

ES5 ES6

```
//Block & Iiffs
//ES5
(function(){
    var a = 1;

})();

//ES6 block
{
    const a = 1;
    console.log(a);
}

//Destructuring
//Es5
var john = ['john', 23];
var name = john[0];
var age = john[1];

//ES6
const [name, age] = ['john', 34];
console.log(name);
console.log(age);

//object er khatre
const obj = {
    firstName: 'john',
    lastName: 'Smit'
};
const {firstName, lastName} = obj;
console.log(firstName);
console.log(lastName);

//or
const {firstName: a, lastName: b} = obj;

// Array
const boxes = document.querySelectorAll('.box');
// ES5
var boxes= Array.prototype.slice.call(boxes);
//ES6
Array.from(boxes);

//loop
//ES5 for
for()
//ES6 for of
for(const _ of _)

//Spread operator
//ES5 .apply array receive kore
//ES6 (...ages);

const familySmit = ['John', 'jane'];
const familyMiller = ['Mary', 'Bob'];
const bigFamily = [...familySmit, ...familyMiller];
```

```

console.log(bigFamily);

// or
const bigFamily = [...familySmit, 'Lily', ...familyMiller];
console.log(bigFamily);


//Rest parameters
// ES5
function isFullAges() { //arguments keyword
    console.log(arguments); //arguments special variable we excess it all function
} //Output: Arguments(3) [1990, 1999, 1980] aita object array na array er moto dekte
isFullAges(1990, 1999, 1980);


// or ES5 main Example
function isFullAges() {
    var argumentsArray = Array.prototype.slice.call(arguments); //Rest parameters in ES5

    argumentsArray.forEach(function(cur) {
        console.log((2021 - cur) >= 18);
    })
}
isFullAges(1990, 1999, 2000, 2021);


// ES6
function isFullAge(...years) { //Rest parameters in ES6
    years.forEach( cur => console.log((2021-cur) >= 18));
}
isFullAge(1990, 2200, 2021);


//Rest operator use when function decleration //Rest operator recevice cupple of single value and transform them into a
array when we call function with multiple parameter
//Spread operator use when function call //Spread operator takes an array and transform in single value


//Default parameter
// ES5
function SmithPerson(firstName, lastName, barthYear){
    barthYear === undefined ? barthYear = 1990 : barthYear = barthYear;
    this.firstName = firstName;
    this.lastName = lastName;
    this.barthYear = barthYear;
}
var john = new SmithPerson("Jhon", "Smith");
console.log(john);


// ES6
function SmithPerson(firstName, lastName = 'Mark', barthYear = 1999){
    barthYear === undefined ? barthYear = 1990 : barthYear = barthYear;
    this.firstName = firstName;
    this.lastName = lastName;
    this.barthYear = barthYear;
}
var john = new SmithPerson('john');
console.log(john);


//ES6 Maps
const question = new Map();


//classes
//ES5
var Person = function(name, job){

```

```

        this.name = name;
        this.job = job;
    }
    Person.prototype.claculateAge = function(){

    }
    var john = new Person('john', 'Teacher');

//ES6
class Person {
    constructor (name, job){
        this.name = name;
        this.job = job;
    }

    claculateAge(){

    }
}
const john = new Person('john', 'Teacher');

// sub Classes
//ES5
var Person = function(name, job){
    this.name = name;
    this.job = job;
}
Person.prototype.claculateAge = function(){

}
var Athlete = function(name, job, olymicGame, medals){//sub Class
this.olymicGame = olymicGame;
this.medals = medals;
}
Athlete.prototype = Object.create(Person.prototype);//connect to function constrator
var johnAthletes = new Athlete('john', 'Teacher', 'swimmer', 2);
johnAthletes.claculateAge();

//ES6
class Person {
    constructor (name, job){
        this.name = name;
        this.job = job;
    }

    claculateAge(){

    }
}

class Athlete extends Person{
    constructor(name, job, olymicGame, medals){
        super(name, job);//Person class er construction parameter gula pass korbo
        this.olymicGame = olymicGame;
        this.medals = medals;
    }
}
const johnAthletes = new Athlete('john', 'Teacher', 'swimmer', 2);

```