

Programa académico CAMPUS



JAVASCRIPT MODULARIZACIÓN

Trainer
Ing. Carlos H. Rueda C.



```
import && export  
{ statements } in  
'./JavaScript'
```

JS

JAVASCRIPT

MÓDULOS

A medida que nuestra aplicación crece, queremos dividirla en múltiples archivos, llamados “**módulos**”.

Un **módulo** puede contener una clase o una biblioteca de funciones para un propósito específico.

JAVASCRIPT MÓDULOS

Un módulo es solo un **archivo**.

Un script es un módulo. Tan sencillo como eso.

Esto hace que sea más **fácil** mantener una base de código.

JAVASCRIPT MÓDULOS

VENTAJAS

1. **Organización**
2. **Reutilización**
3. **Encapsulamiento:** lo que significa que el código dentro de un módulo está aislado del código externo, lo que reduce los efectos secundarios y ayuda a mantener la integridad del programa.
4. **Mantenibilidad:** es más fácil realizar cambios, corregir errores y agregar nuevas funcionalidades sin afectar el funcionamiento de otras partes del programa.

JAVASCRIPT MÓDULOS

¿Cómo se logra?



Se puede intercambiar funcionalidad con las palabras **export** e **import**.

- **export** etiqueta las variables y funciones que deberían ser accesibles desde fuera del módulo actual.
- **import** permite importar funcionalidades desde otros módulos.

JAVASCRIPT MÓDULOS

Ejemplo 1



archivo1.js

```
// Exportar una función desde  
archivo1.js  
export function suma(a, b) {  
    return a + b;  
}
```

```
// Exportar una variable desde  
archivo1.js  
export const PI= 3.1416;
```


JAVASCRIPT MÓDULOS

Ejemplo 1



archivo2.js

```
// Importar la función suma
// y la variable pi desde archivo1.js
import { suma, PI } from './archivo1.js';

// Imprimirá 5
console.log(suma(2, 3));
// Imprimirá 3.1416
console.log(PI);
```


JAVASCRIPT MÓDULOS

Ejemplo 1



Ejemplo-modulos.html

```
<body>  
  <script src="archivo2.js">  
  </script>  
</body>
```

JAVASCRIPT MÓDULOS

Ejemplo 1



Ejemplo-modulos.html

Los módulos también se basan **type="module"** en la etiqueta **<script>**.

```
<body>  
  <script src="archivo2.js" type="module">  
  </script>  
</body>
```

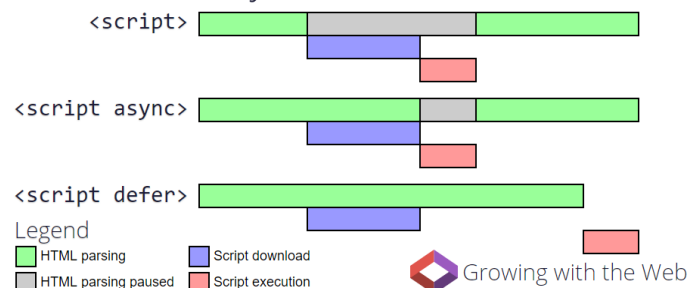
JAVASCRIPT MÓDULOS

Parámetros <script>

async

Indica que el script se descarga de forma asíncrona mientras se continúa procesando la página. Esto significa que no bloqueará la representación de la página mientras se carga el script.

async vs defer



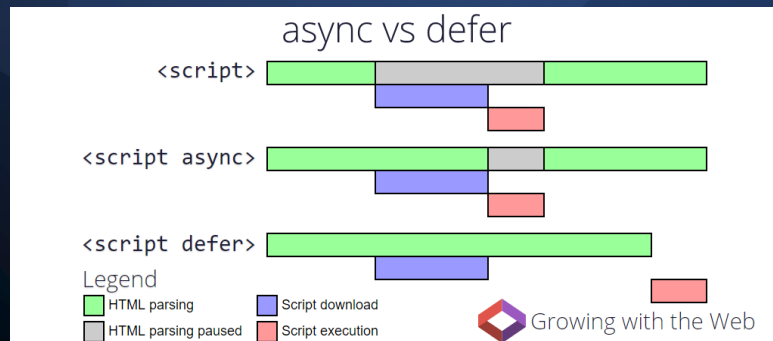
```
<script src="script.js" async></script>
```

JAVASCRIPT MÓDULOS

Parámetros <script>

defer

Indica que el script se descarga de forma asíncrona, pero *no se ejecutará* hasta que la página haya terminado de cargarse. Esto permite que el script se cargue en paralelo con el análisis de la página, pero se ejecute en el orden en que aparece en el documento.

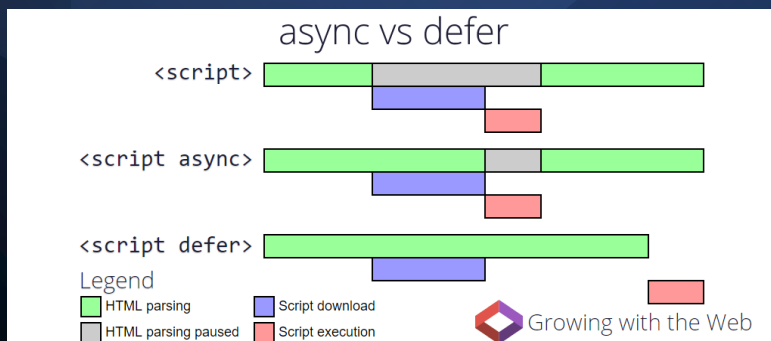
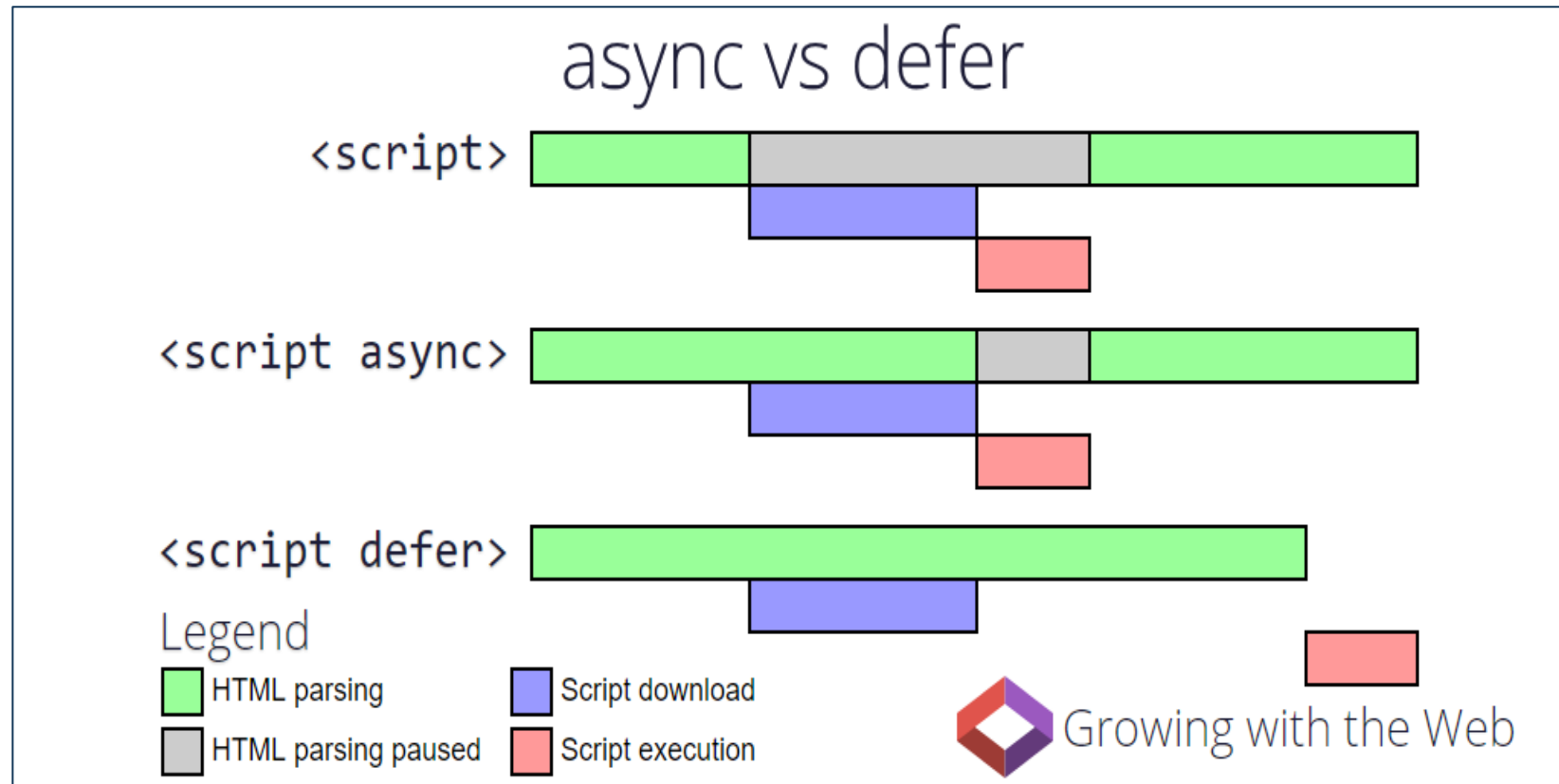


```
<script src="script.js" defer></script>
```

JAVASCRIPT MÓDULOS

Parámetros <script>

async vs defer



JAVASCRIPT MÓDULOS

async vs defer

```
<script src="archivo2.js"  
type="module" defer></script>  
  
</head>  
<body>  
  
</body>  
</html>
```

JAVASCRIPT MÓDULOS

IMPORTAR TODOS

*** as nombre_modulo**

```
// Importar todos con "*" con alias  
// alias "as" nombre_alias  
import * as archivo1 from  
'./archivo1.js';
```

```
console.log(archivo1.suma(2, 3));  
console.log(archivo1.PI);
```


JAVASCRIPT MÓDULOS

CONFLICTOS DE NOMBRES

archivo3.js

```
export const PI = 3.1415926;
```

Importarlo en archivo2.js

```
import * as archivo1 from './archivo1.js';  
// importando PI de archivo3.js  
// usando alias para superar el conflicto  
import {PI as PII} from './archivo3.js'
```

```
console.log(archivo1.suma(2, 3));  
console.log(archivo1.PI);  
console.log(PII);
```

JAVASCRIPT MÓDULOS

USAR MODULOS CON HTTP(s)

Nota

Los módulos solo funcionan con el protocolo HTTP(s).

Una página web abierta a través del protocolo file:// no puede usar la importación/exportación.

JAVASCRIPT MÓDULOS

EXPORTACIONES PREDETERMINADAS

Las exportaciones predeterminadas (**default**) se logran con `export default`.

Solo puede tener ***una*** exportación predeterminada en un archivo.

JAVASCRIPT MÓDULOS

EXPORTACIONES PREDETERMINADAS

Crear el archivo *mensaje.js*

```
const mensaje = () => {  
  const nombre = "Oscar";  
  const edad = 21;  
  return nombre + " tiene " + edad +  
  "años.";  
};  
export default mensaje;
```

JAVASCRIPT MÓDULOS

EXPORTACIONES PREDETERMINADAS

Modificar el *archivo2.js*

```
import * as archivo1 from  
'./archivo1.js';  
import {PI as PII} from './archivo3.js';  
// importando predeterminado: mensaje  
import mensaje from './mensaje.js';
```

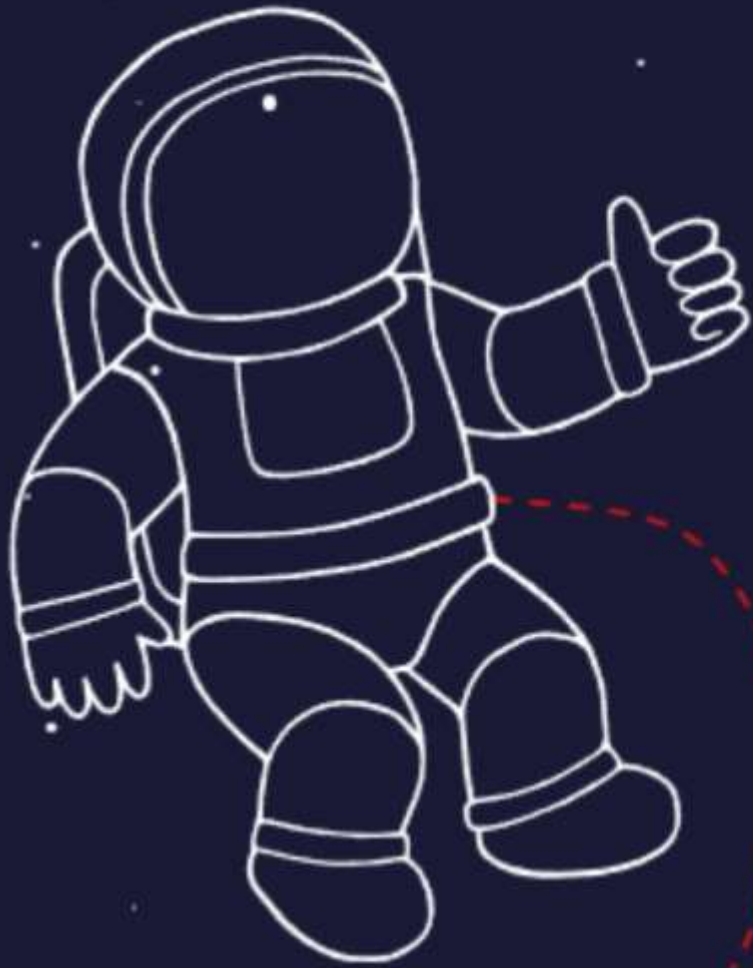
```
console.log(archivo1.suma(2, 3));  
console.log(archivo1.PI);  
console.log(PII);  
console.log(mensaje());
```

JAVASCRIPT MÓDULOS

EJERCICIO

Modificar el archivo de solución de la prueba del viernes pasado.

- ✓ Agregando localStorage para que los productos que se agregan no se pierdan.
- ✓ Modularizar el código JavaScript (por ejemplo separando clases y funciones)



Programa académico CAMPUS



Trainer
Ing. Carlos H. Rueda C.

