**🧩 JavaScript’da Class (Klass) haqida to‘liq ma’ruza**

**1. Kirish: Class nima?**

**Class** — bu **obyekt (object)** yaratish uchun mo‘ljallangan **shablon** (template).  
U yordamida bir xil xususiyat va metodlarga ega bo‘lgan **ko‘p obyektlar** yaratish mumkin.

JavaScript’da class — bu **syntactic sugar**, ya’ni **prototype asosida ishlaydigan**, lekin **o‘qilishi va yozilishi qulay** bo‘lgan yangi sintaksisdir.

**2. Class qanday e’lon qilinadi?**

class Person {

constructor(name, age) {

this.name = name;

this.age = age;

}

sayHello() {

console.log(`Salom, mening ismim ${this.name}, yoshim ${this.age}`);

}

}

// Obyekt yaratish

let person1 = new Person("Ali", 25);

person1.sayHello(); // Salom, mening ismim Ali, yoshim 25

**3. Class qismlari**

| **Qism** | **Tavsif** |
| --- | --- |
| **constructor()** | Klassdan obyekt yaratilganda birinchi chaqiladigan metod |
| **this** | Obyektning o‘ziga ishora qiladi |
| **methods** | Klass ichidagi funksiyalar |
| **properties** | Obyekt xususiyatlari (klass ichida this orqali belgilanadi) |

**4. Class ichida metodlar**

class Calculator {

add(a, b) {

return a + b;

}

subtract(a, b) {

return a - b;

}

}

let calc = new Calculator();

console.log(calc.add(5, 3)); // 8

console.log(calc.subtract(5, 3)); // 2

---------bo’lish ko’paytirish metodlarini qo’shing----------

**5. constructor haqida**

constructor — bu maxsus metod bo‘lib, **obyekt yaratilganda avtomatik ishga tushadi**.

class Car {

constructor(brand, year) {

this.brand = brand;

this.year = year;

}

}

let car1 = new Car("BMW", 2024);

console.log(car1.brand); // BMW

**6. Meros olish (Inheritance)**

Bir klass boshqa klassdan **xususiyat va metodlarni meros qilib olishi** mumkin.  
Buning uchun **extends** va **super()** ishlatiladi.

class Animal {

constructor(name) {

this.name = name;

}

speak() {

console.log(`${this.name} ovoz chiqarmoqda`);

}

}

class Dog extends Animal {

constructor(name, breed) {

super(name); // ota klassning constructorini chaqirish

this.breed = breed;

}

bark() {

console.log(`${this.name} hurmoqda!`);

}

}

let dog1 = new Dog("Sharik", "Ovcharka");

dog1.speak(); // Sharik ovoz chiqarmoqda

dog1.bark(); // Sharik hurmoqda!

**7. Getters va Setters**

Getter va setter metodlar yordamida **xususiyatlarga kirish va ularni boshqarish** mumkin.

class Person {

constructor(name, age) {

this.\_name = name;

this.\_age = age;

}

get age() {

return this.\_age;

}

set age(value) {

if (value < 0) {

console.log("Yosh manfiy bo‘lmasligi kerak!");

} else {

this.\_age = value;

}

}

}

let p1 = new Person("Ali", 25);

console.log(p1.age); // 25

p1.age = -5; // Xato xabar

**8. Static metodlar**

static metodlar — obyektga emas, **klassning o‘ziga tegishli** bo‘ladi.

class MathUtils {

static add(a, b) {

return a + b;

}

}

console.log(MathUtils.add(10, 20)); // 30

🔹 static metodlarni obyekt orqali emas, **klass nomi orqali** chaqiriladi.

**9. Private (maxfiy) xususiyatlar (# bilan)**

ES2022 dan boshlab, # yordamida **maxfiy o‘zgaruvchilar** yaratish mumkin.

class BankAccount {

#balance = 0;

deposit(amount) {

this.#balance += amount;

}

getBalance() {

return this.#balance;

}

}

let account = new BankAccount();

account.deposit(100);

console.log(account.getBalance()); // 100

// console.log(account.#balance); ❌ Xato — private property

-----------withdraw() metodini qo’shing-----------

**10. Class Expression**

Class’larni **o‘zgaruvchiga saqlab e’lon qilish** ham mumkin:

const Rectangle = class {

constructor(width, height) {

this.width = width;

this.height = height;

}

area() {

return this.width \* this.height;

}

};

let rect = new Rectangle(10, 5);

console.log(rect.area()); // 50

**11. Amaliy misol**

class Library {

constructor() {

this.books = [];

}

addBook(book) {

this.books.push(book);

}

getTotalPrice() {

return this.books.reduce((sum, book) => sum + book.price, 0);

}

getBooksByAuthor(author) {

return this.books.filter(b => b.author === author);

}

}

let myLib = new Library();

myLib.addBook({ title: "JS Basics", author: "Ali", price: 30 });

myLib.addBook({ title: "Node.js", author: "Vali", price: 40 });

myLib.addBook({ title: "React", author: "Ali", price: 25 });

console.log("Umumiy narx:", myLib.getTotalPrice());

console.log("Ali kitoblari:", myLib.getBooksByAuthor("Ali"));

**12. Xulosa**

| **Afzallik** | **Tavsif** |
| --- | --- |
| 🔹 Strukturali kod | Kodni toza va o‘qilishi oson qiladi |
| 🔹 Qayta ishlatish | Bir classdan ko‘p obyekt yaratish mumkin |
| 🔹 Meros olish | Katta loyihalarda moslashuvchanlik beradi |
| 🔹 Yangi sintaksis | Zamonaviy JavaScript’da standart hisoblanadi |

**🎯 Yakuniy eslatma**

* class — bu **obyekt yaratish uchun zamonaviy usul**.
* constructor function eski usul, lekin ishlash mantig‘i o‘xshash.
* **Meros olish, getter/setter, static, private field** — class’larni kuchli qiladi.
* Har qanday katta loyiha yoki framework (React, Node.js, Vue) class konseptidan foydalanadi.

**🧩 1. Maosh hisoblash**

**Vazifa:**  
Employee classi: name, salary, position xossalari bilan.  
getAnnualSalary() metodi — yiliga oladigan maoshni hisoblasin.

**🧩 2. Student baholari**

**Vazifa:**  
Student classini yarating:

* name, grades (massiv) xossalari bo‘lsin
* addGrade(grade) va getAverage() metodlari ishlasin.

**🧩 3. Inson oilasi**

**Vazifa:**  
Person → Parent → Child meros tizimi yarating:

* Har biri o‘z metodiga ega bo‘lsin (introduce(), work(), study()).

**🧩 4. Avtomobil parki**

**Vazifa:**  
Car classi → ElectricCar subclass yarating.

* ElectricCar da batteryLevel bo‘lsin.
* charge() metodi batareyani to‘ldirsin.

**🧩 5. Online do‘kon**

**Vazifa:**  
Product class → Shop class.

* Shop da products massivi bo‘lsin
* addProduct(product) va getTotalPrice() metodlari ishlasin.

**🧩 6. reduce bilan class ichida ishlash**

**Vazifa:**  
Company classi:

* employees massivida {name, salary} obyektlar saqlansin
* getHighestSalary() — eng katta maoshli xodimni qaytarsin (reduce bilan).

**🧩 7. To‘liq loyiha: TodoList**

**Vazifa:**  
TodoList classi yarating:

* tasks massiv bo‘lsin
* addTask(task) — yangi vazifa qo‘shadi
* removeTask(task) — vazifani o‘chiradi
* showTasks() — barcha vazifalarni konsolga chiqaradi.