

IIFE in JS

In JavaScript, an IIFE (Immediately Invoked Function Expression) is a function that is defined and executed immediately after it is defined. It is also known as a **self-executing anonymous function**.

IIFEs are useful for creating a new scope, isolating variables and functions from the global scope, and for hiding implementation details.

IIFEs are typically defined using the function keyword, followed by a pair of parentheses to indicate that the function should be invoked immediately. This is then followed by a pair of parentheses to pass any arguments to the function, and a set of curly braces to define the function's body. The entire expression is then wrapped in another set of parentheses to indicate that it is a function expression.

```
(function () {  
    var message = "Hello World!";  
    console.log(message);  
})(); // logs "Hello World!"
```

In the above example, the variable message is only available within the scope of the IIFE and can't be accessed from outside.

IIFEs can also accept arguments, for example:

```
(function (arg) {  
    console.log(arg);  
})("Peter"); // logs "Peter"
```

You can also pass arguments to an IIFE that returns a value:

```
var result = (function (num) {  
    return `The number is ${num}.`;  
})(12);  
console.log(result); // logs "The number is 12."
```