9 video

Let & const in JS Temporal Dead zone

* let and const declaration are hoisted. but they hoisted very differently.

They are in Temporal dead zone.

→ Console.log(b); → undefined
var b = 100;

Memory is allocated to variable & function. even before a single line of code executed

→ console.log(a); o/p
let a = 10; Uncaught: ReferenceError: cannot access 'a' before initialization
var b = 100;

- Throwing Error as let and const are hoisted in a separate execution context with value undefine (Not in global EC and we are accessing in GEC (by default) Hence - throwing an Error).

- let and const variable can't be accessed until they are assigned.

Temporal dead zone The time period between allocating memory and assigning value to let & const.

let & const remain in Temporal dead zone until they are assigned a value. & after that only they can be accessed.

*
```
console.log(x);
console.log(b);
let b = 10;
```

O/P

UnCaught ReferenceError: x is not defined
UnCaught ReferenceError: Cannot access 'a' before initialization

```
let x = 10
console.log(x);
```

O/P : 10

→ let and const are not attached to window(th) object. [i.e window.b will give undefined]

→ one can't redefine let

eg 1
```
console.log("Rahul");
let a = 10;
let a = 100;
```

O/P
- SyntaxError: Even a single of code will not executed.
- Not possible in same scope.

eg 2
```
console.log("UK");
var a = 10;
var a = 100;
```
→ O/P UK perfectly fine.

let can be initialized anywhere in program. However const must be initialized at the time of declaration only. And their value can't be change.

For
```
const a;
```
→ SyntaxError: Missing initialization in const declaration
Invalid JS code.

```
const a = 10;
a = 100;
```
→ TypeError: Assignment to constant variable.

var < let < const

Restriction [preference of use is reverse order]

~~Restriction~~ Errors

- Reference
  - variable is not defined
  - ① may be temporal dead zone
  - ② Not present in scope at all
- Syntax
  - Invalid JS code
  - eg const a;
- Type error
  - assigning again value to const variable.

- To avoid Temporal dead zone all declaration & intialization at the top of the JS code
- We shrinked Temporal zone to zero.