**Shadowing**: If there are two separate variables with the same name, in separate (but nested) scopes.

Ex:

let x = 3;

if(x === 3) {

let x = 4;

console.log(x); // 4

}

console.log(x); // 3

Let’s note two key observations:

Variable shadowing is completely valid JavaScript.

When within the scope of the local x (the if block), there is no way to access the global x.

**Temporal Dead Zone**

If we try to access a **let / const** variable after it has been declared (at the point where it was hoisted to) and before it has been assigned a value, we get an error. This time between the variable being declared and the first time it is assigned a value is called the temporal dead zone (the TDZ).

**Key observation about the TDZ:**

If we attempt to access a let / const variable after it is defined but before it is assigned a value, we get an error.

This differs from var variables, which would return undefined in the same situation.

Ex:

function f() {

let x;

// TDZ begins here

console.log(x); // ReferenceError

// TDZ ends here

x = 3;

console.log(x);

}