

JAVASCRIPT DASTURLASH TILI O'ZGARUVCHILAR, MA'LUMOTLAR TIPLAR VA IFODALAR

Ashurova Gulbahor Umarali qizi

**Qo'qon Davlat Pedagogika Instituti, Fizika-Matematika fakultite, Matematika –
informatika yo'nalishi, 3-kurs, 302-guruh talabasi**

E-mail: bekmirzayovnasrullo@gmail.com

<https://doi.org/10.5281/zenodo.7923170>

JavaScriptda o'zgaruvchilarga hech qanday tip biriktirilmaydi. Biron-bir tipdaqi qiymatni o'zlashtirgandan so'ng o'zgaruvchiga boshqa tipdagi qiymatni bersangiz hech qanday xatolik yuz bermaydi va o'zgaruchi yangi tipni o'zlashtirib oladi. Shuning uchun JavaScriptni ba'zida tipizatsiyaga ega emas deyishadi. Ammo bu mutlaqo noto'g'ri, JavaScript tipizatsiyaga ega va u "dinamik tipizatsiya" qoidalari asosida ishlaydi.

O'zgaruvchidan foydalanish uchun uni avval e'lon qilish kerak. JavaScriptda buni 3 xil usulda amalga oshirish mumkin: var, let yoki const kalit so'zlari orqali. Bularning har biri turli maqsadlarda ishlatiladi.

var orqali e'lon qilish

ES2015 gacha var o'zgaruvchini e'lon qilishning yagona yo'li bo'lgan.

Agar var so'zini qo'yishni unutsangiz, siz e'lon qilinmagan o'zgaruvchiga qiymat berayotgan bo'lasiz va natija siz kutgandek bo'lmaydi. Zamonaviy muhitlarda yoki strict rejimi yoqilgan bo'lsa, yuqoridagi holatda xatolik yuz beradi. Eski muhitlarda (yoki strict rejimi o'chirilgan bo'lsa), o'zgaruvchini initsializatsiya qiladi va global obyektga biriktirib qo'yadi. Qo'shimcha ma'lumot uchun, initsializatsiya — o'zgaruvchiga dastlabki qiymatni o'zlashtirish jarayoni.

o'zgaruvchini e'lon qilganda uni initsializatsiya qilmasangiz, u undefined qiymatini o'zlashtiradi va unga yangi qiymat bermaguningizcha bu holatni saqlab turadi.

var a // typeof a === 'undefined'

Bir o'zgaruvchini bir necha marta e'lon qilishingiz mumkin, bunda oldingisi inkor qilinadi:

var a = 1

var a = 2

Bir qatorda bir nechta o'zgaruvchilarni e'lon qilishingiz ham mumkin:

var a = 1, b = 2

Dasturlashda "scope" tushunchasi mavjud, o'zbek tiliga ko'rinish sohasi deb tarjima qilsak bo'ladi (IMHO). Ko'rinish sohasi deganda o'zgaruvchi ko'rinadigan va foydalansa bo'ladigan dastur qismi tushuniladi.

Funksiyadan tashqarida var orqali e'lon qilingan o'zgaruvchi global obyektga biriktiriladi va global ko'rinish sohasiga ega bo'ladi. Bunday o'zgaruvchidan dasturning istalgan qismida foydalanishingiz mumkin. Agar var orqali funksiya ichida e'lon qilsangiz, o'zgaruvchi funksiya biriktiriladi va u faqat funksiya ichida ishlaydi, funksiya parametri kabi.

Agar funksiya ichida global obyektidagi o'zgaruvchi bilan bir xil nomdagi o'zgaruvchi e'lon qilsangiz, funksiya ichidagi ustunlikka ega bo'ladi va global o'zgaruvchini inkor qilib yuboradi. Shuni ta'kidlash muhimki, { } qavslar bilan ajratilgan blok yaratganiz bilan ko'rinish sohasi (scope) yaratilmaydi. Funksiya yaratganingizdagina ko'rinish sohasi yaratiladi, chunki var blok ko'rinish sohasiga ega emas, balki funksiyaviy ko'rinish sohaga ega.

Funksiya ichida e'lon qilingan o'zgaruvchi funksiya bo'ylab istalgan joyda ishlatilishi mumkin. Hattoki, funksiya ohirida e'lon qilsangiz ham funksiya boshida undan foydalanish mumkin,

chunki JavaScript kodni bajarishdan oldin barcha o'zgaruvchilarni yuqoriga ko'chiradi. Lekin chalg'ib ketmaslik uchun o'zgaruvchini funksiya boshida e'lon qilganingiz ma'qul. let orqali e'lon qilish Yuqorida aytganimizdek, var blok ko'rinish sohasiga ega emas. Bu muammoni hal qilish uchun ES2015 da let qo'shilgan. let orqali e'lon qilgan o'zgaruvchining ko'rinish sohasi faqat o'zi e'lon qilgan blokka hamda undagi ichki bloklargagina tegishli.

Zamonaviy dasturchilar, asosan, let dangina foydalanishmoqda, va var dan butunlay voz kechishmoqda.

Yana bir farqi, funksiya tashqarisida let orqali e'lon qilganizda, var dan farqli o'laroq o'zgaruvchini global o'zgaruvchiga aylantirmaydi.

Open in app

Sign up

Sign In

Qisqa satrlarda JavaScript — 3. O'zgaruvchi, tip va ifodalar.

Follow

Share

Manba Unsplash

O'zgaruvchilar

Indentifikatorga biriktirilgan literal o'zgaruvchi deb ataladi, ulardan dasturning keyingi qismida foydalanishingiz mumkin.

JavaScriptda o'zgaruvchilarga hech qanday tip biriktirilmaydi. Biron-bir tipdaqi qiymatni o'zlashtirgandan so'ng o'zgaruvchiga boshqa tipdagi qiymatni bersangiz hech qanday xatolik yuz bermaydi va o'zgaruchi yangi tipni o'zlashtirib oladi. Shuning uchun JavaScriptni ba'zida tipizatsiyaga ega emas deyishadi. Ammo bu mutlaqo noto'g'ri, JavaScript tipizatsiyaga ega va u "dinamik tipizatsiya" qoidalari asosida ishlaydi.

O'zgaruvchidan foydalanish uchun uni avval e'lon qilish kerak. JavaScriptda buni 3 xil usulda amalga oshirish mumkin: var, let yoki const kalit so'zlari orqali. Bularning har biri turli maqsadlarda ishlatiladi.

var orqali e'lon qilish

ES2015 gacha var o'zgaruchini e'lon qilishning yagona yo'li bo'lgan.

var a = 0

Agar var so'zini qo'yishni unutsangiz, siz e'lon qilinmagan o'zgaruvchiga qiymat berayotgan bo'lasiz va natija siz kutgandek bo'lmaydi. Zamonaviy muhitlarda yoki strict rejimi yoqilgan bo'lsa, yuqoridagi holatda xatolik yuz beradi. Eski muhitlarda (yoki strict rejimi o'chirilgan bo'lsa), o'zgaruvchini initsializatsiya qiladi va global obyektga biriktirib qo'yadi. Qo'shimcha ma'lumot uchun, initsializatsiya — o'zgaruvchiga dastlabki qiymatni o'zlashtirish jarayoni.

o'zgaruvchini e'lon qilganda uni initsializatsiya qilmasangiz, u undefined qiymatini o'zlashtiradi va unga yangi qiymat bermaguningizcha bu holatni saqlab turadi.

var a // typeof a === 'undefined'

Bir o'zgaruvchini bir necha marta e'lon qilishingiz mumkin, bunda oldingisi inkor qilinadi:

var a = 1

var a = 2

Bir qatorda bir nechta o'zgaruvchilarni e'lon qilishingiz ham mumkin:

var a = 1, b = 2

Dasturlashda “scope” tushunchasi mavjud, o‘zbek tiliga ko‘rinish sohasi deb tarjima qilsak bo‘ladi (IMHO). Ko‘rinish sohasi deganda o‘zgaruvchi ko‘rinadigan va foydalansa bo‘ladigan dastur qismi tushuniladi.

Funksiyadan tashqarida var orqali e‘lon qilingan o‘zgaruvchi global obyektga biriktiriladi va global ko‘rinish sohasiga ega bo‘ladi. Bunday o‘zgaruvchidan dasturning istalgan qismida foydalanishingiz mumkin. Agar var orqali funksiya ichida e‘lon qilsangiz, o‘zgaruvchi funksiyaga biriktiriladi va u faqat funksiya ichida ishlaydi, funksiya parametri kabi.

Agar funksiya ichida global obyektidagi o‘zgaruvchi bilan bir xil nomdagi o‘zgaruvchi e‘lon qilsangiz, funksiya ichidagi ustunlikka ega bo‘ladi va global o‘zgaruvchini inkor qilib yuboradi. Shuni ta’kidlash muhimki, { } qavslar bilan ajratilgan blok yaratganiz bilan ko‘rinish sohasi (scope) yaratilmaydi. Funksiya yaratganingizdagina ko‘rinish sohasi yaratiladi, chunki var blok ko‘rinish sohasiga ega emas, balki funksiyaviy ko‘rinish sohaga ega.

Funksiya ichida e‘lon qilingan o‘zgaruvchi funksiya bo‘ylab istalgan joyda ishlatilishi mumkin. Hattoki, funksiya ohirida e‘lon qilsangiz ham funksiya boshida undan foydalanish mumkin, chunki JavaScript kodni bajarishdan oldin barcha o‘zgaruvchilarni yuqoriga ko‘chiradi. Lekin chalg‘ib ketmaslik uchun o‘zgaruvchini funksiya boshida e‘lon qilganingiz ma’qul.

let orqali e‘lon qilish Yuqorida aytganimizdek, var blok ko‘rinish sohasiga ega emas. Bu muammoni hal qilish uchun ES2015 da let qo‘shilgan. let orqali e‘lon qilgan o‘zgaruvchining ko‘rinish sohasi faqat o‘zi e‘lon qilgan blokka hamda undagi ichki bloklargagina tegishli. Zamonaviy dasturchilar, asosan, let dangina foydalanishmoqda, va var dan butunlay voz kechishmoqda. Yana bir farqi, funksiya tashqarisida let orqali e‘lon qilganizda, var dan farqli o‘laroq o‘zgaruvchini global o‘zgaruvchiga aylantirmaydi.

const orqali e‘lon qilish var yoki let yordamida e‘lon qilingan o‘zgaruvchilar keyinchalik qiymatini o‘zgartirishi mumkin. const da e‘lon qilingan o‘zgaruvchi initsializatsiya qilingandan keyin boshqa o‘zgarmaydi va uning qiymati o‘zgarmas qiymatga aylanadi, ya’ni konstantaga.

`const a = 'test'`

a const ga istalgan literalni biriktirish mumkin. Yana bir jihati: agar a konstantaga obyekt biriktirilgan bo‘lsa, undagi funksiyalari orqali obyekt qiymatlarini o‘zgartirsa bo‘ladi.

const ko‘rsatkichni o‘zgartirmaslikni ta’minlaydi. Shuningdek, let kabi blok ko‘rinish sohasiga ega.

Zamonaviy dasturchilar dastur davomida o‘zgarmaydigan o‘zgaruvchilarni const da e‘lon qilishni ma’qul ko‘rishadi, chunki bu keyinchalik uchrashi mumkin bo‘lgan xatoliklarni oldini oladi.

Tiplar

Siz JavaScriptni tipizatsiyaga ega emas deb eshitgan bo‘lishingiz mumkin. Oldin ham ta’kidlaganimdek, bu mutlaqo noto‘g‘ri. JavaScript tipizatsiyaga ega, va aynan “dinamik tipizatsiya” xususiyatida ishlaydi. Bu degani, siz bir o‘zgaruvchiga istalgan turdagi tipni o‘zlashtirishingiz mumkin. JavaScriptda ikki xil tiplar mavjud: oddiy va murakkab tiplar.

Oddiy tiplar

Oddiy tiplarga quyidagilar kiradi:

Sonlar

Qatorlar

Mantiqiy tiplar

Foydalanilgan adabiyotlar:

- 1.G'ulomov S.S. va boshqalar "Axborot tizimlari va texnologiyalari" .Oliy o'quv yurti talabalari uchun darslik. Toshkent-"Sharq"- 2000. 529 b.
- 2.Juraev, Muzaffarjon Mansurjonovich. "Prospects for the development of professional training of students of professional educational institutions using electronic educational resources in the environment of digital transformation." *Academicia Globe: Inderscience Research* 3.10 (2022): 158-162.
- 3.Juraev, Muzaffarjon Mansurjonovich. "The value of open mass competitions in the process of digitalization of extracurricular activities of schoolchildren." *Web of Scientist: International Scientific Research Journal* 3.10 (2022): 338-344.
- 4.Jo'rayev, Muzaffarjon. "Professional ta'lim jarayonida fanlararo uzvilik va uzliksizlikni ta'minlash o'quvchilari kasbiy tayyorgarligining muhim omili sifatida." *Zamonaviy dunyoda amaliy fanlar: Muammolar va yechimlar* 1.29 (2022): 43-46.
- 5.Juraev, M. M. "OA Qo'ysinov Description of the methodological basis for ensuring interdisciplinary continuity of the subject "Computer Science and Information Technology" in vocational education." *JournalNX-A Multidisciplinary Peer Reviewed* 7.10 (2021).
- 6.Mansurjonovich, Juraev Muzaffarjon. "Description of the Methodological Basis for Ensuring Interdisciplinary Continuity of the Subject" *Computer Science and Information TECHNOLOGY" in Vocational Education.* *JournalNX* 7.10: 223-225.
- 7.Khasanov, A. R. (2022). LEARNING IS A COMPETENCY-BASED APPROACH AS A CONTENT UPDATE STEP. *Galaxy International Interdisciplinary Research Journal*, 10(12), 217-223.
8. Khasanov, A. R. (2022). Development of information competence of future informatics teachers as a pedagogical problem. *Open Access Repository*, 9(12), 73-79.
9. Xasanov, A. R. (2021, May). USE OF MODERN PEDAGOGICAL TECHNOLOGIES AND INTERACTIVE METHODS IN TEACHING COMPUTER SCIENCE. In *E-Conference Globe* (pp. 198-199).
10. Mansurjonovich, Juraev Muzaffarjon. "CURRENT STATUS OF THE SCIENCE OF INFORMATICS AND INFORMATION TECHNOLOGIES IN THE PROFESSIONAL EDUCATION SYSTEM, EXISTING PROBLEMS AND SOLUTIONS, PRINCIPLES AND CONTENT OF THE SCIENCE ORGANIZATION." *Galaxy International Interdisciplinary Research Journal* 10.12 (2022): 327-331.
- 11.Mansurjonovich, Juraev Muzaffarjon. "Professional Educational Institutions Theoretical and Practical Basis of Development of the Content of Pedagogical Activity of Teachers of" *Information and Information Technologies*." *Open Access Repository* 9.12 (2022): 85-89.
- 12.Mansurjonovich, Juraev Muzaffarjon. "Experience Of Cambridge Curricula In Ensuring The Continuity Of Curricula In The Field Of "Computer Science And Information Technology" In The System Of Professional Education." *The American Journal of Interdisciplinary Innovations and Research* 3.11 (2021): 26-32.
- 13.Juraev, Muzaffarjon Mansurjonovich. "Theoretical and practical principles of improving the content of the pedagogical activity of ICT teachers of professional educational institutions in the conditions of information of education." (2022).
- 14.Juraev, Muzaffarjon Mansurjonovich. "Methodological foundations for improving the content of training future ict teachers in the conditions of digital transformation of education." (2022): 9-11.

- 15.Melikyzievich, Siddikov Ilkhom, et al. "THE METHOD OF REFERENCE TESTS FOR THE DIAGNOSIS OF DIGITAL DEVICES." International Journal of Early Childhood Special Education 14.7 (2022).
- 16.Mansurjonovich, Juraev Muzaffarjon. "Designing an electronic didactic environment to ensure interdisciplinary integration in the teaching of" Informatics and information technologies" during professional education." Confrencea 11.11 (2023): 78-82.
- 17.Mansurjonovich, Juraev Muzaffarjon, and Aroyev Dilshod Davronovich. "INTERDISCIPLINARY INTEGRATION IS AN IMPORTANT PART OF DEVELOPING THE PROFESSIONAL TRAINING OF STUDENTS." Open Access Repository 9.1 (2023): 93-101.
- 18.Juraev, M. M. "ZY Xudoyberdiyev Theoretical analysis of the continuity model of computer science and information technology in the System of professional education." European Scholar Journal (ESJ)//ISSN (E): 2660-5562.
- 19.Xudayberdiyev, Zayniddin Yavkachevich, and Muzaffarjon Mansurjonovich Juraev. "Theoretical analysis of the continuity model of computer science and information technology in the system of professional education." (2021).

INNOVATIVE
ACADEMY