Life Cycle of Variables

Life Cycle of Variables in the TDZ: `let`, `const`, and `var`

In JavaScript, the life cycle of variables varies depending on their type and presence in the Temporal Dead Zone (TDZ is the period between the creation and declaration of a `let` or `const` variable).

A brief overview of how `let`, `const`, and `var` declarations go through different stages, including TDZ, initialization, and usability is described below:

1. `let` and `const`

- **Creation:** Variables are created during the creation phase but unlike var it is not initialized with “undefined” rather it remain uninitialized.
- **TDZ (Temporal Dead Zone):** Variables enter the TDZ until they are formally declared.
- **TDZ Reference Error:** Accessing or assigning values during the TDZ results in a `ReferenceError`.
- **Declaration:** Variables are declared, initializing them with the assigned value.
- **Usable:** Once declared, variables can be accessed and assigned new values.

2. `var`

- **Hoisting:** `var` variables are hoisted during the creation phase to the top of their scope.
- **Initialization:** `var` variables are initialized with the value `undefined` during hoisting.

- **Usable:** `var` variables can be accessed and assigned values throughout their scope, even before their actual declaration in the code.

Reference: Link