**O‘ZBEKISTON RESPUBLIKASI AXBOROT TEXNOLOGIYALARI VA**

**KOMMUNIKATSIYALARINI RIVOJALANTIRISH VAZIRLIGI**

**MUHAMMAD AL-XORAZMIY NOMIDAGI TOSHKENT AXBOROT TEXNOLOGIYALARI UNIVERSITETI**

**TELEVIZION TEXNOLOGIYALAR FAKULTETI**

**«AUDIOVIZUAL TEXNOLOGIYALAR» KAFEDRASI**

**JAVASCRIPT VA UNING IMKONIYATLARI**

# INDIVIDUAL LOYIHA ISHIGA HISOBOT YOZUVI

Bajardi:

…...............talabasi

...............................

Qabul qildi: kafedra katta o‘qituvchisi

…..........................

Toshkent 2022

Mazkur loyiha ishi “Javascript va uning imkoniyatlari” mavzusida bajarilgan. Loyiha ishida Javascript haqida umumiy ma’lumotlar , uning nimalarga qodirligi , qaysi sohalarda qo’llanilishi rasmlar , amaliy bajarilgan dasturlar yordamida to’laqonli ko’rsatilib o’tilgan

Данная проектная работа выполнена на тему” Javascript и его возможности". В проектной работе подробно изложена общая информация о Javascript , на что он способен , в каких областях может применяться , с использованием практически выполняемых программ

This project work was carried out on the topic ”Javascript and its capabilities". The project work details general information about Javascript , what it is capable of , in which areas it can be used , using practically executed programs

# MUNDARIJA

|  |  |
| --- | --- |
| KIRISH…………………………………….................................................. | 4 |
| I BOB. NAZARIY QISM………………………………………………..… | 4 |
| 1.1. Javascript haqida ma’lumot….…………………...............……….. | 4 |
| 1.2. Javascript ishlatish sferasi..……………………………………….…. | 9 |
| 1.3.Javascriptda amaliy kod ko’rinishi……….... | 12 |
| II BOB. AMALIY QISM…………………………………………………... | 20 |
| 2.1. Javascriptda barcha obyektlari bilan ishlash ……..……….… | 20 |
| 2.2. NODEJS.......................................................................……….…... | 23 |
| 2.3. MERN STACK......................……………….………………..…… | 29 |
| 2.4. VUEJS………….............................….………………………..…... | 35 |
| XULOSA…………………………………………………………………… | 34 |
| FOYDALANILGAN ADABIYOTLAR…………………………………… | 35 |

**KIRISH**

Keling javascriptni nima afzalliklari bor , nima imkoniyatlari bor va boshqa texnologiyalar bilan qanday qilib integratsiya qiladi, shularni ko’rib chiqamiz. Boshida Javascript veb sahifalarni jonlantirish uchun yaratilgan edi

Bu dasturlash tilida yozilgan dasturlar scriptlar deb ataladi

Skriptlar oddiy matn sifatida taqsimlanadi va bajariladi. Ularni ishga tushirish uchun maxsus tayyorgarlik yoki kompilyatsiya kerak emas

Bu Javascriptni boshqa dasturlash tillaridan ajratib turadi , misol uchun “Java” dan

Boshida Javascript yaratilayotganda , uning nomi

<<LiveScript>> edi . Aynan usha paytda “JAVA” dasturlash tili juda mashhur edi , va JavaScript deb shunchaki nomiga qo’shib qo’yishgan , “подход” qilishgan , lekin javascriptni “java” bilan adashtirmaslik lozim.

Vaqt o'tishi bilan JavaScript ECMAScript deb nomlangan spetsifikatsiyasi sababli to'liq mustaqil tilga aylandi va endi Java bilan hech qanday aloqasi yo'q.

Bugungi kunda JavaScript nafaqat brauzerda, balki serverda yoki "JavaScript" deb nomlangan maxsus dasturga ega bo'lgan boshqa qurilmada ham amalga oshirilishi mumkin.

Brauzer o'z dvigateliga ega, ba'zan " virtual JavaScript mashinasi "deb ataladi.

Turli dvigatellar turli "kod nomlari"ga ega. Misol uchun: V8-Chrome va Opera-da.

SpiderMonkey-Firefox-da.

... IE ning turli versiyalari uchun "Trident" va "Chakra", Microsoft

Edge uchun "ChakraCore", Safari uchun "Nitro" va "SquirrelFish" va boshqalar.

**Kurs ishining maqsadi:** Javascriptning imkoniyatlari bilan tanishib chiqish

**Kurs ishining vazifasi:** O’quvchiga javascriptning muhiti qanday ekanligini va u qanday qilib ilovalarga integratsiya qilinishini o’rgatish

# I BOB. NAZARIY QISM

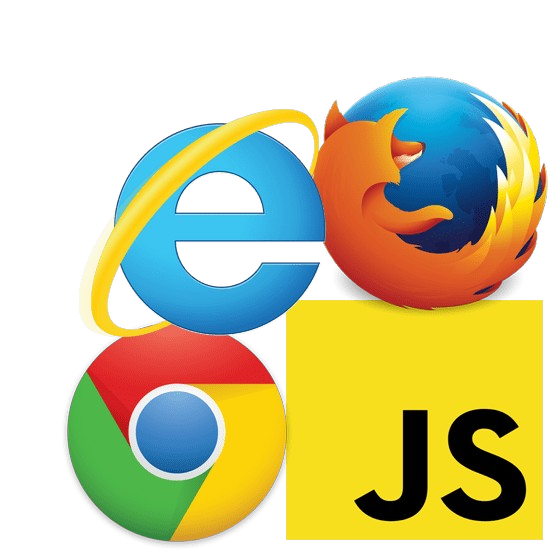
## Javascript haqida ma’lumot

Dvijo’k (o'rnatilgan, agar u brauzer bo'lsa) skript matnini o'qiydi, (“tahlil qiladi”).

Keyin skriptni mashina tiliga aylantiradi ("kompilyatsiya").

Shundan so'ng, mashina kodi ishga tushiriladi va juda tez ishlaydi.

Dvigatel har bir bosqichda optimallashtirishni qo'llaydi. U hatto ish vaqtida tuzilgan skriptni ko'zdan kechiradi, u orqali o'tadigan ma'lumotlarni tahlil qiladi va olingan ma'lumotlarga tayanib, mashina kodiga optimallashtirishni qo'llaydi. Natijada, skriptlar juda tez ishlaydi.



Rasm-1 (aktual browserlarda javascript qo’llaniladi)

Zamonaviy javascript bu xavfsiz dasturlash tilidir Javascriptning imkoniyatlari qaysi muhitda ishlayotganligiga bog’liq. Misol uchun NodeJS , brauzerga fayllarni yuklashga, o’qishga, va tarmoqli so’rovlar yuborishga yordam beradi

Misol uchun Brauzerda javascriptning imkoniyatlari:

Web sahifaga yangi HTML kod qo’shish, tayyor holatdagi sahifani o’zgartirishga, va still berishga imkon beradi

Web sahifadan foydalanayotgan foydalanuvchini ilg’ash , Bu asosan sichqonchani bosganda , kursorni yurg;izganda , yoki klaviaturani bosganda

Serverlarga so’rov yuborish, fayllar yuklab olish , ularni joylashtirish, (asosan AJAX, COMET, AXIOS yordamida qilinadi) ni imkonini beradi

Kukilarni (Cookie ) olish , o’rnatish , foydalanuvchiga savol berish, xabar yuborish, ko’rsatishni imkonini beradi

Klient tarafida ma’lumotlarni saqlash ( localStorage ) imkonini beradi

Veb-sahifadagi JavaScript-ni qattiq diskdagi o'zboshimchalik bilan fayllarni o'qish/yozish, ularni nusxalash yoki dasturlarni ishga tushirish mumkin emas. OS tizimining xususiyatlariga bevosita kirish imkoni yo'q.

Zamonaviy brauzerlar fayllar bilan ishlashga imkon beradi, lekin cheklangan kirish imkoniyati mavjud va foydalanuvchi faqat faylni brauzer oynasiga "sudrab tashlash" yoki <input>tegi bilan tanlash kabi muayyan harakatlar qilsa, uni ta'minlaydi

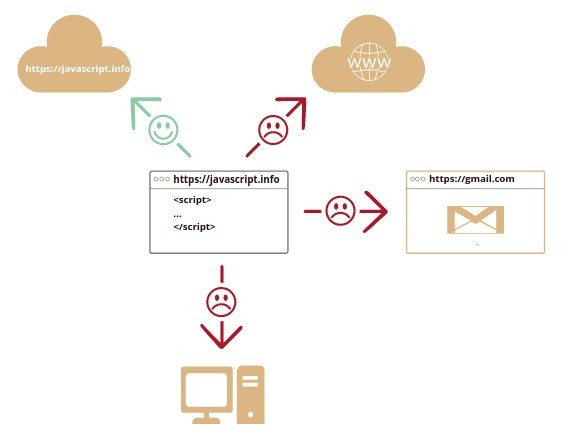
Kamera/mikrofon va boshqa qurilmalar bilan ishlash usullari mavjud, ammo ular aniq foydalanuvchi ruxsatini talab qiladi. Shunday qilib, JavaScript-ni qo'llab- quvvatlaydigan sahifa veb-kamerani sezilarli darajada yoqmaydi, nima sodir bo'lishini kuzata olmaydi va FSB-ga ma'lumot yubora olmaydi.

Turli oyna / yorliqlar bir-birlarini bilishmaydi. Ba'zan JavaScript-ni ishlatib, bitta oyna boshqa Oynani ochadi. Biroq, bu holatda ham, bir sahifadan JavaScript-ni turli saytlardan (boshqa domen, protokol yoki portdan) kelgan bo'lsa, boshqasiga kirish imkoni yo'q.

Bunga "bir xil manba siyosati" (bir xil kelib chiqadigan siyosat) deyiladi. Ushbu cheklovni chetlab o'tish uchun ikkala sahifa ham bunga rozi bo'lishi va maxsus ma'lumot almashadigan JavaScript kodini o'z ichiga olishi kerak.

Ushbu cheklov yana foydalanuvchi xavfsizligi uchun zarur. Sahifa https://anysite.com, foydalanuvchi ochilgan, URL bilan boshqa brauzer yorlig'iga kirish kerak emas https://gmail.com va u erdan ma'lumot o'g'irlash.

JavaScript-ni joriy sahifa kelgan server bilan osongina muloqot qilishi mumkin. Ammo boshqa saytlardan/domenlardan ma'lumotlarni olish qobiliyati cheklangan. Bu printsipial jihatdan mumkin bo'lsa-da, uzoq tomon bilan aniq rozilikni (HTTP sarlavhalarida ifodalangan) talab qiladi. Shunga qaramay, bu xavfsizlikning cheklanishidir.



Rasm-1.1 Javascript kamchiliklari

Nima uchun JAVASCRIPT ajoyib til

Javascriptning minimum 3 ta asosiy kuchli taraflari mavjud

HTML/CSS bilan to’la integratsiya

Oddiy narsalar oddiy amalga oshiriladi

Barcha asosiy brauzerlar tomonidan qo'llab-quvvatlanadi va sukut bo'yicha yoqilgan.

JavaScript-bu uchta narsani birlashtiradigan yagona brauzer texnologiyasi.

Bu JavaScript-ni maxsus va ko’rkam qiladi. Shuning uchun brauzerda interfeyslarni yaratish uchun eng keng tarqalgan vosita hisoblanadi.

Albatta, JavaScript-ni nafaqat brauzerlarda, balki serverda, mobil qurilmalarda va hokazolarda ham ilovalar qilish imkonini beradi. ( bu haqda keyinroq gaplashamiz)

Agar JavaScript brauzerdan tashqarida ishlatilsa, masalan, serverda bunday cheklovlar qo'llanilmaydi. Zamonaviy brauzerlar plaginlar/kengaytmalarni taqdim etadi, ular orqali qo'shimcha ruxsatnomalar talab qilinishi mumkin.



Rasm-2 (HTML web sahifa skeletei, css web sahifa stili , javascript esa web sahifani jonlantiradi)

JAVASCRIPT ustida qurilgan tillar

JavaScript sintaksisi barcha ehtiyojlar uchun mos emas. Turli odamlar turli imkoniyatlarga ega bo'lishni xohlashadi.

Bu tabiiydir, chunki loyihalar boshqacha va ular uchun talablar ham boshqacha.

Shunday qilib, so'nggi paytlarda brauzerda ishga tushirilgunga qadar JavaScript tiliga konvertatsiya qilinadigan ko'plab yangi tillar paydo bo'ldi.

Zamonaviy vositalar transpilatsiyani juda tez va oshkora qiladi, aslida ishlab chiquvchilarga kodni boshqa tilda yozishga imkon beradi, uni avtomatik ravishda JavaScript-ni "qopqoq ostida"o'zgartiradi.

Bunday tillarning namunalari:

**CoffeeScript** JavaScript uchun "sintaktik shakar" qo'shadi. U toza va ixcham kodni yozishga imkon beruvchi qisqa sintaksisni taqdim etadi. Odatda Ruby dasturchilar kabi.

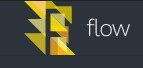


Rasm-3 CoffeScript logotipi

**TypeScript** katta va murakkab tizimlarni ishlab chiqish va qo'llab-quvvatlashni soddalashtirish uchun "qattiq yozish" ni qo'shishga qaratilgan. Microsoft Tomonidan Ishlab Chiqilgan.

Rasm-5 Rasm-4 TypeScript logotipi

**Flow** shuningdek, yozishni qo'shadi, lekin boshqacha. Facebook Tomonidan Ishlab Chiqilgan.



Rasm-5 Rasm-4 TypeScript logotipi

**Dart** alohida turadi, chunki u brauzerdan tashqarida ishlaydigan o'z dvigateliga ega (masalan, mobil ilovalarda). Dastlab Google tomonidan JavaScript-ni almashtirish sifatida taklif qilingan, ammo hozirgi vaqtda uning efirga uzatilishi yuqoridagi tillar bilan bir xil bo'lishi kerak.



Rasm-6 Dart logotip

**Bryton** Pythonni JavaScript-ni tarjima qiladi, bu esa JavaScript-ni holda sof Pythonda ilovalarni yozishga imkon beradi.

Rasm-7 Brython logotipi

Boshqalar ham bor. Ammo, agar biz ushbu tillardan birini ishlatsak ham, nima qilayotganimizni tushunish uchun JavaScript-ni bilishimiz kerak.

Shunday qilib:

JavaScript dastlab faqat brauzer uchun yaratilgan, ammo hozir ko'plab boshqa platformalarda ishlatilgan.

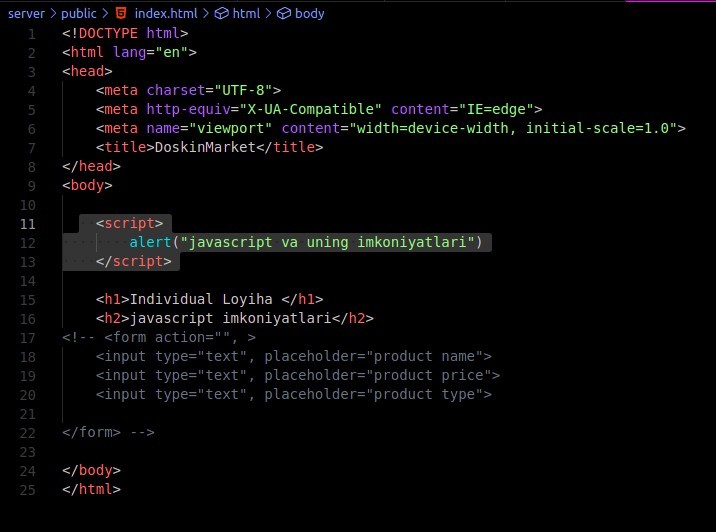
Bugungi kunda JavaScript HTML / CSS bilan to'liq integratsiyaga ega bo'lgan brauzer uchun eng keng tarqalgan til sifatida noyob mavqega ega.

Ko'pgina tillar qo'shimcha funktsiyalarni taqdim etish uchun JavaScript-da "tarjima qilingan" bo'lishi mumkin. JavaScript-ni o'zlashtirganidan keyin ularni kamida qisqacha ko'rib chiqish tavsiya etiladi.

# BOB. NAZARIY QISM

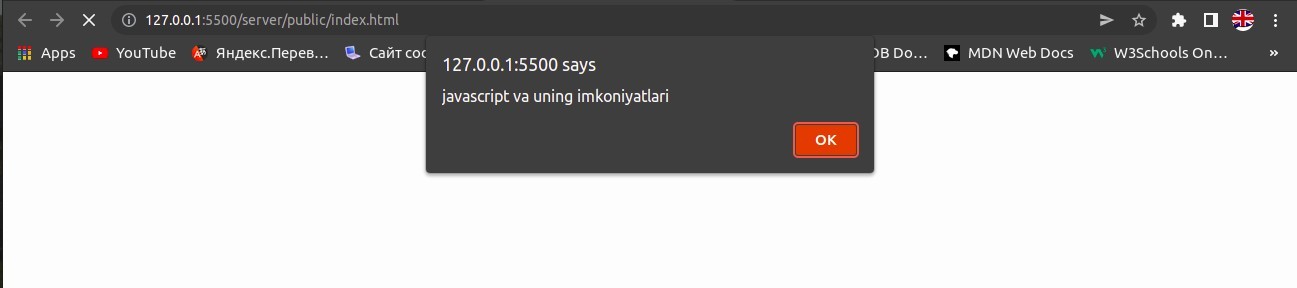
# 2.1 Javascriptda barcha obyektlar bilan ishlash

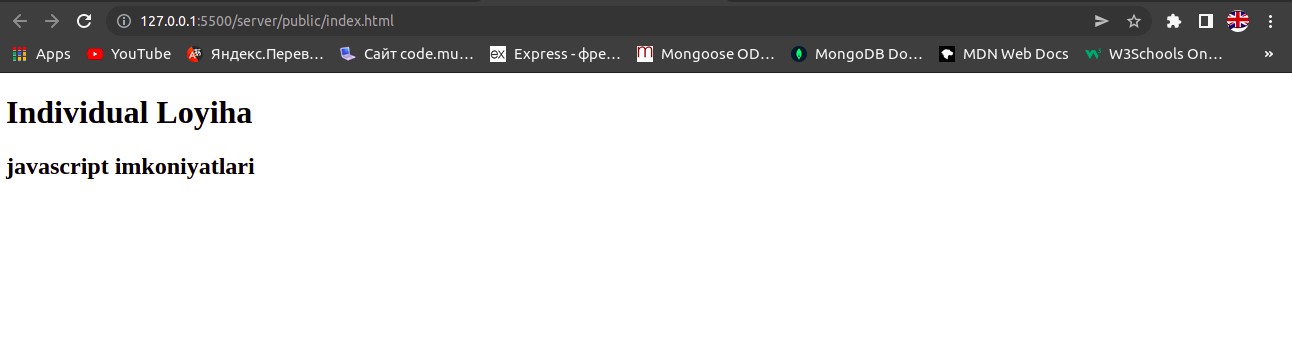
Javascriptni ulash uchun biz script tegidan foydalanamiza



Rasm-10 Javascript kodida alert ni bajaryapmiz

Biz hozir script tegini ichida alert komandasi yordamida “Javascript va uning imkoniyatlari “ degan gapni , paydo bo’ladigan oynaga chiqaramiza:

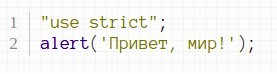


Ok tugmasini bosgandan keyin bizning HTML teglarimiz ishga tushdi

Qattiq rejim (Строгий режим ) — "use strict"

Uzoq vaqt davomida JavaScript-ni qayta muvofiqligi bilan muammosiz ishlab chiqdi. Oldingi funksionallik o'zgarmagan bo'lsa-da, tilga yangi xususiyatlar qo'shildi.

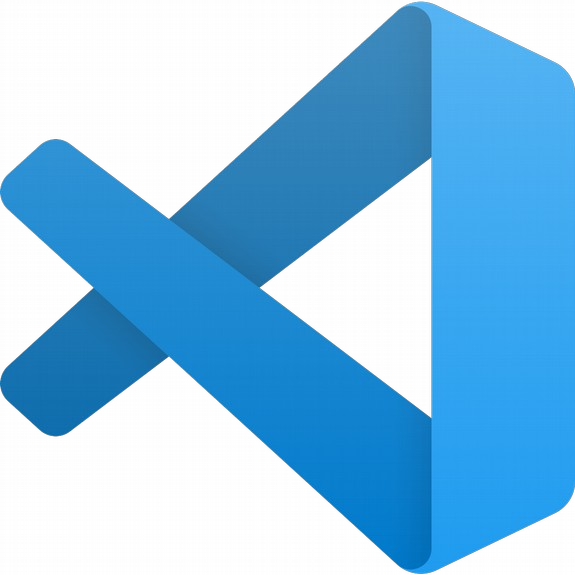
Ushbu yondashuvning afzalligi shundaki, mavjud kod ishlashni davom ettirdi. Va ahvolga tushgan-JavaScript yaratuvchilari tomonidan qabul qilingan har qanday xato yoki nomukammal qaror abadiy tilida qolib ketdi.

Shunday qilib, ECMAScript 5 (ES5) paydo bo'lgan 2009 yilgadan keyin o’zgarish bo’ldi. U tilga yangi imkoniyatlar qo'shdi va mavjud bo'lgan funksionallarni ayrimlarini o'zgartirdi. Eski kodni ishlatish uchun, avvalgidek, bunday o'zgarishlar sukut bo'yicha qo'llanilmaydi. Shuning uchun biz ularni maxsus direktiv bilan aniq faollashtirishimiz kerak:"use strict".

Rasm-11 (salom dunyo)

JavaScript bilan ishlash uchun IDE

Albatta buning uchun bizga Redaktor koda kerak Kod muharriri Kod muharlaridan eng popularnilari bular VsCode, Atom, WebStrom, Sublime text editor .Bular juda qulay , lekin eng ommaboblari bular: VS Code bilan Web Strorm dir.



Rasm-11 VS CODE

**O'zgaruvchilar** dasturda ma'lumotlarni saqlash uchun ishlatiladi . O'zgaruvchilar ish paytida uning qiymatini o'zgartirishi mumkin bo'lgan vaqtinchalik ma'lumotlarni yoki ma'lumotlarni saqlash uchun mo'ljallangan.

### O'zgaruvchilarni e'lon qilish

**Var** va **let** iboralari o'zgaruvchilarni yaratish uchun ishlatiladi . Masalan, o'zgaruvchini e'lon qilaylik username:

|  |  |
| --- | --- |
| bitta | var username; |

O'zgaruvchilar deklaratsiyasi bitta bayonotni ifodalaydi, shuning uchun u nuqta-vergul bilan tugaydi.

**let** iborasi yordamida o'zgaruvchining o'xshash ta'rifi :

|  |  |
| --- | --- |
| bitta | let username; |

Har bir o'zgaruvchining nomi bor. Ism ixtiyoriy harf-raqamli belgilar to'plami, pastki chiziq (\_) yoki dollar belgisi ($) bo'lib, nomlar raqamli belgilar bilan boshlanmasligi kerak. Ya'ni, nomda harflar, raqamlar, pastki chiziqdan foydalanishimiz mumkin. Biroq, boshqa barcha belgilar taqiqlangan.

Masalan, to'g'ri o'zgaruvchilar nomlari:

|  |  |
| --- | --- |
| bitta  2  3  to'rtta  5 | $commision  someVariable  product\_Store  income2  myIncome\_from\_deposit |

Quyidagi nomlar yaroqsiz va ulardan foydalanish mumkin emas:

|  |  |
| --- | --- |
| bitta  2  3 | 222lol  @someVariable  my%percent |

Shuningdek, ajratilgan kalit so'zlarga mos keladigan o'zgaruvchilar nomlarini bermang. JavaScript-da kalit so'zlar ko'p emas, shuning uchun bu qoidaga amal qilish qiyin emas. Misol uchun, quyidagi nom noto'g'ri bo'ladi, chunki **for** JavaScript-da kalit so'zdir:

|  |  |
| --- | --- |
| bitta | var for; |

JavaScript-da zaxiralangan so'zlar ro'yxati:

kutish, sindirish, holat, tutish, sinf, const, davom etish, tuzatuvchi, sukut bo‘yicha, o‘chirish, bajarish, boshqa, enum, eksport, uzaytirish, noto‘g‘ri, nihoyat, uchun, funktsiya, agar, import, in, instanceof, yangi, null, qaytish, super, switch, this, throw, true, try, typeof, var, void, while, with, yield

JavaScript-ning rivojlanishi va yangi til standartlarining qabul qilinishi bilan zahiradagi so'zlar ro'yxatini to'ldirish mumkin, ammo umuman olganda, ularning soni unchalik ko'p emas.

O'zgaruvchilarni nomlashda shuni yodda tutingki, JavaScript katta- kichik harflarni sezgir tildir, ya'ni quyidagi kodda ikki xil o'zgaruvchi e'lon qilinadi **:**

|  |  |
| --- | --- |
| bitta  2 | var username;  var userName; |

Vergul bilan ajratilgan bir nechta o'zgaruvchilarni belgilashingiz mumkin:

|  |  |
| --- | --- |
| bitta  2 | var username, age, heigt;  let a, b, c; |

### O'zgaruvchiga qiymat belgilash

O'zgaruvchi aniqlangandan so'ng, unga qiymat berish mumkin. Buning uchun **tayinlash operatoridan** ( **=** ) foydalaning:

|  |  |
| --- | --- |
| bitta  2 | var username;  username = "Tom"; |

Ya'ni, bu holda, o'zgaruvchi username"Tom" qatorini saqlaydi. O'zgaruvchiga qiymat bergandan so'ng, biz ushbu qiymat bilan biror narsa qilishimiz mumkin, masalan, uni konsolga chop etishimiz mumkin:

<!DOCTYPE html>

<html>

<head>

    <meta charset="utf-8" />

    <title>METANIT.COM</title>

</head>

<body>

    <script>

        var username;           // Определение переменной username

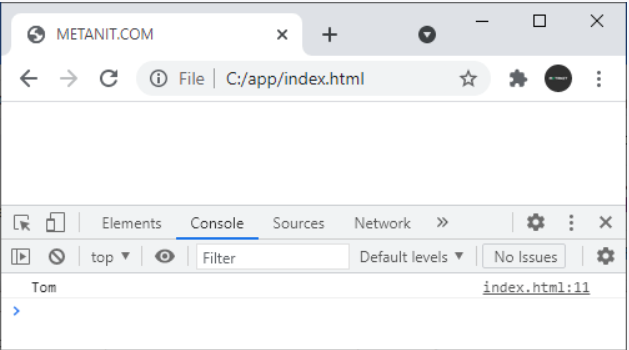
        username = "Tom";       // Присвоение переменной значения

        console.log(username);  // Вывод значения переменной username на консоль

    </script>

</body>

</html>



O'zgaruvchini aniqlaganingizda darhol unga qiymat berishingiz mumkin:

|  |  |
| --- | --- |
| bitta  2 | var username = "Tom";  let userage = 37; |

O'zgaruvchiga boshlang'ich qiymat berish jarayoni **ishga tushirish** deb ataladi .

O'zgaruvchilarning ajoyib tomoni shundaki, biz ularning qiymatini o'zgartirishimiz mumkin:

<!DOCTYPE html>

<html>

<head>

    <meta charset="utf-8" />

    <title>METANIT.COM</title>

</head>

<body>

    <script>

        var username = "Tom";

        console.log("username до изменения:", username);

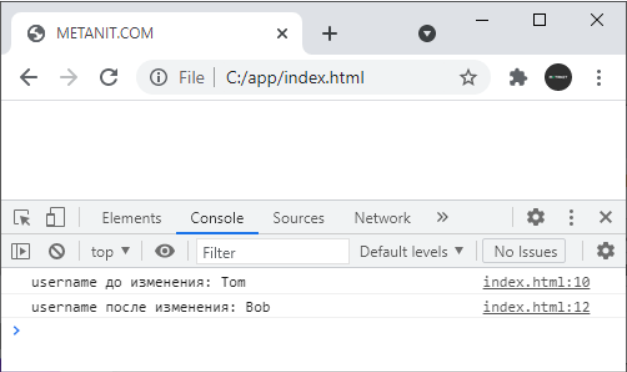
        username = "Bob";

        console.log("username после изменения:", username);

    </script>

</body>

</html>



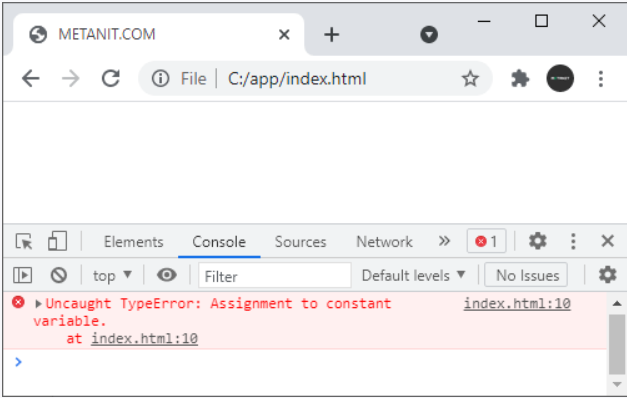
### Doimiylar

**const** kalit so'zi **o'zgaruvchi** kabi qiymatga ega bo'lgan doimiyni aniqlash uchun ishlatilishi mumkin , lekin qiymatni o'zgartirib bo'lmaydi.

|  |  |
| --- | --- |
| bitta | const username = "Tom"; |

Agar biz uning qiymatini o'zgartirishga harakat qilsak, xatoga duch kelamiz:

|  |  |
| --- | --- |
| bitta  2 | const username = "Tom";  username = "Bob";   // ошибка, username - константа, поэтому мы не можем изменить ее значение |



Shuni ham ta'kidlash joizki, biz doimiyning qiymatini o'zgartira olmaganimiz uchun uni initsializatsiya qilish kerak, ya'ni uni belgilashda biz uni boshlang'ich qiymat bilan ta'minlashimiz kerak. Agar buni qilmasak, yana xatoga duch kelamiz:

|  |  |
| --- | --- |
| bitta | const username; // ошибка, username не инициализирована |

Qachon o'zgaruvchilardan va qachon doimiylardan foydalanish kerak? Agar dasturni bajarish jarayonida qiymat o'zgarmasligiga ishonchingiz komil bo'lsa, u holda bu qiymat doimiy sifatida aniqlanadi. Agar qiymat o'zgarishi yoki o'zgarmasligi noma'lum bo'lsa, qiymatni doimiy sifatida belgilash tavsiya etiladi. Va agar siz uni keyinroq o'zgartirishingiz kerak bo'lsa, qiymat ta'rifini **const** dan **var** / **let** ga o'zgartirishingiz mumkin .

Javascriptda ishlatiladigan barcha ma'lumotlar ma'lum bir turga ega. JavaScript sakkizta ma'lumot turiga ega:

* **String** : satrni ifodalaydi
* **Raqam** : raqamli qiymatni bildiradi
* **BigInt** : juda katta butun sonlarni ifodalash uchun mo'ljallangan
* **Mantiqiy** : mantiqiy **to'g'ri** yoki **yolg'onni ifodalaydi**
* **Aniqlanmagan** : bitta maxsus qiymatni ifodalaydi - **aniqlanmagan** va qiymat o'rnatilmaganligini bildiradi
* **Null** : bitta maxsus qiymatni ifodalaydi - **null** va qiymat yo'qligini bildiradi
* **Belgi** : murakkab ob'ektlarning xususiyatlariga kirish uchun tez-tez ishlatiladigan noyob qiymatni ifodalaydi
* **Ob'ekt** : murakkab ob'ektni ifodalaydi

Birinchi yetti tur ibtidoiy ma'lumotlar turlarini ifodalaydi. Oxirgi turdagi **Ob'ekt** ibtidoiy turdagi yoki boshqa ob'ektlarning qiymatlaridan iborat murakkab, murakkab ma'lumotlar turini ifodalaydi. Asosiy ibtidoiy ma'lumotlar turlarini ko'rib chiqing.

### Raqamli ma'lumotlar

#### raqam

Raqam turi JavaScript-dagi **raqamlarni** ifodalaydi, ular butun son yoki kasr bo'lishi mumkin:

* 35 kabi butun sonlar. Biz ham musbat, ham manfiy sonlardan foydalanishimiz mumkin. Ishlatilgan raqamlar diapazoni: -2 53 dan 2 53 gacha
* Kasr raqamlari (suzuvchi nuqtali raqamlar), masalan, 3,5575. Shunga qaramay, ham ijobiy, ham salbiy raqamlardan foydalanish mumkin. Suzuvchi nuqta raqamlari bir xil diapazondan foydalanadi: -253 dan 253 gacha

Masalan:

|  |  |
| --- | --- |
|  | let x = 45;<font></font>  let y = 23.897;<font></font> |

Butun va kasr qismlar orasidagi ajratuvchi sifatida, boshqa dasturlash tillarida bo'lgani kabi, nuqta ishlatiladi.

#### BigInt turi

BigInt turi so'nggi JavaScript standartlarida raqamlar turi doirasidan tashqarida bo'lgan juda katta butun sonlarni ifodalash **uchun qo'shilgan.**Bu biz katta raqamlar bilan raqam turi bilan umuman ishlay olmaymiz degani emas, lekin raqam turida ular bilan ishlash muammolarga to'la bo'ladi. Kichik bir misolni ko'rib chiqing:

|  |  |
| --- | --- |
|  | let num = 9007199254740991<font></font>  console.log(num);       // 9007199254740991<font></font>  console.log(num + 1);   // 9007199254740992<font></font>  console.log(num + 2);   // 9007199254740992<font></font> |

Bu erda num o'zgaruvchisiga maksimal qiymat beriladi. Va keyin biz unga ba'zi qiymatlarni qo'shamiz va natijani konsolda ko'rsatamiz. Va natijalar bizni chalkashtirib yuborishi mumkin, ayniqsa 2 raqamini qo'shganda.

Raqamni turning qiymati sifatida belgilash uchun raqam oxiriga **n**BigInt qo'shimchasi qo'shiladi :

|  |  |
| --- | --- |
|  | let dimension = 19007n;<font></font>  const value = 2545n;<font></font> |

Masalan, oldingi misoldagi raqam turini bigintga o'zgartiramiz:

|  |  |
| --- | --- |
| 5 | let num = 9007199254740991n<font></font>  console.log(num);       // 9007199254740991n<font></font>  console.log(num + 1n);  // 9007199254740992n<font></font>  console.log(num + 2n);  // 9007199254740993n<font></font>  console.log(num + 3n);  // 9007199254740994n<font></font> |

### Boolean turi

Mantiqiy turi mantiqiy yoki mantiqiy qiymatlarni **rost** (to'g'ri) va **noto'g'ri** (noto'g'ri) ifodalaydi:

|  |  |
| --- | --- |
|  | let isAlive = true;<font></font>  let isDead = false;<font></font> |

### String

Turi Stringsatrlarni ifodalaydi. Tirnoqlar satrlarni aniqlash uchun ishlatiladi va siz ikkita, bitta va slash tirnoqlardan foydalanishingiz mumkin. Yagona cheklov shundan iboratki, yopilish taklifining turi ochilish kotirovkasining turi bilan bir xil bo'lishi kerak, ya'ni ikkala juft yoki ikkalasi ham bitta.

|  |  |
| --- | --- |
|  | let user = "Tom";<font></font>  let company = 'Microsoft';<font></font>  let language = `JavaScript`;<font></font>  <font></font>  console.log(user);<font></font>  console.log(company);<font></font>  console.log(language);<font></font> |

Agar satr ichida qo'shtirnoq mavjud bo'lsa, biz ularni qiya chiziq bilan chetlab o'tishimiz kerak. Masalan, bizda matn bor deylik "Бюро "Рога и копыта"". Endi biz tirnoqlardan qochamiz:

|  |  |
| --- | --- |
|  | let company = "Бюро \"Рога и копыта\""; |

Satr ichida boshqa turdagi tirnoqlardan ham foydalanishimiz mumkin:

|  |  |
| --- | --- |
|  | let company1 = "Бюро 'Рога и копыта'";<font></font>  let company2 = 'Бюро "Рога и копыта"';<font></font> |

#### Interpolyatsiya

**To'g'ridan-to'g'ri tirnoq belgilaridan foydalanish bizga interpolyatsiya** kabi texnikani qo'llash imkonini beradi - ma'lumotlarni satrga joylashtirish. Masalan:

|  |  |
| --- | --- |
|  | let user = "Tom";<font></font>  let text = `Name: ${user}`;<font></font>  console.log(text);  // Name: Tom<font></font> |

Ifodaning qiymatlarini (masalan, boshqa o'zgaruvchilar va konstantalarning qiymatlari) satrga kiritish uchun dollar belgisi bilan ifodadan oldin **$** , keyin esa jingalak qavslar ichida ifoda qo'ying. Shunday qilib, yuqoridagi misolda ${user}, bu chiziqning ushbu nuqtasida siz o'zgaruvchining qiymatini kiritishingiz kerakligini anglatadi user.

Xuddi shunday, siz ko'proq ma'lumotlarni joylashtirishingiz mumkin:

|  |  |
| --- | --- |
|  | let user = "Tom";<font></font>  let age = 37;<font></font>  let isMarried = false;<font></font>  let text = `Name: ${user}   Age: ${age}   IsMarried: ${isMarried}`;<font></font>  console.log(text);  // Name: Tom   Age: 37   IsMarried: false<font></font> |

### null va aniqlanmagan

**aniqlanmagan** qiymat aniqlanmagan yoki o'rnatilmaganligini bildiradi. Misol uchun, biz o'zgaruvchini unga boshlang'ich qiymat bermasdan aniqlaganimizda, u aniqlanmagan turni ifodalaydi:

|  |  |
| --- | --- |
|  | let isAlive;<font></font>  console.log(isAlive); // выведет undefined<font></font> |

Nullni belgilash o'zgaruvchining qiymati yo'qligini bildiradi:

|  |  |
| --- | --- |
|  | let isAlive;<font></font>  console.log(isAlive); // undefined<font></font>  isAlive = null;<font></font>  console.log(isAlive); // null<font></font>  isAlive = undefined;  // снова установим тип undefined<font></font>  console.log(isAlive); // undefined<font></font> |

### ob'ekt

Tur objectmurakkab ob'ektni ifodalaydi. Ob'ektning eng oddiy ta'rifi jingalak qavslardir:

|  |  |
| --- | --- |
| bitta | let user = {}; |

Ob'ekt turli xil xususiyat va usullarga ega bo'lishi mumkin:

|  |  |
| --- | --- |
|  | const user = {name: "Tom", age:24};<font></font>  console.log(user.name);<font></font> |

Bunday holda ob'ekt foydalanuvchi deb ataladi va u ikkita xususiyatga ega: nom va yosh. Bu ob'ektlarning qisqacha tavsifi, biz keyingi maqolalarda ob'ektlarni batafsil ko'rib chiqamiz.

### Zaif yozish

JavaScript erkin yozilgan tildir. Bu o'zgaruvchilar turini dinamik ravishda o'zgartirishi mumkinligini anglatadi. Masalan:

|  |  |
| --- | --- |
|  | let id;  // тип undefined<font></font>  console.log(id);<font></font>  id = 45;  // тип number<font></font>  console.log(id);<font></font>  id = "45"; // тип string<font></font>  console.log(id);<font></font> |

Ikkinchi va uchinchi hollarda konsol 45 raqamini ko'rsatishiga qaramay, ikkinchi holatda o'zgaruvchi idraqamni, uchinchi holatda esa satrni ifodalaydi.

Bu e'tiborga olinishi kerak bo'lgan muhim nuqta va dasturdagi o'zgaruvchining xatti-harakati unga bog'liq:

|  |  |
| --- | --- |
|  | let a = 45;  // тип number<font></font>  let b = a + 5;<font></font>  console.log(b); // 50<font></font>      <font></font>  a = "45"; // тип string<font></font>  let c = a + 5<font></font>  console.log(c); // 455<font></font> |

Yuqorida, ikkala holatda ham ao'zgaruvchiga qo'shish (+) amali qo'llaniladi. Lekin birinchi holatda u araqamni ifodalaydi, shuning uchun operatsiya natijasi a + 550 raqami bo'ladi.

Ikkinchi holda, u asatrni ifodalaydi. Ammo satr va 5 raqami o'rtasida qo'shish operatsiyasi mumkin emas. Shuning uchun 5 raqami satrga aylanadi va satrni birlashtirish operatsiyasi amalga oshiriladi. Shuning uchun, ikkinchi holatda, ifodaning natijasi a + 5"455" drenaji bo'ladi.

### operator turi

**typeof** operatoridan foydalanib, o'zgaruvchining turini olishingiz mumkin:

let id;<font></font><font style="vertical-align: inherit;"><font style="vertical-align: inherit;">

console.log (id turi); </font><font style="vertical-align: inherit;">// aniqlanmagan</font></font><font></font>

<font></font><font style="vertical-align: inherit;"><font style="vertical-align: inherit;">

id = 45;</font></font><font></font><font style="vertical-align: inherit;"><font style="vertical-align: inherit;">

console.log (id turi); </font><font style="vertical-align: inherit;">// raqam</font></font><font></font>

<font></font><font style="vertical-align: inherit;"><font style="vertical-align: inherit;">

id = 45n;</font></font><font></font><font style="vertical-align: inherit;"><font style="vertical-align: inherit;">

console.log (id turi); </font><font style="vertical-align: inherit;">// bigint</font></font><font></font>

<font></font><font style="vertical-align: inherit;"><font style="vertical-align: inherit;">

id = "45";</font></font><font></font><font style="vertical-align: inherit;"><font style="vertical-align: inherit;">

console.log (id turi); </font><font style="vertical-align: inherit;">// qator</font></font><font></font>

**NODEJS HAQIDA**

JAVASCRIPT da umuman olganda imkoniyatlar juda ko’p Misol uchun Browser bilan ishlash , web sahifaga o’zgartirish kiritish, uni qayta ishlash.

Oldinlari javascript fayllar bilan ishlash, server yaratish imkoniyati yoq edi , lekin

Node.js — veb-brauzerdan tashqarida JavaScript kodini bajaradigan ochiq manbali, kross-platformali, JavaScript runtime muhiti (inglizcha: JavaScript runtime environment). Node.js 2009-yil 27-mayda Rayan Dal tomonidan yartilingan . C++, C va JavaScriptda yozilgan boʻlib, asosan server uchun foydalaniladi



Rasm-15 Rayan Dal NodeJs asoschisi

1996 yilda Netscape kompaniyasi Server JavaScript-ni (Server — yon JavaScript-SSJS) yaratishga urinishgan, ammo texnologiya tarqalmagan.

Rayan dahl, Node yaratuvchisi.js

Node.JS Server veb-komponentlarini yaratish bo'yicha ikki yillik tajribadan so'ng 2009 yilda Ryan Dal tomonidan ishlab chiqilgan. Tadqiqot davomida, u an'anaviy parallelizm modeli o'rniga, oqimlarga asoslangan holda, voqea yo'naltirilgan tizimlarga murojaat qilish kerak degan xulosaga keldi. Ushbu model soddaligi, past yuk ("har bir ulanish uchun bitta oqim" mafkurasi bilan solishtirganda) va tezligi tufayli tanlangan.

Node ning maqsadi "keng ko'lamli tarmoq serverlarini qurishning oddiy usuli"ni taklif qilishdir. Node Ishlab Chiqish.js Joyent tomonidan homiylik qilinadi

# EXPRESS NODEJS NI FREYMWORKI HAQIDA

Express, NodeJS serverining ochilishini sezilarli darajada

soddalashtirish, sayt tartibini qismlarga ajratish, shablonlashtirish, yo'l-yo'riq berish imkonini beruvchi freymwork hisoblanadi. Odatda, hozirgi vaqtda Node JS bilan ishlash Express orqali amalga oshiriladi. Keling buni ko'rib chiqamiza



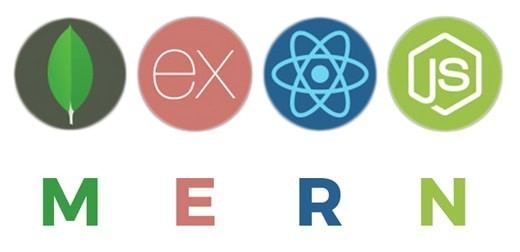
Rasm-16 nodejs va express logotiplari

# MERN- STACK

Umuman hozirgi kunda shunday terminlarni uchratishimiz mumkin , MERN STACK, MEAN STACK bu nima degani MERN stack bu quyidagi texnologiyalarning boshlang’ich nomlaridan olingan stek hisoblanadi, ya’ni

**M**- MongoDb (ma’lumotar ombori NoSQL)

**E**- Expressjs (nodejs kutubxona) **R**- Reactjs (Javascript kutubxona) **N** - NodeJS (Javascript platforma)



Rasm-17 MERN STACK

Hozirgi kunda mern stack yordamida biz katta saytlarni, web Platformalarni , CRM sistemalarni, va hk larni yaratishimiz mumkin Keling endi bular haqida batafsil gaplashamiza



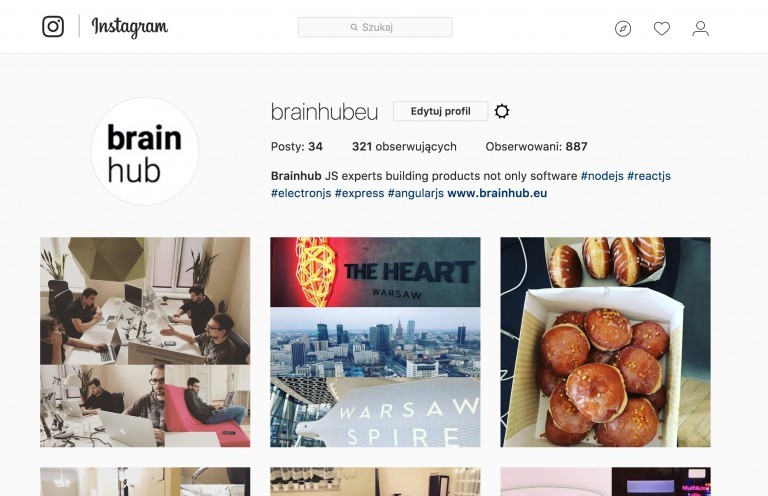
Rasm-18 tavsiya qiladigan kitoblarimdan biri

**REACT JS**

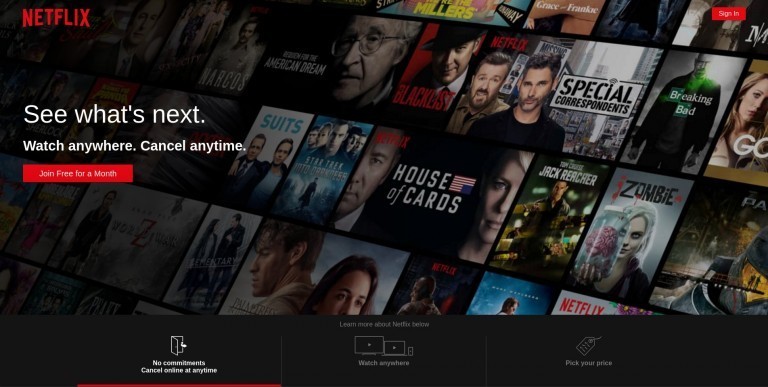
React js nima - JavaScript-maxsus interfeyslarni yaratish uchun kutubxona , NodeJs bizga javascript yordamiza saytning backend qismini yani server qismini yaratish uchun yordam bergan bo’lsa React bizga saytning klient qismini yaratish uchun juda katta imkoniyatlarni beradi

Nega aynan React , hozirgi kunda reactdan tashqari Angular, Vue degan kutubxonalar ham mavjud , reakt bularning eng mashxuri hisoblanadi

Chunki quyidagi proyektlar aynan reaktda qilingan Facebook, Instagram , Netflix, New York Times va boshqalar, bular web saytlar hisoblanadi



Rasm – 19 instagram web sayti



Rasm – 20 netflix web sayti

# MERN STACK DA API CHIQARAMIZ

Buning uchun bizga kerakli narsalar :

NODEJS

NPM

EXPRESSJS

POSTMAN,

MONGODB,

KERAKLI INSTRUMENTLAR

POSTMAN BU so’rovlarni amalga oshirish uchun kerak bo’ladigan dastur hisoblanadi , juda qulay tarzda so’rovlar yuborishimiz mumkin,

MongoDB – bu esa NOSQL turidagi ma’lumotlar ombori hisoblanadi,

Hozirgi proyektda men mongoDB ga ulanishni o’rgatmayman , balki chunki manda o’rnatilib bo;lingan , internetdan qanday qilib o’rnatish kerakigi va qanday ishlatish kerakligi haqida ma’lumotlar juda ko’p , usha yerdan o’rganib olasiz degan umiddaman.

Hozirgi proyekt nima haqida bo’ladi:

Bu proyekt internet magazin hisoblanib unda biz ma’lumotlar omboriga (MB) yangi turdagi mahsulotlarni qo’shamiza

MB ni mahsulotlarga to’ldiramiza , va aynan shu mahsulotlarimiz bilan ishlash uchun har xil API larni yozamiz , Bular asosan REST API deb nomlanadi , va biz bu dasturimizda CRUD amallarini bajaramiza

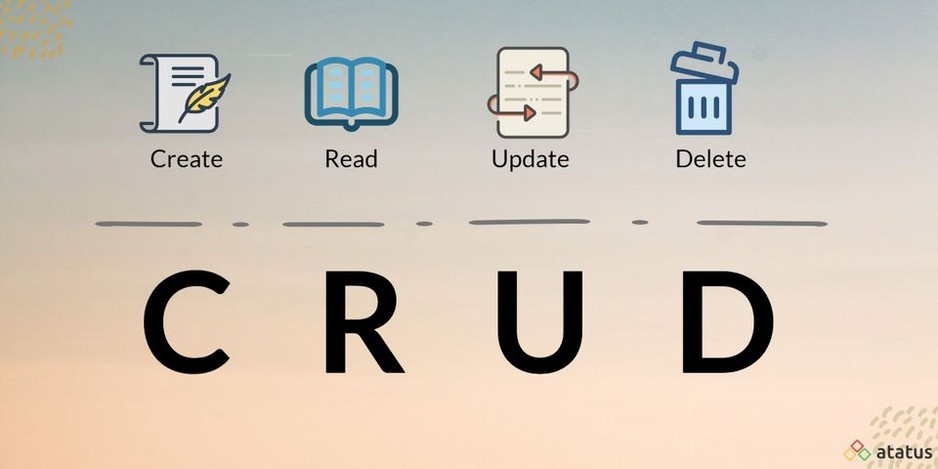
CRUD bu:

C: CREATE (MB ga yangi narsa qo’shish)

R: READ (MB dan yaratilgan narsani o’qish ko’rish) U: UPDATE (Mbdagi yaratilgan narsani o’zgartirish, uni yangilash)

D: DELETE (MB dagi narsani o’chirish, olib tashlash )

Uchun hizmat qiladigan amallar to’plami hisoblanadi Hozir shulardan ba’zilarini ko’rib o’tamiza



Rasm – 22 crud operatsiyalari

VUEJS FRAMEWORKI



Yuqorida biz ReactJS haqida gaplashgan edik , hozir esa biz VUEJS haqida uning imkoniyatlari nima ekanligini bilib o’tamiz

Evan You tomonidan yaratilgan Vue.js - foydalanuvchi interfeyslari (interfeyslari) va bitta sahifali dasturlarni yaratish uchun ochiq manbali progressiv JavaScript ramkasi; u odatda Vue deb nomlanadi. Ushbu framework ishlab chiquvchilarga foydalanuvchi interfeyslarini (UI) bosqichma-bosqich yaratishga imkon beradigan "yuqori ajratish" dan foydalanadi.

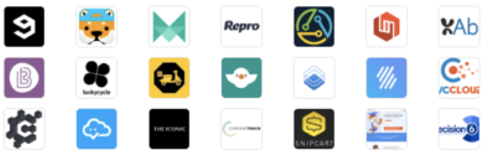
Frameworkdan foydalangan holda kutubxonani modullashtirish frontend rivojlanishida keng tarqalgan. React va Angular modulizatsiyaga ega. Ammo Vue.js-ni boshqa alternativalardan ajratib turadigan narsa - bu "yuqori ajratish", funktsional imkoniyatlarni kengaytirish qanchalik oson va modullar kiritilgandan so'ng barcha qismlarning ishlash darajasi. Masalan, biz kichik vizual komponentlarni tashkil qilishni va ko'rsatishni istasak, bizga Vue.js-ning "asosiy" kutubxonasi kerak; qo'shimcha kutubxonalarni kiritish kerak emas. Ilova o'sib borishi bilan bizda 'vue-router' kabi marshrutlarni boshqarish uchun kutubxonalar, 'vuex' kabi global holatni boshqarish uchun kutubxonalar yoki 'bootstrap-vue' kabi javob beradigan veb-ilovalarni yaratish uchun kutubxonalar mavjud. Bundan tashqari, agar bizning dasturimiz optimallashtirilishi kerak yoki yaxshi SEO kerak bo'lsa, biz "vue-server-rendering" kutubxonasini o'z ichiga olamiz.

Vue.js komponentlari tizimi reaktiv, ya'ni Vue.js asenkron hodisalar orqali qanday aloqa qilishni biladi; masalan, bola komponenti ota-ona komponenti bilan voqealar orqali bog'lanishi mumkin. Vue.js-da boshqa kutubxonalar yoki manbalar bilan ishqalanish bo'lmaydi, boshqacha qilib aytganda, biz eng qulay bo'lgan vositadan foydalanishimiz mumkin. Masalan, biz faqat HTML va JavaScript-ni yozishimiz mumkin yoki agar xohlasak CSS, JSX yoki TypeScript-ni qo'shishimiz mumkin.

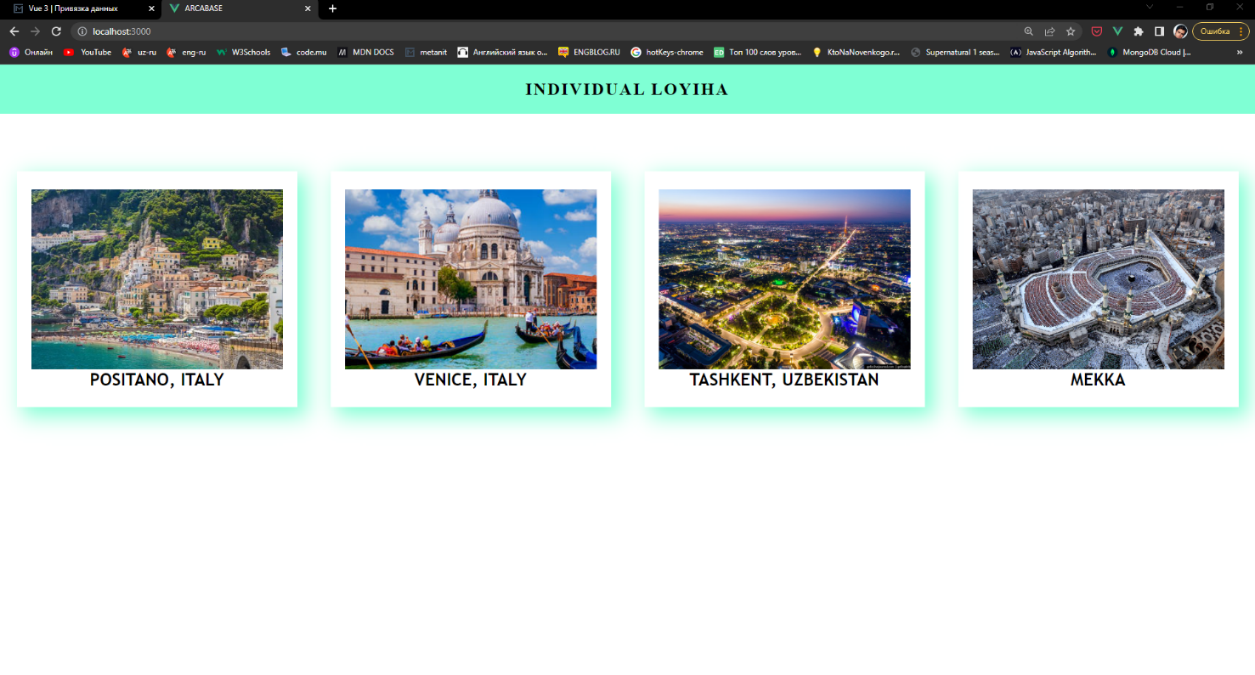
Vue.js-da JS tugunida yaratilgan maxsus buyruq satri (CLI) mavjud. Ushbu vosita bizga qozon plitasi (yoki asosiy shablon) yordamida loyihani boshlashga imkon beradi. Bundan tashqari, Vue.js ishlab chiquvchi guruhi bizning kengaytmamiz daraxtining qanday ko'rsatilishini, voqealar qanday boshlanishi va yozilishini, har bir komponentning ichki holati qanday saqlanishini va komponentning global holatini ko'rishimizga imkon beradigan Chrome kengaytmasini saqlaydi. o'zini tutmoqda.

VUEJS dan kimlar foydalanadi

700 dan ortiq kompaniyalar Vue.js.-dan foydalanmoqda. Ulardan ba'zilari: Xiaomi, Alibaba va Gitlab.



Proyekt turizm



# XULOSA

Shunday qilib aziz do’stlar biz bugun javascript va uning imkoniyatlari degan loyihani ko’rib chiqdik , bunda biz ko’p ma’lumotlarga ega bo’ldik , lekin bu ma’lumotlar , dengizning bir qismi holos , hali ma’lumotlar juda ko’p , shu bilan birga bizni ishimizni osonlashtiradigan texnologiyalar ham juda ko’p, Javascript yilla davomida rivojlanib keldi , hozirgi kunda biron-bir Web saytni javascriptsiz tasavvur ham qilib bo’lmaydi, hozirgi kunda javascript ko’plab sohalarni qamrab oldi,

Shu jumladan server hosil qilish, (NodeJS ExpressJS) , bu haqda ozgina gaplashdik, web saytning frontend qismi uchun kutubxonalar (React, Angular, Vue) , Javascriptda hozirgi kunda Kompyuterlarga dastular tuzish imkoniyati ham bor u ElectronJS orqali amalga oshadi ( siz biladigan VS Code kod muharriri , aynan ElectronJS da yozilgan , undan tashqari Javascript mobil dasturlar tuzish sohasida ham sobitqadamlardan hisoblanadi, mobil telefonlarga dastur yozish uchun javascriptda ReactNative freymworkidan foydalaniladi, qo’lingizdagi telefondagi instagram Ijtimoiy tarmoqi ham aynan ReactNative yordamida yozilgan , Undan tashqari Javascript yordamida telegram uchun botlar tuzish mumkin bu uchun esa TelegrafJS yordamga keladi,

Zamon rivojlangani sayin Javascript Machine Learning sohasiga ham kirib keldi bu uchun esa TensorFlow.Js yordamga keladi, Ko’rib turibsiz Javascript dasturchining yo’llari juda ko’p , asosiysi esa bularni yaxshilab o’rganish zarur. Katta raxmat.

# FOYDALANILGAN RESURSLAR RO’YXATI

1. [http://code.mu](http://code.mu/)
2. https:[//ww](http://www.tensorflow.org/js)w[.t](http://www.tensorflow.org/js)e[nsorflow.org/js](http://www.tensorflow.org/js)
3. https://developer.mozilla.org/en-US/
4. https://expressjs.com/ru/
5. https://translate.yandex.ru/
6. https://ru.reactjs.org/
7. https://learn.javascript.ru/
8. https:[//ww](http://www.pngrepo.com/)w[.pngrepo.com/](http://www.pngrepo.com/)
9. https://docs.mongodb.com/
10. https://[www.google.com/](http://www.google.com/)