**12-mavzu. Java Script tilida ob`еktlar.**

**Reja:**

1. JavaScriptda sinflar..
2. Number sinfi. Sonlar bilan ishlash.
3. String sinfi. Qatorlarni qayta ishlash
4. Array sinfi. Massivlar bilan ishlash va ularni saralash.

|  |  |
| --- | --- |
| **importance_of_education_1** | **O`quv modullari**  *sinf, hususiyat, metod, global sinf, number sinfi, string sinfi, date sinfi, array sinfi, assosiativ massiv, math sinfi* |

*JavaScriptda sinflar.*

**Sinf** — bu o’zgaruvchi va funksiyalarni o’z ichiga qamrab oluvchi bu o’zgaruvcvhilarni boshqarish uchun obyekt tipidir. O’zgaruvchi *hususiyat,* funksiya esa *metod* deb nomlanadi.

Sinflarning metod va hususiyatlaridan foydalanishda sinf nusxasini yaratish zarur. Buning uchun new operatoridan foydalaniladi, keyin nusxa ma’lumotlariga tegishli uning sinf nomi ko’rsatiladi. Sinf nomidan so’ng aylanma qavs ichida bir necha parametrlarni berish mumkin. Umumiy holda quyidagicha yoziladi:

<Sinf nusxasi> = new <Sinf nomi> ([<parametrlar>]);

Sinf nusxasini yaratishda havola (ko’rsatkich) o’zgaruvchida saqlanadi. Foydalanilayotgan havola yaratilgan sinf nusxasi hususiyat va metodlariga murojat qilishi mumkin.

Hususiyatga murojaat quyidagicha amalga oshiriladi:

<Sinf nusxasi>.<Hususiyat nomi>;

Uslubga murojaat ham shunga o’xshash amalga oshiriladi, faqat uslub nomidan keyin aylanma qavs ko’rsatiladi:

<Sinf nusxasi>.<Uslub nomi>();

Qavs ichida ko’pincha uslub parametric ko’rsatiladi.

Global sinfi. Global sinfi hususiyat va uslubidan foydalanish sinf nusxasini yaratishni talab qilmaydi. Berilgan sinfning hususiyat va uslublari JavaScript qurilgan fuksiyalari bo’ladi.

Hususiyat:

* NaN qiymati NaN dan iborat (Not a Number, son emas):

var x = NaN;

* Infinity "plyus cheksizlik" qiymatini qaytaradi:

var x = Infinity;

Uslublar:

* parseInt(<Qator>, [<Asos>]) qatorni asosda ko’rsatilgan sanoq sistemasidagi songa o’zgartiradi. Agar asos berilmasa o’nlik sanoq sistemasidagi songa o’tkaziladi. Agar qatorni songa o’zgartiora olmasa NaN qiymatiga ega bo’ladi. Masalan:

*var Number1 = 15; var Str = "5"; var Str5 = "FF";*

*var Str2 = Number1 - parseInt(Str); // O’zgaruvchi qiymati 10 ga teng*

*var Str3 = Number1 - parseInt(Str5,16); // O’zgaruvchi qiymati -240 ga teng*

*var Str4 = Number1 + parselnt(Str); // O’zgaruvchi qiymati 20 soniga teng*

* parseFloat(<qator>) qatorni suzuvchi vergulli songa o’zgartiradi:

*var Str = "5.2";*

*var Str2 = parseFloat(Str); // O’zgaruvchi qiymati 5.2 soniga teng*

* eval(<qator>) qatorli ifodani JavaScriptdagi oddiy ifoda kabi hisoblaydi:

*var Str = "3 + 5";*

*var Str2 = eval(Str); // O’zgaruvchi 8 sonini o’ziga oladi*

* isNaN(<Ifoda>) ifodani songa tekshiradi. Agar ifoda NaN ga teng bo’lsa true, ifoda qiymati son bo’lsa false qiymatni qaytaradi;
* isFinite(<ifoda>) yakunlovchi sonni tekshiradi. True yoki false qiymat qaytaradi;
* escape(<qator>) qatorni o’n oltilk sistema kodida kodlaydi:

*var Str = escape(" Салом");*

*// Str = %20%u0421%u0430%u043B%u043E%u043C*

* unescape(<qator>) escape() ga teskari uslub:

*var Str = unescape("%20%u0421%u0430%u043B%u043E%u043C"); // Str = Салом*

Izox. ***escape() va unescape() funksiyalari eskirgan. Ularning o’rniga encodeURI() va decodeURI() yoki encodeURIComponent() va decodeURIComponent() dan foydalaniladi.***

* encodeURI(URL-manzil) butun URL-manzil kodlanadi:

*var Str = "test.php?id=5&n=Alisher"; window.alert(encodeURI(Str));*

*//test.php?id=5&n=%D0%90%D0%BB%D0%B8%D1%88%D0%B5%D1%80*

* decodeURI(<Qator>) qator koddan ochiladi, encodeURI() uslubi bilan kod;
* encodeURIComponent(<qator>) qatorni URL-kodlash bajariladi:

*var Str = encodeURIComponent("Qator");*

*//Str = %D0%A1%D1%82%D1%80%D0%BE%D0%BA%D0%B0*

encodeURI() funksiyasidan farqi o’n oltilikdagi barcha maxsus belgilarni o’zgartiradi:

*var Str = "test.php?name=Aliser"; window.alert(encodeURIComponent(Str));*

*//test.php%3Fid%3D5%26n%3D%D0%90%D0%BB%D0%B8%D1%88%D0%B5%D1%80*

Number sinfi. *Sonlar bilan ishlash.* Number sinfi sonli o’zgaruvchi va o’zgarmaslarni ifodalashada ishlatiladi. Sinf nusxasi quyidagi ko’rinishda yaratiladi:

<Sinf nusxasi> = new Number (<Boshlang’ich qiymat>);

Number sinfi hususiyati sinf nuxsxasini yaratmay foydalanish mumkin:

* max\_value — JavaScript da ruxsat etilgan maksimal son:

*var x = Number.MAX\_VALUE; // 1.7976931348623157e+308*

* min\_value — JavaScript da ruxsat etilgan minimal son:

*var x = Number.MIN\_VALUE; // 5e-324*

* NaN —NaN qiymati:

*var x = Number.NaN; // NaN*

* negative\_infinity —"minus cheksizlik":

*var x = Number.NEGATIVE\_INFINITY; // -Infinity*

* positive\_infinity —"plyus cheksizlik":

*var x = Number.POSITIVE\_INFINITY; //Infinity*

***Metodlar:***

* valueOf() sinf nusxasining sonli qiymatini qaytaradi:

*var x = new Number(15);*

*var y = x.valueOf(); // 15*

*document.write(typeof y); // number*

* tostring() sonni qatorli ko’rinishda ifodalaydi:

*var x = new Number (15);*

*var Str = x.toString(); // "15"*

*document.write(typeof Str); // string*

String sinfi. *Qatorlarni qayta ishlash.* String sinfi qatorlarni qayta ishlash uslublariga yo’l ochib beruvchi sinf. Sinf nusxasi quyidagi ko'rinishda yaratiladi:

*<Sinf nusxasi> = new String (<Qator>);*

Siz qatorlar qo’shtirnoq yoki apostrof orqali ifoda qilinishini yaxshi bilasiz:

*var Str1 = "Qator 1";*

*var Str2 = 'Qator 2';*

Bu usul bilan xosil qilingan qatorlar string ma’lumotlar tipiga tegishli bo’ladi, string sinfi nusxasini yaratishda esa ma’lumotlar tipi object tipiga tegishli bo’ladi:

*var Str1 = "Qator 1";*

*var Str2 = 'Qator 2';*

*var Str3 = new String ("Qator 3");*

*document.write(typeof Str1); // string*

*document.write(typeof Str2); // string*

*document.write(typeof Str3); // object !*

Shunga qaramasdan oddiy qatorlarga string sinfi uslublarini qo’llash mumkin:

*var Str = "Qator".toUpperCase(); // belgilarni yuqori registrga o’tkazish*

*document.write(Str); // "QATOR"*

*document.write(typeof Str); // string*

toUpperCase() uslubidan foydalanishda qator string ma’lumotlar tipiga ega bo’ladi. Qator toUpperCase()uslubidan foydalanishda joriy string tipi avtomatik ravishda String sinfi nusxasiga aylantiriladi.

length hususiyati qator uzunligini aniqlaydi:

*var Str=new String ("Hello, world");*

*document.write(Str.length); // 12*

String sinfi ob’yektlarining uslublari quyidagilar:

* tostring() va valueOf() qatorli qiymat qaytaradi:

*var Str = new String ("Hello, world");*

*var Str2 = Str.toString();*

*document.write(Str2); // "Hello, world"*

*document.write(typeof Str); //object document.write(typeof Str2); // string*

* charAt(<belgi raqami>) ko’rsatilgan raqamdagi belgini chiqaradi. Qatorda belgilarni raqamlash noldan boshlanadi:

*var Str = "Hello, world";*

*document.write(Str.charAt(0)); // "Н"*

* charCodeAt(<belgi raqami>) qatordagi ko’rsatilgan parametrdagi belgi kodini chiqaradi. Qatorda belgilarni raqamlash noldan boshlanadi:

*var Str = "Hello, world"; window.alert(Str.charCodeAt(0)); // 72*

* fromCharCode(<кодl>,...,<кодN>) ko’rsatilgan kodlardan qator yasaydi:

*var S = String.fromCharCode(1055, 1088, 1080, 1074, 1077, 1090);*

*window.alert(S); // "Привет"*

* toLowerCase() qator belgilarini quyi regist belgilariga o’zgartiradi:

*var Str = "SALOM, DUNYO";*

*Str = Str.toLowerCase();*

*document.write(Str); // "salom, dunyo"*

* toUpperCase() qatordagi regist belgilarini yuqori registrli belgilarga o’zgartiradi:

*var Str = "salom, dunyo";*

*Str = Str.toUpperCase();*

*document.write(Str); // "SALOM, DUNYO"*

* substr(<boshlang’ich qism>, [<qism uzunligi>]) qatordan berilgan uzunlikdagi qismni chiqarib oladi. Agar ikkichi parametr berilmasa, qator oxirigacha barcha belgilar olinadi:

*var Str = "Salom, dunyo";*

*document.write(Str.substr(0, 5)); // "Salom" document.write(Str.substr(7)); // "dunyo"*

* substring(<Boshlang’ich qism>, <Oxirgi qism>) satrdan ko’rsatilgan pozisiyadagi satr ostini ajratib olish uchun ishlatiladi. Bu holda satrning boshlang’ich va oxirgi pozisiyasini ko’rsatiladi. Qismdagi oxirgi belgi kirmaydi:

*var Str = "Salom, dunyo";*

*document.write(Str.substring(6, 12)); // "dunyo"*

* indexOf(<Satr osti>, [<Qidirishning boshlang’ich pozisiyasi>]) satrdan satr ostining birinchi uchraydigan pozisiyasi, agar ikkinchi parametr berilmasa, qidirish satr boshidan boshlanadi. Agar satr osti umuman uchramasa, -1 qiymatni qaytaradi:

*var Str = "Salom, dunyo";*

*document.write(Str.indexOf("lo")); // 2 document.write(Str.indexOf("lo", 5)); //-1*

* lastIndexOf(<Satr osti>, [<Qidirishning boshlang’ich pozitsiyasi>]) joriy satrdan berilgan satr ostining oxirgi joylashgan pozitsiayasini aniqlaydi. Agar ikkinchi parametr berilmasa, qidirish satr boshidan boshlanadi. Agar satr osti topilmasa -1 qiymatni qaytaradi:

*var Str = "Salom, dunyo";*

*document.write(Str.lastIndexOf("o")); // 11*

* split(<bo’lgich>, [<Limit>]) bo’lgich simvol yordamida sartni satrli massivlarga ajratadi. Agar ikkinchi parametr ishtirok etmasa, natijaviy massiv maksimal elementlar soniga ega bo’ladi:

*var Str = "Salom, dunyo";*

*var Mass = Str.split(",");*

*document.write(Mass[0]); // "Salom" — massivning birinchi elementi*

*document.write(Mass[1]); // " dunyo" — massivning ikkinchi elementi*

* search(<Doimiy ifoda>) doimiy ifoda bilan mos keluvchi satr osti kirish pozitsiyasi tartibini aniqlaydi.

*Array sinfi. Massivlar bilan ishlash va ularni saralash.*

Massiv bu – o’zgaruvchilarning raqamlangan(tartiblangan) to’plamidir. Massivdagi o’zgaruvchilar uning elementi sifatida, ularni o’rni esa indekslar bilan belgilanadi. Massiv elemantlarning tartibi 0 dan boshlanadi. Buni yodda saqlash kerak. Umumi elementlar miqdori massivning ***o’lchami*** deb nomlanadi.

Massiv elementlari kvadrat qavs ichiga o’zaro vergul bilan ajratib yoziladi: mass1 = [1, 2, 3, 4];

Massiv elementining qiymatini biror o’zgaruvchiga o’zlashtirishda massiv nomidan keyin indeksi kvadrat qavs ichida yoziladi:

Str = mass1[0]; // *Str o’zgaruvchisiga massivning 0 chi elementi qiymati – 1 o’zlashtiradi;*

12.1-misolda massiv tuzish va uning elementlarini web-brauzer oynasida chop etish ko’rsatilgan.

***12.1-misol. Massivlar***

*<!DOCTYPE HTML>*

*<html>*

*<head>*

*<title>Massivlar</title>*

*<meta http-equiv="Content-Type" content="text/html; charset=utf-8">*

*</head>*

*<body>*

*<script type="text/javascript">*

*var Mass1, Mass2;*

*Mass1 = [1, 2, 3, 4];*

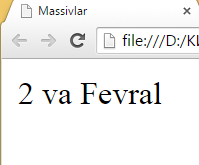
*Mass2 = ["", "Yanvar", "Fevral", "Mart", "Aprel", "May", "Iyun", "Iyul", "Avgus", "Sentabr", "Oktabr", "Nayabr", "Dekabr"];*

*document.write(Mass1[1] + " va " + Mass2[2]);*

*</script>*

*</body>*

*</html>*



*12.1-rasm*

Massivga yangi elementlar qo’shish yoki mavjudlarini o’zgartirish mumkin:

Mass1[5] = 6; Mass1[0] = 0;

Bunda massivning ikkita elementi yaratildi va mavjud bir elementinig qiymati o’zgartirildi. **Nimaga ikkita element yaratildi?** Misolimizda 4 elementdan iborat, oxirgi elementning tartibi (indeksi) 3 bo’lgan massiv mavjud edi. Biz 5-tartibdagi massivni yaratganimizda 4-element avtomatik yaratiladi va unga undefined (aniqlanmagan) qiymat o’zlashtiriladi.

Massivning ixtiyoriy elementiga boshqa massivni o’zlashtirish mumkin:

Mass1[0] = [1, 2, 3, 4];

Bunda holda massivning indeksi ikkita sondan iborat bo’ladi:

Str = Mass1[0][2]; // Str o’zgaruvchisiga 3 o’zlashtiriladi

O’zlashtirish amalida massivlarga bog’lanish saqlanishini xisobga olish kerak. Misol orqali ko’rib chiqalik.

var Mass1, Mass2;

Mass1 = [1, 2, 3, 4];

Mass2 = Mass1; // massivga bo’glanish!!! (Присваивается ссылка на массив!!!)

Mass2[0] = "Yangi qiymat";

document.write(Mass1.join(", ") + "<br>");

document.write(Mass2.join(", "));

***Array sinfi*** massiv yaratishga obyekt sifatida murojaat qilish va uning metodlariga murojaat qilish imkoniyatini yaratadi.

Sinf nusxasi quyidagi usulda yaratilishi mumkin:

*<Sinf nusxasi> = new Array (<Massiv elementlari soni>);*

*<Sinf nusxasi> = new Array (<Vergul bilan ajratilgan massiv elementlari>);*

Agar aylanma qavs ichida hech qanday parameter bo’lmasa, u holda massiv yaratilguncha nol uzunlikda elementsiz bo’ladi. Agar bitta son ko’rsatilsa, bu son massiv elementlari sonini bildiradi. Agar bir nechta elementlar vergul bilan ko’rsatilsa ular massivning elementlari hisoblanadi.

Massiv elementlariga murojaat kvadrat qavslar ichida berilgan indekslar orqalai amalga oshiriladi. Massiv elementlarini tartiblash noldan boshlanadi:

*var Mass=new Array("Bir", "Ikki", "Uch");*

*document.write(Mass[0]); // "Bir"*

*Mass[3] = 4; // Massivning yangi elementini yaratish*

*document.write(Mass.join(", ")); //"Bir, Ikki, Uch, 4"*

**length** massiv elementlari sonini qaytaradi:

*var Mass = ["Bir", "Ikki", "Uch"];*

*document.write(Mass.length + "<br>"); // 3*

*for (var i=0, c=Mass.length; i<c; i++){*

*document.write(Mass[i] + "<br>"); // Bir, Ikki, Uch*

*}*

Massivlar bilan ishlashning qulay metodlari:

* push(<elementlar ro’yxati>) massiv elementlariga ko’rsatilgan elementlar ro’yxatini qo’shadi. Elementlar massiv oxiriga qo’shiladi. Metod massivning yangi uzunligini qaytaradi:

*var Mass = [ "Bir", "Ikki", "Uch"];*

*document.write(Mass.push("To’rtinchi","Beshinchi"));//5*

*document.write(Mass.join(", ")); // "Bir, Ikki, Uch, To’rtinchi, Beshinchi"*

* unshift(<elementlar ro’yxati>) massiv elementlariga ko’rsatilgan ro’yxat elementlarini qo’shish. Elementlar massivning boshidan qo’shiladi:

*var Mass =["Bir", "Ikki", "Uch"];*

*Mass.unshift ("To’rtinchi", "Beshinchi");*

*document.write(Mass.join(","));//"To’rtinchi, Beshinchi, Bir, Ikki,Uch"*

* concat(<elementlar ro’yxati>) joriy massiv va elementlar ro’yxatini birlashtirib natijani boshqa massiv sifatida qaytaradi. Bunda joriy massivga elementlar ro’yxati qo’shilmaydi:

*var Mass = [ "Bir", "Ikki", "Uch" ];*

*var Mass2 = []; // Bo’sh massiv*

*Mass2 = Mass.concat("To’rtinchi","Beshinchi");*

*document.write(Mass.join(","));//"Bir,Ikki, Uch"*

*document.write(Mass2.join(","));//"Bir,Ikki, Uch, To’rtinchi, Beshinchi"*

* join(<bo’luvchi>) massivning barcha elementlarini bo’luvchi orqali ajratilgan qator sifatida qaytaradi:

*var Mass=["Bir", "Ikki", "Uch"];*

*var Str=Mass.join("-");*

*document.write(Str);//"Bir-Ikki-Uch"*

* shift() massivning birinchi elementini o’chiradi va uni qaytaradi:

*var Mass = ["Bir","ikki", "Uch"];*

*document.write(Mass.shift()); //"Bir"*

*document.write(Mass.join(",")); //"Ikki, Uch"*

* pop()massivning oxirgi elementini o’chiradi va uni qaytaradi:

*var Mass = ["Bir", "Ikki", "Uch" ];*

*document.write(Mass.pop()); //"Uch"*

*document.write(Mass.join(",")); //"Bir, Ikki"*

* sort([saralash funksiyasi]) massivni saralaydi. Agar fuksiya ko’rsatilmasa, odatiy saralash (sonlar o’sish, belgilar alfavit tartibida) amalga oshiriladi:

*var Mass = ["Bir", "Ikki", "Uch"];*

*Mass.sort();*

*document.write(Mass.join(", ")); //"Bir, Ikki, Uch" (alfavit tartibida, ya’ni b,i,u)*

Agar standart saralanish tartibini o’zgartirish kerak bo’lsa, buni funksiya yordamida amalga oshirish mumkin. Funksiya ikkita parametrni qabul qiladi va qaytarishi kerak:

* 1 — agar birinchi kata ikkinchidan bo’lsa;
* -1 — agar ikkinchi kata birinchidan bo’lsa;
* 0 — agar elementlar teng bo’lsa.

Masalan, standart saralash simvol registriga bog’liq:

*var Mass = ["alisher","Erkin","Akmal","elbek"];*

*Mass.sort();*

*document.write(Mass.join(","));//"Akmal, Erkin, alisher, elbek"*

Natijada biz xato saralshga erishdik. Endi funksiya yordamida registrlarga bog’liq bo’lmagan saralashni ko’rib chiqamiz (4.6.1-misol).

***12.2-misol. Registrga bog’liq bo’lmagan saralash***

*function f\_sort(Str1, Str2) {*

*var Str1\_1=Str1.toLowerCase();//Quyi registrga o’tkazish*

*var Str2\_1=Str2.toLowerCase();//Quyi registrga o’tkazish*

*if (Str1\_1>Str2\_1) return 1;*

*if (Str1\_1<Str2\_1) return -1;*

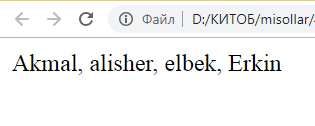
*return 0;*

*}*

*var Mass=["alisher","Erkin","Akmal","elbek"];*

*Mass.sort(f\_sort);//Funksiya nomi qavssiz ko’rsatiladi*

*document.write(Mass.join(", ")); // "Akmal, alisher, elbek, Erkin"*



*12.2-rasm*

Ikkita ozgaruvchi bitta registra o’tkaziladi, keyin standart taqqoslash bajariladi.

Teskari tartibda tartiblash quyidagicha bo’ladi:

***12.3-misol. Registrga bo’g’liq bo’lmagan teskari tartiblash***

*function f\_sort(Str1, Str2) {// Registrga bo’g’liq bo’lmagan teskari tartiblash*

*var Str1\_1 = Str1.toLowerCase(); // Quyi registrga o’tkazish*

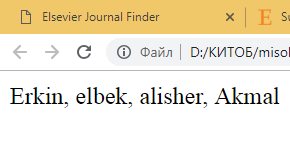
*var Str2\_1 = Str2.toLowerCase(); // Quyi registrga o’tkazish*

*if (Str1\_1>Str2\_1) return -1; if (Str1\_1<Str2\_1) return 1; return 0; }*

*var Mass = ["alisher", "Erkin", "Akmal","elbek"];*

*Mass.sort(f\_sort);*

*document.write(Mass.join(", ")); // "Erkin, elbek, alisher, Akmal"*



*12.3-rasm*

* reverse() massivni teskarilash. Berilgan massiv elementlari o’rinlari teskarisiga almashadi:

*var Mass = ["Bir", "Ikki", "Uch"]; Mass.reverse();*

*document.write(Mass.join(", ")); // "Uch, Ikki, Bir"*

* slice(<boshlanish>, [<Ohir>]) massivning <boshlanish>dan boshlab <Ohir> gacha bo’lgan indeksdagi elementlarini qaytaradi va bunda <Ohir>gi element kirmaydi. Agar ikkinchi parameter ko’rsatilmasa, massivning ohirgi elementigacha olinadi:

*var Mass1=[1, 2, 3, 4, 5 ];*

*var Mass2=Mass1.slice(1, 4);*

*window.alert(Mass2.join(",")); //"2, 3, 4"*

*var Mass3 = Mass1.slice(2);*

*window.alert(Mass3.join(",")); //"3, 4, 5"*

* splice(<boshlanish>, <Miqdor>, [<Qiymatlar ro’yxati>]) massiv elementlarini o’chirish, almashtirish yoki joylashtirishga imkon beradi. Qaytariladigan massiv o’chirilgan elementlardan iborat bo’ladi:

*var Mass1 = [1, 2, 3, 4, 5];*

*var Mass2 = Mass1.splice(2, 2);*

*window.alert(Mass1.join(",")); //"1, 2, 5"*

*window.alert(Mass2.join(",")); // "3, 4"*

*var Mass3 = Mass1.splice(1, 1, 7, 8, 9);*

*window.alert(Mass1.join(",")); //"1,7,8,9,5" window.alert(Mass3.join(",")); // "2"*

*var Mass4 = Mass1.splice(1, 0, 2, 3, 4);*

*window.alert(Mass1.join(","));//"1,2,3,4,7,8, 9, 5"*

*window.alert(Mass4.join(","));//Bo’sh massiv*

* toString() va valueOf() massivni qatorga o’tkazish. Massiv elementlari probelsiz vergul bilan ajratiladi:

*var Mass = ["Bir","Ikki","Uch"];*

*document.write(Mass.toString());//"Bir,Ikki,Uch"*

*Ko’p o’lchamli massiv.* Ko’p o’lchamli massivni yaratish usullarini sanab chiqamiz:

*var Mass = new Array(new Array("Bir", "Ikki", "Uch"), new Array ("To’rt", "Besh", "Olti"));*

*document.write(Mass[0][1]); // "Ikki"*

*var Mass2=[["Bir", "Ikki", "Uch"],["To’rt", "Besh", "Olti"]];*

*document.write(Mass2[1][1]);// "Besh"*

*yoki element bo’yicha:*

*var Mass = new Array();*

*Mass[0] = new Array();*

*Mass[1] = new Array();*

*Mass[0][0] = "Bir";*

*Mass[0][1] = "Ikki";*

*Mass[0][2] = "Uch";*

*Mass[1][0] = "To’rt";*

*Mass[1][1] = "Besh";*

*Mass[1][2] = "Olti";*

*document.write(Mass[1][2]);// "Olti"*

*var Mass2=[];*

*Mass2[0] = [];*

*Mass2[1] = [];*

*Mass2[0][0] = "Bir";*

*Mass2[0][1] = "Ikki";*

*Mass2[0][2] = "Uch";*

*Mass2[1][0] = "To’rt";*

*Mass2[1][1] = "Besh";*

*Mass2[1][2] ="Olti";*

*document.write(Mass2[0][0]); // "Bir"*

Ko’p o’lchamli massiv elementlariga murojaat ikki indeks orqali amalga oshiriladi:

*var Str = Mass[1][2];*

Assosiativ massiv. Assosiativ massivlarni saralash

Oddiy va assosiativ massiv asosiy farqi uning elementlariga murojaat qilishdadir. Assosiativ massivda elementlarga murojaat sonlar orqali emas qatorlar orqali amalga oshiriladi.

var Mass = new Array();

Mass["Bir"] = 1;

var Mass = new Array();

Mass["Bir"] = 1;

Mass["Ikki"] = 2;

Mass["Uch"] = 3;

document.write(Mass["Bir"]); // 1

Massivning barcha elementlarini qanday chop etiladi? Array sinfidagi hech bir metod birdaniga chop etish imkoniyatini bermaydi. O’z vaqtida, length hususiyati ham ishlamaydi. Buni standart sikl operatori for bilan ham amalga oshirib bo’lmaydi.

Bu maqsad uchun maxsus sikl for...in ishlatiladi. U quyidagi formatda beriladi:

for (<O’zgaruvchi> in <Sinf nusxasi>) { <Sikl tanasi>}

for...in siklida har bir iterasiyada <O’zgaruvchi> ga assosiativ massiv elementlari indekslarini navbat bilan o’zlashtirirladi:

var Mass = new Array();

Mass["Bir"] = 1;

Mass["Ikki"] = 2;

Mass["Uch"] = 3;

for (var Name in Mass) {// Name o’zgaruvchisi har bir iterasiyada

// assosiativ massiv qator indeksini o’zlashtiradi

document.write(Name+"= "+Mass[Name]+ "<br>");

}

Natijada biz quyidagi natijani olamiz:

Bir = 1

Ikki = 2

Uch = 3

Math sinfi. Matematik funksiyalardan foydalanish.

Math sinfi o’z ichiga matematik o’zgarmaslar va funksiyalarni oladi. Uning foydalanilishida sinf nusxasini yaratsihni talab qilinmaydi.

Hususiyat:

* e — e, natural logarifm asosi;
* LN2 — natural logarifm 2;
* LN10 — natural logarifm 10;
* LOG2E —2 asosga ko’ra logarifm e;
* LOG10E — 10 asosga ko’ra logarifm e;
* pi —Пи:
* document.write(Math.PI); //3.141592653589793
* sqrt1\_2 —0,5 kvadrat ildiz;
* SQRT2 —2 kvadrat ildiz.

Uslublar:

* abs() — absolyut qiymat;
* sin(), cos(), tan() — standart trigonometrik funksiyalar (sinus, kosinus, tangens). Qiymatlar radianda ko’rsatiladi;
* asin(), acos(), atan() — teskari trigonometric funksiyalar (arksinus, arkkosinus, arktangens). Qiymatlar radianda qaytariladi;
* exp() — eksponenta;
* log() — natural logarifm;
* pow(<Son>, <Daraja>) — <Son> <Daraja> ga ko’tariladi:

var x = 5;

document.write(Math.pow(x, 2)); // 25 (5 ning kvadrat darajasi)

* sqrt() — kvadrat ildiz:

var x = 25;

document.write(Math.sqrt(x)); // 5 (25 dan kvadrat ildiz)

* round() — sonni eng yaqinb butun songacha yaxlitlash. Agar vergulgdan keying birinchi son 0 va 4 oralig’ida bo’lsa, kami bilan yaxlitlanadi aksincha ortig’i bilan:

var x = 2.499; var y = 2.5;

document.write(Math.round(x)); // 2

document.write(Math.round(y)); // 3

* ceil() — sonni eng yaqin katta butun songa yaxlirlash:

var x = 2.499; var y = 2.5;

document.write(Math.ceil(x)); // 3

document.write(Math.ceil(y)); // 3

* floor() — eng yaqin kichik butun songa yaxlitlash:

var x = 2.499; var y = 2.6;

document.write(Math.floor(x)); //2

document.write(Math.floor(y)); // 2

* max(<vergul bilan ajratilgan sonlar ro’yxati>)– ro’yxatdagi qiymatlarning eng kattasi:

document.write(Math.max(3, 10,6)); // 10

* min (<vergul bilan ajratilgan sonlar ro’yxati>) – ro’yxatdagi qiymatlarning eng kichkinasi:

document.write(Math.min(3,10,6)); // 3

* random() —0 dan 1 gacha tasodifiy son:

document.write(Math.random());//masalan,0.9778613566886634

Agar 0 dan 9 gacha tasodifiy butun sonni olish kerak bo’lsa, random() metodidan olingan natijani 9.9999 ga ko’paytirish, keyin floor() metodi yordamida yaxlitlash mumkin bo’ladi:

var x = Math.floor(Math.random()\*9.9999);

document.write(x);

Web sahifani bir necha bor qayta yuklab tekshirib ko’rish mumkin. Tasodifiy sonlar 0 va 9 oralig’ida berilmoqda. Bu nima uchun ishlatiladi? Masalan, bizda to’rtta 468x60 o’lchamdagi bannerni tasodifiy ko’rsatish talab etilsin.

*var x = Math.floor(Math.random()\*3.9999);*

*document.write('<img src="banner' + x + '.gif" width="480" height="60">');*

banner0.gif, banner1.gif, banner2.gif va banner3.gif nomli to’rt banner bir papkada scripda foydalanish uchun joylashgan bo’lishi kerak.

Turli nomdagi bannerli fayllarni massivga olish va ulardan massivnicng tasodifiy elementlari sifatida foydalanish mumkin:

*var Mass = ["banner-red.gif", "banner-blue.jpeg", "banner-gray.gif", "banner-white.png" ];*

*var x = Math.floor(Math.random()\*3.9999);*

*document.write('<img src="' + Mass[x]);*

*document.write(‘width="480" height="60">');*

Date sinfi. Joriy sana va vaqt. Web brauzer oynasida sava va vaqtni chop etish.

Date sinfi vaqt va sana bilan ishlash imkonini beradi. Sing nusxasi quyidagicha yaratiladi:

<Sinf nusxasi> = new Date();

<Sinf nusxasi> = new Date(<Milisekund miqdori>);

<Sinf nusxasi> = new Date(<Yil>, <Oy>, <Kun>, <Soat>, <Minutlar>, <Sekundlar>, <Milisekundlar>);

Sinf quyidagi metodlarni qo’llaydi:

* toString() sanani qatorga o’tkazadi va uni qaytaradi:

var d = new Date();

document.write(d.toString());

// Opera da: Fri,30 Oct 2009 01:07:17 GMT+0300

// Firefox da: Fri Oct 30 2009 01:07:17 GMT+0300

// IE da: Fri Oct 30 01:07:17 UTC+0300 2009

* toLocaleString() tizimda o’rnatilgan sanani qatorga o’tkazadi va uni qaytaradi:

var d = new Date();

document.write(d.toLocaleString());

// Opera da: 30.10.2009 1:11:27

// Firefox da: 30 Октябрь 2009 г. 1:11:27

// IE da: 30 октября 2009 г. 1:11:27

* valueOf()– 01.01.1970 00:00:00 dan boshlab o’tgan vaqtdagi milisekundlar miqdorini aniqlash imkonini beradi:

var d = new Date();

document.write(d.valueOf());// 1256854444062

* getDate() oy kuni tartibini qaytaradi (1 dan 31 gacha):

var d = new Date(); document.write(d.getDate()); // 30

* getDay() hafta kunini tartibini aniqlatydi (yakshanba- 0 dan 6 -shanba gacha):

var Mass =["yakshanba", "dushanba", "seshanba",

"chorshanba", "payshanba", "juma", "shanba"];

var d = new Date();

document.write(Mass[d.getDay()]); // juma

* getMonth() oy tartibini qaytaradi (0- yanvar dan 11-dekabr gacha):

var Mass =["yanvar", "fevral", "mart", "aprel", "may",

"iyun", "iyul", "avgust", "sentabr", "oktabr", "nayabr", "dekabr"];

var d = new Date();

document.write(Mass[d.getMonth()]); // oktabr

Joriy oy tartibini aniqlash uchun funksiya qaytargan qiymatni 1 ga oshirish kerak:

var d = new Date();

var Month = d.getMonth() + 1;

document.write(Month); // 10

* getFullYear() yilni aniqlash imkonini beradi:

var d = new Date();

document.write(d.getFullYear()); // 2009

* getHours() soatni aniqlash imkonini beradi (0 dan 23 gacha):

var d = new Date(); document.write(d.getHours()); // 1

* getMinutes() minutni aniqalsh imkonini beradi (0 dan 59 gacha):

var d = new Date();

document.write(d.getMinutes()); // 23

* getSeconds() sekundni aniqlash imkonini beradi (0 dan 59 gacha):

var d = new Date();

document.write(d.getSeconds()); // 20

* getMilliseconds() milisekundni qaytaradi (0 dan 999 gacha):

var d = new Date();

document.write(d.getMilliseconds()); // 156

* getTime()– 01.01.1970 00:00:00 dan boshlab o’tgan milisekundlar miqdorini aniqlash imkonini beradi:

var d = new Date();

document.write(d.getTime());// 1256855182843

*** 12.4-misol. Joriy sana va vaqt***

*<html> <head>*

*<title>Joriy vaqt va sana</title >*

*<meta http-equiv="Content-Type" content="text/html;*

*charset = windows-1251">*

*<script type="text/javascript">*

*function fDate(Str) {*

*Str += ""; // sanani qatorga o’tkazish*

*if (Str.length==1) return ("0" + Str);*

*else return Str;}*

*function fYear(Year) {*

*Year += ""; // sananani qatorga o’tkazish*

*return Year.substr(2);*

*}*

*</script>*

*</head>*

*<body>*

*<script type="text/javascript">*

*var d = new Date();*

*var msg;*

*var Day=["yakshanba", "dushanba", "seshanba", "chorshanba", "payshanba","juma", "shanba"];*

*var Month=["yanvar", "fevral", "mart", "aprel", "may", "iyun", "iyul", "avgust", "sentyabr", "oktyabr", "noyabr", "dekabr"];*

*msg = "Bugun <br>" + Day[d.getDay()] + " ";*

*msg += d.getDate() + " ";*

*msg += Month[d.getMonth()] + " ";*

*msg += d.getFullYear() + " ";*

*msg += fDate(d.getHours()) +":";*

*msg += fDate(d.getMinutes())+ ":";*

*msg += fDate(d.getSeconds()) + "<br>";*

*msg += fDate(d.getDate())+".";*

*msg += fDate(d.getMonth()+1)+".";*

*msg += fYear(d.getFullYear());*

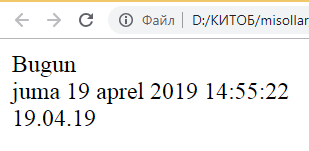
*document.write(msg);*

*</script>*

*</body>*

*</html>*

Web brauzer oynasida quyidagi paydo bo’ladi



*12.4-rasm*

Misolda biz ikkita funksiya yartib undan foydalandik:

* fDate(str) — agar parameter bitta raqamdan iborat bo’lsa, funksiya uning oldiga 0 qo’yib qatorni qaytaradi;
* f\_Year(Year)— funksiya yilning ohirgi ikki raqamini qaytaradi.

Function *sinfi*

Function sinfi funksiyadan sinf nusxasi sifatida foydalanish imkonini beradi. Bu quyidagicha amalga oshiriladi:

<Funksiya nomi> = new Function(<Parametr1>, ... , <Parametrn>, <Funksiya tanasi>);

Masalan, ikki son yig’indisini xisoblasin

function f\_Sum(x, y) {return x + y;}

quyidagicha yozish mumkin:

var f\_Sum=new Function ("x", "y","return x+y");

Funksiya tanasini qator sifatida ko’rsatish juda noqulay. Bu holdan hech kim foydalanmaydi.

Bu o’rinda anonim funksiya qo’llaniladi:

var f\_Sum = function(x, y){ return x + y; };

Funksiyani chaqirish oldingi kabi quyidagicha:

document.write(f\_Sum(5, 6)); // 11

Anonim funktsiyalardan foydalanganda, global o'zgaruvchan funktsiyadagi funksiyani belgilaganingizda, bu o'zgaruvchiga havola saqlanadi, uning qiymati emas:

var x = 5;

var f\_Sum = function() {

return x; // havola saqlanadi, x qiymati emas };

document.write(f\_Sum()); // 5

x = 10; // Qiymatni o'zgartirdi

document.write(f\_Sum()); // 10 ga teng, 5 emas

|  |  |
| --- | --- |
|  | ***MUHOKAMA UCHUN SAVOLLAR VA TOPSHIRIQLAR!*** |

1. *JavaScriptda sinf qanday yaratiladi?*
2. *Metod nima?*
3. *JavaScriptda Global sinf haqida nimalarni bilasiz?*
4. *String sinfi haqida nimalarni bilasiz?*
5. *Array sinfi qanday maqsadlarda ishlatiladi?*
6. Number sinfi haqida nimalarni bilasiz?
7. Date sinfi*dan foydalanish qanday?*