



CONSTRUÏNT HISTÒRIA AMB MEMÒRIA

Taller

GEOLocalITZACIÓ DE LES
STOLPERSTEINE EN UNA APLICACIÓ WEB

Aquest taller forma part del projecte Construïnt història amb memòria del Museu d'història de Sant Feliu de Guíxols inscrit en el programa Àngel Jiménez de Memòria democràtica.

En aquest taller convertirem unes dades sobre les llambordes Stolpersteine a Sant Feliu, que ens han fet arribar en un arxiu word, en una aplicació web, amb un mapa on hi surten geolocalitzades.

Això ens donarà una idea general sobre el funcionament d'una aplicació web i les tecnologies que hi estan involucrades.



En aquest cas construirem una pàgina web, que utilitza el codi HTML

Una fulla d'estils en codi CSS on definirem l'estil dels elements de la pàgina (el color, el tamany, el tipus de lletra ...)

Una part de programació escrita en JS per a mostrar el mapa

I una petita base de dades escrita en format JSON.

Com veieu són molts codis diferents involucrats en la creació d'aquesta aplicació web i no tenim la intenció de explicar en detall com funcionen cadascun d'ells sinó fer un tastet de com es programa una aplicació web.

Si voleu aprendre en detall aquests llenguatges us recomano la pagina [w3schools](https://www.w3schools.com/) on trobareu informació detallada i molt ben explicada amb exemples i tutorials d'aquests llenguatges i molts més.

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

Per a fer aquesta aplicació farem servir una pàgina web que és Codesandbox.io que ens permetrà anar veient com canvia la pàgina a mesura que l'anem construint.

La pàgina inicial te dues opcions gratuïtes:

Podeu fer click en el botó "TRY FROR FREE" per començar a provar-la directament o registrar-vos amb el vostre correu.

Registrar-vos amb el vostre correu us permetrà guardar la vostra pàgina per continuar-hi treballant un altre dia.

Per a la pràctica que farem qualsevol de les dues opcions és vàlida. El vostre professor us dirà quina opció és la que més us convé.

Si trieu registrar-vos haureu de donar el vostre correu i us portarà a la secció Dashboard, que és el taulell on hi haurà tots els projectes que comenceu.

Si heu escollit la opció de provar gratuïtament us demanarà directament quin tipus de projecte voleu crear , però quan acabeu la pràctica recordeu guardar-vos els documents al vostre ordinador, ja que si no, quan tanqueu la pàgina, desapareixeran per sempre més.

Trieu crear un nou sandbox; En el nostre cas farem servir la plantilla HTML + CSS i ens demanarà que li donem un nom : STOLPERSTEINE

Un cop creat el sandbox veureu la pàgina d'edició de codesandbox.

A l'esquerra veureu que la plantilla triada en ha creat varis arxius, un html, una fulla d'estils css i un json amb informació de l'aplicació que en aquest cas no utilitzarem.

Si feu click en qualsevol arxiu a la part central en podreu editar el contingut i a la part dreta tenim una vista prèvia de la nostra pàgina.

PRIMERA PART

Html

Feu click a l'arxiu [index.html](#)

Com veieu en el codi que ens ha generat la plantilla, el llenguatge html està basat en etiquetes, que l'editor ens mostra de color morat.

Html

<nom> bla, bla, bla **</nom>**
PRINCIPI DE LA ETIQUETA FINAL DE LA ETIQUETA

Cada etiqueta comença amb aquest símbols i el nom de l'etiqueta i acaba igual però amb una barra al principi del nom

Així si volem marcar un text amb l'etiqueta p posarem l'etiqueta p al principi i barra p al final.

Tot el que hi hagi entre aquest marcadors es considera part de l'etiqueta, encara que hi hagi salts de línia enmig.

Html

<p>

PRINCIPI DE LA ETIQUETA

<i>

PRINCIPI DE LA ETIQUETA

bla, bla, bla

</i>

FINAL DE LA ETIQUETA

</p>

FINAL DE LA ETIQUETA

De fet una etiqueta pot contenir altres etiquetes com per exemple aquesta etiqueta p conté una etiqueta i.

Per el nom de les etiquetes es pot usar el que vulgueu, però n'hi ha algunes d'estàndard, i és bo usar-les per a facilitar-vos la feina.

Les etiquetes estàndard estan basades en l'estructura d'un diari:

Html

<html>



<head>

document

<body>

L'etiqueta HTML representa el document i té dues parts

El HEAD que és invisible i conté les metadades com per exemple l'autor i l'idioma

i el BODY que és el cos visible de la nostra pàgina.



Dins el BODY hi sol haver-hi un títol <H1> subtítols <H2> i paràgrafs <p>.

També pot haver-hi diverses seccions <DIV>.

Dins el body també hi podem trobar imatges links <a> i altres que avui no usarem.

Anem a veure el document index.html de la plantilla.

Com veieu ens ha creat una etiqueta html amb les dues parts, un head amb unes metadades i un body que conté un sol títol h1 amb el text: this is a simple HTML and CSS template.

Editarem primer el head.

Canviarem la llengua de la pàgina, en el nostre cas ca de català

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

Canviarem el títol a les metadades. recordeu que el head no és visible però el títol apareixerà a la pestanya del vostre navegador.

Veiem que la plantilla ens ha inclòs el link a la fulla d'estils, perfecte, l'editarem després.

```
<link rel="stylesheet" href="style.css" />
```

Ara anem per el body.

Primer ens desfarem del títol que ens dona la plantilla.

Dividirem la nostra pàgina en tres seccions div.

Per tal de diferenciar-les els donarem un identificador.

```
<body>
  <div id="intro">

  </div>

  <div id="map"></div>

  <div id="fitxa">

  </div>
</body>
```

A la primera li direm INTRO a la segona MAP i a la tercera FITXA

La INTRO tindrà un títol Stolpersteine a Sant Feliu de Guíxols i un paràgraf

```
<div id="intro">
  <h1>Stolpersteine a Sant Feliu de Guíxols</h1>
  <p>Un mapa creat per Dani Serrat per al projecte
    Memòria i Història 2024
    del Museu d'Història de Sant Feliu de Guíxols
  </p>
</div>
```

en el mapa de moment res.

La secció fixa contindrà un botó que ens servirà per amagar la fitxa.

Li donarem l'identificador tancar i de text simplement una x

```
<button id="tancar">x</button>
```

també hi posarem un títol amb l'identificador nom però hi deixarem un espai en blanc.

```
<h1 id="nom"></h1>
```

llavors necessitarem quatre etiquetes de paràgraf p amb el text Naixement, Deportat, Camp i Destí.

```
<p>Naixement:</p>
```

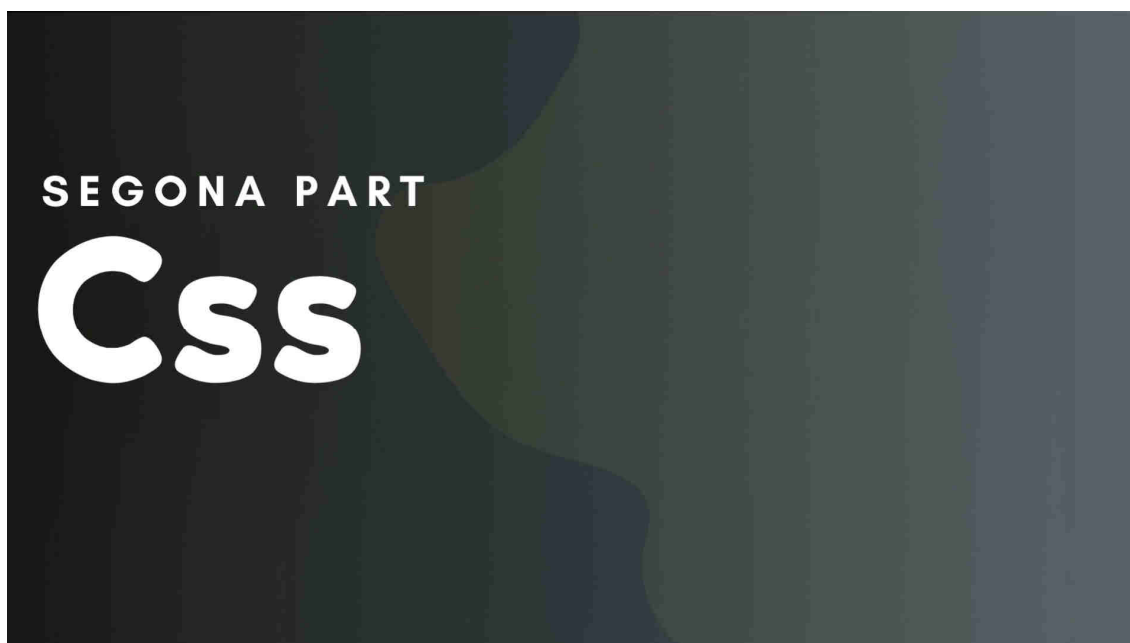
```
<p>Deportat:</p>
<p>Camp:</p>
<p>Destí:</p>
```

Dins de cadascuna d'aquestes etiquetes hi afegirem una etiqueta especial span per a poder injectar-hi el text dinàmicament. Les diferenciarem amb un identificador: la primera naixement, la segona deportat, camp i destí.

```
<p>Naixement: <span id="naixement"> </span></p>
<p>Deportat: <span id="deportat"> </span></p>
<p>Camp: <span id="camp"> </span></p>
<p>Destí: <span id="desti"> </span></p>
```

Fixeu-vos que als identificadors no faig servir majúscules ni accents ni espais ni caràcters estranys com la ç o la ñ. Això ens facilitarà el el css i el js després.

I amb això haurem acabat la part de HTML. Farem control + s per guardar i el preview s'actualitzarà.



La nostra pàgina es veu una mica trista així que donem-hi una mica de color.

Si heu guardat la pagina index amb les tecles control s fem click a l'arxiu styles css

com veieu el codi css és molt senzill, escrivim el nom de l'etiqueta i entre claus els estils que hi volem aplicar.

Aquests estils els defineix el navegador. N'hi ha un munt i us els haureu d'aprendre.

Però amb una mica d'anglès i l'autocompletador de codi veureu que no costen gens d'entendre.

Per exemple veiem que la plantilla ens ha definit que el títol h1 tinguin l'estil de decoració de text subratllat.

```
h1 {  
  text-decoration: underline;  
}
```

Prou fàcil d'entendre, no?

Esborrarem aquesta definició i començarem de cap i de nou.

Primer els estils generals de la pàgina html i body

```
html,  
body {
```

els donarem una alçada del 100% i amplada també del 100%
i volem que la nostra aplicació ocupi tota la finestra del navegador.
donarem un marge de 0 i un padding de 0

```
html,  
body {  
  height: 100%;  
  width: 100%;  
  margin: 0;  
  padding: 0;  
}
```

el margin en css es considera el marge exterior dels objectes és a dir des de la vora fins al següent objectes; el padding és el marge interior, des de la vora fins al contingut.

Definirem també la tipografia, en aquest cas l'escollirem de l'autocompletador

```
font-family: "Courier New", Courier, monospace;
```

per els títols h1 simplement farem la tipografia més gruixuda, font weight 600

```
h1 {  
  font-weight: 600;  
}
```


als paràgraf els donarem una tipografia més primeta, font weight 200;

```
p {  
  font-weight: 200;  
}
```

Ara les seccions; com que els hem donat un identificador i podem accedir amb el caràcter coixinet i el nom

Per a la intró

```
#intro {  
}
```

Una altura d'un 15% de la pàgina serà suficient; No hi volem marge així que margin: 0;

l'hi donarem una mica de marge interior, 10px; El text el volem centrat. text-align: center;

Donarem un fons a tota la secció; El color de la lletra el voldrem blanc, que destaquí: color :

```
#head {  
  height: 15%;  
  margin: 0;  
  padding: 10px;  
  text-align: center;  
  background: rgb(58, 58, 56);  
  color: white;  
}
```

Per al mapa:

li hem dit map

```
#map {  
}
```

li donem un color qualsevol: si la pàgina funciona, aquest color quedarà ocult per el mapa però mentrestant ens donarà una idea del que tenim, per exemple cadetblue.

Com que la intro ocupa un 15% de l'altura volem que el mapa ocupi la resta de la pantalla, és a dir : height: 85% i amplada 100%

I voldrem que la fitxa aparegui sobre el mapa aixó es fa amb z-index: 1

```
#map {
  background-color: cadetblue;
  height: 85%;
  width: 100%;
  z-index: 1;
}
```

el z-index és com la capa on es dibuixen les coses. Si a qualsevol cosa li diem z-index 4 es dibuixarà sobre el que tingui z-index 3 però a sota del que tingui z-index de 5. Quan més gran sigui el número del z-index més a dalt de la pila quedarà.

Només queda la fitxa

```
#fitxa {
}
```

De moment, li donarem un z-index de 10 per estar segurs que queda sobre el mapa.

Li donarem un fons de un color de la gama mitja, així hi podrem escriure tant en color blanc com en negre i tots dos es veuran.

li donarem un marge interior generós, padding: 60px.

Hi voldré una mica de vora, border: 1 pixel solid i potser el mateix color que la intro.

Potser hi quedaria bé que arrodoníssim les cantonades, així que border radius 10px;

De moment ho deixarem així:

```
#fitxa {
  z-index: 10;
  padding: 60px;
  background: rgb(139, 139, 120);
  color: white;
  border: 1px solid #34352d;
  border-radius: 10px;
  box-shadow: 4px 4px 4px 4px rgba(0, 0, 0, 0.2);
}
```

Un truc:

El css té una manera estranya de medir els objectes, el marge interior es compta dins la mida però el marge exterior no. Això complica una mica les mesures però podem escriure el següent a dalt de tot del nostre css:

Per tant a estrella (és a dir qualsevol objecte) li apliquem l'estil box-sizing: border-box;

```
* {  
  box-sizing: border-box;  
}
```

Així la mida total de qualsevol objecte s'agafa sempre des del marge i això ho fa tot més senzill.

Guardem l'arxiu amb control s i veiem el resultat; Anem millorant però la pàgina encara no funciona, falta la programació.



Abans de començar a programar hem de preparar les dades.

En el full que us han donat hi ha l'adreça física de la porta on va instal·lada la llamborda stolpersteine. Per tal de geolocalitzar-la podem anar-hi físicament amb un gps i anotar les coordenades del punt, o bé, podem usar una pàgina com google maps per extreure la informació:

Obrim l'adreça [google.com/maps](https://www.google.com/maps) i cerquem l'adreça; En el primer cas, Carretera de Girona, 124.

Això en posarà un marcador vermell al mapa, feu click amb el botó dret i us sortirà un desplegable amb les coordenades, copieu-les al document o en un full apart.

Feu una pausa al vídeo, quan tingueu totes les coordenades continuem.

Per a l'aplicació farem servir una base de dades en format JSON.

El format JSON (javascript object notation) és una base de dades de text especialment pensada per que la llegeixi el javascript de la web.

És un format molt còmode perquè és senzill i no necessita cap programari especial.

Per crear-lo farem click en el botó `afegir document` per afegir un arxiu i li donarem per nom `bd.json`



En aquestes bases de dades les col·leccions o taules van entre claudàtors i els objectes individuals o fitxes entre claus.

Els noms dels camps van sempre entre cometes i els camps també si és text o directament si és un valor.

Camps i fitxes es separen amb una coma.

Obrim una taula amb uns claudàtors. A dins hi posarem una fitxa per tant unes claus.

```
[
  {
  }
]
```

aquesta fitxa tindrà varis camps,

Per tal de geolocalitzar els stolpersteine necessitarem un camp latitud, així que li donem el nom lat entre cometes, dos punts i el primer valor que us ha donat el google maps, al final una coma per separar els camps.

```
[
  {
    "Lat": 41.78612
  }
]
```

Fixeu-vos que el separador de decimals sigui un punt i no una coma, ja que si tinguéssiu dues comes en el mateix camp us donaria error.

També necessitarem un camp longitud, li donarem per nom Lon també entre cometes,

dos punts i el segon valor que us ha donat el google maps, al final sempre una coma per separar els camps.

```
[
  {
    "Lat": 41.78612,
    "Lon": 3.02493,
  }
]
```

Ara les dades de la persona,

Primer el nom dos punts i com que és un text ho posarem entre cometes.
Josep Coll Sarbosa

Després la data de naixement, li direm naixement i encara que sigui una data també ho posarem entre cometes, el valor és 1908

Tot seguit l'any que va ser deportat en aquest cas 1940

Sabem també el cap on el van portar, Mauthausen.

Finalment sabem el destí que va patir aquesta persona, l'hi inclourem
"Assassinat el 2/11/1941 a Gusen"

I amb aquesta funesta dada acabem la fitxa posem una coma i copiem l'estructura per la següent.

```
[
  {
    "Lat": 41.78612,
    "Lon": 3.02493,
    "Nom": "Josep Coll Sarbosa",
    "Naixement": "1908",
    "Deportat": "1940",
    "Camp": "Mauthausen",
    "Desti": "Assassinat el 2/11/1941 a Gusen"
  },
]
```

Fixeu-vos que després del destí no hi ha coma ja que és l'últim camp de la fitxa

En aquest cas la "Latitud" es 41.78195, la Longitud, 3.03015, el "Nom": "Lluís Murgat Clos",

"Naixement": el "1906", "Deportat": el "1941", també va ser a Mauthausen i morí Assassinat el 28/11/1941 a Gusen"

```
{
  "Lat": 41.78195,
  "Lon": 3.03015,
  "Nom": "Luís Murgat Clos",
  "Naixement": "1906",
  "Deportat": "1941",
  "Camp": "Mauthausen",
  "Desti": "Assassinat el 28/11/1941 a Gusen"
},
```

Feu una pausa al vídeo en aquest punt i completeu la base de dades. Un cop acabada la base de dades, fixeu-vos també que després de l'ultima fitxa no hi va coma.

Ara que tenim la base de dades només queda la última part.

QUARTA PART

Javascript

El javascript és un llenguatge de programació que fem servir per donar contingut dinàmic a les pàgines.

És un llenguatge bastant senzill i es pot entendre amb una mica de paciència ja que també té moltes paraules en anglès.

No m'entretindré massa a explicar-lo, ens limitarem a copiar text de les pàgines d'exemple i adaptar-lo a la nostra aplicació.

Farem servir l'script de leaflet; Leaflet és un codi javascript per a pàgines web, semblant a google maps, però és gratuït i codi obert.

Obrirem la seva pàgina: leafletjs.com

Leaflet està creat a Ucraïna i el creador ens ha deixat un text amb links a diverses organitzacions

de suport als ciutadans. Deixo a la vostra discreció de consultar aquests links.

Un cop entrem a la pàgina anirem a la secció tutorials; Farem click al primer tutorial QUICK START GUIDE

Ens demana que incloem un link a la seva fulla d'estils a la nostra pàgina, seleccionarem el codi i el copiarem al final del head de la nostra pàgina

```
<link
  rel="stylesheet"
  href="https://unpkg.com/leaflet@1.9.4/dist/leaflet.css"
  integrity="sha256-p4NxAoJBhIIN+hmNHzRCf9tD/miZyoHS5obTRR9BMY="
  crossorigin=""
/>
```

De la mateixa manera ens demana que incloem el javascript

Farem el mateix, copiarem el text i el posarem després de la seva fulla d'estils

```
<script
  src="https://unpkg.com/leaflet@1.9.4/dist/leaflet.js"
  integrity="sha256-20nQCchB9co0qIjJZRGuk2/Z9VM+kNiyxNV1vTlZBo="
  crossorigin=""
></script>
```

Ens demana que incloem un div amb l'identificador map, Aixó ja ho tenim fet.

També ens demana que li donem una altura dins l'arxiu css. També ho hem fet abans.

El codi el podríem posar a la nostra pàgina dins una etiqueta script, el copiarem de la pàgina i l'enganxarem així: creem l'etiqueta script i hi posem el codi copiat.

```
<script>
  var map = L.map('map').setView([51.505, -0.09], 13);
</script>
```

Però per ser més endreçats posarem tot el codi en un arxiu extern

A la nostra pàgina deixarem la etiqueta script buida però li donarem l'adreça de un arxiu extern.

Escriurem src , que vol dir source mapa.js que serà el nostre arxiu

```
<script src="mapa.js"></script>
```

al nostre editor crearem un arxiu nou amb el botó i li donarem per nom mapa.js

en aquest arxiu escriurem el nostre codi, enganxarem el codi que hem copiat abans.

Fixeu-vos que les funcions de javascript acaben amb punt i coma.

Les coordenades que ens dona la pàgina d'exemple són les de Londres així que les canviarem per les del centre de Sant Feliu Longitud 41.781518 i latitud 3.031515.

La darrera xifra és el zoom inicial, el canviarem a 15.

```
const map = L.map("map").setView([41.781518, 3.031515], 15);
```


Ens demana també que hi posem un tileLayer. Una capa Gràfica basada en tiles.

Un mapa del món amb aquest detall seria un arxiu enorme, per això el servidor els guarda en arxius més petits anomenats TILES i els carrega quan l'usuari s'apropa a un sector.

En aquest cas els tiles de openmap són molt acolorits, per això farem servir els arxius de CartoDB. Canviarem l'adreça del servidor per {s}.basemaps.cartocdn.com/light_all/

També canviarem el text del copyright per incloure els de Carto DB

Per al símbol del copyright mantindrem la tecla alt i amb el teclat numèric escriurem 0169 en deixar la tecla alt el símbol s'escriurà.

```
L.tileLayer("https://{s}.basemaps.cartocdn.com/light_all/{z}/{x}/{y}.png", {
  attribution: "@OpenStreetMap, @CartoDB",
}).addTo(map);
```

Tornarem a la pàgina de leaflet i seguirem el tutorial.

Tot seguit ens ensenya com posar una marca, copiarem aquest text i l'enganxarem a la pàgina. més endavant l'utilitzarem.

També ens ensenya com fer un quadre de diàleg o popup. Copiarem aquest codi i l'enganxarem a la pàgina. Les coordenades no són les de Sant Feliu, les canviarem.

```
// Popup amb les instruccions bàsiques
const popup = L.popup()
  .setLatLng([41.781518, 3.031515])
  .setContent(
    "Feu Click a qualsevol localització per a obtenir més informació de cada Stoppelsteine"
  )
  .openOn(map);
```

Ara anem per les dades:

Necessitem una variable on guardar els punts o sigui que la inclourem a l'script al principi de tot.

```
var punts;
```

Ara escriurem una funció nova amb el text `async function` i el nom `logBD` per carregar l'arxiu.

El codi d'aquesta funció va entre claus i acaba amb punt i coma.

```
async function logBD() {  
};
```

primer llegim l'arxiu. la variable `resposta` serà igual a `await fetch` i entre parèntesi i cometes el nom de l'arxiu, en el nostre cas `./bd.json`

```
var resposta = await fetch("./bd.json");
```

els nostres punts seran iguals a `await resposta.json()`

```
punts = await resposta.json();
```

Tot aixó vol dir: llegeix l'arxiu i espera, quan el tinguis guarda'l a la variable `punts` com un JSON

Ara que tenim el punts els posarem al mapa, amb el codi

```
punts.forEach( );
```

Aquesta funció repetirà el codi que hi hagi entre parèntesi un cop per cada element del JSON punts.

Dins del parèntesi li direm quin nom donarem a cada element del JSON punts ho escriurem també entre parèntesi i per no complicar-nos li direm `punt`.

Després farem una fletxeta amb el símbol igual i el símbol "més gran que" i entre claus escriurem la funció que s'ha de repetir en aquest cas el que repetirem és el codi que hem copiat del tutorial per a posar un marcador al mapa.

```
punts.forEach((punt) => {  
  const marker = L.marker([punt.Lat, punt.Lon])  
    .addTo(map);  
});
```

Les coordenades les traurem del punt que acaben de llegir per això dins el `marker` canviarem les coordenades per `punt.Lat` i `punt.Lon` fixeu-vos que la `L` és majúscula ja que així ho hem definit a l'arxiu JSON

deixarem igual la part o diu `add to map` però abans del punt i coma i afegirem una altra funció

onMarkerClick La Ema i la C en Majúscules és a dir quan es faci click al marcador executa la funció onMarkerClick.

```
punts.forEach((punt) => {  
  const marker = L.marker([punt.Lat, punt.Lon])  
    .addTo(map)  
    .on("click", onMarkerClick);  
});
```

Abans de acabar la funció passarem les altres dades del JSON al marcador és a dir

marker.dades = punt (punt i coma al final)

Quin embolic de parèntesi, claus i punts i coma! per sort l'editor ens ajuda si fem click abans d'una marca

ens indica l'origen. Repassem que no ens n'haguem deixat cap.

```
punts.forEach((punt) => {  
  const marker = L.marker([punt.Lat, punt.Lon])  
    .addTo(map)  
    .on("click", onMarkerClick);  
  //afegim dades al punt per a llegir-les al fer click  
  marker.dades = punt;  
  
  // afegim una classe marcador per a poder canviar-ne l'estil  
  marker._icon.classList.add("marcador");  
});
```

Ara hem creat aquesta funció, però la pàgina no sap en quin moment l'ha de utilitzar.

Li direm que ho faci en aquest moment del codi escrivint el nom logBD i els parèntesi. punt i coma al final de línia.

```
logBD();
```

Però quan l'usuari faci click a un Punt es cridarà la funció OnMarkerClick i no l'hem definida.

Fem-ho ara:

```
function onMarkerClick(e) {  
}
```

Quan l'usuari faci click seleccionarem del document un element pel seu identificador

document.getElementById("nom") i canviarem el contingut amb text Content amb el que hem passat a l'element.

```
function onMarkerClick(e) {  
    document.getElementById("nom").textContent = e.target.dades.Nom;  
}
```

aixó ho farem cinc vegades

```
function onMarkerClick(e) {  
    document.getElementById("nom").textContent = e.target.dades.Nom;  
    document.getElementById("naixement").textContent = e.target.dades.Naixement;  
    document.getElementById("deportat").textContent = e.target.dades.Deportat;  
    document.getElementById("camp").textContent = e.target.dades.Camp;  
    document.getElementById("desti").textContent = e.target.dades.Desti;  
}
```

Fem control i S per guardar el document i comprovem que funciona: Quant fem click a una marca, la fitxa s'actualitza? Bé.

Només ens queda un últim detall.

A l'arxiu CSS no hem donat estil al botó tancar

té l'etiqueta tancar.

```
#tancar {  
}
```

li donarem una posició fixa, a dalt a la dreta. (Position: absolute, top: 5px i right 5px)

un fons gris (background color gray) sense vora (border: none) color de la lletra blanc (color: white) marge interior (padding: 10px 20px) font size de 22 pixels i les cantonades arrodonides (border-radius: 10 px)

```
#tancar {  
    position: absolute;  
    top: 5px;  
    right: 5px;  
    background-color: #505050; /* Green */  
    border: none;
```

```
color: white;
padding: 10px 20px;
font-size: 22px;
border-radius: 10px;
}
```

també modificarem la fitxa:

li donarem una posició fixa: position : absolute; top 50px i left 50px

i l'ocultarem (opacity: 0)

i pointer-events: none per tal que no es pugui clicar quan estigui amagada.

el farem apareixer poc a poc, en mig segon (transition opacity 0.5s linear)

```
#fitxa {
    position: absolute;
    top: 50px;
    left: 50px;
    z-index: 10;
    padding: 60px;
    background: rgb(139, 139, 120);
    color: white;
    border: 1px solid #34352d;
    border-radius: 10px;
    box-shadow: 4px 4px 4px 4px rgba(0, 0, 0, 0.2);
    opacity: 0;
    transition: opacity 0.5s linear;
    pointer-events: none;
}
```

farem una nova regla per la fitxa amb coixinet fitxa però només quan sigui visible per tant .visible

```
#fitxa.visible {
}
```

Quan la fitxa sigui visible tindrà opacity 1 i pointer events initial per poder-hi fer click.

```
#fitxa.visible {  
  opacity: 1;  
  pointer-events: initial;  
}
```

Només queda tornar al javascript i donar aquesta classe que hem anomenat visible a la fitxa

Per tant aquí quan l'usuari hagi fet click a la marca

```
function onMarkerClick(e) {  
  ...  
  document.getElementById("fitxa").classList.toggle("visible");  
}
```

I per part del botó tancar escriurem una funció

```
document.getElementById("tancar").addEventListener("click", function () {  
  document.getElementById("fitxa").classList.toggle("visible");  
});
```

I amb aixó ja tenim la pàgina funcionant.

Sóc conscient que si no haviu fet mai aquestes coses, he anat molt depressa i potser us heu perdut una mica.

Però espero que us quedeu amb l'experiència de com es fa una aplicació web i els codis que les fan funcionar.

Si voleu guardar-vos aquests arxius al vostre ordinador hi ha un botó a la part superior per a descarregar-vos tot el SandBox, però al moment de fer aquest video no funcionava i descarrega un arxiu buit.

També podeu fer click dret a cada document i seleccioneu download. Recordeu baixar-los tots.

Els codis finals de cada arxiu són :

index.html

```
<!DOCTYPE html>
<html lang="ca">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <meta http-equiv="X-UA-Compatible" content="ie=edge" />
    <title>Stolpersteine a Sant Feliu de Guíxols</title>
    <link rel="stylesheet" href="style.css" />

    <link
      rel="stylesheet"
      href="https://unpkg.com/leaflet@1.9.4/dist/leaflet.css"
      integrity="sha256-p4NxAoJBhIIN+hmNHzRCf9tD/miZyoHS5obTRR9BMY="
      crossorigin=""
    />
    <script
      src="https://unpkg.com/leaflet@1.9.4/dist/leaflet.js"
      integrity="sha256-20nQCchB9co0qIjJZRGuk2/Z9VM+kNiyxNV11vTlZBo="
      crossorigin=""
    ></script>
    <style></style>
  </head>
  <body>
    <div id="head">
      <h1>Stolpersteine a Sant Feliu de Guíxols</h1>
      <p>
        Un mapa creat per Dani Serrat per al projecte Memòria i Història 2024
        del Museu d'Història de Sant Feliu de Guíxols
      </p>
    </div>

    <div id="map"></div>

    <div id="fitxa">
      <button id="tancar">x</button>
      <h1 id="nom"></h1>
      <p>Naixement: <span id="naixement"> </span></p>
      <p>Deportat: <span id="deportat"> </span></p>
      <p>Camp: <span id="camp"> </span></p>
    </div>
  </body>
</html>
```

```
<p>Destí: <span id="desti"> </span></p>
</div>
<script src="mapa.js"></script>
</body>
</html>
```


mapa.js

```
var punts;

const map = L.map("map").setView([41.781518, 3.031515], 15);

L.tileLayer("https://{s}.basemaps.cartocdn.com/light_all/{z}/{x}/{y}.png", {
  attribution: "@OpenStreetMap, @CartoDB",
}).addTo(map);

// Popup amb les instruccions bàsiques
const popup = L.popup()
  .setLatLng([41.781518, 3.031515])
  .setContent(
    "Feu Click a qualsevol localització per a obtenir més informació de cada Stopelsteine"
  )
  .openOn(map);

// aquesta funció llegeix la BD
async function logBD() {
  // llegir la BD i esperar resposta
  var resposta = await fetch("./bd.json");
  // Quant tenim resposta ho assignem a la variable punts
  punts = await resposta.json();

  // Posem cada punt al mapa
  punts.forEach((punt) => {
    const marker = L.marker([punt.Lat, punt.Lon])
      .addTo(map)
      .on("click", onMarkerClick);
    //afegim dades al punt per a llegir-les al fer click
    marker.dades = punt;

    // afegim una classe marcador per a poder canviar-ne l'estil
    marker._icon.classList.add("marcador");
  });
}

logBD();

// Aquesta Funció s'activa quan es clic en un marcador
function onMarkerClick(e) {
```

```
document.getElementById("nom").textContent = e.target.dades.Nom;
document.getElementById("naixement").textContent = e.target.dades.Naixement;
document.getElementById("deportat").textContent = e.target.dades.Deportat;
document.getElementById("camp").textContent = e.target.dades.Camp;
document.getElementById("desti").textContent = e.target.dades.Desti;
document.getElementById("fitxa").classList.toggle("visible");
}
// Aquesta funció s'activa quan es prem el boto de tancar de la fitxa
document.getElementById("tancar").addEventListener("click", function () {
    document.getElementById("fitxa").classList.toggle("visible");
});
```

style.css

```
* { box-sizing: border-box; }

html,
body {
  height: 100%;
  width: 100%;
  margin: 0;
  padding: 0;
  font-family: "Courier New", Courier, monospace;
}

h1 {
  font-weight: 600;
}

p {
  font-weight: 200;
}

#map {
  background-color: cadetblue;
  height: 85%;
  width: 100%;
  z-index: 1;
}

#head {
  height: 15%;
  margin: 0;
  padding: 10px;
  text-align: center;
  background: rgb(58, 58, 56);
  color: white;
}

#fitxa {
  position: absolute;
  top: 50px;
  left: 50px;
  z-index: 10;
}
```

```
padding: 60px;
background: rgb(139, 139, 120);
color: white;
border: 1px solid #34352d;
border-radius: 10px;
box-shadow: 4px 4px 4px 4px rgba(0, 0, 0, 0.2);
opacity: 0;
transition: opacity 0.5s linear;
pointer-events: none;
}

#fitxa.visible {
  opacity: 1;
  pointer-events: initial;
}

#fitxa h1 {
  margin-bottom: 30px;
}

#fitxa span {
  color: #232415;
  margin-left: 40px;
  border-bottom: 1px dashed white;
}

#tancar {
  position: absolute;
  top: 5px;
  right: 5px;
  background-color: #505050; /* Green */
  border: none;
  color: white;
  padding: 10px 20px;
  font-size: 22px;
  border-radius: 10px;
}

img.marcador {
  filter: grayscale(100%);
}
```

i BD.json

```
[
  {
    "Lat": 41.78612,
    "Lon": 3.02493,
    "Nom": "Josep Coll Sarbosa",
    "Naixement": "1908",
    "Deportat": "1940",
    "Camp": "Mauthausen",
    "Desti": "Assassinat el 2/11/1941 a Gusen"
  },
  {
    "Lat": 41.78195,
    "Lon": 3.03015,
    "Nom": "Luís Murgat Clos",
    "Naixement": "1906",
    "Deportat": "1941",
    "Camp": "Mauthausen",
    "Desti": "Assassinat el 28/11/1941 a Gusen"
  },
  {
    "Lat": 41.78406,
    "Lon": 3.02708,
    "Nom": "Francesc Parramon Ponjuan",
    "Naixement": "1912",
    "Deportat": "1941",
    "Camp": "Mauthausen",
    "Desti": "Assassinat el 30/11/1941 a Gusen"
  },
  {
    "Lat": 41.78861691171794,
    "Lon": 3.028505372688729,
    "Nom": "Emili Puigmolé Magrià",
    "Naixement": "1909",
    "Deportat": "1941",
    "Camp": "Mauthausen",
    "Desti": "Assassinat el 10/03/1941"
  },
]
```

```
{
  "Lat": 41.78851769457604,
  "Lon": 3.028172889239772,
  "Nom": "Josep Ribas Juncà",
  "Naixement": "1916",
  "Deportat": "1941",
  "Camp": "Mauthausen",
  "Desti": "Assassinat el 31/01/1941 a Gusen"
},
{
  "Lat": 41.78137110482708,
  "Lon": 3.029177784599707,
  "Nom": "Ricard Rusalleda Lloveras",
  "Naixement": "1902",
  "Deportat": "1941",
  "Camp": "Mauthausen",
  "Desti": "Assassinat el 12/08/1941 a Gusen"
},
{
  "Lat": 41.7856165402158,
  "Lon": 3.028049007676749,
  "Nom": "Francesc Sabater Roquer",
  "Naixement": "1904",
  "Deportat": "1941",
  "Camp": "Mauthausen",
  "Desti": "Assassinat el 20/01/1942 a Gusen"
},
{
  "Lat": 41.78521664488648,
  "Lon": 3.0267193397986176,
  "Nom": "Basilio Polo Pulgar",
  "Naixement": "1910",
  "Deportat": "1941",
  "Camp": "Mauthausen",
  "Desti": "Alliberat"
},
{
  "Lat": 41.78390887842806,
  "Lon": 3.032809214344591,
  "Nom": "Josep Jordà Perich",
```

```
"Naixement": "1913",
"Deportat": "1941",
"Camp": "Mauthausen",
"Desti": "Alliberat"
},
{
  "Lat": 41.78104826623729,
  "Lon": 3.0281300489735505,
  "Nom": "Antoni Moret Barti",
  "Naixement": "1919",
  "Deportat": "1940",
  "Camp": "Mauthausen",
  "Desti": "Alliberat"
}
]
```