

Erronka1. Javascript-II

Web garapena bezeroen ingurunean

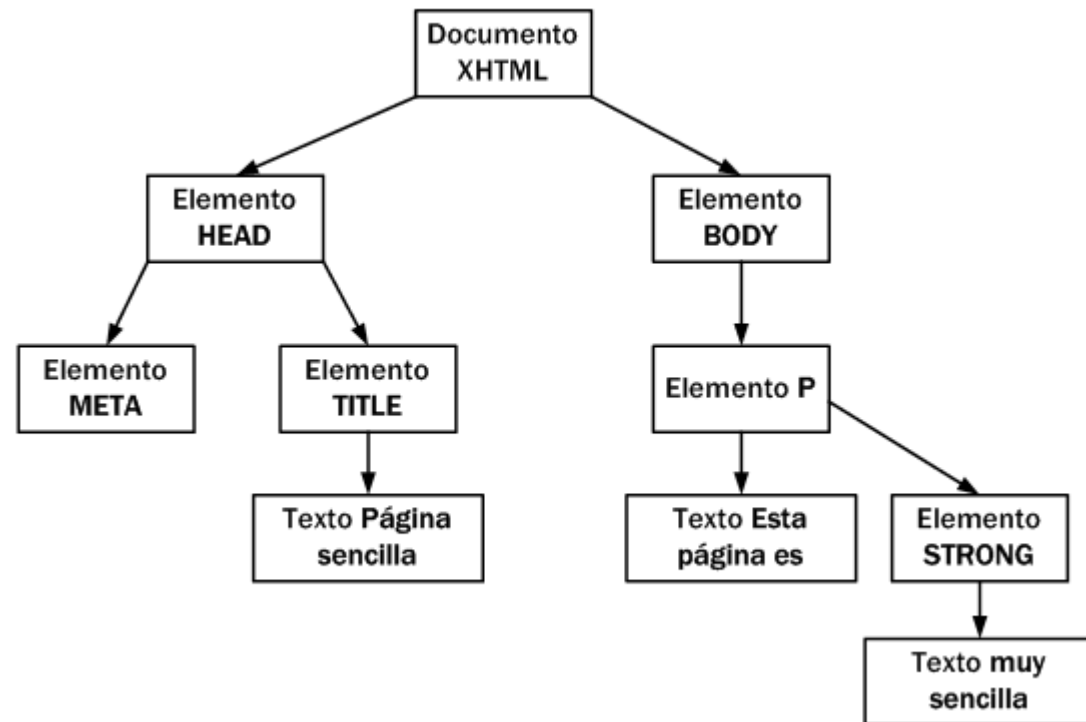
1. DOM erabiltzen
2. Formularioak
3. Gertaerak

Erronka1. Javascript-II

1. DOM erabiltzen

- “Document object model”: XHTML orria aldatzen uzten du, XML fitxategia izango bazen bezala.
- Hau egiteko, XHTML orria hartzen du eta nodoen zuhaitz bat sortzen du. Nodoen zuhaitz hau kontutan hartuta, javascript bezalako lengoaia erabiliz, gure orriko edozein zati aldatu dezakegu.
- **Adibide bezala:**

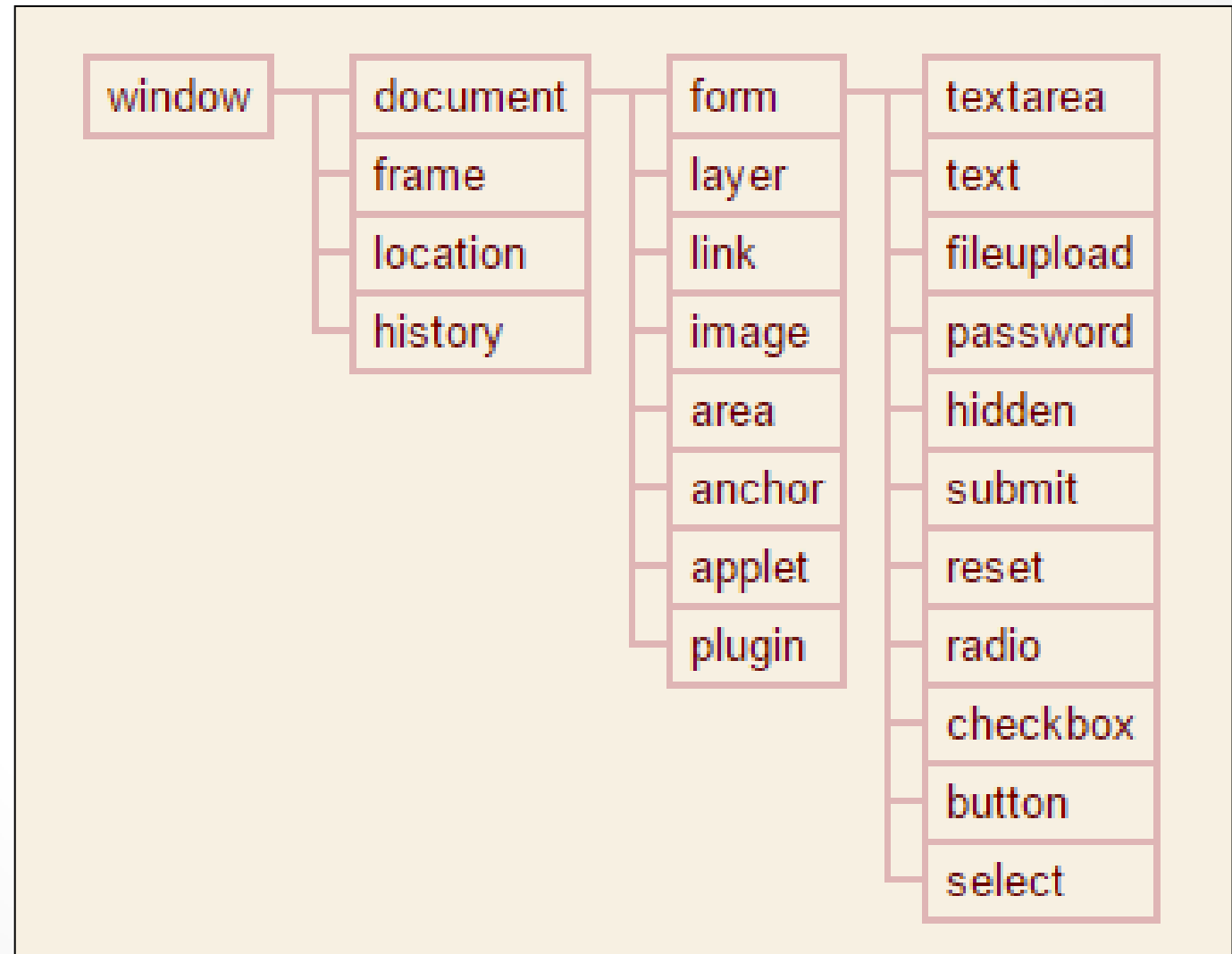
```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0  
Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-  
transitional.dtd">  
<html xmlns="http://www.w3.org/1999/xhtml">  
  <head>  
    <meta http-equiv="Content-Type" content="text/html;  
    charset=iso-8859-1" />  
    <title>Página sencilla</title>  
  </head>  
  <body>  
    <p>Esta página es <strong>muy sencilla</strong>  
    </p>  
  </body> </html>
```



Erronka1. Javascript-II

1. DOM erabiltzen

- DOM teknologia erabiltzeko Javascript-ek hurrengo objektuen hierarkia sortzen du :



Erronka1. Javascript-II

1. DOM erabiltzen

- Window da objektu nagusia. Window barruan beste objektu hauek topatzen ditugu: document, frame, location eta history.
- Document barruan ere beste objektu batzuk daude, estekak (links), irudiak (image) edo formularioak (form).
- Formulario barruan bere osagaiak topatu daitezke.
- Objektu guztiak propietateak dituzte, adibidez izena, eta honela atzitu ditzakegu: objektu_izena.propietate_izena.
- Propietateen balioak ikusi ditzakegu baina baita balio berria itsatsi ere.
- Objektu guztiak metodoak dituzte eta honela atzitu ditzakegu: objektu_izena.metodo_izena(parametroak).
- Metodoak funtzioak dira eta nolabaiteko ekintza egiteko gai dira. Adibidez, leiho bat mugitu edo utzi. Parametroak eduki ahal dituzte.
- **Objektu batzuk array motakoak dira**, adibidez image objektua (document barruan), edo formulario bateko input objektuak.
- Image array batean osagai bakoitzak bere propietate src edukiko du, irudiaren fitxategiaren izenarekin. Input array batean osagai bakoitzak bere type eta value edukiko dute

Erronka1. Javascript-II

1. DOM erabiltzen

- Window objektua

- window objektua momentu honetako leihoa da, hierarkian maila altuenean dago. Beste objektu guztiak bere barnean kokatzen dira.
- Bere propietate esanguratsuenak:
 - location. Momentu honetako URL-a.
Adibidea: `window.location='http://www.uni.eus'`.
 - history. Bisitatutako azkenengo orrien array-a .
 - close(). Leihoa itxi.
 - moveBy(x, y). Leihoa mugitu, *x eta y koordenadak erabiliz*.
 - moveTo(x, y). Leihoa mugitu baina koordenada zehatz batzuetara.
 - open(URL, nombre, características). Leiho berria zabaltzen du, URL batekin, guk jartzen diogun izenarekin eta ezaugarriekin.
 - back(), forward(), aurrera edo atzera joateko historialean.

Erronka1. Javascript-II

1. DOM erabiltzen

- Document objektua
- document objektua orriaren osagai guztiak ditu. Eta bera erabiliz balioak aldatu ditzakegu.
- Propietateak:
 - bgColor, atzealdeko kolorea.
 - fgColor, testuaren kolorea.
 - forms, array-a formularioekin.
 - images, array-a irudiekin.
 - links, array-a estekekin.
 - tittle, orriaren titulua.
- Metodoak:
 - clear(), dokumentua garbitzen du.
 - open(), dokumentua idazteko irekitzen du.
 - write('testua'), pasatzen den testua idazten du.
 - writeln('testua'), testua idatzi eta hurrengo lerroa pasatzen da.
 - close(), dokumentua itxi.
 - **getElementsByTagName()**, etiketa berdineko elementu guztiak
 - **getElementsByTagName()**, izen berdineko elementu guztiak
 - **getElementById()**, ID batekin edozein elementu topatzeko balio du.

Esteka interesgarria: <http://www.genbetadev.com/desarrollo-web/css-arriba-scripts-abajo-y-otras-recomendaciones-para-optmizar-la-carga-de-paginas-web>

Erronka1. Javascript-II

1. DOM erabiltzen
 - Document objektua

Adibide bat probatzeko:

```
<html>
  <head>
    <title>Objektuak eta propietateak</title>
  </head>
  <body>
    <h3 id="intro">Introducción </h3>

    <script>
      var nodo=document.getElementById( "intro" )
      nodo.style.color='red'
      nodo.onmouseover=function()
        { this.style.color='blue' }
    </script>

    <script>
      setTimeout( "document.getElementById('img1').src='2.gif'", 1000 )
    </script>
  </body>
</html>
```

Erronka1. Javascript-II

1. DOM erabiltzen
 - Document objektua

Beste adibide bat probatzeko:

```
<html>
  <head>
    <title>Estekak aldatzen</title>
  </head>
  <body>
    <ul>
      <li><a href="bat.html">Lehenengo esteka</a></li>
      <li><a href="bi.html" target="new">Bigarren esteka</a></li>
      <li><a href="hiru.html">Hirugarren esteka</a></li>
    </ul>
    <script>
      var nodos=document.getElementsByTagName( "a" )
      for ( var i=0; i<nodos.length; ++i )
        { nodos[i].onmouseover=function()
          { this.style.backgroundColor='green' }
          nodos[i].onmouseout=function()
            { this.style.backgroundColor="" }
        }
    </script>
  </body>
</html>
```


Erronka1. Javascript-II

1. DOM erabiltzen

- **InnerHTML**

Zuzenean balioa aldatzeko, ez da DOM-ekoa baina asko erabiltzen da

Adibide bat probatzeko:

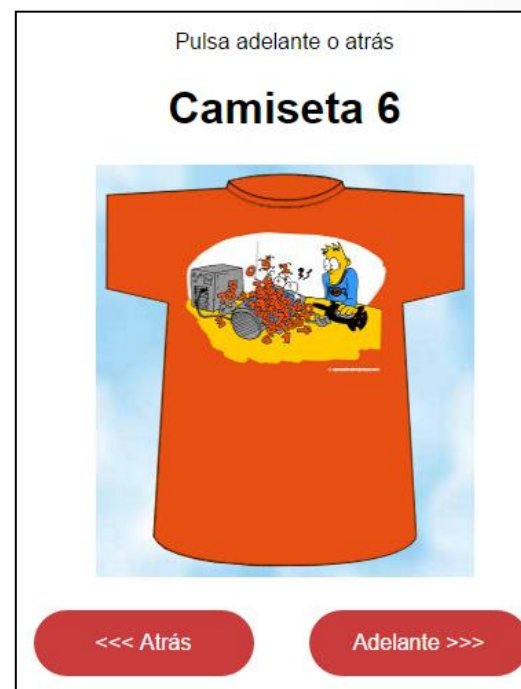
```
<html>
  <head> <title>Accediendo a la propiedad innerHTML</title> </head>
  <body>
    <div id="paragrafoa"> Paragrafo honen barruan gauza asko daude:
      <ul>
        <li>Zerrenda</li>
        <li>bat</li>
      </ul>
      <table border=1>
        <tr>
          <td>Eta</td>
          <td>taula bat</td>
        </tr>
      </table>
    </div>
    <script>
      function aldatu() {
        var el=document.getElementById( "paragrafoa");
        el.innerHTML="Eta osorik aldatzen dut"+
" <ol> <li>nahita</li> <li>zerrenda bat
sartu</li> </ol>";
      }
      setTimeout( "aldatu()", 1000 );
    </script>
  </body>
</html>
```

Erronka1. Javascript-II

1. DOM erabiltzen

- Hurrengo ariketa egingo dugu

21. Programa bat egin kamiseta batzuen irudiak aldatzeko. Kamiseta 6tik, 9ra pasatzen da eta gero 11ra. Beste mugimenduak "Ezin da mugimendu hori egin" mezua aterara.



Erronka1. Javascript-II

2. Formularioak

Formularioak Javascripten gehien erabiltzen diren objektuak dira, bertan objektu estatiko batzuk sartu daitezke, testua eta irudiak. Baita beste objektu dinamiko batzuk ere, erabiltzaileekin interakzioa sortzeko erabiltzen direnak, testu kaxak, aukeratzeko botoiak, checkbox kaxak, etab.

Formularioetan sarrerako balioak ondo dauden ala ez balidatzeko javascript erabili daiteke. Baita beste gauza batzuetarako ere, adibidez, combobox baten informazioa kargatzeko edo egutegi bereziak ikusteko, etab.

Adibidea

```
<html>
<head> <title>Formularioekin lanean</title> </head>
<body>
    Mezu bat sartu eta Google-en bilatuko du baina www.uni.eus domeinuan bakarrik
    <form name="n1" id="i1" action="http://www.google.es/search">
        <label for="q">Bilatzeko testua:</label>
        <input type="text" name="q" size="40">
        <input type="submit" value="Enviar">
    </form>
    <script>
        el=document.getElementById( "i1" )
        el.onsubmit=function test()
            { if ( el.q.value.length>0 )
                { var testua = el.q.value;
                  el.q.value = "site:www.uni.eus " + testua;
                  return true;}
              else { return false; } }
    </script>
</body>
</html>
```

Erronka1. Javascript-II

2. Formularioak

Beste adibide bat formularioa balidatzen

```
<html>
<head> <title>Formularioekin lanean</title> </head>
<body>
    <form id="myform" action="enviar.php" onsubmit="return enviar();">
        <label for="comando">Comando: </label>
        <input type="text" id="comando"/> </br>
        <button>Enviar formulario</button>
    </form>
    <script>
        function enviar(){
            var formulario = document.getElementById("myform");
            var dato = document.getElementById("comando");

            if (dato.value=="enviar"){
                alert("Enviando el formulario");
                formulario.submit();
                return true;
            } else {
                alert("No se envía el formulario");
                return false;
            }
        }
    </script>
</body>
</html>
```

Erronka1. Javascript-II

2. Formularioak

"value" propietatea garrantzitsua da. Objektu gehienek propietate hori dute. Salbuespenak dira checkbox eta radiobutton objektuak. Jakiteko klik egin den ala ez, "checked" propietatea erabiltzen dute. "True" edo "False" izan daiteke.

Adibidea

```
<html>
<head> <title>Trabajando con formularios (II)</title> </head>
<body>
    checkbox automatikoki aldatzen da
    <form name="n1" id="i1" action="http://www.google.es/search">
        <input type="checkbox" name="seleccionar">Aukeratu
denak<br/><br/>
        <input type="checkbox" name="linea">Línea 1<br/>
        <input type="checkbox" name="linea">Línea 2<br/>
        <input type="checkbox" name="linea">Línea 3<br/>
    </form>
    <script>
        el=document.getElementById("i1")
        el.seleccionar.onclick=function aldatu()
            { for( var i=0; i<el.linea.length; i++ )
                { el.linea[i].checked=el.seleccionar.checked; }
            }
    </script>
</body>
</html>
```

Erronka1. Javascript-II

2. Formularioak

Formularioak egiteko laguntza batzuk ditugu, adibidez, bootstrap.
Hau erabiltzeko gure html barruan idatzi behar duguna:

```
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css" integrity="sha384-Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm" crossorigin="anonymous">
```

Eta behar dugun formularioa aukeratu : <http://getbootstrap.com/docs/4.0/components/forms/>

Adibidez:

```
<form>
  <div class="form-group">
    <label for="exampleInputEmail1">Email address</label>
    <input type="email" class="form-control" id="exampleInputEmail1" aria-describedby="emailHelp"
      placeholder="Enter email">
    <small id="emailHelp" class="form-text text-muted">We'll never share your email with anyone
else.</small>
  </div>
  <div class="form-group">
    <label for="exampleInputPassword1">Password</label>
    <input type="password" class="form-control" id="exampleInputPassword1" placeholder="Password">
  </div>
  <div class="form-check">
    <label class="form-check-label">
      <input type="checkbox" class="form-check-input"> Check me out
    </label>
  </div>
  <button type="submit" class="btn btn-primary">Submit</button>
</form>
```

Erronka1. Javascript-II

Ejekutatzerakoan, honelako itxura du:

Email address

We'll never share your email with anyone else.

Password

☐ Check me out

Erronka1. Javascript-II

2. Formularioak

Hurrengo ariketa egin:

22. Web orri bat egin formulario batekin. Bidali botoiari klik egiterakoan **Sartu dituzun datuak** kapan sartu diren datu guztiak agertu behar dira. (**onclick** eta/edo **onsubmit** ebentoak erabili)

DAW klasea

Formularioak eta JavaScript

Zuzentzeko era ☒ Jauna ☐ Anderea

Izena

Abizena

Helbidea

Hiria

Gustokoena ☐ Liburuak ☐ Pelikulak



Sartu dituzun datuak:

Erronka1. Javascript-II

3. Gertaerak

- Javascriptek erabiltzaileen ekintzak kontrolatzeko eta portaera zehatz bat definitzeko gertaerak erabiltzen ditu.
- Web orri batean erabiltzaile batek eragina sortzen du eta horrekin batera gertaerak sortzen dira.
- Nolako portaera izango duen gertaera bakoitzak erabakitzen dugu . Aurreko ariketetan OnClick gertaera ezagutu dugu, botoiekin erabilia gehien bat, baina beste leku batzuetan erabil daiteke ere bai.
- Gertaerak kodifikatzen, adibide bat:

```
<INPUT type=button value="pulsame" onclick="sentencias_javascript...">
```

Beste adibide bat:

```
<SELECT onchange="window.alert('Cambiaste la selección')">  
  <OPTION value="opcion1">Opcion 1  
  <OPTION value="opcion2">Opcion 2  
</SELECT>
```

Erronka1. Javascript-II

3. Gertaerak

Gertaera batean agindu bat baino gehiago jar dezake. Baita funtzio bat sortu agindu guztiak batera jartzeko ere.

Adibidea:

```
<input type=button value=Pulsame onclick="x=30; window.alert(x); window.document.bgColor = 'red'">
```

Eta beste modu batean:

```
<input type=button value=Pulsame onclick="ejecutaEventoOnclick()">
```

```
<script>  
    function ejecutaEventoOnclick()  
    {  
        x = 30;  
        window.alert(x);  
        window.document.bgColor = 'red'  
    }  
</script>
```

Erronka1. Javascript-II

3. Gertaerak

Gertaera batzuk (zerrenda luzea manualean, 54. orrian):

- Onabort: irudi bat kargatzen ez denean
- Onblur: fokoa galtzen duenean
- Onchange: aldaketaren bat sortzen denean
- Onclick: klik egiten denean
- Onfocus: onblur kontrakoa
- Onkeydown, Onkeypress, Onkeyup: teklak sakatzen direnean
- Onload: orria kargatzen denean
- Onmouseover, Onmousedown,..... Saguarekin lotutako gertaerak
- Onselect: osagai bat aukeratzen denean
- Onsubmit: formularioa bidaltzen denean

Erronka1. Javascript-II

3. Gertaerak

Event Listener (gertaerak entzuten)

Nahi badugu edukiak, aurkezpena eta portaera, kapa ezberdinetan banatu, onclick eta beste gertaerak harrapatzeko kodigoa gure aurkezpen kaptik (html) atera beharko dugu, portaera kapakoak baitira. Hau egiteko javascriptek egiten duena "entzutea" da: "EventListener" deitzen zaio.

Metodo bat erabiltzen da: AddEventListener.

Behar den DOM-eko nodoari jartzen zaio, HTML barruan kodigoa jarri barik

Sintaxia: *nodoa.addEventListener('gertaera', egiteko_funtzioa, booleanoa);*
Adibidez:

```
<button onclick="balidatu()">Formularioa bidali</button>
```

Gertaera entzutea teknikarekin aldatzen badugu:

```
<button type="submit" id="bidali"> Formularioa bidali </button>
```

Eta scriptean:

```
document.getElementById('bidali').addEventListener('click',balidatu,false);
```

Erronka1. Javascript-II

3. Gertaerak

Kontutan hartu behar dugu funtzio hori idazteko orduan eta arazoak ez edukitzeko web orria kargatzen dagoen denboran honela prestatu behar dugula:

```
<div id="lekua">  
    <p>klik egiteko lekua</p>  
</div>
```

```
<script>  
    window.onload = function ()  
    {  
  
        document.getElementById('lekua').addEventListener('click',mezua,false);  
  
    }  
</script>
```

Honela, ez dugu uzten script-a exekutatzen html osagaiak kargatu aurretik

Erronka1. Javascript-II

3. Gertaerak

Osagai bat baino gehiagorekin egin daiteke, `getElementsByTagName` edo `getElementsByName` erabiliz:

```
<Form>
    <input type="text"..... / >
    <input type="radio"..... />
</Form>

<script>
    window.onload = function ()
    {
        var elementosDelForm = document.getElementsByTagName('input');

        for (var i = 0; i < elementosDelForm.length; i++)
        {
            if (elementosDelForm[i].type == 'radio')
                { elementosDelForm[i].addEventListener("click", actualizarDatos); }
            else
                { elementosDelForm[i].addEventListener("change",
actualizarDatos); }
        }
    }
</script>
```

Honela, ez dugu uzten script-a exekutatzen html osagaiak kargatu aurretik

Erronka1. Javascript-II

3. Gertaerak

Beste metodo hau erabili daiteke ere bai: removeEventListener

Adibidez behin bakarrik nahi badugu gertaera bat kontrolatu, lehenengo addEventListener erabili eta gero removeEventListener jarri.

Zer gertatzen da gertaera bat baino gehiago programatzen beditugu?

Adibidez:

```
<body>
    <div>
        <button>HAZME CLIC</button>
    </div>
</body>
```

Eta daukagu gertaerak programatuta body etiketarentzat, baita div etiketarentzat eta button etiketarentzat. Zein exekutatzen da lehenengo?

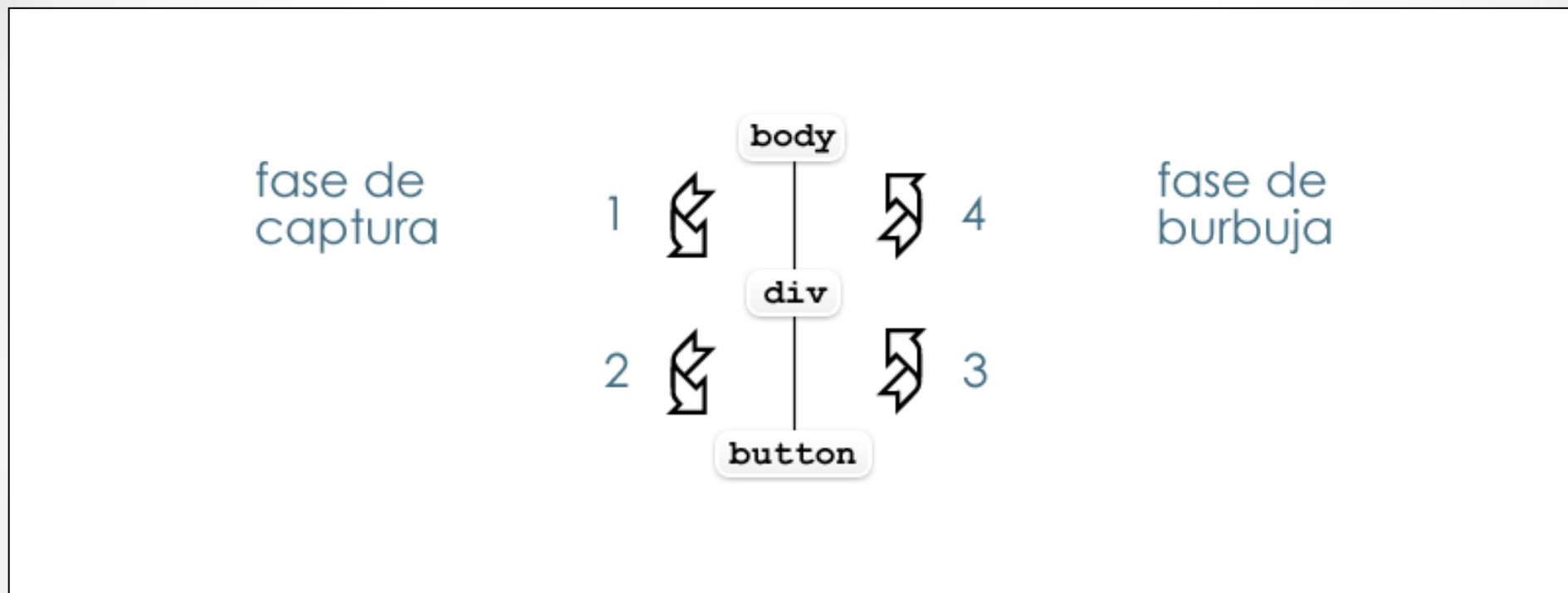
Gertaerak bi modutan exekutatzen dira: "captura" edo "burbuja"

"captura": body-div-button

"burbuja": button-div-body

Erronka1. Javascript-II

3. Gertaerak



Aurreko adibidean:

```
document.getElementById('enviar').addEventListener('click',validar,false);
```

Azkenengo parametroa:

True: “captura” eran izango zen

False: “burbuja” eran izango zen

Erronka1. Javascript-II

3. Gertaerak

Internet Explorer nabigatzaileak (IE9 aurretik), gai hau ezberdin maneiatzen du. Ez ditu metodo hauek erabiltzen, beste bi ditu antzekoak direnak:

```
nodoa.attachEvent('on+gertaera',funtzioa);  
nodoa.detachEvent('on+gertaera',funtzioa);
```

Horrek arazo txiki bat sortzen du eta hori konpontzeko estilo honetako funtzioak sortzea komeni da:

```
function ev (x,y,z)  
{  
    if (document.addEventListener)  
        {  
            x.addEventListener(y,z,false);  
        }  
    else  
        {  
            x.attachEvent('on'+y,z);  
        }  
}
```

Erronka1. Javascript-II

3. Gertaerak

23. Orri bat prestatu (addEventListener erabilita) botoi batekin. Klik egiten den bakoitzean irudi berri bat agertzen da (irudi guztiak array batean jarri). Lau irudi daude, beraz, bostgarren klik egiterakoan, mezua ateratu ("Ezin dira irudi gehiago gehitu"). Irudi bakoitzean klik bikoitza egiterakoan, irudi hori desagertzen da.

```
var irudia = document.createElement("img");  
irudia.src = ".....";  
document.body.appendChild(irudia);
```

```
document.body.removeChild(irudia);
```

Egin klik hemen irudi berria agertzeko



Erronka1. Javascript-II

Estekak:

BOOTSTRAP utilities:

https://www.w3schools.com/bootstrap4/bootstrap_utilities.asp

DIV eta FLOAT:

<https://www.campusmvp.es/recursos/post/display-flow-root-para-limpiar-floats-css.aspx>

Key jakiteko:

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