**What you've learned so far:**

* If an identifier is declared in **global scope**, it's available *everywhere*.
* If an identifier is declared in **function scope**, it's available *in the function* it was declared in (even in functions declared inside the function).
* When trying to access an identifier, the JavaScript Engine will first look in the current function. If it doesn't find anything, it will continue to the next outer function to see if it can find the identifier there. It will keep doing this until it reaches the global scope.
* Global identifiers are a bad idea. They can lead to bad variable names, conflicting variable names, and messy code.

**Directions:**

Write a for (note: not a *function*) loop that prints out the factorial of the number 12:

A **factorial** is calculated by multiplying a number by all the numbers below it. For instance, 3! or "3 factorial" is 3 \* 2 \* 1 = 6

3! = 3 \* 2 \* 1 = 63!=3∗2∗1=6  
4! = 4 \* 3 \* 2 \* 1 = 244!=4∗3∗2∗1=24  
5! = 5 \* 4 \* 3 \* 2 \* 1 = 1205!=5∗4∗3∗2∗1=120

Save your final answer in a variable called solution and print it to the console.

**Your Code:**





/\*

\* Programming Quiz: Factorials (4-7)

\*/

// your code goes here

let solution=1;

for(let i=1; i<=12; i++){

solution \*= i;

}

console.log(solution);

You may have noticed the usage of the keyword let in the solution shown above. In simple words, it's a new way to declare a variable. If you want to know more, read the section below.

## More to Read - Scope and Variable Declaration

At this point, you need to understand that there are ways to declare a variable, by using let, const or var. To understand the difference between the three, first you need to understand the term scope.

### What is a "Scope"?

The scope is defined as a specific portion of the code. There are three types of scope in Javascript

1. Global scope - When a particular variable is visible (can be used) anywhere in the code. Such a variable is generally called as **Global variable**.
2. Function scope - When a particular variable is visible (can be used) within **a** particular function only. Such a variable is generally called as **Local variable**.
3. Block scope - When a particular variable is visible (can be used) within **a** pair of { . . . } only.

Let us see an example of each type of scope:

*/\**

*\* Global scope.*

*\* This variable declared outside of any function is called Global variable.*

*\* Hence, you can use this anywhere in the code*

*\*/*

**var** opinion = "This nanodegree is amazing";

*// Function scope*

**function** **showMessage**() {

*// Local variable, visible within the function `showMessage`*

**var** message = "I am an Udacian!";

*// Block scope*

{

**let** greet = "How are you doing?";

*/\**

*\* We have used the keyword `let` to declare a variable `greet` because variables declared with the `var` keyword can not have Block Scope.*

*\*/*

} *// block scope ends*

console.log( message ); *// OK*

console.log( greet ); *// ERROR.*

*// Variable greet can NOT be used outside the block*

console.log( opinion ); *// OK to use the gobal variable anywhere in the code*

} *// function scope ends*

### Variable Declaration

There are three ways to declare a variable:

1. let - It a new way to declare a variable in any scope - Global, Local, or Block. The value of this variable can be changed or reassigned anytime within its scope.
2. const - It is also a way to declare constants in any scope - Global, Local, or Block. Once you are assigned a value to a const variable, the value of this variable CANNOT be changed or reassigned throughout the code.
3. var - This is the old way of declaring variables in only two scope - Global, or Local. Variables declared with the var keyword can not have Block Scope. The value of this variable can be changed or reassigned anytime within its scope.