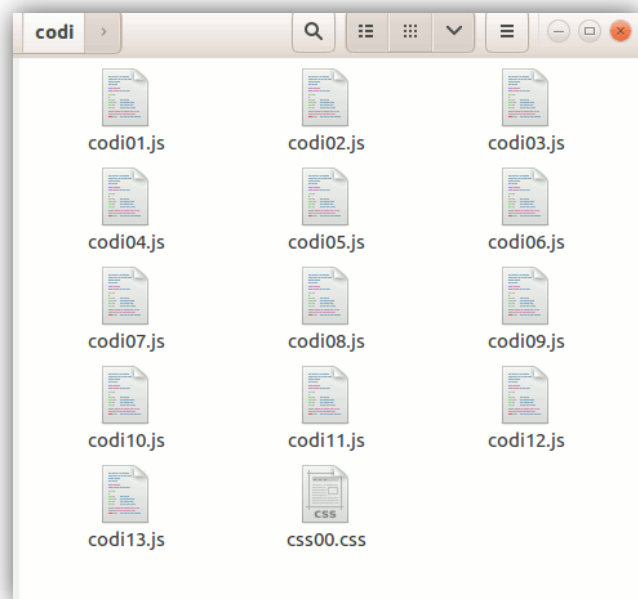


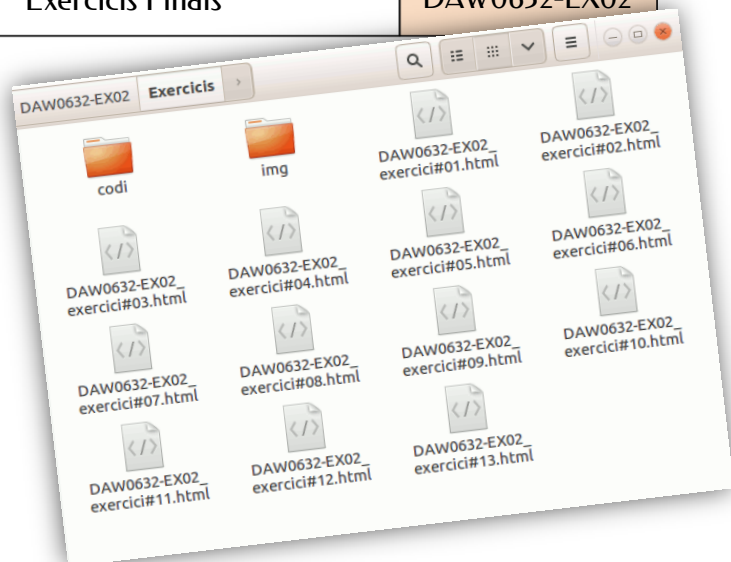


Per fer aquests exercicis utilitzarem els documents **HTML5**, continguts dins la carpeta **/Exercicis**. No podeu afegir res a aquests documents. Ni noves etiquetes, ni cap atribut. És un document intocable. El que heu de fer és afegir un nou arxiu amb el codi **JavaScript** pur (**RAW**), sense cap mena de *framework*. Aquests arxius els heu de posar en la carpeta **/Exercicis/codi**, on ja trobareu l'arxiu **css** anomenat **css00.css**, que tampoc el podeu tocar.



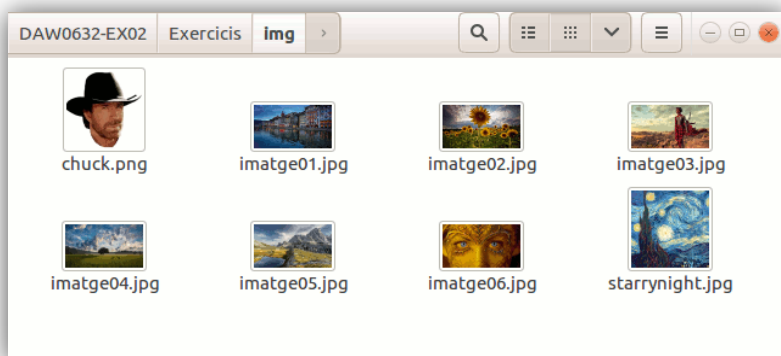
Disposem, també, dels vídeos amb els funcionament dels exercicis per que us sigui més fàcil la comprensió dels enunciats.

Hem d'utilitzar tots els nostres coneixements adquirits fins ara, i especialment els d'aquesta **UF**, el **DOM** i els esdeveniments (**EVENT**).



Segons l'exercici els heu de anomenar com **codi01.js**, **codi02.js**fins a **codi13.js**. Com veieu en la imatge de l'esquerre.

Dins de la carpeta **/Exercicis** també trobareu la carpeta **/img**, on tenim totes les imatges que utilitzarem amb els exercicis.



CONSIDERACIONS BÀSIQUES PER REALITZAR ELS EXERCICIS:

Utilitzeu per a navegar o obtenir els objectes de l'estructura del document, preferentment, els objectes **DOM**: **getElementById**, **getElementsByTagName**, **getElementsByName**, **getElementsByClassName** i molt especialment, els objectes de l'**API HTML5**: **querySelector** i **querySelectorAll** que facilitaran aquesta tasca d'una forma increïble, sobre tot si ja hem realitzat les pràctiques del **Mòdul 9** sobre el marc de treball **jQuery**.

Utilitzarem el mètode estàndard pels esdeveniments i els seus gestors, això vol dir utilitzar el mètode d'esdeveniment **addEventListener** que hem estudiat.

Considerar, per últim, que al fer-se el codi en un arxiu extern, i aquest posar-lo en el **<head>**, haurem d'utilitzar el mètode **load** per carregar, abans de tot, el document **HTML** sencer.

Per exemple:

```
window.onload = function() { // Carreguem tot el document
    ..... codi
    .....
}
```

O la forma més moderna, amb la funció fletxa:

```
window.onload = () => { // Carreguem tot el document
    ..... codi
    .....
}
```

Encara que dins d'aquesta estructura podem afegir definicions de funcions, fora més assenyat fer-ho fora, a l'igual que la definició de variables globals. Per exemple, aquesta seria una bona estructura per fer els exercicis:

```
// Definició de variables globals
var unaVariable,
    unaAltreVariable;

// Definició de funcions
function unaFuncio() {
    .....
}

function unaAltreFuncio(){
    .....
}

// Estructura principal
window.onload = () => { // Carreguem tot el document
    ..... codi
    unaVariable =
    unaFuncio();
    ..... codi
}
```

Considereu també d'utilitzar, si cal, les iteracions per índex o per elements en els conjunts d'elements recollits, per exemple, amb el mètode **getElementsByTagName** o qualsevol altre que retorni un conjunt d'elements.

Això vol dir utilitzar els bucles **for...in** i **for...of**, el primer amb l'índex i el segon amb l'element.

Encara que possiblement sigui més efectiu i elegant utilitzar les múltiples formes del mètode, per matrius, del **forEach** on podem utilitzar els dos paràmetres, **element** i **índex** a la vegada per utilitzar-los com vulguem. La forma més encertada fora aquesta:

```
// Per cadascun dels elements en el conjunt d'objectes
[].forEach.call(objectes,function(element, index){

    ..... codi
    .....

});
```

Hem d'observar que quan el mètode **addEventListener** crida a la funció, aquest li passa l'esdeveniment, així quan definim aquesta funció, i si volem utilitzar aquest esdeveniment, l'hauréu d'afegir. Per exemple:

```
function gestionemEsdeveniment (e){
    ..... codi
    .....
}
```

Amb l'esdeveniment passat podem obtenir l'element que ha cridat a la funció. Per exemple:

```
var element = e.target;
```

també podem accedir a l'atribut **ID** d'aquest element, si aquest en tingués. Per exemple:

```
var id = e.target.id;
```

A més tingueu en compte que si el mètode **addEventListener** és dins de la iteració **forEach** i hem passat tant l'element com l'índex, la forma més idònia de transmetre només l'índex, possiblement, es aquesta:

```
element.addEventListener("click", function(){ // Afegim l'esdeveniment
    return cridaGestor(index); // Cridem al gestor amb l'índex de l'element
}, false);
```

En quan els valors **CSS** d'un element, si volguéssim saber un valor concret d'una regla de full d'estil, no hem d'utilitzar aquesta fórmula:

```
var color = element.style.color; // Obtindrem 'undefined'
```

Això no funcionarà normalment, només si l'estil a estat definit dins de l'etiqueta **HTML**. Per obtenir el valor hauríem d'utilitzar la fórmula següent:

```
var color = window.getComputedStyle(element,null).getPropertyValue("color");
```

En canvi per canviar-la o afegir-la si la podem fer servir. Per exemple:

```
element.style.fontSize = "20px"; // Això funcionarà.
```

Per últim, considereu, si voleu fer un "**reload**" del document, per inicialitzar-lo, podeu fer:

```
window.history.go();
```

DESCRIPCIÓ DELS EXERCICIS:

Fixeu-vos en el vídeo de cada exercici per més comprensió.

01

Feu un nou arxiu i anomeu-lo **codi01.js** i afegiu-lo a la carpeta **\codi**. Aquest **JavaScript** serà utilitzat pel document **HTML** anomenat **DAW0632-EX02_exercici#01.html**.

Hem de construir un detector d'esdeveniments que cridi a una funció quan es faci clic en el botó. Aquesta funció recuperarà els valors actuals dels elements d'entrada del formulari i amb ells construirem una història que enviarem al **div** de sortida **id="sortida"**. Per exemple: "A la Pamela li agraden els cogombres". Fixeu-vos en el botó Neteja Dades.

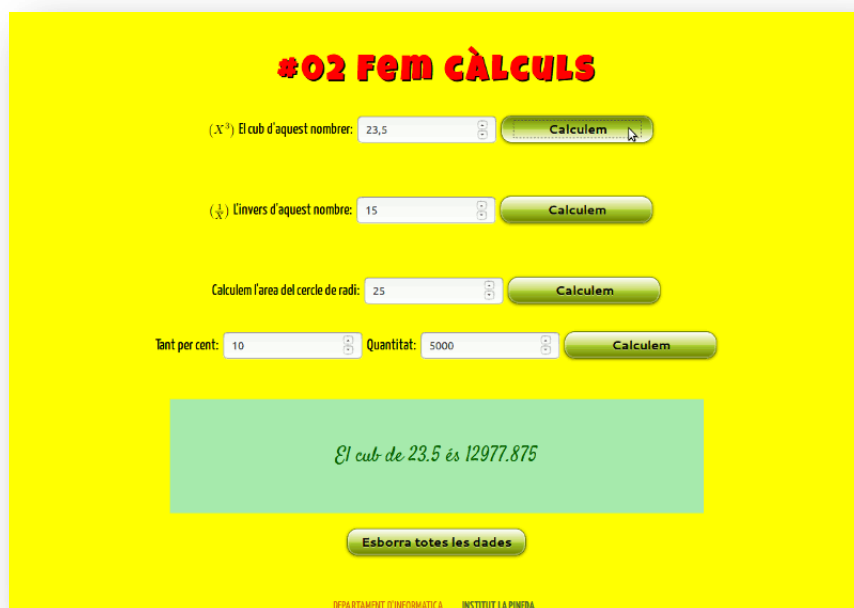


02

Feu un nou arxiu i anomeu-lo **codi02.js** i afegiu-lo a la carpeta **\codi**. Aquest **JavaScript** serà utilitzat pel document **HTML** anomenat **DAW0632-EX02_exercici#02.html**.

Hem de fer una mini calculadora per fer petits càlculs. Heu de crear el mateix gestor d'esdeveniment (molt important) per quan es faci clic en cadascun dels botons. Heu de trobar el valor de l'entrada adequat i mostrar el resultat del càlcul en el **div id="sortida"**.

Fixeu-vos en el botó Esborra totes les dades.



03

Feu un nou arxiu i anomeu-lo **codi03.js** i afegiu-lo a la carpeta **\codi**. Aquest **JavaScript** serà utilitzat pel document **HTML** anomenat **DAW0632-EX02_exercici#03.html**.

Heu de calcular l'Índex de massa corporal (**IMC**). Aquest índex és una xifra que ens permet avaluar la corpulència d'una persona, tot relacionant la seva massa (en Kg.) amb la seva alçada (en metres).

$$IMC = \frac{m}{h^2}$$

On **m** és la massa de la persona i **h** la seva alçada; ambdues introduïdes en unitats del **S.I.**, és a dir, **kg** i **m**. La seva interpretació és la següent (aplicable només a persones adultes, per infants la classificació és diferent):

IMC (Kg./m ²)	Interpretació
menys de 18	magror
18 a 25	corpulència normal
25 a 30	sobrepès
30 a 40	obesitat
més de 40	obesitat mòrbida

Heu de posar el seu **IMC** dins de la etiqueta **** del primer paràgraf, i el seu tipus de cos en la etiqueta **** del segon paràgraf.

04

Feu un nou arxiu i anomeu-lo **codi04.js** i afegiu-lo a la carpeta **\codi**. Aquest **JavaScript** serà utilitzat pel document **HTML** anomenat **DAW0632-EX02_exercici#04.html**

Es tracta d'un convertidor de lliures en kilograms i a l'inversa. Convertiu en la unitat seleccionada al quadre de l'esquerra i convertiu-la en la unitat seleccionada al quadre desplegable de la dreta. La unitat ha de ser representada dins el paràgraf. El factor de conversió de lliures a kilograms és **0.45359237**, i el factor de conversió de kilograms a lliures és **2,20462262**.

#03 CÀLCUL DE L'IMC

Índex de la Massa Corporal

Pes (en kgr.):

Alçada (en metres):

El seu IMC és de 37.037

El tipus de cos és de: obès

DEPARTAMENT D'INFORMÀTICA INSTITUT LA PINEDA

#04 CONVERTIDOR LLIURES/KG.

De lliures a kilograms i a l'inversa

Converteix a

45 kilograms son, aproximadament, 99.208 lliures

DEPARTAMENT D'INFORMÀTICA INSTITUT LA PINEDA

05

Feu un nou arxiu i anomenau-lo **codi05.js** i afegiu-lo a la carpeta **\codi**. Aquest **JavaScript** serà utilitzat pel document **HTML** anomenat **DAW0632-EX02_exercici#05.html**

Es tracta de retornar el canvi de la quantitat introduïda en euros (€) en bitllets (**500, 200, 100, 50, 20, 10, 5**) i en monedes de (**2 i 1**) euros i cèntims de (**50, 20, 10, 5, 2, 1**) de tal manera que tot el canvi sumi el total consignat. Els valors del paper i les monedes més altes són preferibles als valors del paper i la moneda més baixos. (És a dir, **2** euros en lloc de dues monedes d'**1**, i així successivament.

#05 CALCULA EL TEU CANVI

Suma:

Resultat: $500+500+500+500+500+500+200+200+20+5+2+0.5+0.1+0.05+0.02+0.01 = 3427.68€$

6 bitllets de 500€
2 bitllets de 200€
1 bitllet de 20€
1 bitllet de 5€
1 moneda de 2€
1 moneda de 50 cèntims
1 moneda de 10 cèntims
1 moneda de 5 cèntims
1 moneda de 2 cèntims
1 moneda d'1 cèntim

DEPARTAMENT D'INFORMÀTICA INSTITUT LA PINEDA

#05 CALCULA EL TEU CANVI

Suma:

Resultat: $2+1+0.1+0.05+0.02+0.01 = 3.18€$

1 moneda de 2€
1 moneda d'1€
1 moneda de 10 cèntims
1 moneda de 5 cèntims
1 moneda de 2 cèntims
1 moneda d'1 cèntim

El missatge amb el càlcul ha de ser introduït en el paràgraf, en el primer **** i el total en el segon ****.

El missatge mitjançant lletres l'hem d'introduir dins l'etiqueta **<h5>**. Per exemple, si una persona entra la quantitat de **"3.18"** com a suma, el programa ha de mostrar **"1 moneda de 2 euros, 1 moneda de 1 euro, 1 moneda de 10 cèntims, 1 moneda de 5 cèntims, 2 moneda de 2 cèntims i una moneda d'1 cèntim"**.

Hem de controlar, com veieu en la imatge, els singulars i els plurals i posar-los en columna.

IMPORTANT: És molt possible que en el càlcul d'aquesta quantitat, la decimal), tinguem problemes degut al que varem explicar al principi de curs amb les quantitats molt petites. Heu de trobar una solució per resoldre aquest entrebanc i que surti en el missatge, sempre, les quantitats esperades.

06

Feu un nou arxiu i anomeu-lo **codi06.js** i afegiu-lo a la carpeta **\codi**. Aquest **JavaScript** serà utilitzat pel document **HTML** anomenat **DAW0632-EX02_exercici#06.html**.

Es tracta d'un formulari de comanda de pizzas per a calcular el preu final. Una pizza petita val **8,99€**, una pizza mitjana val **10,99€** i una pizza gran val de **12,99€**. Els impostos són un **9%** addicional a la comanda. L'usuari també ha de pagar una propina entre el **10-20%** del cost total de la comanda després dels d'impostos. Hem d'escriure el codi necessari

JavaScript per calcular i mostrar la propina i el cost total de la comanda utilitzem, del segon paràgraf, el primer **** i en el segon **** respectivament. Fixeu-vos com canvien aquestes xifres segons la propina donada. Per cada entrada de nombres de pizzas disposem d'uns elements **** per que indiquem el preu parcial. En el primer paràgraf i en els diferents **** indiquem, per aquest ordre, la suma de les pizzas, els impostos (**9%**), i el total de sumar el primer amb el segon.

#06 FEM UNA COMANDA



07

Feu un nou arxiu i anomeu-lo **codi07.js** i afegiu-lo a la carpeta **\codi**. Aquest **JavaScript** serà utilitzat pel document **HTML** anomenat **DAW0632-EX02_exercici#07.html**.

En un quadrat (**div**) de **50px**, hem d'escriure el codi **JavaScript** de manera que la mida de l'element **div** s'incrementarà cada vegada que l'usuari faci clic sobre ell. Tant l'alçada com l'amplada del **div** s'ha augmentaran un quart de la seva grandària anterior. Per exemple, si el **div** inicialment té una amplada i alçada de **50px**, el seu augment d'amplada i alçada passarà a ser de **62.5px** després del primer clic, **77.5px** després del segon, i així successivament fins que el quadrat superi els **500px**, a partir d'aquest moment anirà minvant, un quart de la seva grandària anterior de la mateixa forma que abans anava augmentant, fins arribar a ser més petit de **50px**, aleshores, s'iniciarà el procés invers.

Dins del quadrat hem inclòs les diferents mides que va obtenint. Poseu-les vosaltres també.

#07 GRAN I PETIT



08

Feu un nou arxiu i anomeu-lo **codi08.js** i afegiu-lo a la carpeta **\codi**. Aquest **JavaScript** serà utilitzat pel document **HTML** anomenat **DAW0632-EX02_exercici#08.html**.

Es tracta d'un exercici força similar a l'anterior però aquesta vegada amb mides de lletra. Mitjançant els diferents **** amb **id="mespetit"** i **id="mesgran"** hem d'aconseguir canviar la mida de la font del text en els dos paràgrafs del document, mitjançant fer clic en la "a" minúscula o en la "A" majúscula.

Per cada pulsació tots dos paràgrafs han de disminuir o augmentar la mida, segons on es polsi, de la font, en **2 pt.** exactament.

#08 FON GRAN I PETITA

a | A

El pop és una criatura marina de 8 potes. Habita en moltes regions de l'oceà. Contràriament a la creença popular, el pop a la gallega no és la millor manera de cuinar-los, potser, alguns argumenten, que els pops a la llauna és on es treu el millor sabor de l'octòpode.

DEPARTAMENT D'INFORMÀTICA INSTITUT LA PINEDA

09

Feu un nou arxiu i anomeu-lo **codi09.js** i afegiu-lo a la carpeta **\codi**. Aquest **JavaScript** serà utilitzat pel document **HTML** anomenat **DAW0632-EX02_exercici#09.html**.

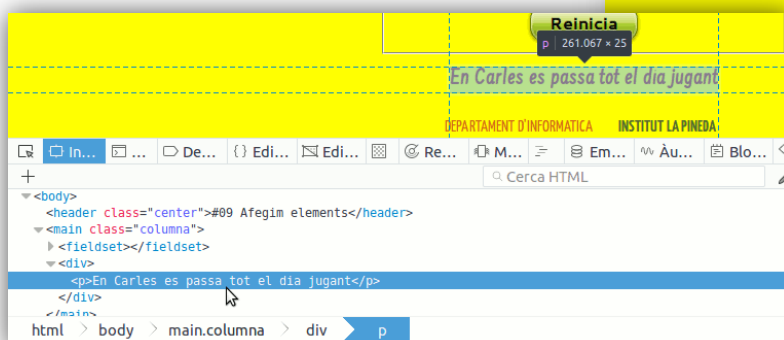
Quan l'usuari faci clic al botó "**Afegim-lo**", hem de crear un nou element de paràgraf dins de l'element **div** amb el text contingut dins de l'àrea de text.

#09 AFEGIM ELEMENTS

Text a inserir:

En Carles es passa tot el dia jugant

DEPARTAMENT D'INFORMÀTICA INSTITUT LA PINEDA



Molt important: El text de l'àrea de text s'ha d'eliminar immediatament després que es polsi per introduir un nou text

10

Feu un nou arxiu i anomenau-lo **codi10.js** i afegiu-lo a la carpeta **\codi**. Aquest **JavaScript** serà utilitzat pel document **HTML** anomenat **DAW0632-EX02_exercici#10.html**.

En aquesta ocasió hem de crear un nombre especificat per l'usuari, mitjançant un **input** del tipus **range**, de **divs** de forma rectangular i colors aleatoris quan l'usuari faci clic al botó amb **"Explotar"**. Els costats dels rectangles han d'estar entre **5px** i **300px** de longitud, aleatòriament, i la localització del rectangle han de ser, també aleatòriament, entre **5px** i **850px** en l'eix **x** i entre **5px** i **500px** en l'eix **y**. Habilitarem un nombre màxim de **50** rectangles i un mínim de **10**.

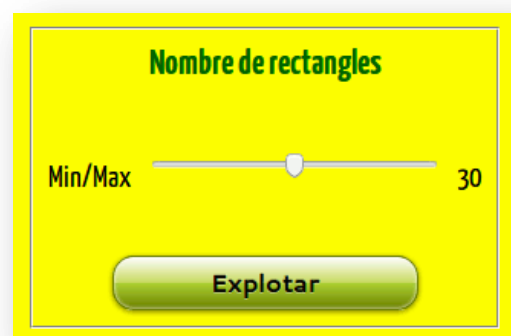
Els colors aleatoris que hem emprat han estat els següents:

red, cyan, gold, magenta, green, blue, orange, coral, darkcyan, greenyellow, lime, olive, peru, salmon, violet i **silver**. A una opacitat del **0,9**.

Per posar el valor al lliscant de l'element **range**, com encara no ho hem estudiat, us el donem fet. Fixeu-vos:

```
objRange.addEventListener("input", function(){
    output.value = this.value;
}, false);
```

Com veieu utilitzem l'esdeveniment **"input"**, de l'element **range**, que haurem d'obtenir (**objRange**), encara no estudiat en aquest curs, per posar el valor dels lliscant en l'element **output**.



11

Feu un nou arxiu i anomeneu-lo **codi11.js** i afegiu-lo a la carpeta **\codi**. Aquest **JavaScript** serà utilitzat pel document **HTML** anomenat **DAW0632-EX02_exercici#11.html**.

Es tracta de mostrar un "fet" d'en **Chuck Norris** aleatòriament quan el ratolí es mou per sobre la imatge de **Chuck Norris** i ocultar-ho quan el ratolí es mogui per fora de la imatge.

Hem de posar el fet dins el paràgraf, substituint el text existent quan tinguem el ratolí dins de la imatge, i retornar el text original quan aquest surti.

Utilitzeu aquesta matriu com a font de dades "**fets**" per mostrar-los:

```
var fets = [
  "Chuck Norris va guanyar una cursa de fórmula 1 empenyent el cotxe.",
  "Chuck Norris no fa servir gasolina, orina al dipòsit.",
  "Chuck Norris agafa les corbes en línia recta.",
  "Chuck Norris va destruir la taula periòdica, perquè només reconeix l'element de sorpresa.",
  "El satèl·lit Meteosat no existeix, els meteoròlegs pregunten a Chuck Norris el temps que vol que faci.",
  "Chuck Norris no només té un xalet a la lluna, també té un al Sol.",
  "No hi ha bojos, sinó gent a la qual Chuck va mirar fixament als ulls.",
  "Chuck Norris no talla la gespa. Simplement li adverteix que si creix, es penedirà.",
  "Chuck Norris pot dibuixar un polígon amb 2 costats.",
  "Chuck Norris escriu codi que s'optimitza sol.",
  "Chuck Norris pot resoldre les Torres de Hanoi en un sol moviment",
  "Qualsevol que toqui la guitarra de Chuck Norris, morirà d'una puntada giratòria abans que pugui escoltar algun so.",
  "Chuck Norris va recórrer un bucle infinit.",
  "El temps en l'univers es regeix pel rellotge de Chuck Norris."
];
```

Teniu la imatge en la carpeta **/img (chuck.png)**

#11 FETS D'EN CHUCK NORRIS



Chuck Norris va destruir la taula periòdica, perquè només reconeix l'element sorpresa.

DEPARTAMENT D'INFORMÀTICA INSTITUT LA PINEDA



12

Feu un nou arxiu i anomenau-lo **codi12.js** i afegiu-lo a la carpeta **\codi**. Aquest **JavaScript** serà utilitzat pel document **HTML** anomenat **DAW0632-EX02_exercici#12.html**.

Es tracta de crear una eina interactiva de revelació d'imatge. Inicialment la pàgina apareix en negre. Quan l'usuari col·loca el ratolí sobre un àrea de color negre de la imatge, la pàgina web revela la imatge de fons.

El **CSS** per al **div** contenidor dona una imatge de fons, que és de **500px x 500px**.

Quan es carrega la pàgina, el **JavaScript** ha de generar una quadrícula de **10 x 10 divs** de **50px x 50px**, de color negre, que són fills del **div** contenidor.

Quan l'usuari desplaça el ratolí sobre els petits **divs** per on passa, aquests s'han de transparentar d'una manera no sobtada, hem posat una petita animació, en el **CSS**, per que ho faci més bonic.

```
@keyframes apareix {from {opacity: 0;} to {opacity: 1;}}
```

Fixeu-vos be en l'efecte de la animació. Evidentment vosaltres podeu fer la vostra, però sempre utilitzant **JavaScript**. Procureu que sigui prou efectiva.

#12 REVELEM UNA IMATGE



DEPARTAMENT D'INFORMÀTICA INSTITUT LA PINEDA



#12 REVELEM UNA IMATGE



La nit estelada de Vincent van Gogh

Finalment, quan la imatge estigui revelada del tot, sortirà un missatge dins el paràgraf amb el nom del quadre i el seu autor.

Fixeu-vos en la imatge de l'esquerre i el missatge:

"La nit estelada d'en Vincent van Gogh"

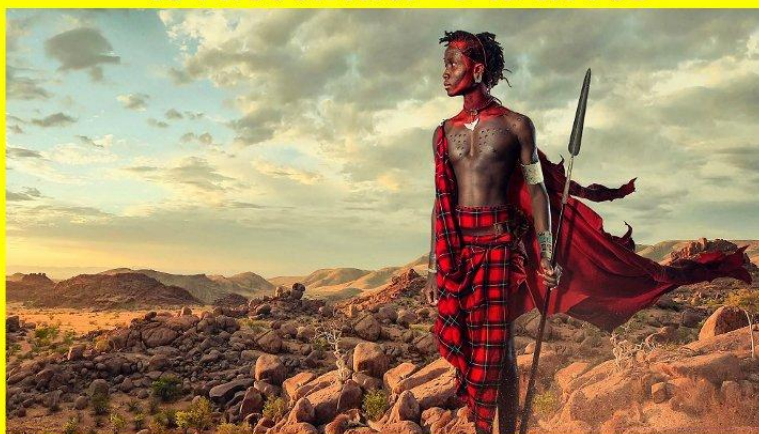
13

Feu un nou arxiu i anomenau-lo **codi13.js** i afegiu-lo a la carpeta **\codi**. Aquest **JavaScript** serà utilitzat pel document **HTML** anomenat **DAW0632-EX02_exercici#13.html**

Es tracta de fer una presentació de diapositives mitjançant la rotació a través de diferents imatges cada **3** segons. Assumiu que te sis imatges, les trobareu en la carpeta **/img**, cadascuna nomenada "**imatge01.jpg**" fins a "**imatge06.jpg**".

Per iniciar la rotació només caldrà passar el ratolí per sobre la imatge. Per aturar-la tornem a passar-lo, una altre vegada, per la imatge.

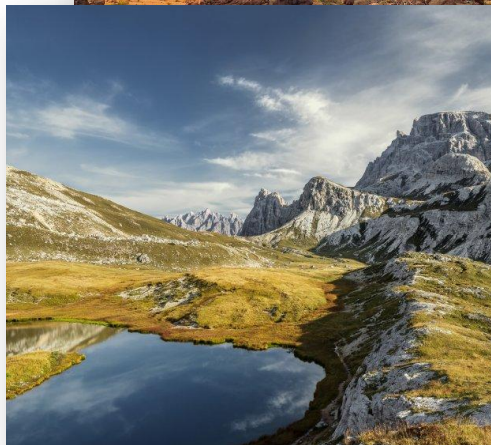
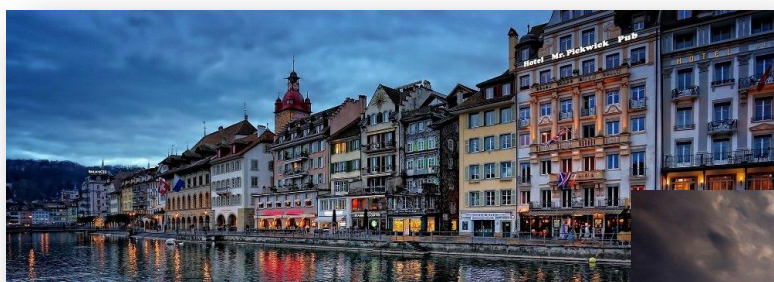
#13 PRESENTACIÓ D'IMATGES



imatge03.jpg
FUNCIONANT

DEPARTAMENT D'INFORMÀTICA INSTITUT LA PINEDA

És important, tant el missatge de "**FUNCIONANT**" i "**ATURAT**" en l'**span**, com el nom de la imatge que estem visualitzant en el paràgraf.



EXERCICIS:

Heu d'intentar resoldre tots els exercicis, fixeu-vos be en l'explicació i en els vídeos d'ajuda.

PREMISSES:

1. Heu de procurar fer el codi JavaScript el més optimitzat possible, funcional i sobre tot efectiu.
2. Llegiu-vos, mol atentament, les consideracions bàsiques que hem introduït al principi de la pràctica.
3. És molt important que comenteu el codi, línia a línia.
4. Col·loqueu cada arxiu de codi al lloc que li pertoca, en la carpeta **/codi**.
5. Al final compriu, només, la carpeta **/codi**, en un sol arxiu de format **7z**.

LLIURAMENT DE LA PRÀCTICA:

Comprimiu la carpeta **/codi**, amb tots els exercicis fets, en format **7z**, anomeneu-lo:

"DAW0632-EX02-vostrenom.7z" i lliureu-lo a l'aula virtual.

A on "vostrenom" correspon al vostre primer cognom. Exemple: Si el vostre nom fos Pere Beltran, l'arxiu el podríeu anomenar: **"DAW0632-EX02-beltran.7z"** o també **"DAW0632-EX02-pbeltran.7z"**.

