

JavaScri<sup>c</sup>t

The End

- Higher Order Function :-

Higher Order function are the functions which accept A function In a parameter or return a function or Both.

for eg :-

```
function abcd(val){  
      
    abcd(function (){})  
}
```

Phele toh aap ek function bana lo or aap usse call karne vali jisse andr function pass kardo. toh aapka abcd function aap ka higher Order function ho jaata hai.

Aisa function jisse andr function pass karne ke liye aap accept kar sakte ho. i.e (val). toh abcd() aap ka higher order function ho jaata hai.

Dusra tadika kya hai? toh chaalo jante hai!

Dusra hadha yeh hai ki, Hoy aesa function jo  
ki aapne andar se return karde Hoy function.

for eg:-

function abcd() {

return function () {}

}

abcd()

Hya aese code likhey hai? Yes, Hab jab hum  
Closures banaate hai.

aisa function jo accept karne ke liye aur function ya  
fir wo return karne ke liye aur function

• Constructor function :-

function SaanchaofBiscuit() {} } Hya yeh Constructor function hai  
} Nahin, yeh to he normal function hai.

Normal function jismein this ka istemal ho and  
aap function ko call karte waqt new keyword  
ka use karein

Thoda sa this pad lete hai, age detail mai pade ge

function mai jab tum this likh te ho toh byDefault  
function this matlab hota hai Window.

for eg:-

```
function SaanchaofBiscuit()
    this.width = 12;
    this.height = 22;
    this.color = "brown";
    this.taste = "Sugary";
```

}

accha Saancha  
Uske anda this  
hai. acap usko nahi  
use kar sakte  
(Kha-salte) toh  
Mesa khaye ge?

Var biscuit1 = new SaanchaofBiscuit()

Var biscuit2 = new SaanchaofBiscuit()

Var biscuit3 = new SaanchaofBiscuit()

Result:-

biscuit1

SaanchaofBiscuit { width: 12, height: 22, color: 'brown',  
taste: 'Sugary' }

biscuit2

SaanchaofBiscuit { width: 12, height: 22, color: 'brown',  
taste: 'Sugary' }

Toh yaha baat ati hai aap Constructor function  
Hyu use Hante ho?

jab aapke paas aisa koi bhi maufha ho he aapko  
ek jaisi properties waale abhut sare elements  
banaane hai us waqt aap Constructor function  
use hao sakte ho.

For eg:-

function CircularButtonBanao (Color) {

this.radius = 2;

this.color = color;

this.icon = false;

this.pressable = true;

}

Var redButton = new CircularButtonBanao ("red")

Var GreenButton = new CircularButtonBanao ("green")

Result :- redButton

CircularButtonBanao { radius: 2, color: 'red',  
icon: false, pressable: true }

GreenButton

CircularButtonBanao { radius: 2, color: 'green',  
icon: false, pressable: true }.

## First Class Function :-

We already discuss in "Javascript"

The master art".

## New Keyword :-

jab bhi new lagta hai humesha ek blank object apne maan mein bana lo.

{

// Blank object.

}

For eg :-

function abcd() {

this.age = 12;

y

new abcd()

{

age : 12;

}

z

New Keyword always create a blank object for the constructor function which is getting called just after New Keyword.

iife :-

iife  $\Rightarrow$  immediately invoked function expression.

जिनी भी Jquery, GSAP, Angularjs, library हैं जब भी तुम इन्हें use करते हो तुम एक variable मिला हो।  
Like :- Jquery, \$(),  $\Rightarrow$  शावल है। ऐसे मिल जाते हैं?

उसका naam है iife (immediately invoked function expression).

iife है function को तुमने chalane की Raison is tareek se कि हम log को private variable bana पाये।

Toh aap Const bana lijiye  $\Rightarrow$  baat shif change करने की nahi है। aap use direct access bhi nahi kar sakte.

For eg :-

Function abcd() {} } Yeh hua normal function.

} Small bracket

This is iife. { Function () {} } Phir is par hi chiz ko small brackets se cover kar lijiye।

} ()  $\downarrow$  call  
small bracket

or is par hi chiz ko call kar dijiye dijiye।

(function() {

    var a=12; }  
})()

} Variable

iife

Result:-

a is not defined

jab padane time pe or ajo bhi code likhte hai

We never want ke log aptke much variables

ko access kar paaye or agar kaane hai

access ya change hum unhey alag se methods

dege use change update kaane ke liye or access

karne ke liye.

Aisa kya? voh aapke code ko jyada safe or jyada "private" banata deta hai.

Example:-

1. Var ans=function() {

2.     Var privateval=12;

3.     return {

4.         getters: function() {

5.             Console.log(privateval);

6.         },

7.         },

8.         },

9.         Getters: function() {

10.         Console.

Yeh fnc chala  
hai. chalte

hai ek Var  
bara hai

fir return  
hui hai

object kaha  
gaya hoga

(ans) mai.

- 11.  $\text{privateval} = \text{val};$
- 12. 3
- 13. 3
- 14. 3) (

} ans mai object hai. obj  
 } He and & kya hai getters  
 } Getters ek value mag raha  
 } hai jo value send hogi Get he

iife K and & aap Ho variable banate ho voh auto  
 matic private ho jate hai Voh aap access nahi  
 Kar sakhte jii hag, iske and & variable banu ga  
 voh access nahi Kar sakhte jii hai.

Yeh variable uske and & use Kar sakte ho yani ki  
 iife ke and &. i.e line no (02 to 13 )

Result:-

// for access variable.

ans. getters() => 12.

// for Get a value (change).

ans. getters(24).

// again see the value.

ans. getters() =>

24.

Toh yeh hai Nahein iife hi.

## Prototype :-

Prototype is basically the collection of build in help of function and help of methods jo apte huye use karne ke liye available hain.

for eg :-

Var obj = {

name: "haash"

}

Result :- obj

Open  $\leftarrow \rightarrow$  { name: 'haash' } } We create this but,  
name: "haash" }

$\rightarrow$  [ [ prototype ] ] : object. } We don't create this

JavaScript by default adds a property called [ [ prototype ] ] to every object. jitne bhi objects tum create karo ge [ [ prototype ] ] yeh banta hai.

When you open [ [ prototype ] ] it contains many helper properties for eg :-

Constructor, hasOwnProperty, isPrototypeOf, toString,  
Valueof and so on.

obj Ke andar Kya Kya hai?  $\Rightarrow$  Name toh mujhe check  
Kas na obj Ke andar hai ya nahi.

obj. hasOwnProperty('name') } Yeh property hum ne toh  
true. } nahi banay hai toh  
↓ } Maha se aata hai.  
Prototype Ki Property That's what prototype  
is all about.

## Prototype inheritance :-

Example :-

Var human = {

CanFly: false,

CanTalk: true,

canWalk: true,

haveemotions: true,

hasFourLegs: false

3

Var ShehryansStudent = {

CanMakeAmazingWebsite: true,

CanMakeAwesomeAnimations: true,

CanMakeWorldClassAwardWebsites: true,

3

Shediyans Student. -- proto-- = human;

II This is the way to inheritance.

Human → Shediyans Student ka part nahi hai but  
Voh borrowed hai toh voh uska part bhi hai  
Voh uske prototype ke through uagaya.  
This is called Prototypal inheritance.

This call apply bind :-

jab bhi kuchh iske dahe ho  
check karo kya usmein haai bhi ho function, object  
ya fir scope bana hai ky (toh fir tum Global  
scope mai nahi ho aap local scope mai ho). andar hai.

jab bhi ho cheej ky brackets

He andar nahi hoti to hum use global scope hante  
hai

for eg :-

Var a=12; // Global Scope hai.

For eg :-

function abcd() {

Var a=12; // abcd() He local Scope mai  
hai.

Global Scope mai this Ki value Hya hai?

Console.log(this);

Result:- ► Window. // Gives Window

function Scope mai this Ki value Hya hai?

Function abcd() {

    Console.log(this);

}

Result:- abcd()

► Window. // Gives Window

Method Scope mai this Ki value Hya hai?

elt function jo object He andha hog use method  
Mahte hai.

Var obj = {

    baatHao : function () {

        Console.log(this);

}

    obj.baatHao();

Result:- { baatHao: f } // Gives object (obj).

↓

Function.

## • Event listeners :-

Note :- To understand event listeners you must know DOM. Toh jao or paado DOM.

for eg :-

```
{HTML}
<head>
<title> Event listeners mai this ki value kya
        hai </title>
</head>
<body>
<button> click here to check what is this
        value in event listeners </button>
</body>
</html>
```

Script.js =>

```
Var button = document.querySelector("button");
button.addEventListener("click", function () {
    console.log(this);
```

)

Result :-

```
<button> click here to check what is this
        value in event listeners </button>
```

Event listeners mai this Ho value jo bhi addEventlistener phele jo bhi littta hoga (button) toh this Ho se defter Rahega humesha button Ho.

### Call :-

agar tumhaare paas Ho function hai and Ho object hai and tumhe function chalana hai by default jo this Ho value window hai use window naa zakh Raat point Raawana hai/isi object Ho tadaf.

for eg :-

function abcd() {

    console.log(this);

}

abcd()

Result :- ► Window.

ab hum chate hai Ho this Ho value window na ho lie this Ho value obj ho jo abhi hum banaye ge.

For Eg :-

function abcd() {

    console.log(this);

}

Var obj = { age: 24 };

abcd.call(obj);

Result :-

► {age:24}

• Apply :-

for eg :-

function abcd (val1, val2, val3) { }

console.log (this, val1, val2, val3);

y

Var obj = {age:24}

abcd.apply (obj, [1, 2, 3])

Result :- ► {age:24} 1 2 3

Apply aap ko kہتا ہے کہ humesha apko two arguments use karne ہے phelatoh jo this ke liye or dusra ho jata ہے vo values jo apko parameters mai pass haware ہے or vo ek se jyada ho sakti ہے toh aap adday he dup mai likhiye i.e [ ].

Note :- Yeh use Hab hota ہے ?

Jab bhi apko ek function chalana ہے jis mai this ki value kuchh or ہے (window) par aap chahete ہے ki this ki value kuchh or object raha tab .call, .apply, .bind use mai ate ہے.

bind :- object function

Yeh bind jyada use hota hai React mai.

For eg :-

```
function abcd() {  
  console.log(this);  
}
```

Var obj = {age: 24};

abcd.bind(obj);

Result :- // answer nahi aya

Answer Kyu nahi aya? Hyuki bind kabhi bhi chalata nahi hai.

Yeh Hya Maata hai bind Maata hai. Yeh abcd ko obj se bind karega or ek naya function dedega aapko dega binded function.

Abhi dikh to nahi zaha, aapko use save karne padega.

```
Eg :- function abcd() {  
  console.log(this);  
}
```

Var obj = {age: 24};

Var bindedFnc = abcd.bind(obj);  
bindedFnc();

Result :-

```
bindedfnc // check what is inside bindedfnc.  
f abcd() {  
    console.log(this); } } eli function hai.
```

Result :- ➔ 24y // bindedfnc(); Call.

Kabhi Kabhi asa hota hai tuant nahi chalana hai. Future mai chalana hai. Jab aap React paooge, toh aapko eventlistened banane hote hai. eventlistened mai function tuant nahi chalane hai. Jab event "click" ho tab chalana hai, tab use hota hai yeh.

### Pure & impure functions

For eg :-

```
function abcd(val) {  
    return Math.random() * val;  
}  
  
Var ans1 = abcd(2);  
Var ans2 = abcd(2);
```

Same value pass  
kaadne par humesha  
agali value  
different mil dahi  
hai.

Result :- ans1

|            |            |
|------------|------------|
| Same input | 1.926503   |
| But        | ans2       |
| Different  | 0.37059718 |
| Output     |            |

Toh yeh Pure Function  
nahi huy. Yeh ek  
impure function hai.

esa function jo same input dena par same output de karbaaz or voh Global Variable ki value change na kare use hum karte hai Pure function.

For eg:-

```
// var abcdef = 12; //Global variable  
function abcd(a,b){  
    return a*b; // abcdef = 24; //update variable.  
}  
  
var ans1 = abcd(2,1);  
var ans2 = abcd(3,1);  
var ans3 = abcd(2,1);
```

Pure function.

Results - ans1

2

ans2

2

ans3

2

Same Input  
&  
Same Output.

So, This is called Pure Function.

Abhi Global mai ek variable banate hai or use function mai value ko update karte hai toh voh Pure Function (X) nahi hai. Voh Impure Function hai.

See an example (↑) i.e ~~written~~ in Comments. (11).

written

II JavaScript The End II.