

JavaScript Module Pattern es uno de los patrones de diseño más típicos en JavaScript y nos ayuda a organizar y aislar las distintas funciones y clases con las que trabajamos. JavaScript por defecto no soporta el concepto de Package/NameSpace. Vamos a ver como implementar en JavaScript este patrón de diseño.

```
<html>
<head>
<script type="text/javascript">

(function(window){

var suma = function(x, y){
return x + y;
}

var resta = function(x, y){
return x - y;
}

var matematicas = {
suma: function(a, b){
return suma(a,b);
},
resta: function(a, b){
return resta(a, b);
}
}

window.matematicas = matematicas;
```

```
})(window);

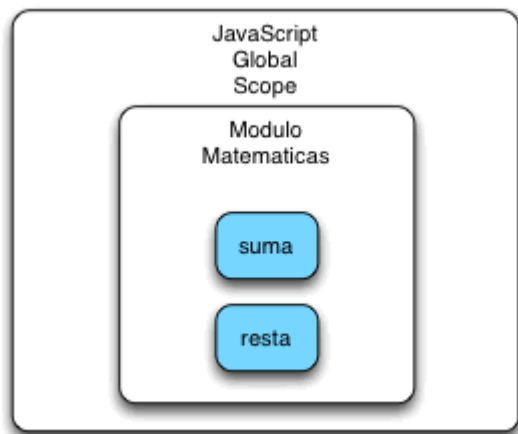
console.log(matematicas.suma(5, 5));
console.log(matematicas.resta(5, 5));

</script>
</head>
<body>

</body>
</html>
```

## JavaScript Module Pattern y IIFE

El código define dos funciones `suma` y `resta` las cuales se agrupan en un objeto que denominamos `matemáticas`, realizado este primer paso, englobamos todo dentro de una función IIFE (Immediate Invoked Function Expression) que es la encargada de aislar este bloque de código del resto de código de JavaScript y tiene la peculiaridad de que se autoinvoca. Esta funcionalidad es muy típica de frameworks como jQuery que también usa una función IIFE. De esta forma se consigue diseñar las funciones sin afectar al global scope.



Otros artículos relacionados: [JavaScript Iteradores](#) ,[JavaScript Console](#) ,[JavaScript Map y JSON](#)