**Javascript scope?**

JavaScriptda turli xil qamrov turlarini bilaman, ular uchta global, mahalliy va blokli qamrovdir. Ushbu qamrovlar kodimning turli qismlaridagi o'zgaruvchilarga kirish imkoniyatini nazorat qiladi va kodimni tashkil qilishni saqlashda va o'zgaruvchan to'qnashuvlarning oldini olishda muhim rol o'ynaydi.

**Global scope?**

JavaScript-da global miqyos mavjud bo'lgan eng keng qamrovdir. Global miqyosda e'lon qilingan o'zgaruvchilarga kodingizning istalgan joyidan kirish mumkin, xoh u ichki funktsiyalar, shartli bayonotlar, sikllar yoki boshqa kod bloklari bo'ladimi.

**Local scope?**

JavaScript-ning mahalliy doirasi bino ichidagi shaxsiy xonaga o'xshaydi - bu o'zgaruvchilarga faqat ma'lum bir xonadan kirish mumkin bo'lgan yopiq joy.

O'zgaruvchini mahalliy miqyosda e'lon qilganingizda, u aniqlangan kod bloki, funksiya yoki shartli bayonot bilan cheklangan bo'ladi.

Mahalliy miqyosdagi o'zgaruvchilar o'z doirasidan tashqaridagi kod bilan aralashish yoki o'zgartirishdan himoyalangan bo'lib, izolyatsiya darajasini ta'minlaydi.

**Block scope?**

JavaScript-da blokirovka doirasi har biri o'z o'zgaruvchilar to'plamiga ega bo'lgan kattaroq konteyner ichidagi bir qator ichki qutilarga o'xshaydi.

Funktsiyalar yoki global kontekst bilan belgilanadigan global va mahalliy doiralardan farqli o'laroq, blok doirasi shartli bayonotlar (if, else, switch) va tsikllar (for, while) kabi maxsus kod bloklari ichida yaratiladi.

Blok doirasida e'lon qilingan o'zgaruvchilar yuqori darajadagi izolyatsiyani ta'minlovchi blok bilan chegaralangan.

**Optional Chaining (?.) nima?**

ECMAScript 2020 da joriy qilingan ixtiyoriy zanjirlash bu oraliq xususiyatlar null yoki undefined bo‘lishi mumkin bo‘lgan hollarda ichki o‘rnatilgan obyektlar yoki massivlarning xossalari va usullariga kirish jarayonini soddalashtiradigan xususiyatdir.

Ixtiyoriy zanjirlash operatori (?.) aniq null yoki aniqlanmagan tekshiruvlarsiz xususiyatlar yoki usullarga kirish imkonini beradi. Agar zanjirdagi har qanday oraliq xususiyat nol yoki undefined bo'lsa, ifoda qisqa tutashuvga olib keladi va natija undefined deb o'rnatiladi.

Dasturlashda "qisqa tutashuv" - bu xususiyat yoki usullar zanjiri bo'ylab nol yoki undefined qiymat paydo bo'lishi bilanoq ifodani baholash to'xtatiladigan xatti-harakatni anglatadi. Ifodani baholashni davom ettirish o'rniga, natija darhol undefined qilib o'rnatiladi va har qanday keyingi xususiyat yoki usulga kirish o'tkazib yuboriladi.

**Nullish Coalescing?**

ES11 ikkita savol belgilari bilan belgilanadigan nullish coalescing operatorni qo'shdi: ??.

Nima uchun JavaScript-ga Nullish Coalescing operatori kerak edi

|| operator ajoyib ishlaydi, lekin ba'zida biz faqat birinchi operand null yoki aniqlanmagan bo'lsa, keyingi ifodani baholashni xohlaymiz.

Shuning uchun, ES11 null birlashtiruvchi operatorni qo'shdi.

x ifodasida ?? y,

Agar x null yoki aniqlanmagan bo'lsa, faqat natija y bo'ladi. Agar x null yoki aniqlanmagan bo'lsa, natija x bo'ladi.Bu shartli tekshiruvlar va disk raskadrovka kodini oson vazifaga aylantiradi.

Nullish Coalescing operatori har qanday o'zgaruvchi uchun faqat null yoki noaniq qiymat haqida qayg'urganingizda juda foydali bo'ladi.

ES6 dan boshlab, JavaScript-ga juda ko'p foydali qo'shimchalar mavjud

ES6 tuzilmasini buzish

Import and Export Syntax

Arrow functions

Promises

Async/await

Optional chaining operator

va yana ko'p narsalar.

ES6+ ning barcha funksiyalari haqida hamma narsani “Zamonaviy JavaScript-ni oʻzlashtirish” kitobida batafsil oʻrganishingiz mumkin.