

wawiwa

CIERRE

CIERRE

Un **cierre** (o "closure" en inglés) es la combinación de una función junto con las referencias a su entorno circundante (el **entorno léxico**). En otras palabras, un cierre te da acceso al ámbito de una función exterior desde una función interior. En JavaScript, los cierres se crean cada vez que se crea una función, en el momento de la creación de la función.

Cierre

```
function init() {  
  var name = 'Mozilla'; // name is a local variable created by init  
  
  function displayName() { // displayName() is the inner function, a closure  
    alert(name); // use variable declared in the parent function  
  }  
  
  displayName();  
}  
  
init();
```

init() crea una variable local llamada “name” y una función llamada displayName(). La función displayName() es una función interna que está definida dentro de init() y está disponible solamente dentro del cuerpo de la función init() function.

Ten en cuenta que la función `displayName()` no tiene variables locales propias. Sin embargo, dado que las funciones internas tienen acceso a las variables de las funciones externas, `displayName()` puede acceder a la variable `name` declarada en la función padre - `init()`.

Cierre

Correr esta función tiene exactamente el mismo efecto que el ejemplo previo de la función `init()`. Lo que es diferente (e interesante) es que la función interna `displayName()` es retornada de la función externa antes de ser ejecutada.

```
function makeFunc() {  
  var name = 'Asaf';  
  
  function displayName() {  
    alert(name);  
  }  
  
  return displayName;  
}  
  
var myFunc = makeFunc();  
  
myFunc();
```


Cierre

```
function adding(x) {  
  return function(y) {  
    return x + y;  
  };  
}  
  
var add10 = adding(10);  
var add15 = adding(15);  
  
console.log(add10(2)); // ?  
console.log(add15(2)); // ?
```

Cierre

```
function adding(x) {  
  return function(y) {  
    return x + y;  
  };  
}  
  
var add10 = adding(10);  
var add15 = adding(15);  
  
console.log(add10(2)); // 12  
console.log(add15(2)); // 17
```

En este ejemplo, definimos una función **adding**(x), que tome un solo argumento x, y retorna una nueva función. La función que retorna toma un único argumento y, y retorna la suma de x y y.

En esencia, **adding** es una fábrica de funciones. Crea funciones que pueden agregar un argumento específico a su argumento. En el ejemplo, la fábrica de funciones crea dos nuevas funciones: una agrega cinco a su argumento y la otra agrega 10.

add5 y **add10** son ambas cierres. Comparten la misma definición de cuerpo de la función, pero almacenar diferentes entornos léxicos. En el entorno léxico de **add5**, x es 5, mientras que el entorno léxico de **add10**, x es 10.

Cierre práctico

```
body {  
  font-family: Helvetica, Arial, sans-serif;  
  font-size: 12px;  
}
```

```
h1 { font-size: 1.5em; }  
h2 { font-size: 1.2em; }
```

```
function makeSizer(size) {  
  return function() {  
    document.body.style.fontSize = size + 'px';  
  };  
}
```

```
var size12 = makeSizer(12);  
var size14 = makeSizer(14);  
var size16 = makeSizer(16);
```

Los cierres son útiles porque te permiten asociar datos con una función que opera en esos datos. Utilizamos un cierre en cualquier lugar donde normalmente usarías un objeto con solo un método.

Cuando programas aplicaciones web, defines un comportamiento y luego lo adjuntas a un evento que es desencadenado por el usuario (como un click o pulsar una tecla). El código se adjunta como una callback (una única función que se ejecuta en respuesta al evento).

Por ejemplo, si queremos agregar botones a una página, para ajustar el tamaño del texto, especificamos el font-size del elemento body (en píxeles) y luego establecemos el tamaño de los otros elementos en la página (como los encabezados) usando la unidad relativa "em".

Cierre práctico

```
body {
  font-family: Helvetica, Arial, sans-serif;
  font-size: 12px;
}
```

```
h1 { font-size: 1.5em; }
h2 { font-size: 1.2em; }
```

```
function makeSizer(size) {
  return function() {
    document.body.style.fontSize = size + 'px';
  };
}
```

```
var size12 = makeSizer(12);
var size14 = makeSizer(14);
var size16 = makeSizer(16);
```

size12, size14, y size16 ahora son funciones que ajustan el tamaño del texto del body a 12, 14, y 16 pixels, respectivamente. Puedes adjuntarlas a botones (en este caso hipervínculos) como se muestra en el siguiente ejemplo de código.

```
document.getElementById('size-12').onclick = size12;
document.getElementById('size-14').onclick = size14;
document.getElementById('size-16').onclick = size16;
```

Or

```
<a href="#" id="size-12">12</a>
<a href="#" id="size-14">14</a>
<a href="#" id="size-16">16</a>
```


Cierre avanzado - bueno saberlo

```
const counter = (function () {  
  let privateCounter = 0;  
  
  function changeBy(val) { privateCounter += val; }  
  
  return {  
    increment () { changeBy (1); },  
    decrement () { changeBy (-1); },  
    value () { return privateCounter; }  
  }  
})();  
  
console.log(counter.value()); // logs 0  
counter.increment();  
counter.increment();  
console.log(counter.value()); // logs 2  
counter.decrement();  
console.log(counter.value()); // logs 1
```

A nighttime photograph of a city skyline. In the foreground, a large, curved skyscraper with a grid-like facade is illuminated with blue and white lights. To its right, another tall building with a similar grid facade is visible. In the background, other skyscrapers are lit up, including one with a 'SQUARE' sign and another with a 'WAWIWA' sign. The street in the foreground is dark, with some streetlights and a blue sign visible on the right side.

wawiwa

Questions?