

*Web dasturlashga kirish

Javascript asoslari. Javascript ssenariy tilining xususiyatlari

- JavaScript asoslari. JavaScript ssenariylarini HTML-hujjat bilan bog'lash
- O'zgaruvchilar, ma'lumot turlari
- Massivlar, operatorlar

* JavaScript asoslari

- * **JavaScript** – web sahifalarni klient tomonida ko'rish ssenariylarini boshqarish uchun mo'ljallangan dasturlash tili.
- * ***JavaScript dasturlash tili*** yordamida HTML –dokumentlarning ko'rinishi va tuzilishini dinamik ravishda boshqarish mumkin.
Ekranda tasvirlanayotgan dokumentga brauzer tomonidan yuklangan dokumentning sintaktik tahlil qilish jarayonida istalgan HTML-kodlarni joylashtirish mumkin.

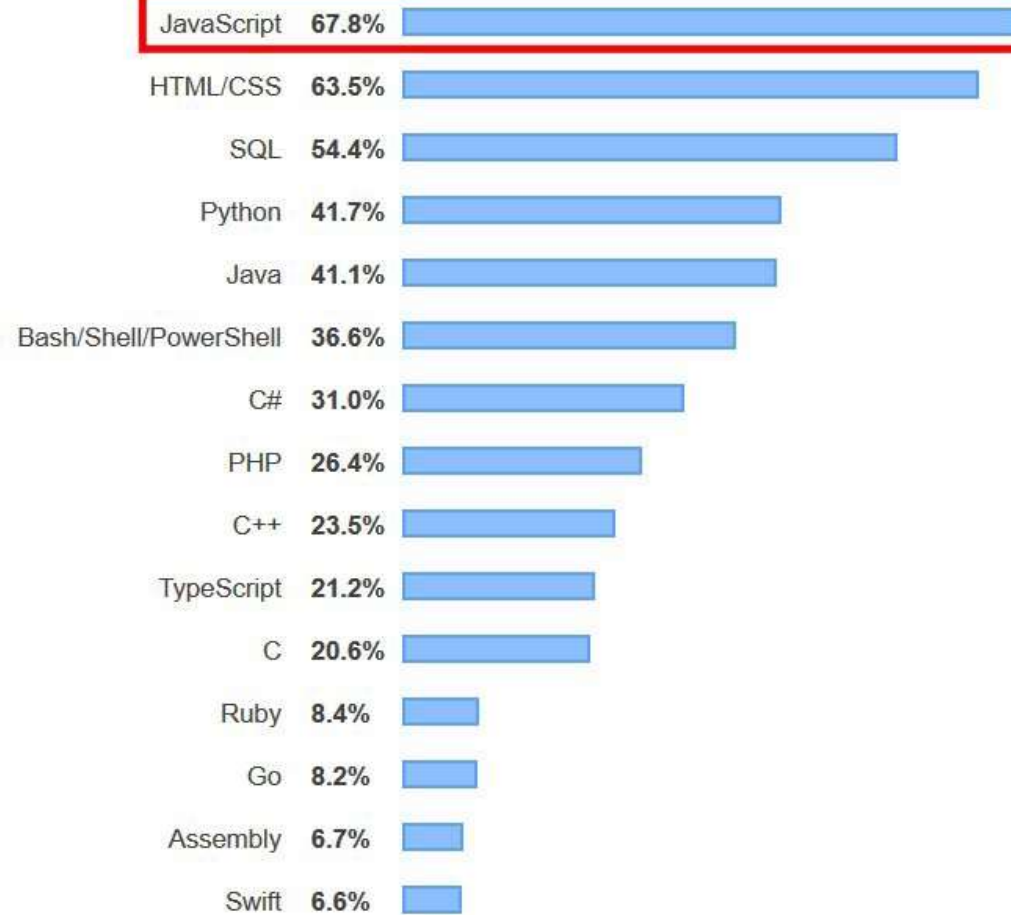


Most Popular Technologies

Programming, Scripting, and Markup Languages

All Respondents

Professional Developers



* JavaScript ssenariylarini HTML-hujjat bilan bog'lash

HTML sahifada bevosita JavaScript kodni ishlatish uchun **<script> </script>** tegi orasiga yoziladi.

Ushbu **<script>** tegi **<head>** qismda yoki **<body>** qism ichida yozilishi mumkin.

```
<script>
function myFunction() {
    document.getElementById("demo").innerHTML = "Paragraph changed.";
}
</script>
```

```
<!DOCTYPE html>
<html>
<head>
<script>
function myFunction() {
    document.getElementById("demo").innerHTML = "Ana xolos!!!";
}
</script>
</head>
<body>

<h2>JavaScriptni qo'llashga namuna</h2>

<p id="demo">Bu yozuv o'zgarishi kerak</p>

<button type="button" onclick="myFunction()">Tugmani bos!!!</button>

</body>
</html>
```


JavaScript da natijani chiqazish uchun quyidagilardan foydalaniladi:

- Brauzer xabar oynasiga: `window.alert("Hello World");` yoki `alert("Hello World");`
- Brauzer oynasiga: `document.write("Hello World");`
- HTML element ichiga: `document.getElementById("p1").innerHTML = "Hello World";`
- Brauzer konsol oynasiga: `console.log("Hello World");`

***javascript Output**

*Tashqi .js fayl orqali bog'lash

HTML fayldagi bog'lanish kodi:

```
<script src="myScript.js"></script>
```

myScript.js fayli:

```
function myFunction() {  
    document.getElementById("demo").innerHTML = "Paragraph changed.";  
}
```


Dastur kodiga yoziladigan izoh aksariyat dasturlash tillari singari JavaScript da ham bir qatorli va ko'pqatorli turlarga bo'linadi:

```
// bir qatorli izoh  
document.getElementById("div1").innerHTML = "Salom JavaScript!!!";
```

```
/*  
JavaScript da  
ko'p satrli  
izoh shu tarzda yoziladi!!!  
*/  
document.getElementById("div1").innerHTML = "Salom JavaScript!!!";
```

*O'zgaruvchilar (variables)

Biror ma'lumot qiymatini o'zida saqlovchi nomlangan belgi o'zgaruvshidir.

O'zgaruvchilar quyidagi talablar asosida bo'lishi shart:

```
<script>  
  
  var a = 7;  
  var b = 9;  
  var c = x+y;  
  
</script>
```

- o'zgaruvchi nomi unikal nom bilan e'lon qilinadi
- o'zgaruvchi nomi sifatida raqam. Harf, tag chiziq va dollar belgisi kabilar ishlatish mumkin
- o'zgaruvchi nomi harf, \$ belgisi yoki tag chiziq bilan boshlanishi mumkin
- JavaScript da o'zgaruvchi nomlarida kata va kichik harflar farqlanadi
- JavaScript da mavjud kalit so'zlardan o'zgaruvchi nomi sifatida foydalanish mumkin emas

var, let, const (variables declaration)

O'zgaruvchilarni e'lon qilishda asosan **var** kalit so'zidan foydalaniladi. Biroq, 2015-yilgi ECMA (European Computer Manufacturer's Association) ning ECMAScript ga kiritilgan o'zgarishlaridan so'ng, **let** hamda **const** kalit so'zlari ham o'zgaruvchilar e'lon qilishda ishlatiladigan bo'ldi.

var, let, const bir-biridan farqi:

- o'zgaruvchi qiymatini qayta o'zlashtirishda var da muammo yo'q, biroq let yordamida qiymatni qayta o'zlashtirib bo'lmaydi.

```
var a = "Apple";  
var a = "Pear";  
  
let b = "Samsung";  
let b = "Apple";  
  
console.log(a); // Pear  
console.log(b); //Error (xatolik)
```

- var orqali o'zgaruvchi faqatgina funksiya ichida e'lon qilinsa, u local o'zgaruvchi hisoblanadi, va faqat shu funksiya ichidagina indan foydalanish mumkin.

- let hamda const bilan e'lon qilingan o'zgaruvchi esa istalgan biror blok ichida e'lon qilinsa u local hisoblanadi.

Blok bu { } orasida yozilgan dastur qismi
(if, for, while va h-k kabilar)

- const bu o'zgarmas qiymatni o'zida saqlaydi, ya'ni const bilan e'lon qilingan ma'lumot qiymatini qayta o'zgartirish mumkin emas;

```
var a = -9;  
if(a<0){  
    let c = 8;  
    var d = 7;  
}  
console.log(c); // error  
console.log(d); // 7
```

```
function myFunction(){  
    var x = 1.27;  
}  
myFuntion();  
console.log(x); // error
```

```
const a = "Manchester United";  
a = "Manchester City";  
  
console.log(a); // error;
```

Primitiv turlar:

`var a = "Salom";` - satr

`var pi = 3,14;` - sonli

`var b = true;` - mantiqiy

`var c = false;` - mantiqiy

`var x;` - noaniq tip

Kompleks turlar:

`var a = {name: 'John', age: 34};` - obyekt

`var pi = [1,2,3,4,5];` - massiv obyekt

`var null;` - qiymatsiz obyekt

`function myFunction() {}` - funksiya

* **Ma'lumot turlari**

*Massiv (array)

Bir nechta o'zgaruvchilarning bitta nom bilan birlashtirilishi massiv hisoblanadi.

```
var cars = ["Toyota", "Volvo", "BMW", "Audi"];
```

```
var cars = new Array("Toyota", "Volvo", "BMW", "Audi");
```

```
var cars = [  
    "Toyota",  
    "Volvo",  
    "BMW",  
    "Audi"  
];
```

Massivlarga murojaat indeks orqali amalga oshiriladi

```
var fruits = ["Banana", "Orange", "Apple", "Mango"];  
var first = fruits[0];
```

Massiv uzunligi:

```
var fruits = ["Banana", "Orange", "Apple", "Mango"];  
var len = fruits.length;
```

Massivga yangi element qo'shish (yangi element push funksiyasi orqali massiv oxiridan qo'shiladi)

```
var fruits = ["Banana", "Orange", "Apple", "Mango"];  
fruits.push("Lemon");
```

*Massiv metodlari (array methods)

Metodlar bu - biror vazifani bajaruvchi tayyor funksiya qolipidir.

toString() – massiv elementlarini satrli tipga o'girish

join("*") – massiv elementlarini satrli tipga orasini malum belgi yordamida ajratib o'tkazish (m-n: *, +, _ va h-k)

pop() – massiv oxirgi elementini o'chirib tashlash

push(*new element*) – massiv oxiridan yangi element qo'shish

shift() – massiv birinchi elementini o'chirib tashlash

sort() – o'sish tartibida saralash

reverse() – elementlarini teskari tartibda joylashtirish

*Massiv metodlari (array methods)

delete massiv_nomi[indeks] – massiv biror indeks elementini o'chirish

massiv1.concat(massiv2) – 1- massivga, 2-massiv elementlarini birlashtirish

slice(i, j) – i-indeksdan j-indeksgacha elementlarni ajratib olish

valueOf() – massiv elementlari primitiv tipli qiymatini qaytaradi

isArray() – o'zgaruvchi yoki obyektning aynan massiv yoki massiv emasligini aniqlaydi

- *Arifmetik operatorlar
- *Qiymat o'zlashtirish operatorlari
- *Taqqoslash operatorlari
- *Mantiqiy operatorlar
- *Bitlar bilan ishlash operatorlari

*operatorlar

+ qo'shish
- ayirish
***** ko'paytirish
/ bo'lish
% qoldiq olish
++ inkrement
-- dekriment

+	Addition
-	Subtraction
*	Multiplication
/	Division
%	Modulus (Division Remainder)
++	Increment
--	Decrement

* Qiymat o'zlashtirish operatorlari

=	x = y	x = y
+=	x += y	x = x + y
-=	x -= y	x = x - y
*=	x *= y	x = x * y
/=	x /= y	x = x / y
%=	x %= y	x = x % y

*Taqqoslash operatorlari

Operator	Tavsifi
==	aynan teng
===	qiymat va tip o'zaro teng
!=	Teng emas
!==	qiymat yoki tip turlicha (teng emas)
>	... dan katta
<	...dan kichik
>=	katta yoki teng
<=	Kichik yoki teng
?	ternar amal

*Mantiqiy operatorlar

Operator	Tavsif
&&	mantiqiy va
	mantiqiy yoki
!	mantiqiy emas

*Bitli operatorlar

Operator	Tavsif	Namuna	Misol	Natija	O'nlik
&	AND	5 & 1	0101 & 0001	0001	1
	OR	5 1	0101 0001	0101	5
~	NOT	~ 5	~0101	1010	10
^	XOR	5 ^ 1	0101 ^ 0001	0100	4
<<	Bit bo'yicha chapga surish	5 << 1	0101 << 1	1010	10
>>	Bit bo'yicha o'ngga surish	5 >> 1	0101 >> 1	0010	2

 **E'tiboringiz uchun tashakkur!!!**