# ES6

## Difference between let and var

var is not not in use

## Function default param

```
function add(first, second = 0){
    const total = first + second;
    return total;
}
```

## Template string

```
const newMultiLine = `First Line of text
Second LIner of text
third line of string
fourth line of code`;

const newSummary = `sum of ${a} and ${b} is: ${a+b}`;
```

## Arrow fucntion

```
// no parameter arrow function
const getPie = () => 3.14;

// one parameter
const doubleIt = (num) => num * 2;

// one parameter simple version
const fiveTimes = num => num * 5;

// multi line arrow function
// if you want to return something, use the return
const doMath = (x, y, z) => {
    const firstSum = x + y;
    const secondSum = y + z;
    const multiplyResult = firstSum * secondSum;
    const result = multiplyResult / 2;
    return result;
}
```

## Spread operator

```
const largest = Math.max(...numbers);
const numbers2 = [...numbers];
const numbers4 = [444, 78, ...numbers, 111, 33]
```

## Destructuring

```
const {age, name} = {name: 'Almas', age: 56, profession: 'makeup artist'}
const [first, another] = [44, 99, 88, 456];
```

## Nested Object, Optional chaining

```
const user = {
    id: 5001,
    name: 'Thomas alba edison',
    address: {
        street: {
            first: '35/A kochukhet lane',
            second: 'third floor',
            third: 'right side'
        },
        postOffice: 'cantonment',
        city: 'Dhaka'
    }
}

const userFloor = user.address?.stret?.second;
console.log(userFloor);
```

## Array looping

```
// for loop
let array = [10,20,30,40];
for(let i = 0; i <= array.length; i++)
{
    console.log(array[i]);
}

//do while
let i = 0;
do{
    console.log(i);
    i++;
} while(i<5)

//while loop
```

```
let i = 0;
while(i<array.length){
    console.log(array[i]);
    i++;
}

//for...of
for(let i of array)
{
    console.log(i);
}

//map - returns an array
const names = {'lincoln','daniel','tabitha','mclead'}
let newArr = names.map((index)=>console.log(index));

//forEach() - doesn't return anything
array.forEach((index)=>{console.log(index)});
```

## Filter, Find, Reduce

```
//filter - returns all matching value
const even = numbers.filter( num => num % 2 === 0);

//find - returns first matching value
const fives = numbers.find(num => num % 5 === 0)

//reduce
const numbers = [1, 2, 3, 4, 5];
// .reduce (accumulatorFunction, initial value)
const total = numbers.reduce( (previous, current) => {
    console.log(previous, current);
    return previous + current
} , 0);
console.log(total);
```

## Notation

```
const student = {
    name: 'Kolim Uddin',
    age: 15,
    class: 'Ten',
    marks: {
        math: 78,
        physics: 89,
        chemistry: 65
    }
}
```

```
const chemistry = student['marks']['chemistry'];
```

## Class

```
class Instructor{
    name;
    designation = 'Web Course Instructor'
    team = 'web team'
    location;
    constructor(name, location){
        this.name = name;
        this.location = location;
    }
    startSupportSession(time){
        console.log(`start the support session at ${time}`)
    }
    createQuiz(module){
        console.log(`please create quiz for module ${module}`)
    }
}
const aamir = new Instructor('aamir', 'mumbai')
console.log(aamir);
aamir.startSupportSession('9.00');
aamir.createQuiz(60);
```

## Inheritance

```
class TeamMember{
    name;
    location;
    constructor(name, location){
        this.name = name;
        this.location = location;
    }
    provideFeedback(){
        console.log(`${this.name} thank you for your feedback.`)
    }
}
class Developer extends TeamMember{
    designation = 'Web Course Instructor'
    team = 'web team'
    tech;
    constructor(name, location, tech){
        super(name, location);
        this.tech = tech;
    }
    developFeature(feature){
        console.log(`Please develop the  ${feature}`)
```

```
    }
    release(version){
        console.log(`please release the version ${version}`)
    }
}

const alia = new Developer('Alia Bhatt', 'Dhaka', 'React');
console.log(alia);
alia.provideFeedback();
```