

03

Desenvolupament Web en entorn Client

Cicle: Desenvolupament d'aplicacions web

Curs 2025-2026

©José Masip Alonso

1. Introducció

En aquest document aprendrem a utilitzar cookies, programació orientada a objectes i a manipular el DOM.

NOTA: ALS EXERCICIS DE DOM TENIM QUE UTILITZAR LA METODOLOGIA APRESA A AQUEST DOCUMENT.

2. Contingut

2.1. Cookies

Les cookies permeten emmagatzemar informació de l'usuari a les pàgines web. És guarden a l'equip del client.

A les cookies podem guardar informació de l'usuari per a quan torne a visitar la pàgina.

La cookie es guarda en format nom_variable=valor.

```
function crear(){
    document.cookie="galleta=ieeeee";
}

function llegir(){
    alert (document.cookie);
}
```

Podem fer que la cookie caduque afegint l'atribut expires.

```
document.cookie="galleta=ieeeee;expires=data a la que caduca"

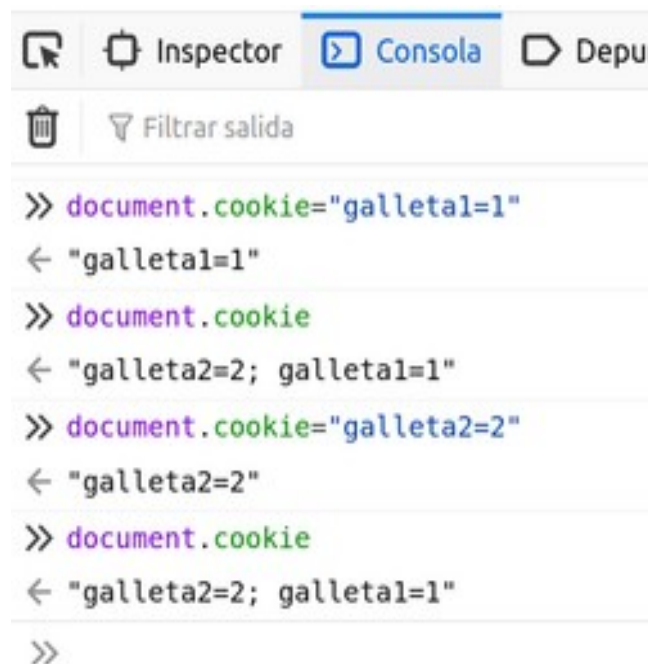
var datacad= new Date();
```

```
datacad= new Date(datacad.getTime()+1000*60*5); //5minuts en ms  
document.cookie="galleta=ee;expires="+datacad.toGMTString()+";";
```

Com podem guardar 2 cookies distintes? Com accedim a elles?

Com eliminem una cookie?

Per a guardar 2 cookies simplement les guardarem en distint nom.



The screenshot shows a web browser's developer console with the 'Consola' tab selected. The console displays a series of commands and their outputs:

```
>> document.cookie="galleta1=1"  
← "galleta1=1"  
  
>> document.cookie  
← "galleta2=2; galleta1=1"  
  
>> document.cookie="galleta2=2"  
← "galleta2=2"  
  
>> document.cookie  
← "galleta2=2; galleta1=1"  
  
>>
```

Si ens fixem en exemple anterior podem deduir que les cookies es mostren com a un string. Per tant, per a accedir a aquestes cal obtindre els valors separant aquest string en parts. Si dividim la cadena pels ; conseguirem les distintes cookies. Posteriorment dividirem aquestes per = per a separar el nom de la cookie del seu valor.

Per a eliminar una cookie simplement la tornem a crear amb una data actual a aquest moment. Així la creem caducada i la eliminem.

2.2. HTML5 LocalStorage

Amb localStorage, les aplicacions web poden emmagatzemar dades localment en el navegador de l'usuari.

Abans d'HTML5, dades d'aplicacions havien de ser emmagatzemada en les cookies, inclosos en cada petició de servidor. localStorage és més segura, i grans quantitats de dades poden ser emmagatzemats localment, sense afectar el rendiment web.

A diferència de les cookies, el límit d'emmagatzematge és molt gran (almenys 5 MB, cookies 4Kb) i la informació mai es transfereix al servidor (en cookies necessita transferir tota la informació).

L'emmagatzematge local és per domini. Totes les pàgines, d'un domini, poden emmagatzemar i accedir a les mateixes dades.

window.localStorage - magatzems de dades sense data de caducitat

code.sessionStorage - emmagatzema les dades per a un període de sessions (les dades es perden quan la llengüeta està tancat)

```
// guardar valor
localStorage.setItem("nom", "Rodrigo de la Fuente Luis");

// Obtindre valor
localStorage.getItem("nom");

//eliminar valor
localStorage.removeItem("nom");
```

2.3. Programació Orientada a Objectes

Les classes JavaScript són plantilles per a objectes JavaScript.

Utilitzem la paraula clau `class` per crear una classe. Amb `constructor()` definim l'objecte (aquest mètode es crida automàticament).

```
class NomClasse {  
  constructor() {  
    this.tribut=...;  
    this.tribut2=...;  
  }  
  metode1(){...}  
  metode2(){...}  
}  
  
let variable= new NomClasse();  
variable.tribut2; //accedir a tribut  
variable.metode1(); //cridar a un metode
```

Exemple de classe: client d'un banc.

Un client d'un banc té un compte bancari amb un saldo. Les operacions que pot fer sobre el compte es crear-lo (constructor), depositar i extraure diners.

```
class client {  
  constructor(nom,saldo) {  
    this.nom=nom;  
    this.saldo=saldo;  
  }  
  
  depositar(diners){  
    this.saldo=this.saldo+diners;  
  }  
  
  extraure(diners){  
    this.saldo=this.saldo-diners;  
  }  
}
```

```
}  
}
```

this ----- fa referencia al objecte. Per tant, this.at fa referencia a l'atribut at de l'objecte que definim.

```
<script language="javascript">  
  var client1;  
  client1=new client('didac',1200);  
  document.write('Nom del client:'+client1.nom+'<br>');  
  document.write('Saldo actual:'+client1.saldo+'<br>');  
  client1.depositar(120);  
  document.write('Nou saldo---->'+client1.saldo+'<br>');  
  client1.extraure(1000);  
  document.write('Nou saldo---->'+client1.saldo+'<br>');  
</script>
```

Exemple complet

```
<!doctype html>  
<html lang="es">  
<head>  
  <meta charset="utf-8">  
  <title>titol</title>  
  <meta name="description" content="HTML5">  
  <meta name="author" content="Jose Masip">  
  <link rel="stylesheet" href="css/estilos.css">  
  
<script language="javascript">  
  class client {  
    constructor(nom,saldo) {  
      this.nom=nom;  
      this.saldo=saldo;  
    }  
  
    depositar(diners){  
      this.saldo=this.saldo+diners;  
    }  
  
    extraure(diners){  
      this.saldo=this.saldo-diners;  
    }  
  }  
</script>
```

```
var client1;  
client1=new client('didac',1200);  
document.write('Nom del client:'+client1.nom+'<br>');  
document.write('Saldo actual:'+client1.saldo+'<br>');  
client1.depositar(120);  
document.write('Nou saldo---->'+client1.saldo+'<br>');  
client1.extraure(1000);  
document.write('Nou saldo---->'+client1.saldo+'<br>');  
</script>  
</head>  
<body>  
  
</body>  
</html>
```

2.4. Propietats d'un event

Podem detectar quina tecla hem polsat consultant propietats de l'event, coordenades en que hem provocat l'event, etc. Aquesta informació l'obtenim de l'objecte event.

```
event.keyCode  
event.charCode  
event.shiftKey  
event.clientX  
event.screenX
```

Recordar que cal definir un event sobre l'element (onkeypress, onkeydown,).

2.5. DOM.

Quan es carrega una pàgina web, el navegador crea un «**DOCUMENT Object Model**» de la pàgina.

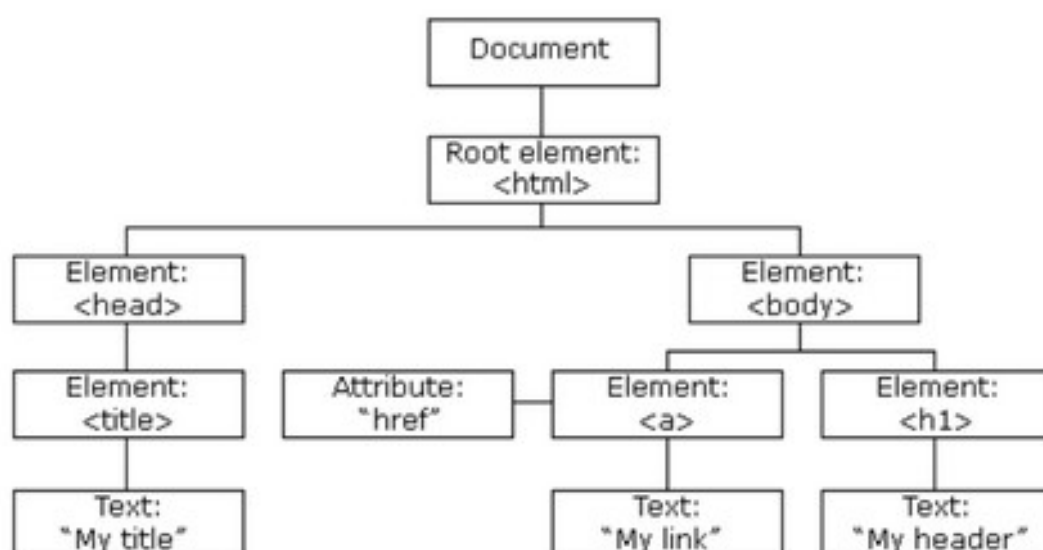
Defineix un estàndard per a accedir i manipular documents HTML.

Presenta un document html com una estructura d'arbre.

"El documento de W3C Object Model (DOM) es una interfaz de plataforma y lenguaje neutro que permite que los programas y scripts acceder y actualizar dinámicamente el contenido, la estructura y el estilo de un documento".

Els nodes d'un arbre tenen una relació jeràrquica entre sí.

- En un arbre de nodes, el node superior s'anomena arrel.
- Cadascun dels nodes té exactament un pare, excepte l'arrel (que no té pare)
- Un node pot tindre qualsevol número de fills
- Els germans son els nodes amb el mateix pare.



2.6. Número de fills

Amb el vector `childNodes` podem saber el número de fills i accedir a ells.

```
<!doctype html>
<html lang="es">
<head>
  <meta charset="utf-8">
  <title>Plantilla</title>
  <meta name="description" content="descripcio">
  <meta name="author" content="elteunom">
  <link rel="stylesheet" href="css/estilos.css">
</head>

<body>
  <h1>DOM</h1>
  <div id="capa">
    <p style="fontSize:10px">p1</p>
    <p style="fontSize:10px">p2</p>
  </div>
  <script language="JavaScript">
    alert("numero fills"+document.getElementById("capa").childNodes.length);
  </script>

  <p>tots navegadors generen mateix numero de fills?</p>
</body>
</html>
```

Si modifiquem la capa de la manera mostrada a continuació i mostrem el número de fills. **És el mateix?**

```
<div id="capa"><p>p1</p><p>p2</p></div>
```

Al següent exemple d'utilització del vector de fills augmentem la mida del text dels paràgrafs:

```
<script language="JavaScript">
function tamanytext(){
  var obj=document.getElementById('capa');
  text=14;
  for(f=0;f<obj.childNodes.length;f++)
  {
    obj.childNodes[f].style.fontSize=text+'px';
    text=text+4;
  }
}
</script>
```

Si deixem l'opció del div en vots de línia per a cadascun dels paràgrafs, aquesta solució genera un error. Si utilitzem sense espais ni vots de línia funciona correctament.

2.7. parentNode

Permet accedir al pare d'un node.

Sobre l'exemple anterior:

```
var paragrafs=document.getElementsByTagName('p');
var pare=paragrafs[0].parentNode; //estariem en el div
```

2.8. nodeType.

Aquesta propietat mostra el tipus d'un node.

Node type	Description	Children
1 Element	Represents an element	Element, Text, Comment, ProcessingInstruction, CDATASection, EntityReference
2 Attr	Represents an attribute	Text, EntityReference
3 Text	Represents textual content in an element or attribute	None
4 CDATASection	Represents a CDATA section in a document (text that will NOT be parsed by a parser)	None
5 EntityReference	Represents an entity reference	Element, ProcessingInstruction, Comment, Text, CDATASection, EntityReference
6 Entity	Represents an entity	Element, ProcessingInstruction, Comment, Text, CDATASection, EntityReference
7 ProcessingInstruction	Represents a processing instruction	None
8 Comment	Represents a comment	None
9 Document	Represents the entire document (the root-node of the DOM tree)	Element, ProcessingInstruction, Comment, DocumentType
10 DocumentType	Provides an interface to the entities defined for the document	None
11 DocumentFragment	Represents a "lightweight" Document object, which can hold a portion of a document	Element, ProcessingInstruction, Comment, Text, CDATASection, EntityReference
12 Notation	Represents a notation declared in the DTD	None

Utilitzant aquesta propietat podem solucionar el problema del vots de línia anterior.

```
<script language="JavaScript">
function fexecutar(){
    var obj=document.getElementById('capa');
    text=14;
    for(f=0;f<obj.childNodes.length;f++)
    {
        //alert(obj.childNodes[f].nodeType);
        if(obj.childNodes[f].nodeType==1){
            obj.childNodes[f].style.fontSize=text+'px';
            text=text+4;
        }
    }
}
</script>
```

2.9. Crear Nodes.

Depenent del tipus de node que volem crear tenim que utilitzar:

- `appendChild()`. Annexa el node passat com a paràmetre a l'objecte sobre el que executem.
- `createElement()`. Crea un node que representa una etiqueta.
- `createTextNode()`.
- `cloneNode()`. Clona un node

Exemple on creem un paràgraf:

```
var para=document.createElement("p"); // crear etiqueta
var node=document.createTextNode("NOU PARÀGRAF."); //crear text
para.appendChild(node); // afegir el text al paràgraf

document.body.appendChild(para); // afegir el paràgraf al body
```

Altra funció que ens podria servir es `insertBefore()`.

2.10. Eliminar Nodes.

Per a eliminar un node cal utilitzar:

- `removeChild()`. elimina el node passat com a paràmetre.

Exemple on eliminem un paràgraf:

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

document.body.removeChild(document.body.childNodes[document.body.childNodes.length-1])
```

2.11. Atributs.

Per a crear o eliminar un atribut cal utilitzar:

- `setAttribute()`. Crea el atribut passat com a primer paràmetre amb el valor passat com a segon.
- `removeAttribute()`. Elimina l'atribut passat com a paràmetre.
- `getAttribute()`. Obte el valor de l'atribut.

Exemple :

```
Node.setAttribute('id','hiper');
node.getAttribute('id');
node.removeAttribute('id')
```

DOM

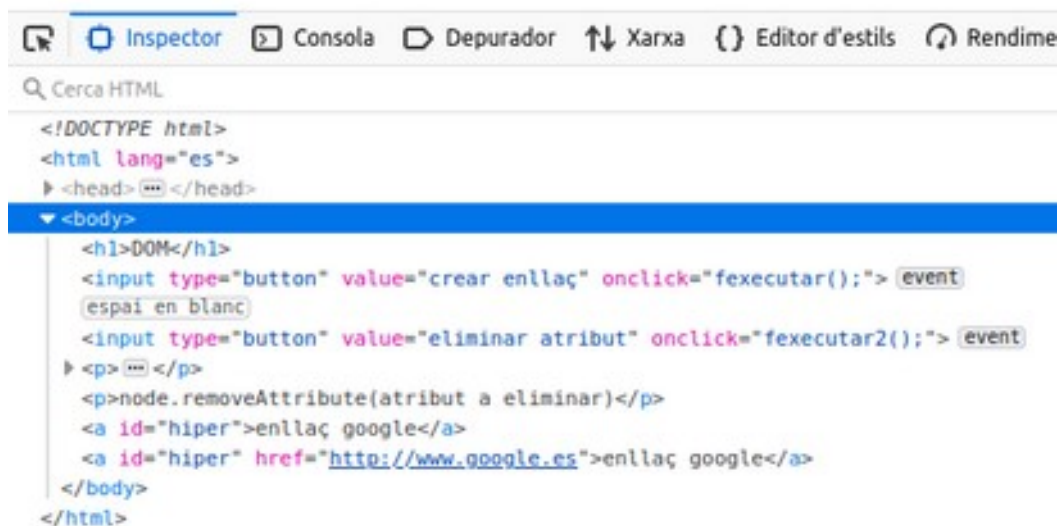
crear enllaç

eliminar atribut

node.setAttribute(atribut a definir, valor assignem)

node.removeAttribute(atribut a eliminar)

enllaç google [enllaç google](#)



2.12. Events.

Per a crear o eliminar un event cal utilitzar:

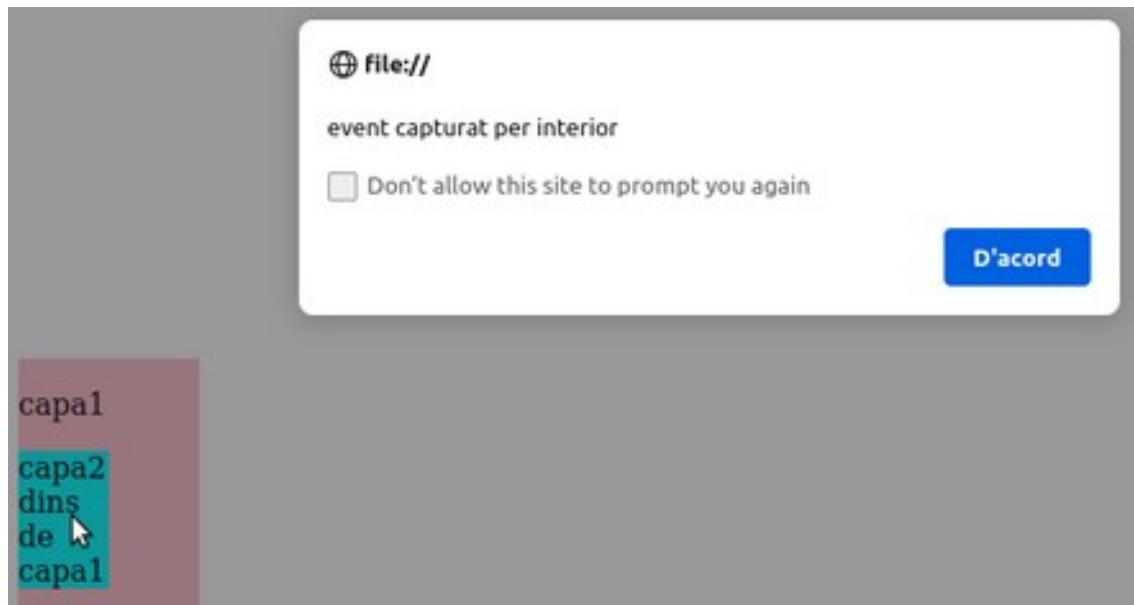
- `addEventListener()`. Crea l'event passat com a primer paràmetre que executa la funció passada com a segon paràmetre. Amb el tercer paràmetre indiquem l'ordre en que es captura l'event.
- `event.target`. Ens indica l'objecte genera event
- `event.type`. Ens indica el nom de l'event

- event.button. Ens indica quin boto del ratolí hem pulsat (0 esquerra, 1 mig, 2 dreta)
- event.stopPropagation(). no es propagaria l'event als demes objectes.

Exemple:

```
<!doctype html>
<html lang="es">
<head>
  <meta charset="utf-8">
  <title>Plantilla</title>
  <meta name="description" content="descripcio">
  <meta name="author" content="Jose Masip">
  <link rel="stylesheet" href="css/estilos.css">
  <script language="JavaScript">
    window.addEventListener('load',ini,false);
    function ini(){
//interior i despres es propaga a la exterior
document.getElementById('capa1').addEventListener('click',fexecutar,false);
document.getElementById('capa2').addEventListener('click',fexecutar2,false);
    }

    function fexecutar(){
      alert("event capturat per exterior")
    }
    function fexecutar2(e){
      alert("event capturat per interior");
    }
  </script>
</head>
<body>
  <h1>DOM</h1>
  qui captura l'event?
  <div id="capa1" style="background-color:pink;position:absolute;width:100px;top:400px;">
    <p>capa1</p>
    <div id="capa2" style="background-color:cyan;width:50px;">
      <p>capa2 dins de capa1</p>
    </div>
  </div>
</body>
</html>
```



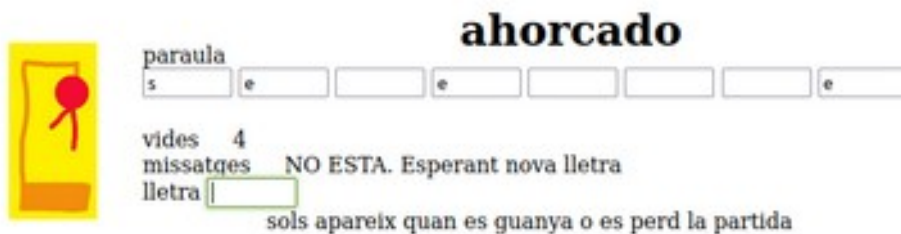
Si modifiquem el false per un true, aconseguirem que primer capture l'event la capa exterior i després la interior.



Si a la funció que s'executa primer afegim el `e.stopPropagation()` provocarem que l'event no es propague i la segon funció no s'execute.

3. Activitats.

- E01. Crea un ahorcado:



- Les paraules per al joc les traguem d'un vector de 8 posicions d'on es seleccionerà una aleatòriament per a cadascuna de les partides.
 - Les paraules son de longitud variable (max 15)
 - una lletra pot aparéixer mes d'una volta en la paraula.
 - paraules: moble, cua, novembre, porta, setembre, quadre, quatre, hipervincle.
 - al fallar 7 voltes el joc acaba i es dona la possibilitat de tornar a escomençar mostrant la paraula de la sol·lucio. es mostra en el camp vides, les opcions que et queden
 - Si existeix la lletra en alguna posicio, escriurela en la casella corresponent i es mostra un missatge en el camp felicitant al jugador. si no existeix la lletra, es resta una vida i s'indica l'error.
- E02. Crea una pàgina amb la següent funcionalitat:
 - crea un contador del número de voltes entres a la pagina en un minut.
 - Si no eixsteix la cookie, hi ha que crear-la.
 - si ja existeix, incrementar el número de visites.
 - Mostrar el nùmero de visites actual dins d'un h1.

- E03. Crea una pàgina amb la següent funcionalitat:
 - crea una cistella de la compra amb cookies.
 - utilitza una (o mes) cookie per a emmagatzemar els productes i les unitats seleccionades per a la compra.
 - Incorporà un boto per a eliminar o buidar el carrito.
- E04. Modifica l'exercici 01 amb aquesta funcionalitat:
 - No es pot repetir una lletra. Si he dit la «a» i aquesta ja havia sigut utilitzada restarà una vida.
 - Mostrar un camp d'ajuda on es mostren les lletres que ja hem dit.
 - Les lletres les detectem automaticament al polsar el teclat sobre qualsevol lloc de la pantalla.



- E05. Crea una pàgina amb DOM amb la següent funcionalitat:

DOM

Crea 3 paragrafs. afegeix el paragraf al principi

NOU PARAGRAF.

p1

p2

p3

executar

- Afegir un paràgraf al principi de la capa.

- E06. Crea una pàgina amb DOM amb la següent funcionalitat:

DOM

Crea 3 paragrafs. eliminar el primer paragraf. Crear al final

p2

p3

NOU PARAGRAF 0.

NOU PARAGRAF 1.

crear

eliminar

- Eliminar el primer paràgraf.
- Afegir al final numerant el número de paràgraf creat.

- E07. Crea una pàgina amb DOM amb la següent funcionalitat:

DOM

`node.setAttribute(atribut a definir, valor assignem)`

`node.removeAttribute(atribut a eliminar)`

enllaç google [enllaç google](#)



- Crea un hipervincle amb un identificador al polsar el boto.
- Elimina l'atribut href al polsar el boto corresponent.

- E08. Crea una pàgina amb DOM amb la següent funcionalitat:

The image shows two examples of a web form. The left example shows a dropdown menu with the value '3' selected, a button labeled 'crear formulari', and three input fields labeled 'camp 1', 'camp 2', and 'camp 3'. The right example shows a dropdown menu with the value '6' selected, a button labeled 'crear formulari', and six input fields labeled 'camp 1' through 'camp 6'.

- Crea un select amb X valors (variable al principi del document).
 - Crea tants camps com el número seleccionat al camp anterior.
 - Aquest camps cal que siguin funcionals per a rebre's quan enviarem el formulari.
- E09. Crea una pàgina amb DOM amb la següent funcionalitat:

☐ obrir en un altra pagina

[google](#)

[yahoo](#)

[terra](#)

- Si marquem la casella de verificació obrim l'enllaç a una altra pestanya.
 - Si no està marcada obrim a la finestra actual.
 - El número d'hipervincles pot variar i funcionarà exactament igual el nostre codi.

- E10. Crea una pàgina amb DOM amb la següent funcionalitat:

[google](#)

[yahoo](#)

[terra](#)

[google](#)

[yahoo](#)

[terra](#)

[google](#)

[yahoo](#)

[terra](#)

[obrir tots](#)



- Si fem click sobre un enllaç, obrim aquest.
- Si fem click sobre «obrir tots» obrim tots els enllaços a distintes pestanyes (cal evitar que aquest hipervincle ens redirigisca a altra pàgina).
- El número d'hipervincles pot variar i funcionarà exactament igual el nostre codi.

- E11. Modifica el «ahorcado» fet anteriorment per a:
 - Els camps de les lletres els creem amb DOM. Sols crearem els camps necessaris per a la paraula de la partida. Es a dir:
 - si la paraula es «cua» crearem 3 camps.
 - Si la paraula es «porta» crearem 5 camps.

4. Activitats entregables.

Entrega les següents activitats:

- E03
- E10
- E11

5. Bibliografia.

- <https://www.w3schools.com/js/>

6. Aclariments

No intentes copiar exercicis d'altres companys o internet. ni tans sols pegar una miradeta. Si que pots consultar la sintaxis d'una funció, etc. Però no et serveix per a res intentar buscar la solució.

És el major error que pot realitzar qui comença a programar. Sols et deixarà una falsa sensació d'aprenentatge i després no sabràs solucionar els problemes per tu mateix.

Cal intentar que el resultat que genere el vostre codi siga el mes paregut possible a l'exercici (si hi ha imatge d'exemple).

Com els Navegadors no son compatibles entre sí al 100% jo corregire els examens amb FIREFOX. Per tant, es recomanable utilitzar aquest per als nostres exercicis.

Crea uns arxius per a cada exercici o apartat. Així un error a un no afectarà a la resta.

Per a entregar els exercicis crear una carpeta amb el vostre nom i compriu aquesta (el nom de l'arxiu comprimit serà elteunom.zip o elteunom.tar.gz). Si entregueu arxius solts o arxius genèrics sense identificar el vostre nom puc confondre'm i no puntuar-vos algun exercici al no saber de qui és.

Cal entregar tots els arxius necessaris per a l'exercici (fotos, llibreries, etc).