var` gako-hitz bakarrarekin.
 - `var i, kont;`
- Erazagupenak eta hasieraketak batera egin daitezke.
 - `var i = 0, kont = 0;`
- Aldagaien esparrua:
 - Aldagai orokorrek esparru orokorra hartzen dute (kodeko edozein tokitan definituta dago).
 - Funtzioen baitan erazagututako aldagaiek eta funtzio-parametroek esparru lokala hartzen dute (funtzioaren gorputzean baizik ez daude definituta).
 - Funtzio batean, beraz, aldagai orokorrak eta lokalak erabil daitezke.
 - *Funtzio bateko aldagaia kanpoan ere erabil daiteke, var hitz erreserbatuarekin erazagutu ez bada.*
- `this` *aldagaia: uneko objektua adierazteko.*

22



Aldagaiak (II)

- Aldagaiak globalak

```
<script>
  var aldagaiGlobal
</script>;
```

- Eta lokak:

```
<script> function nireFun(){
  var aldagaiLokala ...}
</script>;
```

- Aldagaien erazagupenean ez da mota esleitzen
- Array motakoko aldagaiak defini litezke, elementu bakoitza objektu bat izanik

```
var nireTau = new Array(10)
nireTau[0]=2015
```

- `this` aldagaia: uneko objektua adierazteko.

23



Espressoak

- Espressoen ebaluazioa egitean, JavaScript saiatzen da beharrezko bihurketak egiten.
- Oro har: karaktere-kateak beste edozerekin "+" eragiketaren bidez lotzen direnean, dena karaktere-kate bihurtzen da.
 - Aldiz, "-", "*" edo "/" eragiketekin konbinatuz gero, zenbaki bihurtzen ditu, ahal bada.
 - Adb.:
 - "a" * "b" espresioa ez da legezkoa
 - "3" * "4" espresioa legezkoa da!, eta 12 zenbakia da bere balioa, ez "12" string-a.

24



Eragileak

- Aritmetikoak: +, -, *, / (0/0 NaN gisa ebaluatzen da), % (modulua), ++ (1 gehitu), -- (1 gutxitu)
- Berdintzak: == (berdintza), === (identitatea)
 - Zenbakiak, string-ak eta boolearrak balio bidez konparatzen dira
 - Objektuak, array-ak eta funtzioak erreferentzia bidez konparatzen dira
 - Mota desberdineko balioak berdinak izan daitezke, baina ez identikoak
 - Desberdintzak: !=, !==
- Erlazionalak: <, >, <=, >=
 - Eragigaiak edozein motatakoak izan daitezke, baina konparazioak bakarrik zenbakiekin eta string-ekin egiten dira. Beraz, eragigaien bihurketa egiten da, behar denean eta ahal denean
- String eragileak: + (kateamendua)
- Logikoak: && (and), || (or), ! (ukapena)
- Bit mailakoak: & (and), | (or), ^ (xor), ~ (not), << (ezkerrera), >> (eskuinera)
- Asignazioak: =, +=, -=, *=, /=, %= ...

25



Eragileak (II)

- Bestelakoak
 - Baldintzazko eragilea: ? :
 - Eragile hirutarra: baldintza ? bloke1 : bloke2
 - Adb.: x>0 ? x*y : -x*y
 - typeof: eragile bakuna, eragigaiaren mota string moduan itzultzen duena.
 - new: objektuak eraikitzeko eragilea.
 - delete: eragile bakuna, eragigai (objektu baten propietatea, array baten elementua edo aldagai bat) ezabatzen duena. true itzultzen du ezabaketa arrakastatsua izan bada, false bestela.
 - void: eragile bakuna, eragigaiaren ebaluazioak itzultzen duen balioa baztertzen duena.
 - Adb:
- Ireki leiho berria

26



Bestelakoak

- Badaude funtzio aurredefinituak, adibidez, elkarrizketa leihoetarako.

```
alert ("leiho hau ohar bat da");  
prompt ("leiho hau ohar bat da");  
confirm ("leiho hau ohar bat da");
```

- Nabigatzaileen elementuen gainean objektu-hierarkia bat dago definituta.

```
window.location.href = window.history.next;
```

27



Kontrol egiturak

- if...else
- while ... break ... continue
- for
- for in
- function ...return

28



Kontrol egiturak: baldintzazkoak

- **if...else**

```
if(baldintza){
    aginduak}
[ else{
    aginduak}]
```

```
(baldintza)?
    (agindu-bloke1):
    (agindu-bloke2)
```

- **switch**

```
switch(n) {
    case 1:    // 1 blokea exekutatu
                break;
    case 2:    // 2 blokea exekutatu
                break;
    ...
    default: // blokea exekutatu
                break;
}
```

29

Kontrol egiturak: iterazioak



- **while**

```
while(baldintza){
    aginduak
}
```

- **do...while**

```
do {
    aginduak
} while(baldintza)
```

- **break eta continue**

- Bigizta-etenak eragiten dituzte
- **break:** iterazio-agindutik irten
- **continue:** hurrengo begiztara pasa

30



Kontrol egiturak: iterazioak

- for

```
for (HasierakoEspresioa; Baldintza; Inkrementua){  
    aginduak  
}
```

- for...in

```
for (aldagaia in objektua) {  
    aginduak  
}
```

31

Kodearen egituraketa. Funtzioak



- Kodea funtzioen bidez egituratzen da.
- Funtzioetatik kanpo dagoen kodea orria kargatu ahala exekutatzen da.
- Funtzioetan argumentuak jartzen ahal dira.
- Funtzioek itzultzen duten balioaren mota ez da erazagutzen

```
function funtzio_izena (parametro_lista){  
    ...  
    [ return dena_delakoa; ]  
}
```

- Parametroak balioz pasatzen dira

32

JS kodeen adibidea:

Formularioak egiaztatzeko kodea



```
ikusBaliok.js  adFormularioaBalidatzen.html
1 <html>
2 <head>
3 <script language="javascript">
4     function egiaztatu() {
5         /*var izena = document.formulario.izena.value;
6         var taldea = document.formulario.taldea.value;*/
7         var izena=document.getElementById("izena").value;
8         var taldea=document.getElementById("taldea").value;
9         if (izena.length < 5 )
10            { alert("Errorea izenean"); return false; }
11         if (taldea < 11 || taldea > 33)
12            { alert("Errorea taldearen identifikazioan");
13              return false; }
14         return true;
15     }
16 </script>
17 </head>
18 <body>
19 <form action="http://ji.ehu.es/webs" method="get" id="formulario" onsubmit="return egiaztatu()">
20     Izena: <input type="text" id="izena" value=""> <br/> <br/>
21     Taldea: <input type="text" id="taldea" value="" size="3" maxlength="3"><br/> <br/>
22     <input type="submit" value="Bidali"><br/>
23 </body>
24 </html>
```

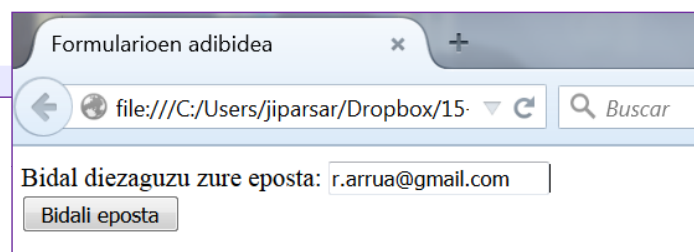
33

JS kodeen adibidea:

Formularioak maneiatzen



```
adEvents.html  aDFormularioakManeiatzen.html
1 <html>
2 <head>
3 <title>Formularioen adibidea</title>
4 <script language="JavaScript">
5     function egiaztatu(helbidea) {
6         if (helbidea.indexOf("@") != -1) return true;
7         else { alert("Helbide zuzen bat izan behar du");
8               return false; }
9     }
10 </script>
11 </head>
12 <body>
13 <form name="nireFormularioa" method="POST" action="mailto:r.arruabarrena@gmail.com"
14     enctype="text/plain" onSubmit="return egiaztatu(this.email.value)">
15     Bidal diezaguzu zure eposta: <input name="email" type="text"><br>
16     <input type="submit" value="Bidali eposta">
17 </form>
18 </body>
19 </html>
20
```



34

JS kodeen adibidea: Gertaeren manipulazioa



The screenshot shows a code editor with a file named `adEvents.html`. The code is as follows:

```
1 <html>
2 <head>
3 <script language="JavaScript">
4   <!-- JavaScript ez duten nabigatzaileentzat kodea ezkutuan uzten dute iruzkinek
5   function Alarma() { alert("Aldendu! Aireara behar dut eta"); }
6   -->
7 </script>
8 </head>
9 <body>
10  <a href="eventos.html" onMouseOver="Alarma()" >
11    Hemen gaitetik igaro </a>
12 </body>
13 </html>
```

Below the code editor, there are two browser windows. The left window shows the URL `file:///C:/Users/jiparsar/Dropbox/15-16-WebSistemak/garapenak/eventos.html` and the text "Hemen gaitetik igaro". The right window shows the same URL and displays an alert box with the message "Aldendu! Aireara behar dut eta" and an "Aceptar" button.

JS kodeen adibidea: Leihoen manipulazioa



The screenshot shows a code editor with a file named `adLeihoaSortzen.html`. The code is as follows:

```
<html>
<head>
<script language="JavaScript">
  function LeihoaIreki()
  { NireLeihoa=window.open("", "NireLeihoPropioa", "toolbar=no, directories=no, menubar=no, status=yes");
    NireLeihoa.document.write("<head><title>Leiho berri bat</title></head>");
    NireLeihoa.document.write("<center><h1><b>"
    + "Hau zuk nahi duzuna izan liteke</b></h1></center>");
  }
</script>
</head>
<body>
  <form>
    <input type="button" name="Boton1" value="Zabal nazazu!" onClick="LeihoaIreki()" >
  </form>
</body>
</html>
```

Below the code editor, there are two browser windows. The left window shows the URL `file:///C:/Users/jiparsar/I.../adLeihoaSortzen.html` and a button labeled "Zabal nazazu!". The right window shows the URL `localhost:1234/wsra/adLeihoaSortzen.html` and displays a new window titled "Leiho berri bat - Mozilla Firefox" with the text "Hau zuk nahi duzuna izan liteke".



JS kodeen adibidea:

Taulak eta elementuen sorrera

ikusBaliok.js

aDTableRow.html

```
1 <html>
2 <head>
3 <script language='javascript'>
4 function addRow(a,b) {
5     var r = document.createElement('tr');
6     var ca = document.createElement('td');
7     var cb = document.createElement('td');
8     var ta = document.createTextNode(a);
9     var tb = document.createTextNode(b);
10    var t = document.getElementById('tbl');
11    ca.appendChild(ta);
12    cb.appendChild(tb);
13    r.appendChild(ca);
14    r.appendChild(cb);
15    t.appendChild(r);
16 }
17 </script>
18 </head>
19
20 <body>
21 <form>
22 Column A: <input type='text' name='a' >
23 Column B: <input type='text' name='b' >
24 <input type='button' value='add Row'onClick='javascript:addRow(a.value, b.value)' >
25 </form>
26
27 <table id='tbl'>
28 <tr><td>Column A</td><td>Column B</td></tr>
29 </table>
30
31 </body>
32 </html>
```

Column A: Bizkaia

Column B: Bilbo

add Row

Column A Column B

Gipuzkoa Donostia

Bizkaia Bilbo

Hyper Text Markup Language filelength : 737 lines : 32 Lr