

DAW CFGS Desenvolupament d'aplicacions web
Mòdul 6 bloc 1 – Desenvolupament web en entorn client
UF2 – Estructures definides pel programador. Objectes
U2 – Estructures definides pel programador
EAC2
(Curs 2020–21 / 1r semestre)

Presentació i resultats d'aprenentatge

Aquest exercici d'avaluació continuada (EAC) es correspon amb els continguts treballats a la unitat 2, Estructures definides pel programador.

El **resultats d'aprenentatge** que es planteja és:

- Programa codi per a clients web analitzant i utilitzant estructures definides per l'usuari.

Criteris d'avaluació

La puntuació màxima assignada a cada activitat s'indica a l'enunciat.

Els criteris que es tindran en compte per avaluar el treball de l'alumnat són els següents:

- Classifica i utilitza les funcions predefinides del llenguatge.
- Crea i utilitza funcions definides per l'usuari.
- Reconeix les característiques del llenguatge relatives a la creació i ús d'arrays.
- Crea i utilitza arrays.
- Depura i documenta el codi.


Forma i data de lliurament

Per respondre les qüestions heu d'utilitzar aquest mateix fitxer. Elimineu els apartats *Presentació i resultats d'aprenentatge*, *Criteris d'avaluació* i *Forma i data de lliurament*.

Un cop finalitzat l'exercici d'avaluació continuada heu d'enviar el document des de l'apartat **M6B1 Lliurament EAC2** de l'aula, dins del termini establert. Tingueu en compte que el sistema no permetrà fer lliuraments després de la data i hora indicades.

El nom del fitxer serà el següent: **DAW_M06B1_EAC2_Cognom1_Inicial del cognom2**. Els cognoms s'escriuran sense accents. Per exemple, l'estudiant *Joan García Santos* posaria el següent nom al seu fitxer de l'EAC2: **DAW_M06B1_EAC2_Garcia_S**.

Substituïu **Nom i cognoms** de la capçalera per les vostres dades personals.

	Codi: I71	Exercici d'avaluació contínua 2	Pàgina 1 de 10
	Versió: 01	DAW_M06B1_EAC2_Enunciat_2021S1	Lliurament: 18/11/2020

El termini de lliurament finalitza a les **23:55 h** del dia **18/11/2020**. La proposta de solució de l'EAC es publicarà el dia 20/11/2020 i les qualificacions el dia 26/11/2020.

Enunciat

Per publicar la solució heu d'utilitzar aquest mateix model, eliminant els apartats *Presentació i resultats d'aprenentatge*, *Criteris d'avaluació* i *Forma i data de lliurament* i substituint *Enunciat* i *Nom i cognoms* pels vostres.

Cal lliurar un fitxer comprimit amb format .zip que contindrà:

- Les respostes als exercicis de l'apartat 1 en un fitxer que s'anomenarà:
DAW_M06B1_EAC2_Cognom1_Inicial del cognom2.odt

Aquest fitxer serà el mateix que conté aquest enunciat, però, amb les respostes als exercicis de l'apartat 1 i eliminant els apartats indicats al començament d'aquesta secció.

- Els fitxers **exercici2.js** i **doc2.zip**, necessaris per resoldre l'exercici 2.

APARTAT 1 – Funcions (4 punts)

Als exercicis d'aquest apartat es treballa amb una pàgina web que té l'estructura indicada a l'exercici 1.1.: una part HTML que només conté la part JavaScript (tancada entre les etiquetes <script> i </script>). Aquesta és la part que fa tota la funcionalitat de la pàgina i presenta els resultats.

Aquest format s'utilitza per simplicitat: us permet provar directament el codi amb "copiar i enganxar" i, com que no hi ha contingut HTML, no és tan important separar els dos codis en fitxers diferents.

1.1. (0,8 punts) Completeu el codi del següent fitxer HTML perquè, en obrir la pàgina web, demani un nom de persona en una finestra emergent, una data també en una nova finestra emergent i tot seguit mostri en una alerta el nom d'usuari de la següent manera:

- El nom d'usuari comença per les dues primeres lletres del nom, seguit de l'última lletra del nom
- Després s'hi ha d'afegir el mes de la data de naixement.
- No cal validar les dades, i ens asseguren que les dates s'entraran en el format dd/mm/aa (separades per barres).
- Per exemple, si el nom és Núria i la data de naixement és 20/05/90, el nom d'usuari a generar és Nua05.

Codi:


```
<!DOCTYPE html>
<html lang="ca">

<head>
  <meta charset="utf-8">
  <title>Generació Usuari</title>
  <script>
    <!-- Afegiu el codi aquí -->
  </script>
</head>

<body>
</body>

</html>
```

Resposta:

	Codi: I71	Exercici d'avaluació contínua 2	Pàgina 2 de 10
	Versió: 01	DAW_M06B1_EAC2_Enunciat_2021S1	Lliurament: 18/11/2020

1.2. (0,8 punts) Teniu la següent pàgina web:

```
<!DOCTYPE html>
<html lang="ca">
  <head>
    <meta charset="utf-8">
    <title>Cub</title>
  </head>
  <body>
    <script>
      var x=5;
      var y;
      function cub(x){
        x=x*x*x;
        return x;
      }

      y=cub(x);
      alert(x);

    </script>
  </body>
</html>
```

a) En obrir-la amb el navegador, què mostra l'*alert*?

b) Modifica la funció cub, en la línia `x=x*x*x`; per `this.x=x*x*x`; Que apareix ara a l'alerta?

c) Explica per què en l'apartat *a* es mostra un resultat, i al *b* se'n mostra un altre. Explica també, a què fa referència el *this* que s'ha afegit a l'apartat anterior.

1.3. (0,8 punts) Teniu la següent pàgina web:


```
<!DOCTYPE html>
<html lang="ca">
  <head>
    <meta charset="utf-8">
    <title>Funcions</title>
  </head>
  <body>
    <script>
      var x=10;
      var y=20;
      function f1(y){
        x=x+y;
        f2();
        function f2(){
          y=y+9;
        }
      }
      f1(x);
      alert(x);
      alert(y);
    </script>
  </body>
</html>
```

a) Què s'escriu per pantalla? Raona la resposta.

b) Què hauríem de modificar a f2 perquè l'alerta final mostrés 19? Raona la resposta.

1.4. (0,8 punts) Teniu la següent pàgina web:

```
<!DOCTYPE html>
<html lang="ca">
  <head>
    <meta charset="utf-8">
    <title>Disc</title>
  </head>
  <body>
    <script>
      var x=new Producte("planxa",40);
      function Producte(n,p){
        this.nom=n;
        preu=p;
        iva=function(){preu=preu*1.21;}
      }
      alert(x.nom);
      alert(x.preu);
      x.iva();
      alert(x.preu);
    </script>
  </body>
</html>
```

	Codi: I71	Exercici d'avaluació contínua 2	Pàgina 4 de 10
	Versió: 01	DAW_M06B1_EAC2_Enunciat_2021S1	Lliurament: 18/11/2020

a) En obrir-la amb el navegador, què mostra l'alerta? Què es mostra per consola?

b) Quins canvis caldria fer a la funció *Producte* perquè en la segona alerta es vegi el preu?

c) Quins canvis s'haurien de fer perquè el mètode iva deixi de fallar, i augmenti el preu del producte correctament?

1.5. (0,8 punts) Teniu la següent pàgina web:

```
<!DOCTYPE html>
<html lang="ca">
  <head>
    <meta charset="utf-8">
    <title>Torneig</title>
  </head>
  <body>
    <script>
function Jugador(cognom,ranquing){
  this.cognom=cognom;
  this.ranquing=ranquing;
}

var torneig=[];
torneig.push(new Jugador("Aldana",7));
torneig.push(new Jugador("Hill",12));
torneig.push(new Jugador("Santana",6));
torneig.push(new Jugador("Benet",12));
torneig.push(new Jugador("Houari",5));
torneig.push(new Jugador("Pujades",6));

//EL CODI VOSTRE AQUÍ
function criteri(j1,j2)
{
  if(j1.ranquing==j2.ranquing) return j2.cognom<j1.cognom;
  return (j1.ranquing-j2.ranquing);
}
function criteri2(j1)
{
  if(j1.ranquing<=10) return true;
}
torneig=torneig.sort(criteri);
for(i=0;i<torneig.length;i++)
{
  alert(torneig[i].cognom+", "+torneig[i].ranquing)
}
torneig=torneig.filter(criteri2);
for(i=0;i<torneig.length;i++)
{
  alert(torneig[i].cognom+", "+torneig[i].ranquing)
}
    </script>
  </body>
</html>
```

```
</script>  
</body>  
</html>
```

a) Escriviu el codi de la funció *crteri1* perquè el mètode *sort()* ordeni els elements de l'array per rànkung de petit a gran, i en cas d'empat al rànkung, per cognom, també de petit a gran (alfabèticament). Podeu modificar el nombre de paràmetres que rep la funció.

b) Escriviu el codi de la funció *crteri2* perquè el mètode *filter()* retorni només els jugadors fins al rànkung 10 inclòs. Podeu modificar el nombre de paràmetres que rep la funció.

APARTAT 2 – Arrays (6 punts)

2. (6 punts: 5 punts per l'aplicació + 1 punt per la documentació) Completeu l'aplicació compresa al fitxer exercici2.zip, annex a aquest enunciat i genereu la documentació del codi resultant.

Un cop descomprimiu la carpeta, trobareu aquesta estructura de directoris:

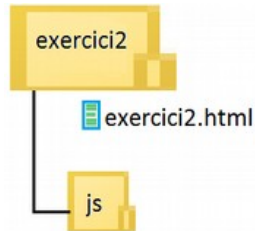


Figura 1

Als fitxers exercici2.html hi ha la pàgina web de l'aplicació que heu de completar.

Quan hàgiu acabat l'exercici, en obrir exercici2.html al navegador, us trobareu amb una pantalla amb un aspecte similar a aquest:

Caçadors de bolets

Gestió de Municipis

La seu

Gestió d'especies


Espècie

Resultats de la caça

Municipi Espècie Molts ☐ Bastants ☐ Pocs ☐

Extracció de resultats

Diuen els experts que aquest any serà una bona temporada de bolets. Per poder gestionar les múltiples sortides que fem a la muntanya, volem una aplicació per enregistrar els llocs on hem anat, les espècies de bolets que hem trobat, i la quantitat que n'hem trobat cada cop. A partir d'aquí, volem poder extreure uns llistats, que ens ajudin a decidir la propera sortida.

	Codi: I71	Exercici d'avaluació contínua 2	Pàgina 7 de 10
	Versió: 01	DAW_M06B1_EAC2_Enunciat_2021S1	Lliurament: 18/11/2020

Així doncs, aquesta aplicació, un cop acabada, ha de mantenir un array de troballes (quantitat i tipus de bolets que heu trobat en un lloc concret). A cada posició de l'array, es guardarà, separat per arroves @, el municipi, l'espècie i la quantitat (1-molts, 2-bastants, 3-pocs). Si per exemple, trobem molts Rovellons, a Vic, l'emmagatzemareu així: "Vic@Rovello@1".

Si us fixeu en la captura de l'aplicació, veureu quatre seccions ben definides: la de **gestió de municipis**, la de **gestió d'espècies**, la de **resultats de la caça (les troballes)**, i la **d'extracció de resultats**. En la primera part, haureu de donar d'alta i eliminar municipis. En la segona, haureu de donar d'alta i eliminar espècies. En la tercera, seleccionarem el municipi, l'espècie, i la quantitat, i ho guardarem al vector. La darrera part ens permetrà mostrar resultats.

- Gestió dels Municipis: control dels municipis.


- **Nou Municipi:**
 - Funcionalitats a implementar:
 - Afegeix un nou municipi al desplegable de municipis
 - Es controlen els errors i es mostra un missatge en una finestra emergent:
 - Si no s'ha introduït nom.
 - En cas que el municipi ja estigui donat d'alta.
- **Elimina Municipi:**
 - Funcionalitats a implementar:
 - Esborra totes les troballes de l'array de troballes corresponents a aquell municipi, i indica quantes troballes s'han eliminat.
 - Esborra el municipi del desplegable de municipis.
 - Es controlen els errors i es mostra un missatge en una finestra emergent:
 - Si no s'ha introduït nom.
 - Si el municipi no existeix.

- Gestió d'espècies: control de les espècies de bolets.

- **Nova Espècie:**
 - Funcionalitats a implementar:
 - Afegeix una nova espècie al desplegable d'espècies.
 - Es controlen els errors i es mostra un missatge en una finestra emergent:
 - Si no s'ha introduït nom.
 - En cas que l'espècie ja estigui donada d'alta.
- **Elimina Espècies:**
 - Funcionalitats a implementar:
 - Esborra totes les troballes de l'array de troballes corresponents a aquella espècie, i indica quantes troballes s'han eliminat.
 - Esborra l'espècie del desplegable d'espècies.
 - Es controlen els errors i es mostra un missatge en una finestra emergent:
 - Si no s'ha introduït nom.
 - Si l'espècie no existeix.

- Resultats de la caça: alta de registres al vector troballes.

- **Afegir Troballa:**
 - Funcionalitats a implementar:
 - Afegeix una troballa a l'array de troballes, amb el municipi, l'espècie i la quantitat trobada.
 - Els camps han d'anar separats per una arrova, com s'indica al principi de l'exercici.
 - Es recomana que guardeu un 1 pel "molts", un 2 pel "bastants" i un 3 pel "pocs", ja que us anirà bé pel següent apartat.
 - Es controlen els errors i es mostra un missatge en una finestra emergent:
 - Si no s'ha seleccionat la quantitat.
 - Si al desplegable de municipis o espècies no hi ha cap opció.

	Codi: I71	Exercici d'avaluació contínua 2	Pàgina 8 de 10
	Versió: 01	DAW_M06B1_EAC2_Enunciat_2021S1	Lliurament: 18/11/2020

- **Extracció de resultats:** secció destinada a mostrar les dades introduïdes de tres maneres diferents. Els resultats es mostren a la divisió anomenada llistats.

- **Llista per espècies:** Genera un llistat ordenat per espècies ordenades alfabèticament.
- **Llista per quantitat:** Genera un llistat ordenat per quantitat on primer apareixen els que n'hi ha molts, després els que n'hi ha bastants i per últim el que n'hi ha pocs.
- **Llista ordenada per població espècie i quantitat:** Genera el llistat ordenat pels tres camps, primer ordenat per població, dins de la mateixa població ordenat per espècie, i dins de l'espècie ordenat per quantitat (primer molts, després bastants i per últim pocs).

La vostra feina consistirà a construir el fitxer `./js/exercici2.js` perquè l'aplicació realitzi totes les tasques indicades i generar la documentació del codi.

En concret, el fitxer `exercici2.js` haurà de tenir:

- La declaració del vector `troballes` esmentat abans. No pot tenir més variables globals que les que ja apareixen al fitxer `exercici2.js`
- La declaració de la funció `inicialitza()`, que es crida en carregar el formulari i inicialitza les variables i fa que es mostri la pantalla tal com veieu a la captura.
- La declaració de les funcions explicades al punt anterior (el nom de les quals trobareu dins del fitxer `exercici2.HTML`)
- La declaració de les funcions addicionals que necessiteu, com per exemple una per dibuixar els vectors, una per cercar si existeix una espècie o no, una per afegir un element a un desplegable...)
- No cal dir que s'espera que utilitzeu amb difusió les funcions que incorpora JavaScript, sobretot les referides a `arrays`.

Haureu de lliurar el fitxer **exercici2.js** dins del comprimit, tal com s'ha indicat a l'inici de l'enunciat.

Vídeo del programa

A l'enllaç següent trobareu un vídeo on podeu veure el funcionament del programa que se us demana:

https://www.loom.com/share/9ec66041aa534e4c9f724a6f4115f0f2?from_recorder=1


Per documentar el codi de l'aplicació, fareu servir l'eina JSDoc. Per utilitzar-la, cal posar en el codi comentaris similars als que es posen a Java per generar els javadoc. Un cop posats, només cal executar el programa i generar la documentació. El resultat d'aquesta execució és un fitxer comprimit que lliurareu amb el nom `doc_eac2ex-.zip`, canviant "-" pel número de l'exercici documentat.

A continuació trobareu una sèrie d'indicacions i enllaços per generar la documentació:

- Documentació oficial (en anglès): <http://usejsdoc.org/> A l'enllaç [Getting Started with JSDoc 3](#) hi trobareu el més bàsic. A la secció de la pàgina principal "Block Tags" trobareu la referència de totes les anotacions que poden posar-se en aquests comentaris. També hi trobareu un enllaç al projecte a github.
- Instal·lació de JSDoc. JSDoc s'ha escrit en JavaScript. Per tant, caldrà un motor de JavaScript per executar-lo. Concretament, JSDoc és compatible amb Node.js. Node.js és un entorn basat en el motor de JavaScript V8, elaborat per Google. Aquest entorn permet executar programes JavaScript autònoms. Sol utilitzar-se per executar servidors web -al bloc 2 d'aquest mòdul es tracten alguns casos-, però també pot executar programes autònoms, com JSDoc. Les passes per utilitzar-lo són:

1) Instal·lar Node.js:

- a) En Windows o Mac Os: cal descarregar l'instal·lador de nodejs.org/en/download i executar-lo. És important assegurar-se que s'instal·la la característica `npm`.

	Codi: I71	Exercici d'avaluació contínua 2	Pàgina 9 de 10
	Versió: 01	DAW_M06B1_EAC2_Enunciat_2021S1	Lliurament: 18/11/2020

- b) En un sistema GNU Linux basat en Debian o Ubuntu podeu instal·lar-lo executant directament les ordres:

```
sudo apt-get update  
sudo apt-get install nodejs
```

- c) Per altres distribucions de GNU Linux podeu consultar aquest enllaç:

<https://nodejs.org/es/download/package-manager/>

Per verificar que s'ha instal·lat tot correctament, podeu executar les següents ordres, que mostren la versió instal·lada de cada programa:

```
nodejs -v  
npm -v
```

Si la versió de Nodejs instal·lada no contingués la utilitat npm (és el gestor de paquets) caldria executar:

```
apt-get install npm
```

En aquest [enllaç](#) del material podeu trobar el procés amb més detall.

- 2) Instal·lar JSDOCS: amb la instrucció `npm install -g jsdoc`

- 3) Generar la documentació: cal executar l'ordre que hi ha a continuació. Els paràmetres són en **blau**. El paràmetre *origen* tant pot referir-se a un únic fitxer .js com a una carpeta que contingui els fitxers font.


```
jsdoc origen -d carpeta-destí
```

Si ometeu l'opció -d, els fitxers es generaran en una subcarpeta del directori actual que es crearà amb el nom d'*out*.

En aquest enllaç podeu trobar més opcions de l'ordre `jsdoc`:

<http://usejsdoc.org/about-commandline.html>

Documentar el codi d'aquesta manera tindrà un pes de 1 **punt** sobre els 6 punts de l'apartat.

	Codi: I71	Exercici d'avaluació contínua 2	Pàgina 10 de 10
	Versió: 01	DAW_M06B1_EAC2_Enunciat_2021S1	Lliurament: 18/11/2020