

An **Immediately Invoked Function Expression (IIFE)** is a JavaScript programming technique that involves defining and executing a function immediately after its creation. This is achieved by enclosing the function within parentheses and immediately invoking it with an additional set of parentheses. IIFEs are commonly used to create private scopes for variables and functions, preventing them from cluttering the global scope. For example, you can encapsulate a block of code within an IIFE to ensure that any variables or functions defined within it do not interfere with or pollute the global scope. This is particularly useful in scenarios where you want to avoid naming conflicts and maintain a clean, organized code structure. IIFEs can also accept arguments and return values, making them versatile tools for creating reusable code blocks while keeping the internal logic hidden from the global scope. Overall, IIFEs are a valuable pattern in JavaScript for enhancing code encapsulation, reducing scope-related issues, and promoting modularity.

```
(function () {  
    var localVar = 'This is a local variable';  
    console.log(localVar);  
})();
```