

## ANSWER 2-

When you use the `let` or `const` keywords to declare a variable in JavaScript, the TDZ is created for that variable. The TDZ starts at the beginning of the scope (such as a function or block) where the variable is declared and continues until the actual declaration statement is encountered in the code.

During the TDZ, if you try to access the variable, either by reading its value or assigning a new value to it, JavaScript will throw a runtime error. This is because the variable exists in the TDZ, but it is not yet initialized or accessible. The error is meant to prevent you from accidentally using a variable before it has been properly declared and initialized.

It's important to note that the TDZ only applies to variables declared with `let` and `const`, not with `var`. Variables declared with `var` are hoisted to the top of their scope and can be accessed with an undefined value before their declaration. To avoid TDZ errors, make sure to always declare your variables before using them within the same scope. By doing so, you ensure that the variables are properly initialized and accessible when needed.

In summary, the TDZ is a period in JavaScript where variables declared with `let` or `const` exist but cannot be accessed or assigned any value until they are properly declared in the code. It helps catch potential issues with variable usage and promotes better code organization and readability.