

Advanced js

Execution context :

3properties of EC:

- 1.var obj
- 2.scope chain
- 3.this.var

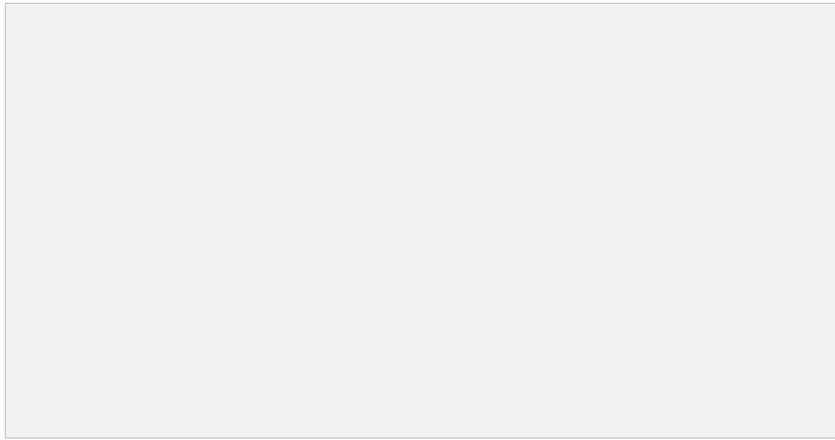
EC 2 phases :

1.creation phase :

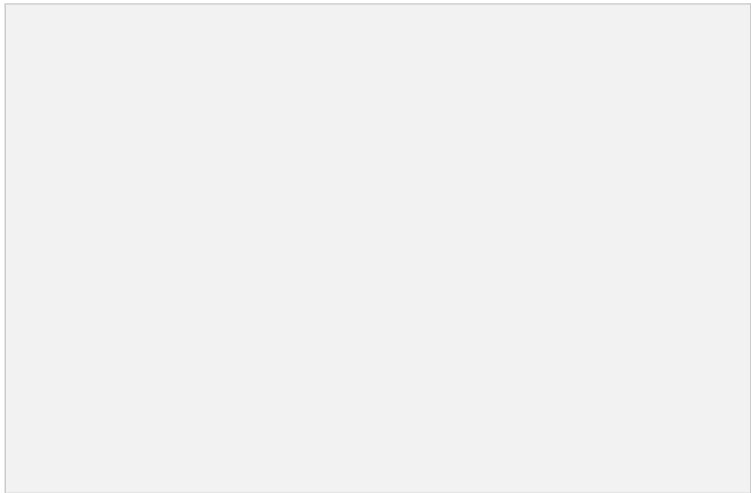
var obj are created, scope chains are created, this var determine

First check function dec and for each fun property is created in the var object .which is pointing to that function.

For each var property is crated in var obj which is set to undefined.



Before execution phase starts function and var available to us(hoisting)



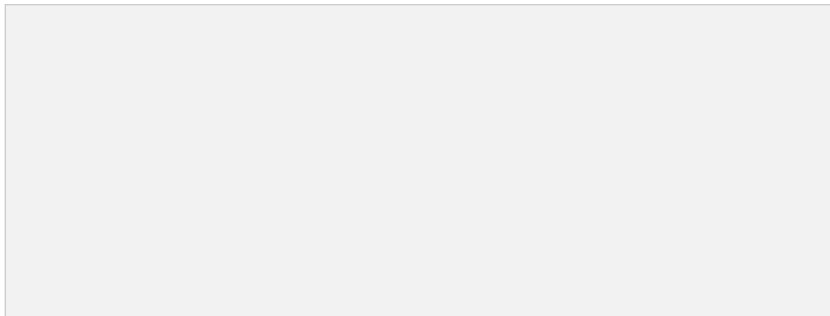
This is called hoisting in js

In Create phase all fun dec and var move to the top to their scope called hoisting.

With function expression hoisting is not working(it assumes as var)

Hoisting is work only with var (not with let or const)

Hoisting is not working here coz var is define in first with undefined value



Scope chain :

2.Execution phase :

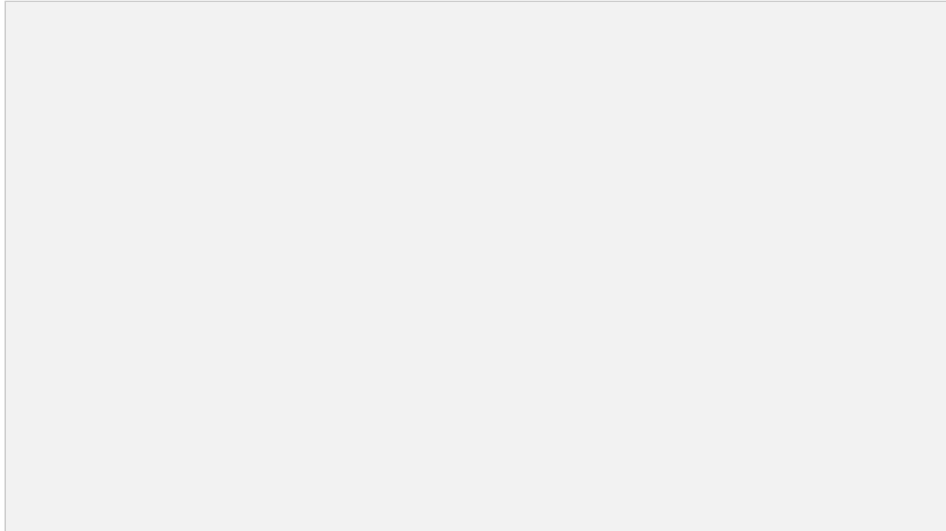
current execution line by line

Use Strict :

It must write in your program.

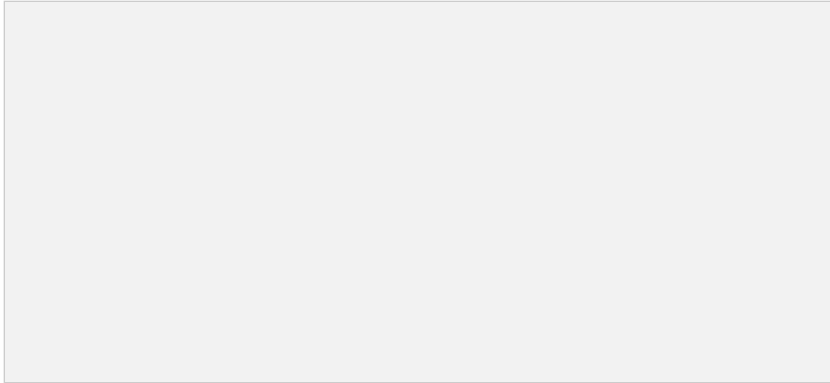
It shows warnings and errors.

O/p is displayed but difficult to understand to browser.

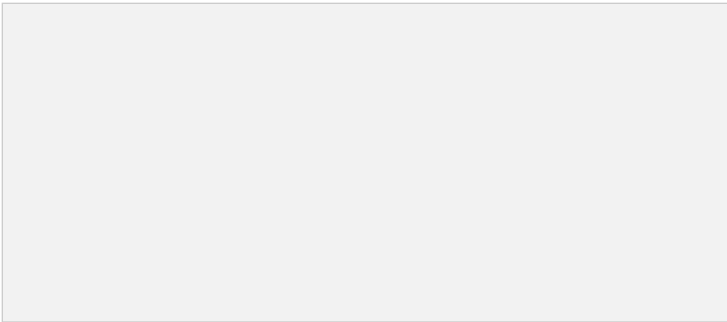


This keyword:

- This keyword is refer to current class of object.
- It refers to global object(ie. Window object)



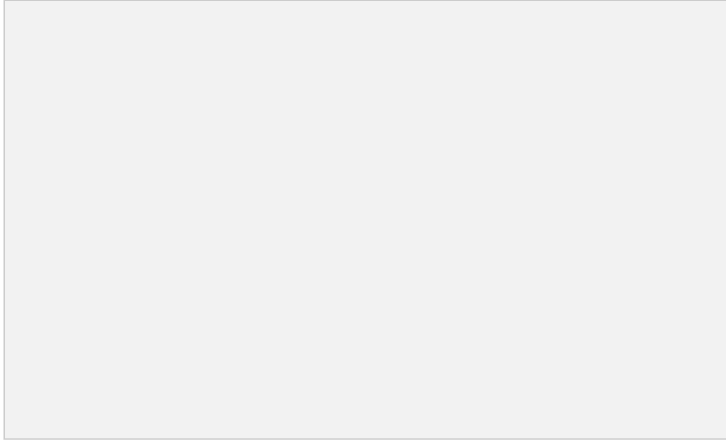
- In regular function , this refers to the global obj:



4

window.object

- In method / object this refer to owner obj:



4

Vinod thapa
Owner object

- In strict mode this is undefined

❖ Type is js:

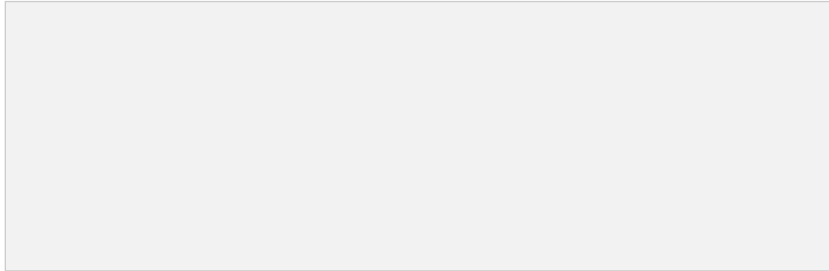
- Primitive datatype: which have values
- Non-primitive datatype: which have references
- Type of null is object

When u will check `console.log(typeof(function({})))`:

o/p : function

But , In js everything is object

It showing it is function but it is object



o/p: allright i am an obj too

How to prove type is array ?

```
const arr=[1,2];
```

```
console.log(Array.isArray(arr))
```

Pass by value or Pass by reference:

Pass by value :

Both var assign separate memory

It works with only primitive datatype.

Program:

```
Let a=5;
```

```
Let b=a+5;
```

```
  console.log(a);
```

```
console.log(b);
```

o/p :

5

10