```
//CLOSURE:
//Closure é uma memória de contexto léxico (o local onde um elemento é escrito) que existe dentro de todas as funções.
//É o closure que possibilita que funções sempre referenciem a variáveis e outros elementos que são visíveis ao seu contexto léxico, enquando elementos que estão encapsulados dentro de outros contextos léxicos não são visíveis para as funções.
//Entender esse conceito é muito importante para evitar bugs quando tentamos referenciar variáveis que estão fora das nossas funções, pois, dependendo o contexto onde a variável se encontra, ela pode ser visível para a função, ou não.

//FUNÇÃO REFERENCIANDO SEU VERDADEIRO CONTEXTO LÉXICO POR CAUSA DO SEU CLOSURE:
const valor = 'Fora'; //Perceba que essa mesma constante é declarada dentro da função funcaoArmazenadora, mas com valor diferente...

function funcaoFora() { //Função fora tem que retornar o valor da variável "valor", mas ela foi escrita do contexto global, por isso seu contexto léxico é somente o contexto global...
    return valor;
}

function funcaoArmazenadora() {
    const valor = 'Dentro'; //A constante "valor", existe só no contexto da função "funcaoArmazenadora"...
    return funcaoFora(); //Mesmo chamando a funcao "funcaoFora" dentro do contexto da função "funcaoArmazenadora" o closure dela não permite que ela referencie a constante valor que está no contexto da função "funcaoArmazenadora", ela vai referenciar somente ao seu contexto léxico...
}

console.log("1) Função chamando a variável valor que é visível para ela por ter sido criada no contexto global...")
console.log(funcaoArmazenadora());



//FUNÇÃO REFERENCIANDO VARIÁVEL Á PARTIR DE UM ESCOPO DE FUNÇÃO, POIS, FOI CRIADA DENTRO DESTE ESCOPO, POR ISSO SEU CLOSURE É DIRENTE:
//Agora ao cenário mudou, temos uma constante criada globalmente....
const valor2 = 'Fora';

function armazenaECria() {
    const valor2 = 'Dentro'; //E temos a mesma constante criada localmente com o seu valor diferente...
    function criadaDentro() { //Agora que a função foi criada dentro de outra função, o seu contexto léxico não é mais o global, passou a ser o contexto existe dentro da função, isso significa que as variáveis geradas dentro da função e fora dela podem ser visíveis
```

```
agora, mas a variável que estiver mais próxima do seu closure (ou seja, da sua memória de contexto léxico) será que vai ser referenci
ada...
        return valor2;
    }
    return criadaDentro();
}

console.log("\n2) Função chama variável local que é a única visível para ela por ter sido criada no contexto local...")
console.log(armazenaECria());
```

RESULTADO NO CONSOLE...

```
[Running] node "c:\Users\User\Documents\JAVASCRIPT\ARQUIVOS_DAS_AULAS\072-Closure.js"
1) Função chamando a variável valor que é visível para ela por ter sido criada no contexto global...
Fora

2) Função chama variável local que é a única visível para ela por ter sido criada no contexto local...
Dentro

[Done] exited with code=0 in 0.131 seconds
```