Promise in javascript

- The promise object represent the eventual completion of an asynchronous operation and its resulting value

- promises occurs in three steps
  (i) pending
  (ii) Fulfilled
  (iii) Rejected

```
const promise1 = new promise()
                    ↓
        new promise (function(resolve, reject)
        {

        setTimeout (function()
            {
        console.log("promise1"); resolve()
        }, 1000)
        })

promise.then(function() {
        console.log('consume');
        })
```

output → Promise1

Add

output
Promise1
Consume

Other method

```
Const promise = new promise(function
                (resolve, reject){
    Settimeout(function(){
resolve({ username: "Ankit",
          password: "6325" })
    }, (000)

})

promise.then(function(user){
        console.log(user);
    });
```

output

username: "Ankit password: "632

Fetch API

- The fetch API provides an interface for fetching resources (including across the network). It is a more powerful and flexible replacement for XMLHttpRequest.
- fetch API provides a method to send and receive resources.

using promises

```
function getjokes()
    {
    fetch(URL).then(ANKIT) =>
        {
        Returns ANKIT.json()
    }).then((data) => {

        Console.log(data);
    })
}

Btn.addeventlistener("click", getjokes)
```

Output → New joke.

OOPs

(Object oriented programming)

Classes

- classes are a way to create object templates.

creation of class

Syntax:

```
class myclass {
    constructor() {}
    method() {}
}
```

```
let myobject = new myclass()
```

eg:

```
class car {
    start() {
        console.log("start");
    }
    drive() {
        console.log("drive");
    }
    stop() {
        console.log("drive");
    }
}
```

```
let bolero = new car()
output
Car()
  > Bolero
    prototype : object

    constructor : class car
    drive : f drive()
    start : f start()
    stop : f stop()
```

We can define various other parameter in this

eg: for Car
several parameter are
Speed, brands etc...

Inheritance

- Inheritance are the property that passes characters from parents to child.

- Some of the characters are property and method

Super keyword

- Superword is used to call the constructor of its parents class to access the parent's properties and methods