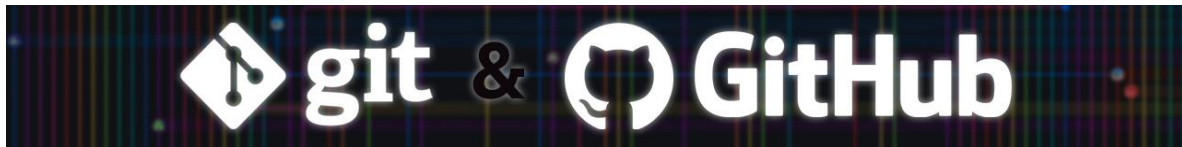




Clase 1:



Ejercicio:

Empecemos

1. Abrir desde VSC (Visual Studio Code) nuestra carpeta de trabajo

▼ EJERCICIOPRACTICO

2. Dentro de nuestra carpeta de trabajo creamos un archivo llamado index.html e insertamos una estructura base de HTML.

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Document</title>
7 </head>
8 <body>
9
10 </body>
11 </html>
```

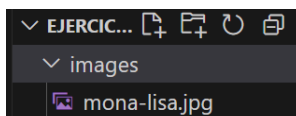
3. Creamos un archivo llamado index.js y lo vinculamos a nuestro archivo index.html

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Document</title>
7 </head>
8 <body>
9   <script src="./index.js"></script>
10 </body>
11 </html>
```

4. Dentro de nuestra estructura de trabajo creamos una carpeta llamada images y dentro de ella guardaremos las imágenes que requerimos.

OJO:

Se deben renombrar las imágenes para poder trabajar con nombres cortos fáciles de usar.





5. Dentro de nuestra estructura de trabajo creamos una carpeta llamada data y dentro de ella creamos un archivo llamado data.js el cual tendrá la siguiente estructura.

```
data > JS data.js > pinturas
1  export const pinturas = [
2      {
3          id:1,
4          name: "LA GIOCONDA, LEONARDO DA VINCI",
5          image:"./images/mona-lisa.jpg"
6      }
7  ]
```

6. Esta data la debemos exportar e importarla en el archivo index.js

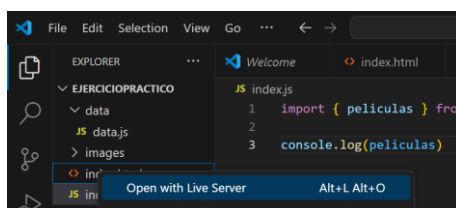
Realicemos lo siguiente: En nuestro archivo index.html relacionamos nuestro archivo data y tanto al archivo data como a index.js les indicamos type="module"

```
< index.html > html > body > script
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Document</title>
7  </head>
8  <body>
9      <script src="./index.js" type="module"></script>
10 </body>
11 </html>
```

7. Dentro del archivo index.js importaremos el archivo data y verificaremos por consola la obtención de la data.

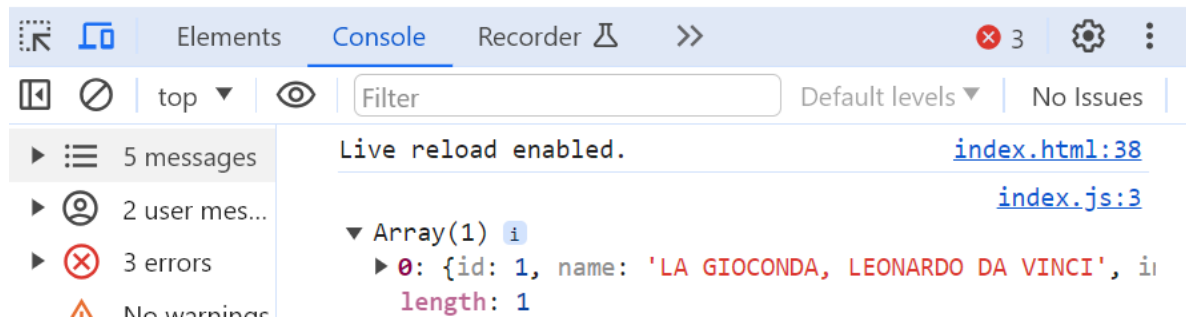
```
JS index.js
1  import { pinturas } from "./data/data.js";
2
3  console.log(pinturas)
```

Ejecutamos con Open with Live Server





En la consola nos debe de mostrar el arreglo con 1 objeto



Ya que tenemos nuestra data, empecemos a pintar la galería de pinturas en HTML.

MOSTRAR PINTURAS EN HTML...

1. Crear una división como contenedor principal con un id: "container"
2. Poner un título principal como el nombre: **Portafolio de pinturas famosas**
3. Agregar etiqueta de línea separadora - `<hr/>`

```
8 <body>
9   <div id="container">
10     <h1>Portafolio de pinturas famosas </h1>
11     <hr>
12   </div>
13
14   <script src="./index.js" type="module"></script>
15 </body>
16 </html>
```

4. Agregar una tarjeta para mostrar las pinturas

```
11 <body>
12   <div id="container">
13     <h1>Portafolio de pinturas famosas </h1>
14     <hr>
15
16     <div class="tarjeta">
17       <h5 class="titulo">Una tarjeta básica</h5>
18       <div class="cuerpo">
19         
20       </div>
21     </div>
22
23   </div>
24
25   <script src="./index.js" type="module"></script>
26 </body>
```



5. Crear un archivo index.css y vincularlo en el archivo index.html.

```
> data      1  <!DOCTYPE html>
> images    2  <html lang="en">
# index.css  3
< index.html 4  <head>
JS index.js 5  <meta charset="UTF-8">
              6  <meta name="viewport" content="width=device-width, initial-scale=1.0">
              7  <link rel="stylesheet" href="./index.css">
```

6. En el archivo index.css agregar los siguientes estilos.

```
1
2  .tarjeta{
3    display: flex;
4    flex-direction: column;
5    justify-content: space-between;
6    text-align: center;
7
8    width: 400px;
9    border: 1px solid lightgray;
10   box-shadow: 2px 2px 8px 4px #d3d3d3;
11   border-radius: 15px;
12   font-family: sans-serif;
13 }
14 .titulo{
15   font-size: 24px;
16   padding: 10px 10px 0 10px;
17 }
18 .cuerpo{
19   padding: 10px;
20 }
21 img{
22   border-radius: 50%;
23   width: 350px;
24   height: 350px;
25 }
26
```

7. Agregar en el archivo HTML una etiqueta <div> que permita mostrar los elementos enviados desde el archivo index.js



```
10
11 <body>
12   <div id="container">
13     <h1>Portafolio de pinturas famosas </h1>
14     <hr>
15
16     <div id="item"></div>
17
18     <template id="template-card">
19       .....<div class="tarjeta">
20       .....<h5 class="titulo">Una tarjeta básica</h5>
21       .....<div class="cuerpo">
22       .....
23       .....</div>
24       .....</div>
25     .....</template>
26   </div>
27
28   <script src="./index.js" type="module"></script>
29 </body>
```

8. a toda la estructura de la tarjeta la vamos a contener en una itiqueta llamada <template> y le vamos a poner un id="template-card"

```
10
11 <body>
12   <div id="container">
13     <h1>Portafolio de pinturas famosas </h1>
14     <hr>
15
16     <div id="item"></div>
17
18     <template id="template-card">
19       .....<div class="tarjeta">
20       .....<h5 class="titulo">Una tarjeta básica</h5>
21       .....<div class="cuerpo">
22       .....
23       .....</div>
24       .....</div>
25     .....</template>
26   </div>
27
28   <script src="./index.js" type="module"></script>
29 </body>
```

9. en el archivo index.js capturamos algunos elementos desde el HTML, tales como: la división con id ítems y el template con id template-card. Adicional adicionaremos un fragmento



```
JS index.js > ...
1  import { pinturas } from "../data/data.js";
2
3  console.log(pinturas)
4
5  const items = document.getElementById('items');
6  const templateCard = document.getElementById('template-card').content;
7  const fragment = document.createDocumentFragment();
8
```

Crearemos nuestra función para cargar la data y llamamos al DOMContentLoaded para el momento que se ejecute nuestra aplicación nos cargue la data con las pinturas en la estructura que definiremos

10. La función la llamaremos loadData y recibirá como parámetro la data que importamos desde el archivo data.js

```
9  const LoadData = pinturas =>{
10    pinturas.forEach(data => {
11      const {id,name,image} = data
12      templateCard.querySelector('h5').textContent =name;
13      templateCard.querySelector('img').setAttribute('src',image);
14      const clone = templateCard.cloneNode(true);
15      fragment.appendChild(clone)
16    });
17    items.appendChild(fragment)
18  }
19
20  document.addEventListener('DOMContentLoaded', ()=>{
21    LoadData(pinturas)
22  })
```

Esta función recibirá las pinturas y la debemos recorrer, lo haremos con un foreach. Cuando ingresemos al foreach y nos quede el objeto, lo desestructuraremos y asignamos a la estructura del template los atributos. Clonamos los nodos del template card, empezamos a adicionar los nodos al fragmento (uno tras otro) y luego mostramos el fragmento en la división con id ítems

Texto original:

```
const LoadData = pinturas =>{
  pinturas.forEach(data => {
    const {id,name,image} = data
    templateCard.querySelector('h5').textContent =name;
    templateCard.querySelector('img').setAttribute('src',image);
    const clone = templateCard.cloneNode(true);
    fragment.appendChild(clone)
  });
  items.appendChild(fragment)
}
```



```
document.addEventListener('DOMContentLoaded', ()=>{  
  LoadData(pinturas)  
})
```

Para finalizar:

Vamos a agregar mas elementos en el archivo data.js de la siguiente manera.

```
data > JS data.js > [?] pinturas  
1  export const pinturas = [  
2    {  
3      id:1,  
4      name: "LA GIOCONDA, LEONARDO DA VINCI",  
5      image:"./images/mona-lisa.jpg"  
6    },  
7    {  
8      id:2,  
9      name: "IMPRESIÓN, SOL NACIENTE, CLAUDE MONET",  
10     image:"./images/sunrise.jpg"  
11   }  
12 ]
```

Resultado Final:

Portafolio de pinturas famosas

LA GIOCONDA, LEONARDO DA
VINCI



IMPRESIÓN, SOL NACIENTE,
CLAUDE MONET





Formulario de entrega:

<https://forms.gle/ircE93xEyikCKZkQ6>