MODULE 05

TYPESCRIPT VARIABLES AND SCOPE

MODULE TOPICS

let and const keywords
Block Level Scope
Hoisting Shadowing
Understanding Boxing and UnBoxing

LET AND CONST KEYWORDS

- TypeScript, like ES2015, support the let and const keywords
  - let declares a variable with block level scope
  - const declares a variable that cannot be changed

BLOCK LEVEL SCOPE

- The JavaScript var keyword declares a variable with function level scope
- Even if the variable is declared in a code block (if, loop, etc), that variable lasts the duration of the function
- The let keyword declares a variable with block level scope, so it will go out of scope when the code block it is declared in ends

HOISTING

When a variable is declared using the var keyword in a code block, the declaration will be hoisted to the top of the function

SHADOWING

Shadowing is when one variable in a nested scope "shadows" a variable with the same name from an outer scope

UNDERSTANDING BOXING AND UNBOXING

- Primitive values don't have properties or methods, so to access things like .length or .toString() an object wrapper around the value is needed
- JavaScript will automatically box (aka wrap) the primitive value to fulfill such accesses

WALKTHRU

TypeScript Variables and Scope

```
function globalFunction1() {
    console.log("globalVar1 in globalFunction() is " + globalVar1);

    var localVar1 = "localVar1";
    console.log("localVar1 in globalFunction() is " + localVar1);

    let localLet1 = "localLet1";
    console.log("localLet1 in globalFunction() is " + localLet1);

    // mysteryVar = "mysteryVar";
}
```

ANY QUESTIONS?