ACÁMICA

TEMA DEL DÍA

Módulos

Practiquemos cómo crear módulos para importar y exportar código en una aplicación de Javascript.

Agenda

Daily

Contexto

La variable var

El objeto window

Módulos

import / export

Actividades

Daily

Daily

Sincronizando...

Bitácora

¿Cómo te ha ido?
¿Obstáculos?
¿Cómo seguimos?

Challenge

¿Cómo te ha ido?
¿Obstáculos?
¿Cómo seguimos?

Contexto (scope)

Contexto

Es el espacio en el que una variable está disponible para ser consultada.

En inglés, a este concepto se lo conoce como *scope*.

En Javascript, un contexto se define con un par de llaves `{}` y todas las variables que se creen dentro de este par serán sólo accesible dentro de las llaves.

La variable `var`

Es el primer tipo de variable que tuvo el lenguaje Javascript.

La variable **`var`** a diferencia de **`let`** y **`const`**, tiene por defecto un contexto global.

```
if(5 > 1) {
  var algo = 9;
  console.log(algo)
}


console.log(algo) // ¿que muestra?
// 9
```

El objeto `window`

Es un objeto que todos los navegadores tienen en su memoria. Contiene información sobre la ventana en cuestión.

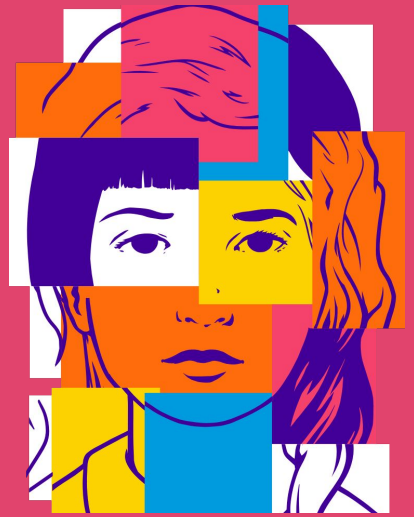
Programo

squad lead

Programo

Veamos una demostración de los conceptos explicados previamente en la bitácora.

Módulos *(repaso)*

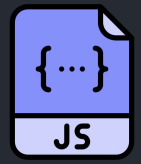


Módulos

Proveen una forma de compartir códigos contenidos en contextos locales.

Modular una aplicación asegura un crecimiento estructurado al independizar funcionalidades.

index.html

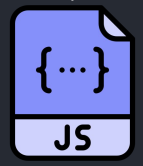


script.js

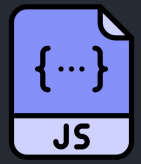
Contexto
mezclado

index.html

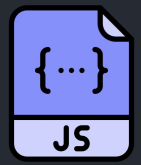
```
<script src="script.js"></script>
```



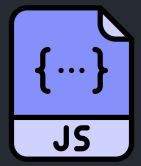
script.js



scriptA.js



scriptB.js



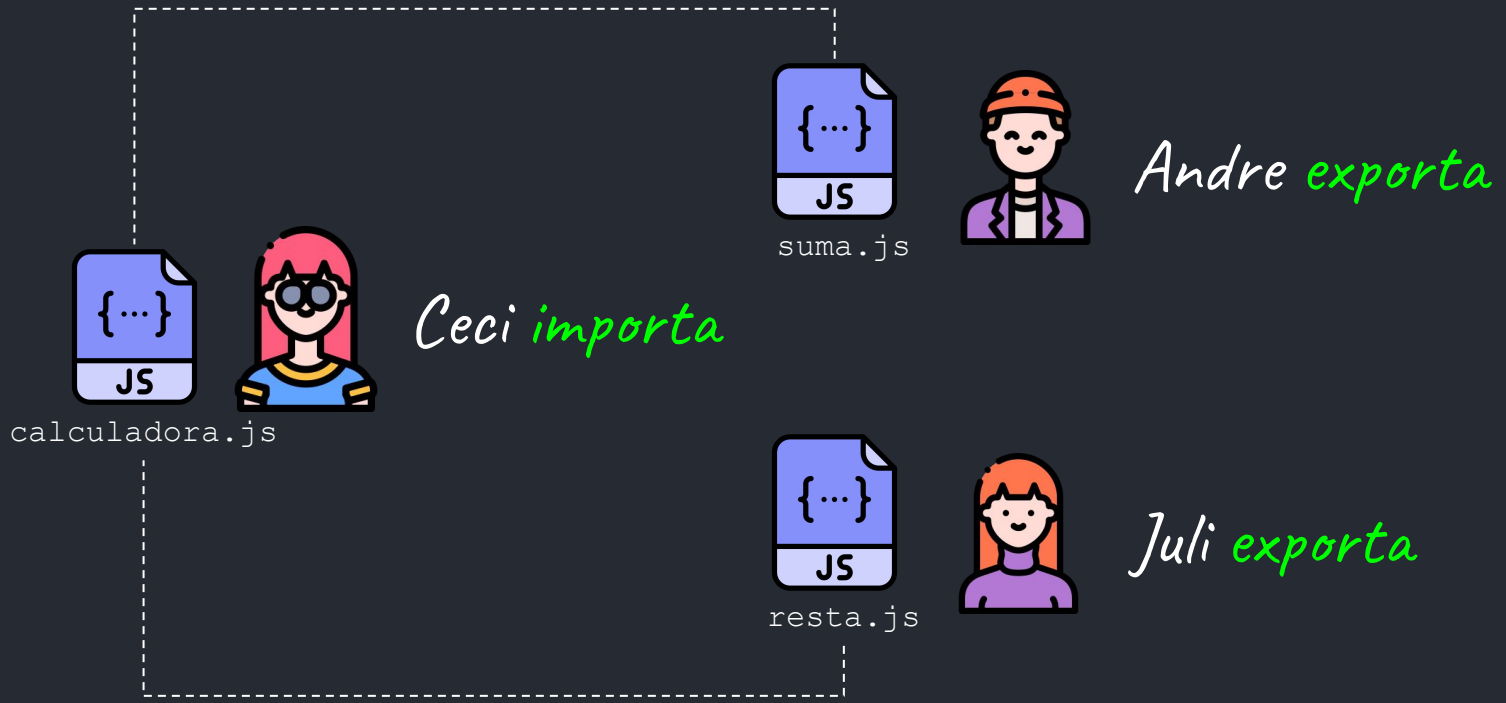
scriptC.js

`import`

Sirve para importar funciones o variables a un archivo.

`export`

Sirve para exportar funciones o variables a otros archivos.


JS
suma.js
Andre exporta
JS
calculadora.js
Ceci importa
JS
resta.js
Juli exporta

`import` *variables y/o funciones* `export`

export*

```
1  exportar función
2  export default function nombreFuncion () {
3    // ...
4  }
5  exportar variables
6  export let edad = 40;
7  export const nombre = "Juli";
8
```

*solo un export default por archivo

import*

```
1  importar un export default
2  import nombreFuncion from "./direccionAlArchivo";
3
4  importar varios exports normals (los no default)
5  import { edad, nombre } from "./direccionAlArchivo";
6
```

*se hacen en el principio del archivo

Programamos

¡todxs!

Calculadora

¡Necesitamos **6 personas** voluntarias para esta práctica!

En esta práctica vamos a construir una calculadora pequeña.

Una persona sera encargada de iniciar un sandbox nuevo, de tipo Vanilla, y deberá crear 4 archivos dentro de la carpeta src: (tendrá en total 5, más el index.js)

- suma.js
- resta.js
- multiplicacion.js
- division.js

Cada archivo deberá tener una función exportada, la cual toma 2 números como parámetros y retornara la operación correspondiente.

Cuatro personas deberán ser invitadas al sandbox para que construyan estas funciones, en paralelo.

Calculadora

Al terminar las 4 funciones, la última persona voluntaria deberá importar las funciones en index.js e invocarlas al menos dos veces cada una, con parámetros distintos, por ejemplo:

```
console.log(suma(2,4));
console.log(suma(12,-64));
console.log(resta(55,40));
console.log(resta(234,22));
console.log(multiplicacion(11,76));
console.log(multiplicacion(36,49));
console.log(division(22,443));
console.log(division(233,47));
```

Tip: envuelve las funciones en un console.log para ver de inmediato el resultado retornado.

DWFE

Nos tomamos unos minutos para completar esta encuesta.

Queremos saber cómo valoran mi tarea hasta acá. ¡Les va a llevar solo un minuto!

¡BREAK!

Ph. Credit: Drew Coffmann

Programan

estudiantes

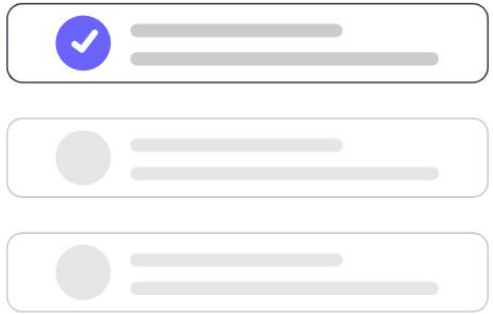
Taskineitor v0.2

Construye la aplicación Taskineitor. Crea una estructura de archivos que te permita separar por módulos la siguiente información:

- Mostrar información de una tarea
- Mostrar todas las tareas
- agregar una tarea
- borrar una tarea
- modificar una tarea

Importa todos los módulos a `index.js` para poder ejecutar todas las funciones desde ahí.

A continuación te recordamos las funcionalidades:



Taskineitor v0.2

La función **`mostrarInfoTarea()`** recibe como argumento el número del id, y dentro de esta se deberá iterar sobre el array de tareas hasta encontrar el mismo id.

Al encontrarlo, muestra por consola todas las propiedades del objeto.

```
let tareas = [
  {
    titulo: "lavar los platos",
    completada: false,
    fechaCreada:  "9/22/2020" ,
    id: 1
  },
  {
    titulo: "ordenar la ropa",
    completada: true,
    fechaCreada:  "9/22/2020" ,
    id: 2
  },
  {
    titulo: "ir al gym",
    completada: true,
    fechaCreada:  "9/22/2020" ,
    id: 3
  },
]

mostrarInfoTarea(2);
```

Taskineitor v0.2

La función **`agregarTarea()`** debe tomar un objeto de tarea como argumento y modificar el array de tareas al agregar, por último, el nuevo objeto. La fecha deberá estar especificada en formato "mes/dia/año"

Adicionalmente, deberás generar un id nuevo para el objeto tarea, que será la continuación del anterior. Por ejemplo, si la última tarea en el array tiene el id 2, la nueva tarea tendrá el id 3.

```
let tareas = [
  {
    titulo: "lavar los platos",
    completada: false,
    fechaCreada:  "9/22/2020" ,
    id: 1
  }
];

let nuevaTarea = {
  titulo: "ir al super",
  completada: false,
  fechaCreada: "12/03/2020"
}

agregarTarea(nuevaTarea);
```

Taskineitor v0.2

La función **`modificarTarea()`** debe tomar un objeto de tarea como argumento y reemplazar la tarea, con el mismo id, dentro del array.

```
let tareas = [
  {
    titulo: "lavar los platos",
    completada: false,
    fechaCreada:  "9/22/2020" ,
    id: 1
  }
];

let tareaModificada = {
  titulo: "lavar los platos",
  completada: true,
  fechaCreada:  "9/22/2020" ,
  id: 1
}

modificarTarea(tareaModificada);
```

Taskineitor v0.2

La función **`borrarTarea()`** debe tomar un id como argumento y borrar el objeto que tenga el mismo id.

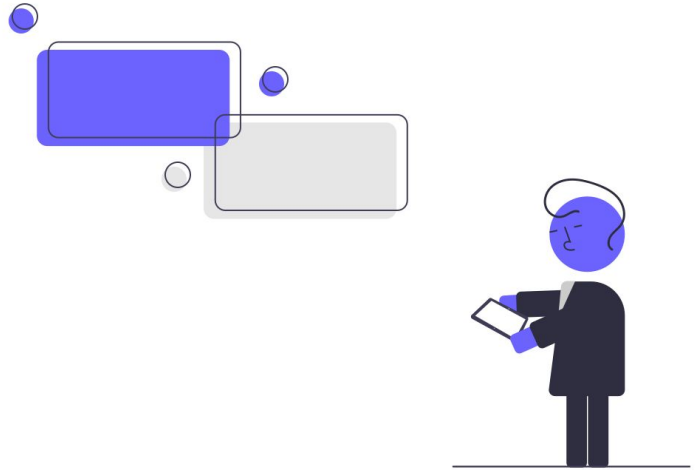
```
let tareas = [
  {
    titulo: "lavar los platos",
    completada: false,
    fechaCreada: "9/22/2020" ,
    id: 1
  },
  {
    titulo: "buscar precio del jean",
    completada: false,
    fechaCreada: "9/22/2020" ,
    id: 2
  },
  {
    titulo: "ir al super",
    completada: false,
    fechaCreada:  "9/22/2020" ,
    id: 2
  },
];

borrarTarea(2);
```

Soluciones

Soluciones

A continuación te damos una solución a los ejercicios planteados, asegurate de revisarlos una vez que completes los tuyos, para poder comparar.

Revisa las soluciones aquí.



Para la próxima

1) Termina los ejercicios del encuentro de hoy.
2) Lee la toolbox 25

ACÁMICA