

O Closure está muito relacionado ao escopo léxico da função. Vamos criar 2 funções uma função que retorna uma função para que eu consiga ver isso no navegador:

```
function returnFunction() {  
  const name = 'Leonardo'; // é variável do escopo léxico  
  return function () {  
    return name;  
  }  
}
```

Então no escopo dessa função eu tenho a variável name, mas eu vi na aula anterior que eu posso por exemplo retornar uma função que tem acesso ao escopo do mãe dela que é a returnFunction, então essa função anônima tem acesso a variável name então essa função anônima pode retornar, falar o nome e por aí vai aqui eu vou retornar o nome.

Agora eu vou criar uma variável que recebe a função returnFunction(), se eu chamar essa função a minha variável vai receber a função anônima onde eu estou retornando o nome e eu dou um console.log na variável função:

```
const funcao = returnFunction(); // vai receber a função anônima,  
porque a função returnFunction está retornando essa função anônima e o  
que essa função retorna está indo para a variável funcao.  
console.log(funcao);
```

Resultado: [Function]

A gente tem uma Function sem nome porque ela é uma função anônima;

Além de console.log a gente também tem console.dir que não faz a mínima diferença no node então o resultado é o mesmo. Mas se eu abrir o meu arquivo index.js no navegador a gente sabe que a função anônima que foi retornada pela função returnFunction tem acesso a 3 escopos -> *Scopes*.

Por que 3 escopos?

R: Porque a função anônima tem acesso ao escopo da função mãe dela que é a returnFunction, ela tem acesso ao escopo da própria função dela mesmo onde é retornado o nome, e do global que é o de fora da função.

```
// Global  
function returnFunction() {
```

```
const name = 'Leonardo'; // é variável do escopo léxico
return function () {
    return name; // escopo aqui
}
}
```

Mas o que é o closure?

O closure na verdade é a habilidade dessa função anônima em acessar o seu escopo léxico. Então ela tem a habilidade de acessar o seu escopo léxico que é o da mãe dela e o global. O closure é somente a variável que está definida fora dela que ela está utilizando.

Por que Closure?

R: Porque ela está fechando um escopo quando eu executo essa função, então quando eu executar essa função em algum lugar ela vai fechar o escopo ela vai retornar a variável name do jeito que ela está definida.

Então ela tem que relembrar quando eu declaro ela na variável funcao ela tem que relembrar o closure, o escopo léxico dela, ou seja, ela lembra dela e dos vizinhos dela no momento em que ela foi declarada. Então resumindo ela lembra do escopo léxico dela.

Ela tem acesso ao script que é local onde ela foi declarada então se tivesse outra função ela teria acesso a essa outra função vizinha dela também no script. E no escopo global eu posso acessar qualquer coisa no navegador que estiver dentro de window no escopo global.

Agora, imagine que defini uma variavel com valor real eu não mudo esse valor e então o Closure de função para função não vai mudar, mas se eu colocasse essa variável name como parâmetro e no momento da declaração dessa função eu mandasse no Closure dela no parâmetro dela o meu nome seria mostrado normal, mas se eu tivesse uma outra variável e na declaração eu chamaria outro nome a eu teria 2 Closures:

```
// Global
function returnFunction(name) {
    return function () {
        return name; // escopo aqui
    }
}

const funcao = returnFunction('Leonardo');
```

```
const funcao2 = returnFunction('Spare me!'); // Vai ser sempre este
valor
console.dir(funcao);
console.dir(funcao2);
```

Eu tenho uma função que no Closure dela o nome vai ser o meu e no outro Closure que está vindo do mesmo moo nome vai ser Gregory. Então o Closure pode ser alterado de função para função. Então ela precisa fechar o escopo onde eu chamo ela na variável funcao2, ela está fechando o escopo com o parâmetro name chamando o nome então ela está fechando o escopo com o Closure, com o nome sendo Gregory, então neste caso ela está mudando o contexto léxico dela, ela está mudando o valor da variável name.

Então isso a gente chama de Closure quando a gente declara a função e o nome não vai ser mais mudado vai ser sempre Leonardo na variável funcao2 e isso serve o mesmo para a variável funcao2.

```
console.log(funcao(), funcao2());
```

Então isso caracteriza o Closure da função;

Resumindo: Closure é a habilidade que a função tem em acessar o escopo léxico dela que a gente viu na aula anterior.