Module 5: Advanced JavaScript: Arrays, Functions, Scopes, and Modules

Practice Activity 5.4 Scope of Variables - Solution

Activity 1: Global Scope

```
const globalVar = "Global variable";

function printGlobal() {
    console.log(globalVar); // Global variable is logged
}
printGlobal();
```

Activity 2: Local Scope

```
function localScope() {
    const localVar = "Local variable";
    console.log(localVar); // Local variable is logged
}

localScope();
console.log(localVar); // ReferenceError: localVar is not defined
```

Activity 3: Block Scope (ES6)

Solution Code using let:

```
if (true) {
    let blockVar = "Block variable";
    console.log(blockVar); // Block variable is logged
}


console.log(blockVar); // ReferenceError: blockVar is not defined
```

Solution Code using var:

```
if (true) {
    var blockVar = "Block variable";
    console.log(blockVar); // Block variable is logged
}
console.log(blockVar); // Block variable is logged
```

Activity 4: Nested Scope

```
function outerFunction() {
    const outerVar = "Outer variable";

    function innerFunction() {
        const innerVar = "Inner variable";

        console.log(outerVar); // Outer variable is logged
        console.log(innerVar); // Inner variable is logged
    }

    innerFunction();

    console.log(outerVar); // Outer variable is logged
    console.log(innerVar); // ReferenceError: innerVar is not defined
}

outerFunction();
```

1. **What happens when you try to access a global variable inside a function?**
   When you try to access a global variable inside a function, the function has access to the global variable. It can read and modify the global variable's value (if declared using var or let and not const).

2. **What is the scope of a variable declared inside a function?**
   A variable declared inside a function has local scope. This means that the variable is accessible only within that function and cannot be accessed outside of it.

3. **How is block scope different from function scope?**
   In JavaScript, variables declared with var have function scope, meaning they are only accessible within the function where they are declared. Variables declared with let and const have block scope, meaning they are accessible only within the block (enclosed by curly braces) where they are declared.

4. **What happens when you declare variables with let inside and outside a block?**
   Variables declared with let outside a block have global or function scope, depending on where they are declared. Inside a block, variables declared with let have block scope, meaning they are only accessible within that block.

5. **Can a variable declared in an inner function be accessed from an outer function?**
   No, a variable declared in an inner function cannot be directly accessed from an outer function. Each function establishes its own scope, and variables declared inside a function are local to that function. However, an inner function has access to variables declared in its outer function due to the concept of closure.