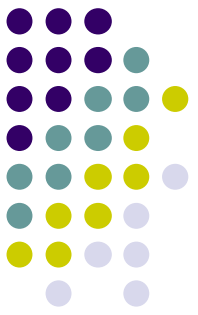


# **G3.2(e): Bezeroaren aldeko teknologiak. DOM. (JS. JQuery)**

Rosa Arruabarrena, Jose Ángel Vadillo  
LSI, UPV/EHU

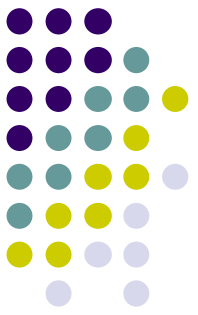


# DOM. Zer da? (I)

- *Document Object Model*
- Dokumentuko Objektuen eredua, edo Dokumentuen errepresentaziorako objektuen eredua

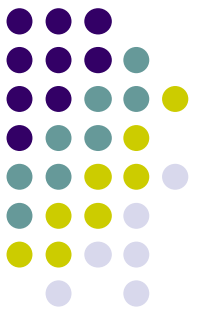
## Zer da?

- Plataformekiko independentea (S.E., Web arakatzaileretatik, lengoaieretatik) izanik, plataforma-arte konbentzio bat da.
- Hitzarmenak HTML, XHTML eta XML dokumentuetako objektuak nola adierazi edota elkar eragin ditzakegun zehazten du, bai dokumentuetan bai arakatzaileretan
- Dokumentuetako edukia, egitura eta estiloa dinamikoki kontsultatzeko eta aldatzeko aukera emanez



## DOM. Zer da? (II)

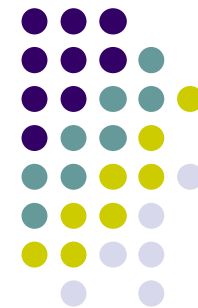
- Web Aplikazioak Programatzeko Interfazea (API) da esentzialki
- DOMak dokumentuko edukiak zuhaitz egitura baten bidez adierazten du, *DOM zuhaitza*.  
Dokumentuko elementuek DOM zuhaitzeko nodoak erlazionatuak dituzte.
- Zuhaitzeko nodo bakoitza objektu bat da
  - Atzigarri, maneiaragarri, aldagarri dena
  - Bere edukia nahiz atributuakObjektuko *metodoak* erabiliz
- DOMaren interfaze publikoa bere API-an zehaztua dago



## DOM. Zer da? (III)

- Objektuetan Oinarritutako eredua jarraitzen da
- Programazio-lengoaiek DOMa inplementatzen duten liburutegiak dituzte
- Javascript-ez programatzean HTMLeko DOM espezifikoa erabiltzen da, HTML dokumentuak manipulatzeko, nabigatzeko
- Arakatzailleak dokumentuaren egitura (dagokion DOMa) sortzen du orri bat kargatzen den aldioro
  - DOM inspector (Firefox extension)

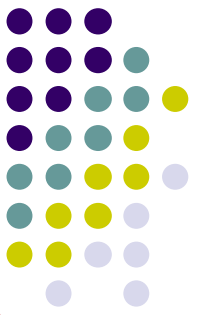
# Beraz, Zertarako balio du?



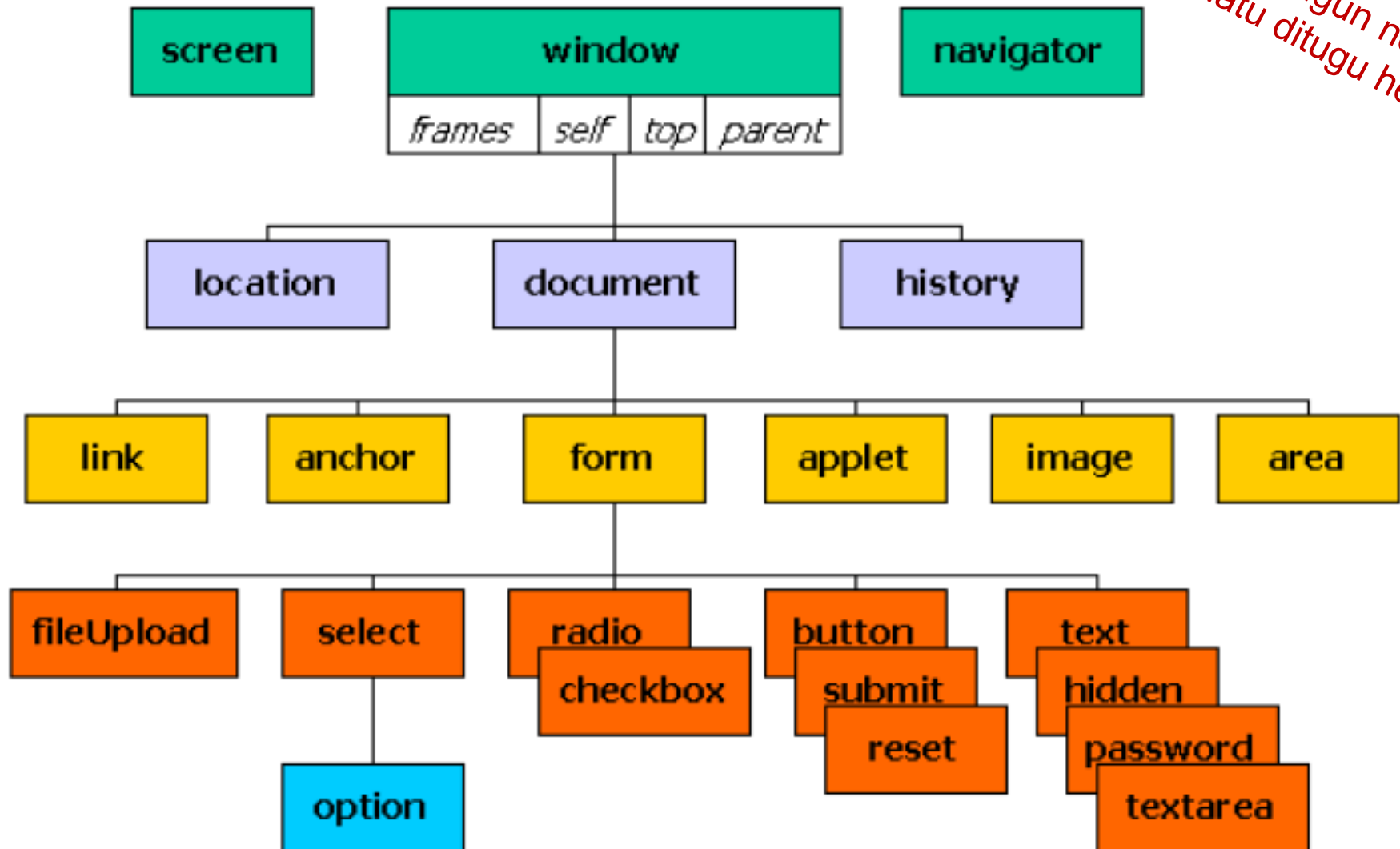
Web orriei funtzionalitatea eta elkarrekintza gehitzeko. Adibidez:

- Orriko elementuen edukia dinamikoki aldatu ahal izateko
- Orriko elementuen balio desberdinak atzitu ahal izateko.
- Orrian elementu berriak sortzeko
- ...

# OBJEKTUEN EREDUA



Irakasgaian interesantzen  
zaizkigun nodoak  
soilik islatu ditugu hemen



document

Root element:  
<html>

Element:  
<head>

Element:  
<title>

Text:  
"My title"

Element:  
<body>

Element:  
<h1>

Text:  
"A heading"

Element:  
<a>

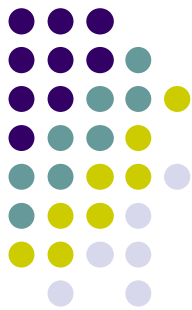
Attribute:  
href

Text:  
"Link text"

DOM

Document Object Model

[https://en.wikipedia.org/wiki/Document\\_Object\\_Model](https://en.wikipedia.org/wiki/Document_Object_Model)



# DOMak aztertzen

1erro2

Hau HTML dokumentu *simplea* da

1erro3

beste div eremu bat zerrenda bat

- BAT
- BI
- HIRU esteka batekin

1erro4

Inspe Cons Depu Editor d Rend Mem Red Almacena

+

```
<!DOCTYPE html>
<html>
  <head>
    <title>DOM-ak aztertzen</title>
    <meta charset="utf-8">
  </head>
  <body>
    <div id="gelal">
      <div>
        beste div eremu bat zerrenda bat
        <ul>
          <li>BAT</li>
          <li>BI</li>
          <li>
            HIRU
            <a href="http://www.ehu.eus">esteka batekin</a>
          </li>
        </ul>
      </div>
      lerro4
    </body>
  </html>
```

html > body > div > ul > li

## Reglas

Calculado

## Disposición

## Animaciones

## Tipografía

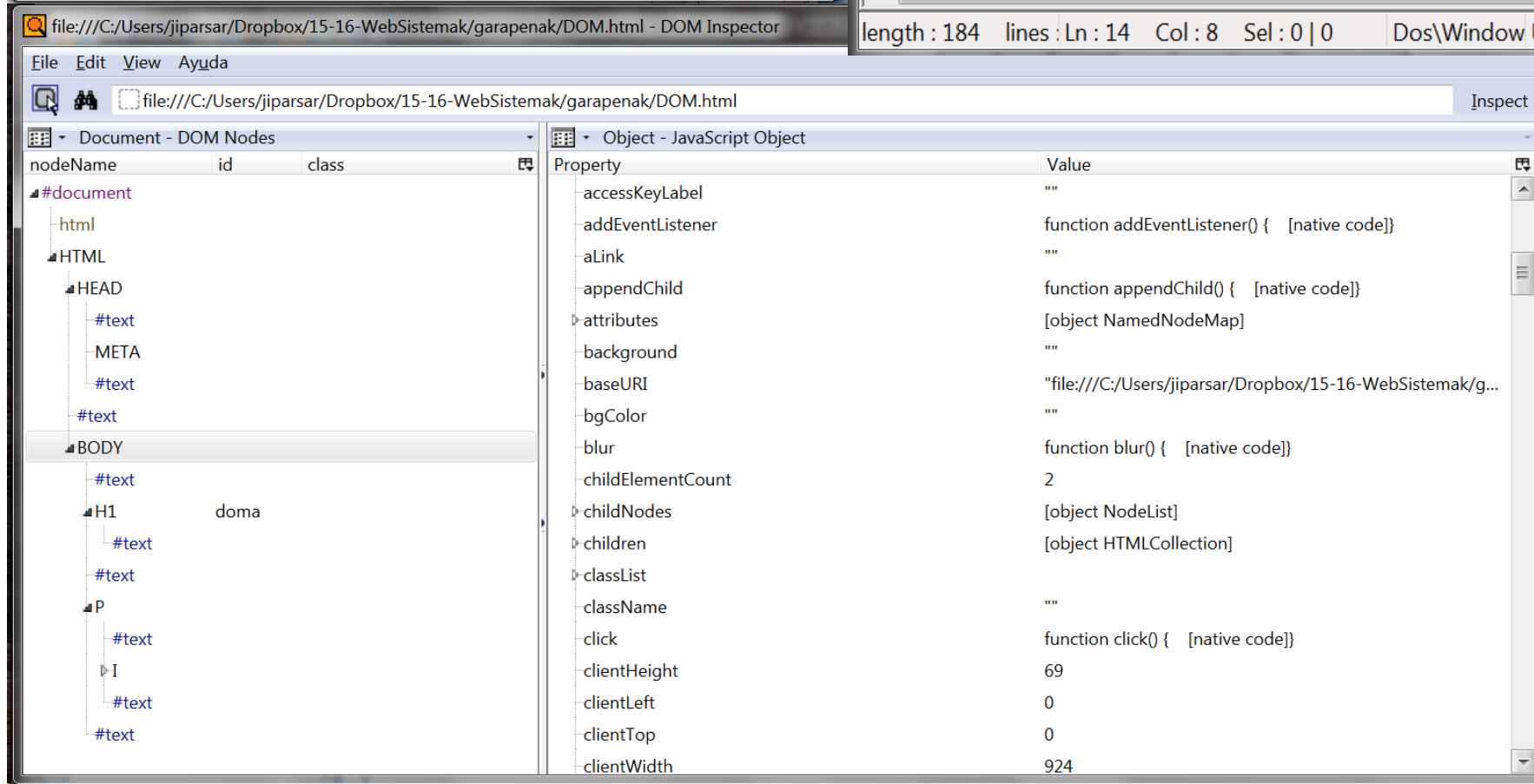
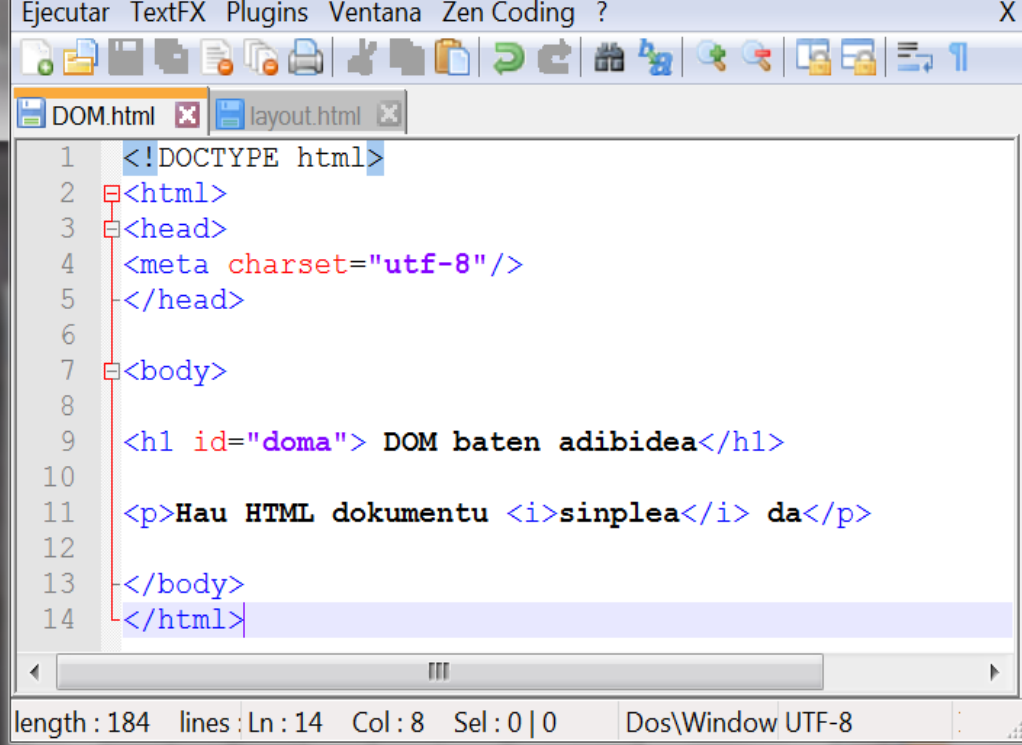
 Filtrar estilos

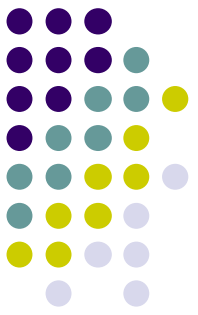
```
elemento {
```

+ ↗ ... .cls

en línea



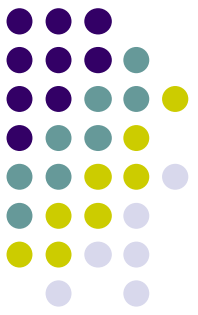




# Objektuak atzitzen

- DOMak definitzen du nola elkar erlazionatzen diren objektuak/elementuak arakatzaille eta dokumentu artean
- Web diseinurako egokia den edozein programazio lengoaiak erabil lezake DOM eredia
- JavaScript-en kasuan, objektu bakoitzak izen bat izan dezake. Balu, bakarra litzateke, eksklusiboa, hura erabiliz elementua atzigarri litzateke.
- Mota bereko objektu bat baino gehiago dagoenean web dokumentu batean, orduan bektore moduan antolatzen dira (0 lehenengoa).

# Objektuak atzitzen



```
<div id="gela1"> .... </div >
```

Atzitzeko aukerak:

```
document.div[0]
```

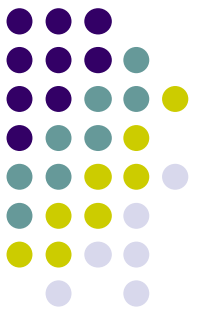
```
document.div["gela1"]
```

```
document.div.gela1
```

```
document.getElementById("gela1")
```

---

```
document.getElementsByTagName("div") → [divObj]
```



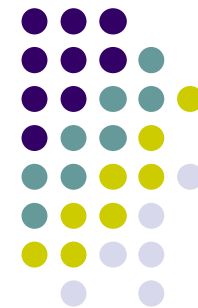
# Objektuen propietateak

- Objektuek propietateak eta funtzionalitateak dituzte:

```
objektua.propietatea = balioa;
```

- Objektuen manipulaziorako, erabiltzen den programazio lengoaiaren arauak jarraitu behar dira.
- Objektuak normalki zerbaitetarako sortuak direnez, funtzionalitate bat/batzuk dituzte.
  - JavaScripten objektuek funtzio aurredefinituak badituzte ere, web garatzaileak funtzio berriak defini ditzake orria bere gustura dinamikoki alda dadin

# Gertaerak (I)



- (Informatikan) egoera aldaketa bat suertatzen denean sortzen dira **gertaerak** :
  - Saguaren posizioa aldatu da
  - Teklaren bat sakatu da
  - Pantailaren egoera aldatu da
  - ..
- Web orrien sorreran, erabiltzaile eta konputagailuaren elkarrekintza adierazten dute gertaerek
- Honela, gertaera baten ondorioz, maiz efektu bat sor dezala konputagailuak espero da, adibidez, botoi bat sakatu ostean, testu kutxa bat uzterakoan,...

# Gertaerak (II)



- HTMLrenkin erlazionatutako gertaerak
  - Saguarekin edo idazten denarekin  
`onClick, onChange, onMouseOver`
  - Saguarekin egiten diren ekintzen ondorioz eta kurtsoare bidez pantailan ikusten direnak  
`onBlur, onFocus`
  - Testu aukeraketarekin, formularioen igortzearekin, orriaren kargarekin,...  
`onSelect, onSubmit, onLoad, ...`

# Gertaerak (III)



- “event handlers” (gertaera-kudeatzaileak) dira gertaerei erantzuna ematen dieten objektuak

```
<element onEvent="script"> .... </element>
```

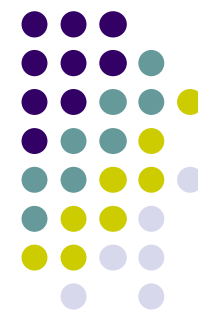
- Gerta maneia liteke, adibidez:

```
<div id="gela1" onClick="javascript:ikusBaliabideakG1();">
```

Hemen bestelako testua legoke

```
</div>
```

# Gertaerak (IV)



## Formularioetako gertaerak

- onBlur** – eremuak fokua galtzean
- onChange** – fokua duen elementuak balioz aldatzean
- onFocus** – eremuak fokua lortzean
- onReset** – erabiltzaileak formularioari reset egitean
- onSelect** – testu bat aukeratua izatean
- onSubmit** – erabiltzaileak formularioa igortzea aukeratzean

## Esteketako gertaerak

- onClick** – hautatutako elementuan klik egitean
- onMouseOut** – sagua botoiaren edo estekaren kanpora mugitzean
- onMouseOver** sagua botoiaren edo estekaren gainetik pasatzean

## Teklatuko gertaerak

- onKeyDown** – erabiltzaileak tekla sakatzean
- onKeyPress** – erabiltzaileak tekla sakatua mantentzean
- onKeyUp** – erabiltzaileak tekla sakatua askatzean

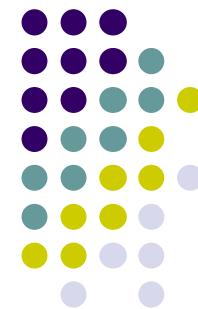
## Leihoetako gertaerak

- onBlur** – leiho edo markoak (frameak) fokua galtzean
- onError** – errore bat gertatzean
- onFocus** – leihoak edo frameak fokua lortzean
- onLoad** – objektua erabat kargatzean
- onUnload** – leihoa ixtean
- onResize** – leihoa edo framea birdimentsionatzean



# Gertaerak maneiatzen.

## Beste aukera batzuk



- Gertaerak ere orriko objektuen *propietateak balira* moduan trata litezke:

```
objektua.gertaera = funtzioa;
```

```
document.div.gela1.onClick = egizuZerbait;
```

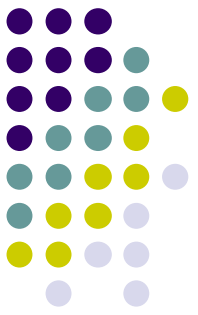
```
document.getElementById("gela1").onClick = egizuZerbait;
```

- DOMak gertaera web orritik kanpo gertatzen dela kontsideratzen du, eta nolabait hedatzen dela orriko barne elementuetara. Nabigazio honetan, zeharkatzen diren objektuek adi egon beharko lukete, eta gertaera “entzuteko” funtzio bat dute:

```
objektua.addEventListener (gertaera, funtzioa, unea);
```

```
document.getElementById("gela1").addEventListener("click",  
egizuZerbait,false);
```

# Gertaerak maneiatzen. Beste aukera batzuk (II)



DOMak gertaera web orritik kanpo gertatzen dela kontsideratzen du, eta nolabait hedatzen dela orriko barne elementuetara. (nabigatuz). Hedapenaren adibide bat:

Gertaera → Leihoa → DOCUMENT → HTML → BODY → DIV → jomuga

Erantzuna → DIV → BODY → HTML → DOCUMENT → Leihoa → Gertaera

Sekuentziazioaren interpretazioak 3 etapa ditu:

- *caption* (harrapaketa): gertaera jomugarantz doanean
- *hit* (jotzea): jomugara iristean. Jomuga objektu bat da, gertaera jasotzearen ondorioz zerbait egingo duena
- *bubbling* (burbuilak): etapa honetan gertaera jatorrizko posiziora itzultzen da.

JS Global

JS Conversion

## Browser BOM

Window

Navigator

Screen

History

Location

## HTML DOM

DOM Document

DOM Elements

DOM Attributes

DOM Events

## HTML Objects

&lt;a&gt;

&lt;abbr&gt;

&lt;address&gt;

&lt;area&gt;

&lt;article&gt;

&lt;aside&gt;

## HTML DOM Events

**DOM:** Indicates in which DOM Level the property was introduced.

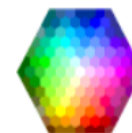
## Mouse Events

Event	Description	DOM
<a href="#">onclick</a>	The event occurs when the user clicks on an element	2
<a href="#">oncontextmenu</a>	The event occurs when the user right-clicks on an element to open a context menu	3
<a href="#">ondblclick</a>	The event occurs when the user double-clicks on an element	2
<a href="#">onmousedown</a>	The event occurs when the user presses a mouse button over an element	2
<a href="#">onmouseenter</a>	The event occurs when the pointer is moved onto an element	2
<a href="#">onmouseleave</a>	The event occurs when the pointer is moved out of an element	2
<a href="#">onmousemove</a>	The event occurs when the pointer is moving while it is over an element	2
<a href="#">onmouseover</a>	The event occurs when the pointer is moved onto an element, or onto one of its children	2
<a href="#">onmouseout</a>	The event occurs when a user moves the mouse pointer out of an element, or out of one of its children	2
<a href="#">onmouseup</a>	The event occurs when a user releases a mouse button over an element	2

Google

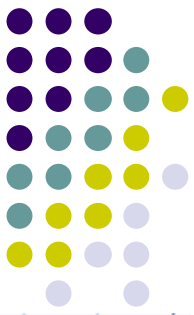
W3SCHOOLS  
EXAMSHTML, CSS,  
JavaScript, PHP,  
jQuery, and XML  
Certifications

COLOR PICKER



SHARE THIS PAGE

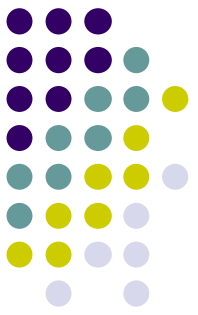




# Gertaeren koadroa

Evento	Descripción	Elementos que lo admiten
onLoad	Terminar de cargarse una página	<BODY...> <FRAMESET...>
onUnLoad	Salir de una página (descargarla)	<BODY...><FRAMESET...>
onMouseOver	Pasar el ratón por encima	<A HREF...><AREA...>
onMouseOut	Que el ratón deje de estar encima	<A HREF...><AREA...>
onSubmit	Enviar un formulario	<FORM...>
onClick	Pulsar un elemento	<INPUT TYPE="button, checkbox, link, radio"...> <SELECT>
onBlur	Perder el cursor	<INPUT TYPE="text"...> <TEXTAREA...> <SELECT>
onChange	Cambiar de contenido o cambiar de selección	<INPUT TYPE="text"...> <TEXTAREA...> <SELECT>
onFocus	Conseguir el cursor	<INPUT TYPE="text"...> <TEXTAREA...> <SELECT>
onSelect	Seleccionar texto	<INPUT TYPE="text"...> <TEXTAREA...>

# Objects (adibideak) ★



- `document.write(string)`
  - String-a orrian idazten du
- `window.history.back()`
  - aurreko orriaren URLa itzultzen du
- `window.status`
  - Egoera barrako testua
- `document.form1.control1`
  - `Document.getElementById("control1")`
  - Formulario barruko kontrol bat
  - Formularion balioztatzea
- `document.CreateElement(elem)`



HTML

CSS

JAVASCRIPT

SQL

PHP

BOOTSTRAP

TUTORIALS ▾

REFERENCES ▾

&lt;abbr&gt;

&lt;address&gt;

&lt;area&gt;

&lt;article&gt;

&lt;aside&gt;

&lt;audio&gt;

&lt;b&gt;

&lt;base&gt;

&lt;bdo&gt;

&lt;blockquote&gt;

&lt;body&gt;

&lt;br&gt;

&lt;button&gt;

&lt;canvas&gt;

&lt;caption&gt;

&lt;cite&gt;

&lt;code&gt;

&lt;col&gt;

&lt;colgroup&gt;

&lt;datalist&gt;

&lt;dd&gt;

&lt;del&gt;

&lt;details&gt;

&lt;dfn&gt;

&lt;dialog&gt;

&lt;div&gt;

# Document Object Properties and Methods

The following properties and methods can be used on HTML documents:

Property / Method	Description
<u><a href="#">document.activeElement</a></u>	Returns the currently focused element in the document
<u><a href="#">document.addEventListener()</a></u>	Attaches an event handler to the document
<u><a href="#">document.adoptNode()</a></u>	Adopts a node from another document
<u><a href="#">document.anchors</a></u>	Returns a collection of all <a> elements in the document that have a name attribute
<u><a href="#">document.applets</a></u>	Returns a collection of all <applet> elements in the document
<u><a href="#">document.baseURI</a></u>	Returns the absolute base URI of a document
<u><a href="#">document.body</a></u>	Sets or returns the document's body (the <body> element)
<u><a href="#">document.close()</a></u>	Closes the output stream previously opened with document.open()
<u><a href="#">document.cookie</a></u>	Returns all name/value pairs of cookies in the document
<u><a href="#">document.createAttribute()</a></u>	Creates an attribute node
<u><a href="#">document.createComment()</a></u>	Creates a Comment node with the specified text
<u><a href="#">document.createDocumentFragment()</a></u>	Creates an empty DocumentFragment node

# Orrien DOMak. Adibideak

Archivo Editar Ver Historial Marcadores Herram

file:///F:/SW-2015...rearelementos.html

← JavaScript/ejemplosjs/crearele ↻ Google

Column A:

Column B:

add Row

Column A	Column B
----------	----------

file:///F:/SW-2015/Transparencias/Tema3-Tecnolog%C3%AD... Inspect

File Edit View Ayuda

Document - DOM Nodes

nodeName	id	class
#document		
HTML		
HEAD		
#text		
BODY		
#text		
FORM		
#text		
INPUT	a	
#text		
BR		
#text		
INPUT	b	
#text		
BR		
#text		
INPUT		
#text		
#text		
TABLE	tbl	
#text		
TBODY		
TR		
TH		
#text		
TH		
#text		
#text		

Object - DOM No

Column A



# Orrien DOMak. Adibideak (II)

Archivo Editar Ver Historial Marcadores Herr

file:///F:/SW-2015...rearelementos.html +

← file:///F:/SW-2015...rearelementos.html ↻ Google

Column A:

Column B:

Column A	Column B
ValorA	ValorB

file:///F:/SW-2015/Transparencias/Tema3-Tecnol...

File Edit View Ayuda

file:///F:/SW-2015/Transparencias/Tema3-Tecnolog%C3%... Inspect

Document - DOM Nodes

nodeName	id	class
HEAD		
#text		
BODY		
#text		
FORM		
#text		
INPUT	a	
DIV		
#text		
BR		
#text		
INPUT	b	
#text		
BR		
#text		
INPUT		
#text		
#text		
TABLE	tbl	
#text		
TBODY		
TR		
TH		
TH		
#text		
TR		
TD		
#text		
TD		
#text		
#text		

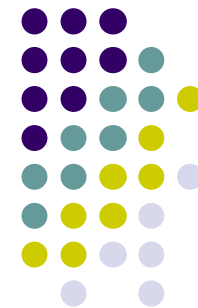
Object - D

ValorA

anonymous-...



# Iturburu interesgarriak



<http://www.w3.org/DOM/>

[https://developer.mozilla.org/en-US/docs/Web/API/Document\\_Object\\_Model/Introduction](https://developer.mozilla.org/en-US/docs/Web/API/Document_Object_Model/Introduction)

[https://en.wikipedia.org/wiki/Document\\_Object\\_Model](https://en.wikipedia.org/wiki/Document_Object_Model)

DOM tutoriala

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

Gertaerak//Events

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

DOM txuleta:

<http://www.wait-till-i.com/stuff/JavaScript-DOM-Cheatsheet.pdf>