LET

①
```
let a = 10;
console.log(a);
var b = 100;
```

* memory is assigned to b in the var declaration and the variable b is attached with global object.

but in case of let and const (hoisting) are also allocated memory but in a different memory space.

& we can't access this memory space unless we have put a value in them.

Temporal dead zone — It is the time since when this let variable was hoisted and till it is initialized some value.

- We can't access values in temporal dead zone. They can only be initialized after value is assigned.

②
```
this.b // window.b
>> 100
this.a // window.a
>> undefined.
```

```
Console.log("Hello dbh")
let a = 10;
let a = 100;
```
>> identifier a has already been declared. // Syntax error.

③
```
let a = 100        | let a = 100
let a = 10         | var a = 10
```
>> Syntax Error:
We can't reinitialize let variable but we can do it with var.

CONST

(very much similar to let but more strict)

④
```
Const a;
a=100;
console.log(a);
```

>> Uncaught Syntax error: Missing initializer in const declaration

// Needs to be initialized in the same line.

⑤
```
const a=10;
a=100;
console.log(a);
```

>> Uncaught TypeError: Assignment to a constant variable.

⑥ Type error – It is of type const so it needs to be initialized as well as declared together.

Syntax error – If we declare a variable and don't initialize it. (missing syntax)

Reference Error –
```
console.log(a);
let a=1900;
```

We try to access a even before we initialize. In this stage a is in temporal dead zone.

- Another if we try to access a random variable which is not in the memory.

7 What should be use generally?

① We should use const first place. Wherever we can use const. We should use it. Whenever we want to put in some value which is not changed later and we don't have to assign anything else for some variable.

Best one. Won't run into unexpected errors

② If not const, try to use let wherever possible. Let has a temporal dead zone and we won't run into unexpected errors like undefined.

③ Keep var aside. Don't use it in day to day coding Use it very consciously.

Always prefer to use const and let.