**¿Qué es la Modularización en JavaScript?**

La modularización es el proceso de dividir tu código en módulos o archivos separados para organizarlo mejor y facilitar su mantenimiento. Cada módulo puede contener funciones, clases, objetos, o cualquier otra parte del código que se necesite. La idea es que cada módulo tenga una responsabilidad específica y pueda ser reutilizado en diferentes partes de la aplicación.

**Exportar e Importar en JavaScript:** JavaScript moderno utiliza módulos ES6 (también conocidos como módulos ECMAScript) para manejar la modularización. Esto se hace a través de las palabras clave export e import.

**Exportar:** Hay dos formas principales de exportar en JavaScript: exportaciones nombradas y exportación por defecto.

* **Exportación Nombrada:** Permite exportar múltiples valores desde un módulo. Cada valor exportado debe ser importado con el mismo nombre.

Texto

Descripción generada automáticamente

* **Exportación por Defecto:** Permite exportar un solo valor por módulo. Este valor puede ser una función, una clase, un objeto, etc.

**Texto

Descripción generada automáticamente con confianza media**

**Importar:** Para utilizar los módulos exportados, debes importarlos en el archivo donde los necesitas.

* **Importar Exportaciones Nombradas**

**Texto

Descripción generada automáticamente**

* **Importar Exportación por Defecto**

**Texto

Descripción generada automáticamente**

* **Importar Todo un Módulo**

**Texto

Descripción generada automáticamente**

**Utilización del Atributo type="module":** Para que el navegador entienda que estás utilizando módulos ES6, debes indicar esto en la etiqueta <script> de tu HTML usando el atributo type="module".

**Características del Atributo type="module"**

* **Soporte para import y export:** Permite utilizar la sintaxis de módulos.
* **Modo Estricto:** Los módulos siempre se ejecutan en modo estricto.
* **Carga Asíncrona:** Los scripts con type="module" se cargan de manera asíncrona.
* **Caché:** Los módulos se cargan una sola vez y se reutilizan, evitando ejecuciones múltiples del mismo código.