

## LES IIFIEs “Immediately Invoke Function Expression”

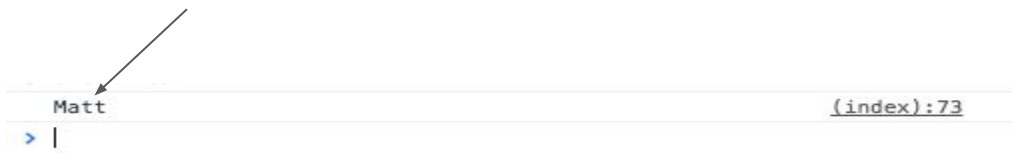
Pour faire simple c'est **une fonction anonyme** qui est déclarer et immédiatement invoqué a la volé.

Les variables situé à l'intérieur d'une IIFEs ne sont pas accessible dans le contexte global car il dépende du scope de fonction donc il ne sont pas accessible dans les autres page.js **cela sert à privatiser une variable / à la protéger.**

```
1 (function(){  
2  
3 // Voici le code d'une IIFE's  
4  
5     var myName = "Matt";  
6  
7     console.log(myName);  
8  
9 })();  
10
```

La fonction déclarer est aussitôt invoqué.

Résultat.



```
Matt  
(index):73  
> |
```

# Créer un objet qui nous retourne dès fonction ou d'autre objet grâce au closure et aux IIFEs "Immediately Invoked Function Expression".

```
<> index.html JS module1.js x JS module2.js JS module3.js
1  const myClosure = (function(){
2
3      var myPassword = "12345";
4
5      function setPassword(newPassword){
6          myPassword = newPassword;
7      };
8
9      function getPassword(){
10         return myPassword;
11     };
12
13     return {
14         getPassword : getPassword
15     }
16
17 })();
18
19
20
21
22
```

La fonction IIFEs mis dans une closure retourne un objet avec un alias de la fonction a lequel on veut accéder.

```
<> index.html JS module1.js JS module2.js JS module3.js x
1
2  console.log(myClosure.getPassword());
```

On peut donc appeler la fonction getPassword dans un autre module en le préfixant du module dans lequel elle est.

```
top Filter Default levels Group similar
12345 module3.js:2
```

Résultat