1)

**Meros olish (Inheritance)** obyektga yoʻnaltirilgan dasturlashda (OOP) mavjud bo‘lgan sinfdan (class) yangi sinf (farzand sinf) yaratish jarayonidir. Yangi sinf eski sinfning xususiyatlari va metodlarini o‘zida saqlab, ularni kengaytirishi yoki o'zgartirishi mumkin. Bu konsepsiya kodni qayta ishlatishni ta'minlaydi.

2)

Encapsulation malumotlarni yashirib qoyishlik unga dostub bermaslik classlarini berkitish uchun kerak. # ishorasi orqali beriladi

3)

**Polimorfizm** obyektga yo‘naltirilgan dasturlashda (OOP) muhim tushunchalardan biri bo‘lib, bir xil nomdagi funksiyalar yoki metodlarning turli sinflar yoki obyektlar bilan turlicha ishlashiga imkon beradi. Bu tushuncha dasturiy ta'minotni yanada moslashuvchan va kengaytirishga yaroqli qiladi.

4)

Abstraksiya

JavaScript'da abstraksiya sinflar va obyektlar orqali amalga oshiriladi. Ota sinflar umumiy xatti-harakatlarni ta’minlaydi, farzand sinflar esa ularni aniq maqsadlarga moslashtirishi mumkin.

5)

JavaScript'dagi **Map** obyektlari — bu **kalit-qiymat** juftliklarini saqlashga mo‘ljallangan ma’lumot tuzilmasidir. **Map** obyektlarida kalit sifatida har qanday turdagi ma’lumotni ishlatish mumkin (primitive yoki object), bu esa uni **Object** obyektiga nisbatan ko‘proq moslashuvchan qiladi, chunki **Object** kalit sifatida faqat string va symbol turini qabul qiladi.

6)

JavaScript'dagi **Set** — bu ma’lumotlar tuzilmasi bo‘lib, unda **takrorlanmaydigan** qiymatlar to‘plami saqlanadi. **Set** primitiv turlar (raqamlar, stringlar) va obyektlar kabi har xil turdagi qiymatlarni saqlashi mumkin, lekin har bir qiymat to‘plamda faqat bir marta mavjud bo‘ladi.