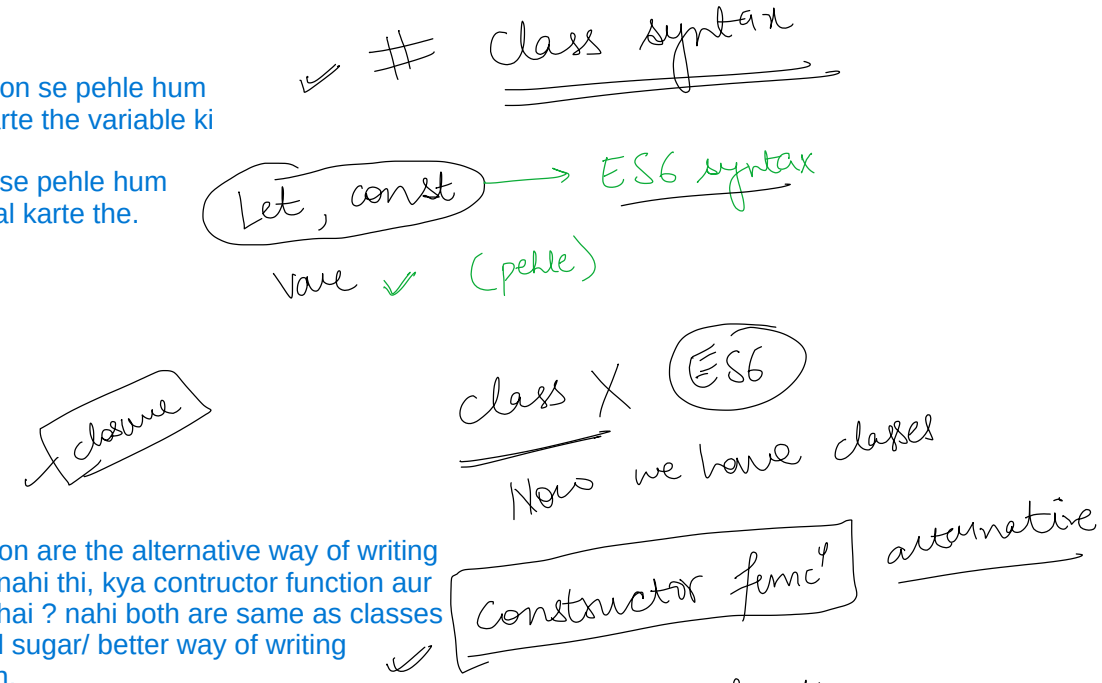
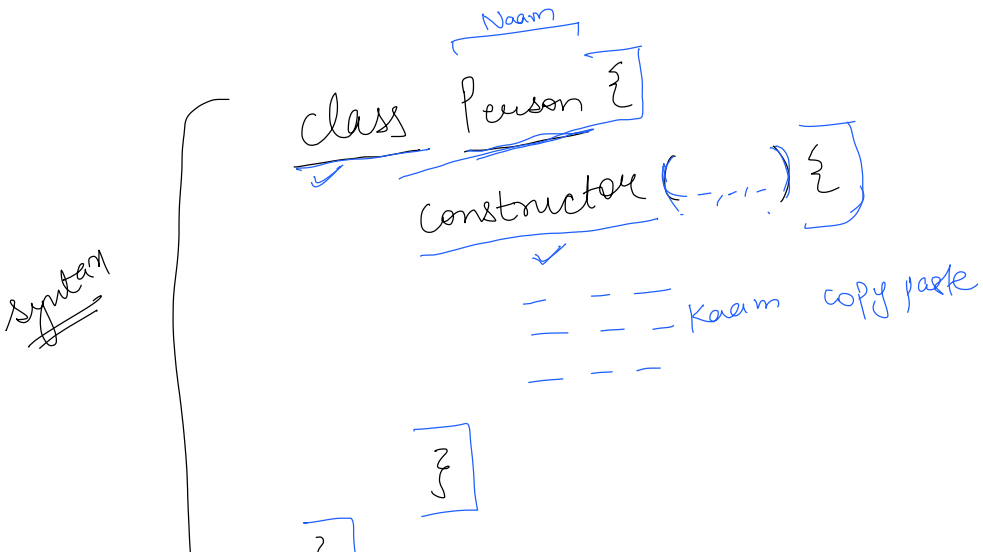
Day26(class, super ,constructor , this , call , bind , apply)

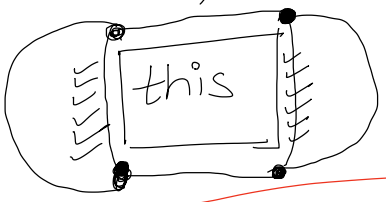
Wednesday, 12 April 2023 7:42 PM

1. Jaise ES6 version se pehle hum var ka istemaal karte the variable ki tarah.
2. Aise hi classes se pehle hum closure ka istemaal karte the.



Constructor function are the alternative way of writing classes jab class nahi thi, kya contructor function aur class m koi anter hai ? nahi both are same as classes are the syntactical sugar/ better way of writing contructor function.





this kisko point kar raha hai ye decide hoga agar function calling / invocation se.

Kaise ho rahi hai

5 types

1. regular func' invocation

```
function fun() {
}
fun();
```

Whenever JS code is executed a GEC is created along with that a global object is also created and in case of browser this global object is window object. so this refers to this global or window object in 1st case.

2. method invocation

```
let obj = {
    fn : function() {
        clg --
    }
}
obj.fn();
```

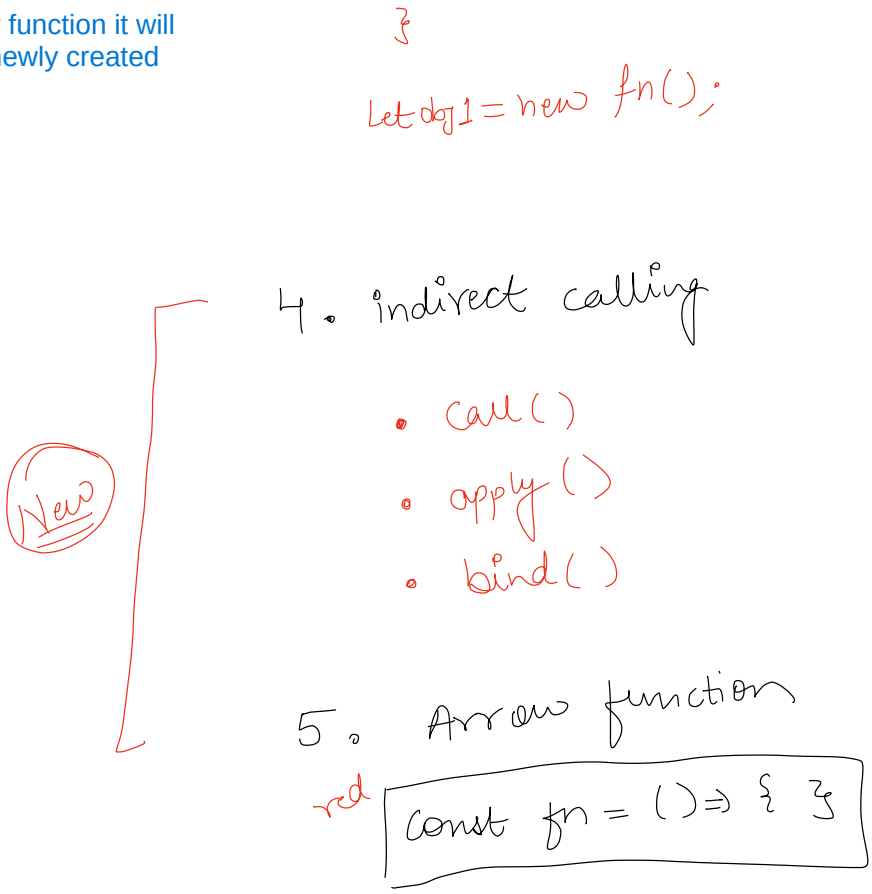
In this case this is pointing to the object in which it is invoke.

Note : this hameesa ye dekhega ki mai kis ke ander call / access kiya ja raha hu aur woh function kaise run kiya ja raha hai agar regular call points to window object and call with the help of object then points to object.

3. Constructor Invocation

```
function fn() {
```

In case of constructor function it will always points to the newly created object.




```
let obj = {
    a : 20,
    fn: function{
        log(this);
    }
}

let obj2 = {
    a : 20,

}
```

In case of arrow function this will always point to the window object.

```
let obj = {
    a : 20,
    fn: () => {
        console.log(this); // points to window object
    }
}
```

obj.fn.call(obj2) => it indicate that ki ab humara this points karega first argument means this points karega obj2 ko jiski help se hum obj2 ke ander jitne bhi variable aur function hai unhe acces kar payenga.

call(), bind(), apply() :

https://www.interviewbit.com/javascript-interview-questions/#call-apply-and-bind-methods
https://www.geeksforgeeks.org/explain-call-apply-and-bind-methods-in-javascript/

Arrow Function :

https://dev.to/hyemiie/understanding-arrow-functions-syntax-features-and-use-cases-1a6m