

Funções imediatas ou funções auto invocadas (IIFE) que significa uma expressão de função invocada automaticamente.

IIFE -> *Immediately invoked function expression*

Imagina que quando a gente está trabalhando no Browser a gente quer fugir do escopo global a gente não quer que as nossas funções, nossas variáveis que a gente definiu toquem o escopo global porque elas podem ser alteradas, enfim, pode acontecer algum problema e além da gente não querer outras bibliotecas atrapalhem o nosso código a gente também não quer poluir o escopo global com todas as nossas funções, as nossas variáveis etc. Então o que a gente faz para fugir do escopo global é criar uma função qualquer e jogar tudo dentro dela:

```
function anything() {  
  console.log(2242);  
}  
anything();
```

A função é executada imediatamente mas a gente precisou dar um nome para ela e imagina que a gente queira criar uma função anônima porque nesse caso a função anything toca o escopo global e então criando uma função anônima ela não vai ter como tocar o escopo global.

Para a gente criar uma função que é invocada automaticamente quando ela é criada eu posso envolver ela em ()

```
(function() {  
  console.log(12345);  
})(); // aqui eu chamo a função
```

Tudo o que escrever neste código vai ser executado imediatamente e nada do que está dentro toca o escopo global.

```
(function() {  
  const nome = 'Leonardo';  
  console.log(nome);  
})();
```

```
const nome = 'Giovana';  
console.log(nome);
```

Eu não tenho conflito porque o meu código está protegido por uma função que foi invocada imediatamente. Eu poderia ter qualquer coisa no escopo global que ele

não iria afetar o meu código, então dentro dele eu posso escrever o meu código normalmente como eu escreveria de qualquer maneira no escopo global.

Exemplo:

```
(function() {  
  const sobrenome = 'Machado';  
  function criaNome(nome) {  
    return nome + ' ' + sobrenome;  
  }  
  function falaNome() {  
    console.log(criaNome('Leonardo'));  
  }  
  falaNome();  
})(); // Leonardo Machado
```

E se eu quiser passar parâmetros para dentro dessa função eu faço assim:

```
(function(idade, peso, altura) {  
  const sobrenome = 'Machado';  
  function criaNome(nome) {  
    return nome + ' ' + sobrenome;  
  }  
  function falaNome() {  
    console.log(criaNome('Leonardo'));  
  }  
  falaNome();  
  console.log(idade, peso, altura);  
  
})(18, 66, 1.75); // eu já estou invocando a função e eles são os argumentos porque eu já estou executando a minha função aqui.
```

A gente envolve a função dentro de () e eu posso ter várias funções dessa maneira, mas fica mais complicado de eu fazer a função anônima conversar com outra função, mas dá para fazer também através dos parâmetros caso eu precise.

Guia: <https://developer.mozilla.org/en-US/docs/Glossary/IIFE>