

# IIFE

La expresión `(function(){ /* código */ })();` en JavaScript se conoce como una función autoejecutable o función autoinvocada (IIFE, por sus siglas en inglés, Immediately Invoked Function Expression). Se utiliza para encapsular un fragmento de código dentro de una función anónima y ejecutarlo inmediatamente después de su definición. Esto se hace envolviendo la función en paréntesis y luego agregando ()` al final.`

La principal razón para utilizar una IIFE es crear un ámbito o scope local para las variables dentro de la función. Esto ayuda a prevenir la contaminación del ámbito global y evita conflictos de nombres con otras variables en el código. Antes de la introducción de los bloques de alcance (con `let` y `const` en JavaScript, las IIFE eran una forma común de lograr el encapsulamiento de variables.`

Aquí tienes un ejemplo básico de una IIFE:

```
(function() {  
    var x = 10;  
    console.log(x); // Se puede acceder a x dentro de la IIFE  
})();  
  
console.log(x); // Esto arrojará un error, ya que x no está definido en el ámbito global
```

En este ejemplo, la variable `x` está definida dentro de la IIFE y es inaccesible desde el ámbito global. Esto ayuda a evitar conflictos de nombres con otras partes del código que pueden usar la misma variable x`.`

Hoy en día, con la introducción de `let` y `const`, el uso de IIFE es menos común para el encapsulamiento de variables, ya que estas declaraciones de bloque pueden lograr el mismo resultado. Sin embargo, las IIFE todavía se utilizan en situaciones específicas donde se necesita un ámbito local temporal o para mantener la compatibilidad con código más antiguo.`