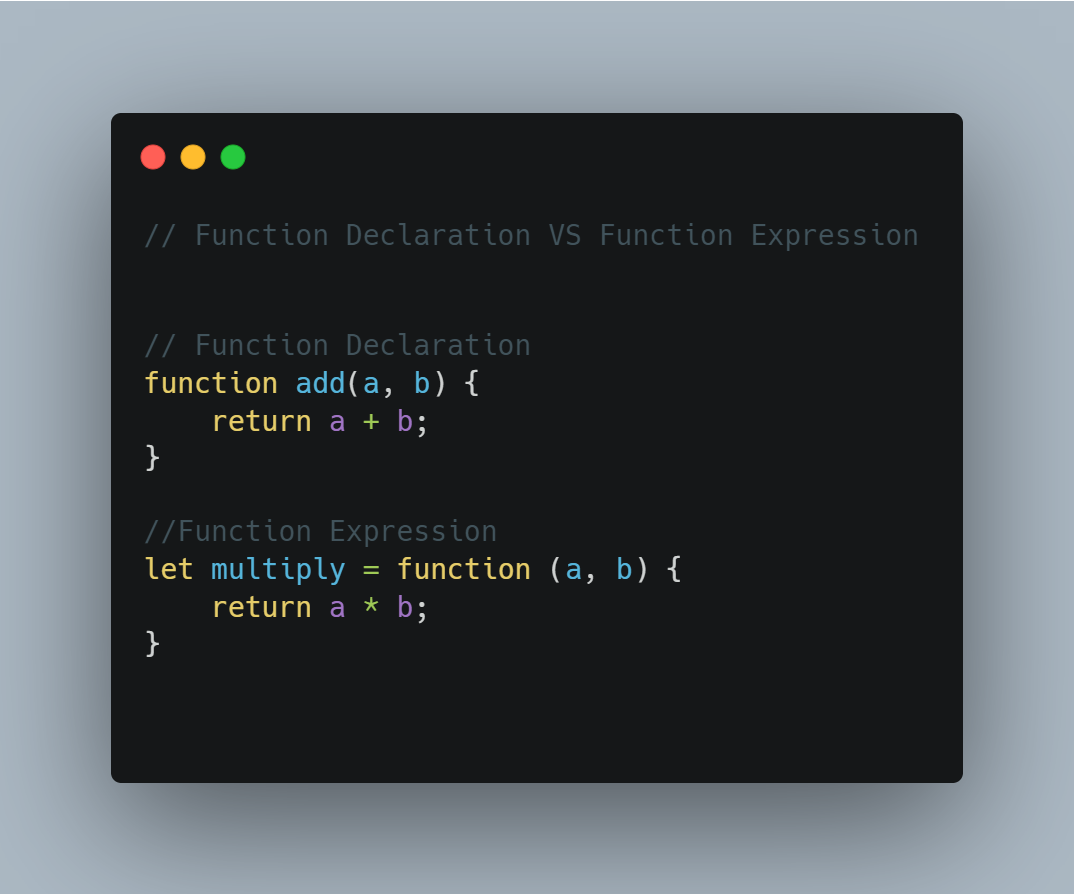
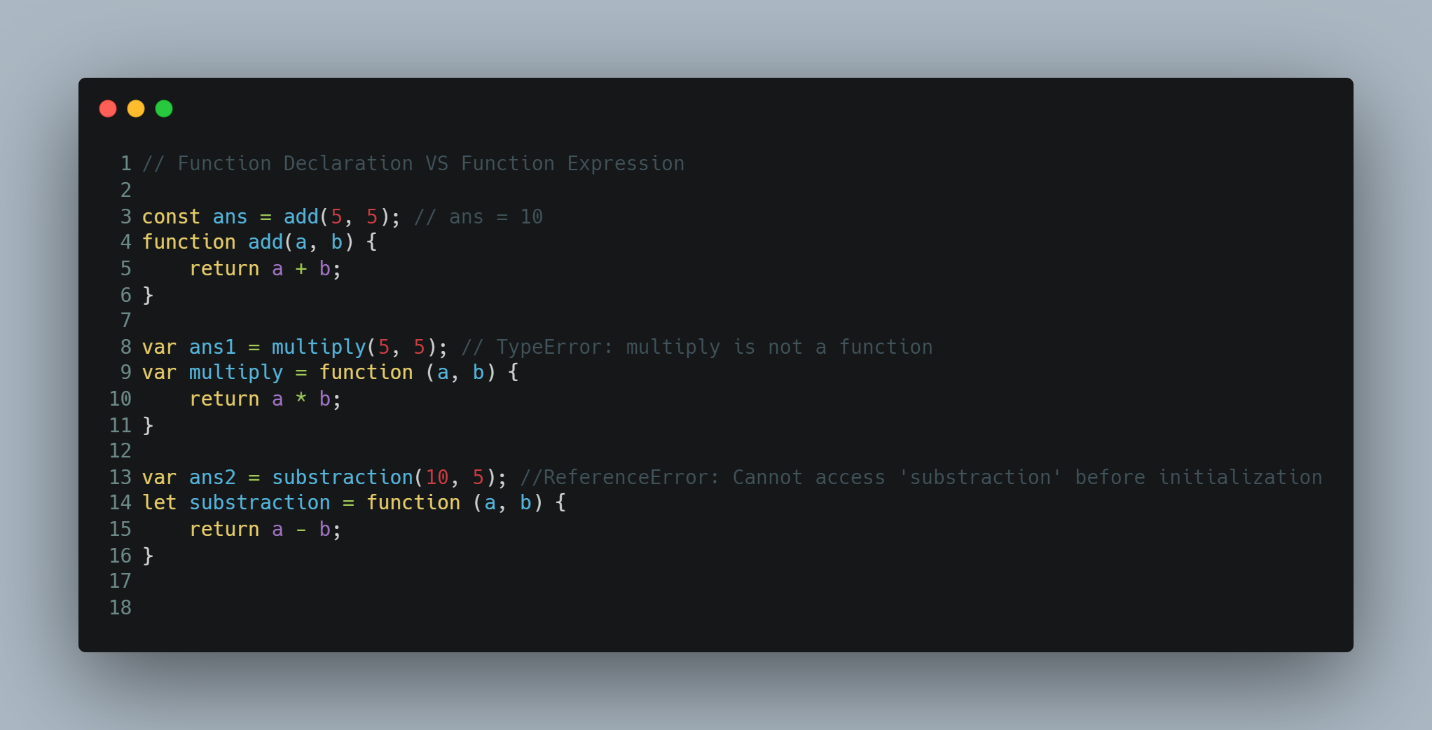
**Function Declaration:** Creating a function with a function keyword followed by a function name is *Function Declaration*. The add function is an example of function declaration in the code snippet. After the function keyword, there is a name of the function which is ‘add’.



**Picture:** Example of function declaration & function expression.

**Function Expression**: Creating a function without a name after the function keyword and assigning it to a named variable declared with [var, let, const] is Function Expression. The ‘multiply’ variable contains a function expression in the above code snippet. This assignment operation happens in the runtime. We can see there is no name after the function keyword here. So, it is an anonymous function assigned to multiply variable declared with let.

In the case of function declaration, we can call the function even before declaration of that function. This is possible because of the hoisting. In the time of memory creation phase of execution context memory gets allocated for the declared function and put the whole function body inside that memory location. So, in the time of code execution phase JavaScript can call that function even before declaration. This scenario is also known as hoisting of function.



In the case of function expression, we can not call the function before initialization of function.  
Because here we assign function into a variable and hoisting of variable is different from hoisting of function. In the memory creation phase JavaScript creates memory location for variable and put ‘undefined’ on that location.