



## Blocks :—

- Blocks are defined by "{ }".
- It is also known as compound statement.
- Block is used to ~~define~~ combine multiple JS statements into one group.

e.g. {

```
var a = 10;  
console.log(a);  
}
```

do

Q. Why <sup>to</sup> we need group multiple statements together?  
need to in a block

We group these statements together so that we can use multiple statements in a place where JS expects only one statement.

Q. What is Block Scope?

Block scope means what all variables & functions we can access inside this block is known as block scope.



e.g.

{

var a = 10;

let b = 20;

const c = 30;

}

} These two variables are inside block scope.

Note: —

①

We cannot access "let" & "const" outside the scope block.That's why let & const are block scoped.

②

"var" is declared in global scope. So it can be used outside the block.

Q.

What is shadowing in JS?

If we have same named variable outside the block, so the variable tries to shadow inside variable.

e.g.

```
var a = 100;
{
  var a = 10;
  let b = 20;
  const c = 30;
  console.log(a);
  console.log(b);
  console.log(c);
} console.log(a);
```

→ o/p: — 10

20

30

10

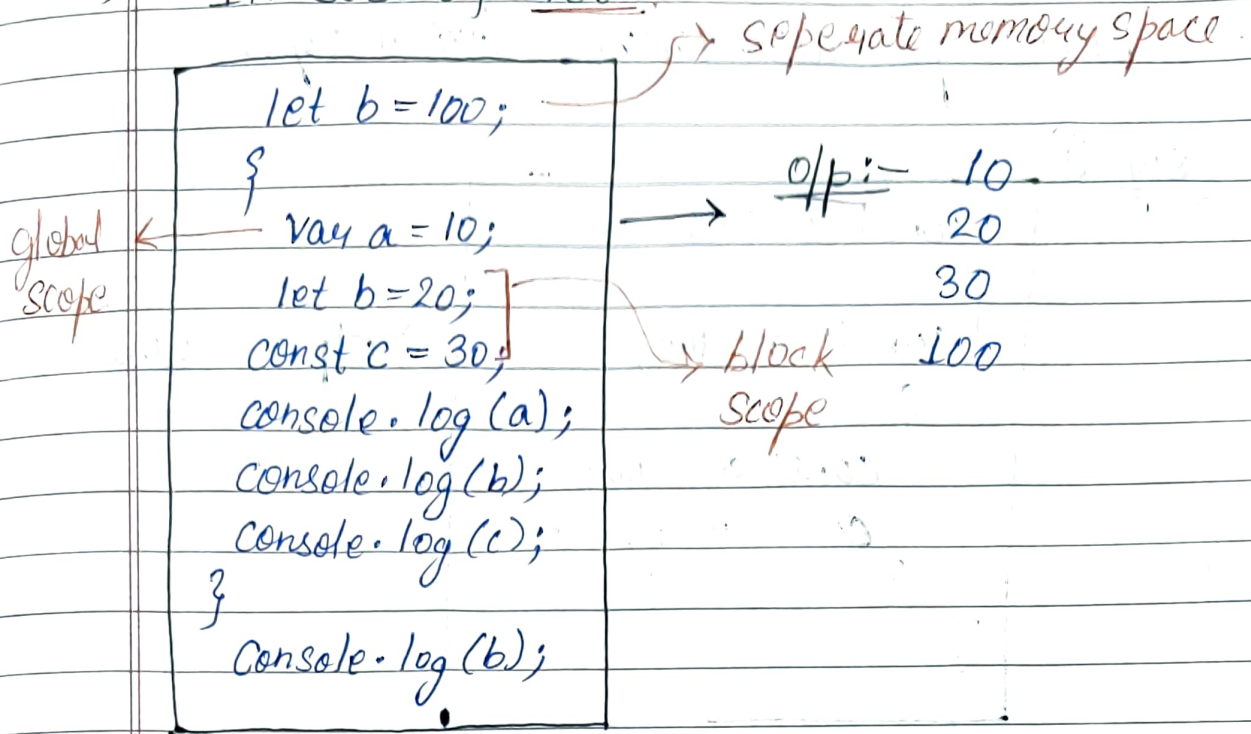
→ outside the block.

No

Q

e.g.

→ In case of "let"



→ Similar in case of "const"

Note:- Shadowing works in the same way in "function" as it works in block.

Q. What is illegal shadowing?

We can not "shadow" let using ~~var~~ "var". It is illegal shadowing.

eg.

```

let a = 20;
{
  var a = 20;
}
        
```

→ Error

```

var a = 20;
{
  let a = 20;
}
        
```

→ It's true in this shadowing -g take place.

→ Const is also same.





→ "Block Scope" also follows "lexical scope".

e.g. ①

```
const a = 20;
{
  const a = 100;
  {
    const a = 200;
    console.log(a);
  }
}
```

→ o/p: - 200

②

```
const a = 20;
{
  const a = 100;
  {
    const a = 200;
  }
  console.log(a);
}
```

→ o/p: - 100

③

```
const a = 20;
{
  const a = 100;
  {
    const a = 200;
  }
  console.log(a);
}
```

→ o/p: - 20