Mündericat

1 js nedir ---------------------------------------------------------------------------------------------

2 js de datatypelar --------------------------------------------------------------------------------

3 js de let const var ferqi-------------------------------------------------------------------------

4 js de Operatorlar (Arithmetic, Assignment, Comparison, Logical və s.) --------------

5 js adlandirma qaydalari (ornek camleCase,pascalCase,snakeCase,

funksiyalarin adi feil olmalidir kimi)---------------------------------------------------------

6 js de if else switch case (ternary)------------------------------------------------------------

7 js de for while do While break continue -------------------------------------------------

8 js de arraylar , length , push ve s -------------------------------------------------------------

9 js de object nedir nece yazilir-----------------------------------------------------------------

10 js de funksiyalar ve orneklerle izahi (arrow , decleration,expression)----------------

10 js de string templete literal orneyi----------------------------------------------------------

11 js de strring metodlari--------------------------------------------------------------------------

12 js de hoisting nedir------------------------------------------------------------------------------

Java Script nədir?

JavaScript internetin proqramlaşdırma dilidir. Həm HTML, həm də CSS-ni yeniləyə və dəyişə bilər. *JavaScript*  "veb səhifələrini dinamik" üçün yaradılmışdır. Javascripti aşağıdakı kimi HTML-ə qoşuruq:

<body>

 <script src="/asess/index.js"></script>

</body>

2.Java scripdə datatype-lar

JavaScript-in 8 verilənlər növü var:String, Number, Bigint, Boolean, Undefined, Null, Symbol, Object

String-JavaScript-də sətir dırnaq işarələri ilə əhatə olunmalıdır.

// Strings:  
let str = "Hello";

let str2 = 'Single quotes are ok too';

let phrase = `can embed another ${str}`;

JavaScript-də 3 növ sitat var.

1. İkiqat sitatlar: "Hello".
2. Tək sitatlar: 'Hello'.
3. Geri işarələr: `Hello`.

İkiqat və tək dırnaqlar “sadə” sitatlardır. JavaScript-də onlar arasında praktiki olaraq heç bir fərq yoxdur.

Geri işarələr "genişlənmiş funksionallıq" sitatlarıdır. Onlar bizə dəyişənləri və ifadələri bükərək sətirdə yerləşdirməyə imkan verir ${…}, məsələn:

let name = "John";

// embed a variable

alert( `Hello, ${name}!` ); // Hello, John!

// embed an expression

alert( `the result is ${1 + 2}` ); // the result is 3

İçindəki ifadə ${…}qiymətləndirilir və nəticə sətirin bir hissəsinə çevrilir. Biz oraya hər şeyi qoya bilərik: kimi dəyişən namevə ya arifmetik ifadə kimi 1 + 2və ya daha mürəkkəb bir şey.

// Numbers:  
let length = 16;  
let weight = 7.5;

// Booleans  
let x = true;  
let y = false;

Bu tip adətən bəli/xeyr dəyərlərini saxlamaq üçün istifadə olunur: true“bəli, düzgün” və false“yox, yanlış” deməkdir.  
// Object:  
const person = {firstName:"John", lastName:"Doe"};

obyektlər məlumat kolleksiyalarını və daha mürəkkəb obyektləri saxlamaq üçün istifadə olunur.

3.Java Scripte let var const ferqi

**let**: Dəyəri verib sonra çağırmaq olar, qiymətini sonradan dəyişmək mümkündür.

**var**: Dəyəri vermədən əvvəl də çağırmaq mümkündür, amma qiymətini dəyişmək olar.

**const**: Dəyəri verib çağırmaq mümkündür, amma qiymətini sonradan dəyişmək olmaz.

4.Java Scripte Operatorlar (Arithmetic, Assignment, Comparison, Logical)

**Arifmetik operatorlar** ədədlər üzərində hesab aparmaq üçün istifadə olunur(arifmetik)

* + Toplama
* - Çıxma
* \* Vurma
* \*\* Qüvvət
* / Bölmə
* % Bölmədə alinan qalıq
* ++ Artma
* -- Azalma

**Addition Assignment Operator** (+=) dəyişənə dəyər əlavə edir.(teyinat)

* = 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
* \*\*= x\*\*=y x=x\*\*y

  Comparison Operators (muqayise)

* == bərabərdir
* === dəyəri və tipi bərabərdir
* != bərabər deyil
* !== dəyəri və tipi bərabər deyil
* > Böyükdür
* < Kiçikdir
* >= Böyük bərabırdir
* <= Kiçikdir bərabərdir

Lokal operators (məntiqi)

* && ve
* || vey a
* ! eks

5 js adlandirma qaydalari (ornek camleCase,pascalCase,snakeCase, ve .s funksiyalarin adi feil olmalidir kimi)

snake\_case: : Sözlər arasında alt xətt (\_) istifadə olunur və bütün sözlər kiçik hərflə yazılır.

user\_age, total\_amount, calculate\_total\_price

camelCase: İlk söz kiçik hərflə yazılır, sonra gələn hər bir sözün ilk hərfi böyük yazılır. userAge, getUserInfo, calculateTotalAmount

PascalCase**:** Hər sözün ilk hərfi böyük yazılır: Person, CarModel, UserProfile

* Kebab-case: Sözlər arasında tire (-) istifadə olunur və bütün sözlər kiçik hərflə yazılır. user-profile, total-amount, calculate-total-price

6 js de if else switch case (ternary)

**if**: Əgər şərt doğru olarsa, müəyyən etdiyin işləri görmək üçün bunu istifadə et.

**else**: Əgər şərt yanlış olarsa, başqa işləri görmək üçün bunu istifadə et.

**else if**: İlk şərt yanlış olduqda, başqa bir şərt yoxlamaq üçün bunu istifadə et.

let age = 20;

if (age >= 18) {

console.log("Yaşınız yetkindir."); // Əgər yaş 18 və ya daha böyükdürsə, bu yazı çıxacaq

} else if (age >= 13) {

console.log("Yaşınız yeniyetmədir."); // Əgər yaş 13 və ya daha böyük, amma 18-dən kiçikdirsə, bu yazı çıxacaq

} else {

console.log("Yaşınız uşaqlıq dövründədir."); // Əgər heç biri düzgün deyilsə, bu yazı çıxacaq

}

Swich case if elsenin ekvivalentidi daha qisa yazilisdir

switch (new Date().getDay()) {  
  case 0:  
    day = "Sunday";  
    break;  
  case 1:  
    day = "Monday";  
    break;  
  case 2:  
     day = "Tuesday";  
    break;  
  case 3:  
    day = "Wednesday";  
    break;  
  case 4:  
    day = "Thursday";  
    break;  
  case 5:  
    day = "Friday";  
    break;  
  case 6:  
    day = "Saturday";  
}

7. js de for while do While break continue

for dövrü: Müəyyən bir sayda təkrarlama etmək üçün istifadə olunur.

for (let i = 1; i <= 5; i++) {

console.log(i); // 1-dən 5-ə qədər sayar}

while dövrü: Şərt doğru olduğu müddətcə işləyir

let i = 1;

while (i <= 5) {

console.log(i); // 1-dən 5-ə qədər sayar

i++;}

do while dövrü: İlk əvvəl kodu bir dəfə işlədib, sonra şərti yoxlayır. Yəni, ən azı bir dəfə işləyəcək.

let i = 1;

do {

console.log(i); // 1-dən 5-ə qədər sayar

i++;

} while (i <= 5);

break: : Dövrü dayandırmaq üçün istifadə olunur. Əgər şərt uyğun gələrsə, dövr dərhal dayanır.

for (let i = 1; i <= 5; i++) {

if (i === 3) {

break; }

console.log(i);}

continue: Dövrün cari təkrarlamasını atlayır və növbəti təkrarlamaya keçir.

for (let i = 1; i <= 5; i++) {

if (i === 3) {

continue; // 3-ü atlayacaq }

console.log(i);}

8.js de arraylar , length , push ve s

Array bir neçə qiyməti bir yerdə saxlamağa imkan verən obyekt növüdü

let fruits = ["alma", "banan", "portakal"]

length uzunluğu ,

push sonuna əlavə edir,

pop sonundan silir

shift əvvəldən əlavə edir

unshift əvvəldən silir

9 js de object nedir nece yazılır

bir çox məlumatı bir yerdə saxlamağa imkan verir. Məsələn, bir insanın adı, yaşı

let obj= {

name: "Aygün",

surname: "Bayramova",

age: 20

};

console.log(obj.name);

console.log(obj.age);

10 js de funksiyalar ve orneklerle izahi (arrow , decleration,expression)

Declaration function- tekce burda hoisting var

function liveage(age) {

return age \* 12;

}

console.log(liveage(3));

Expression Funktion

const sumExpressionway = function (age) {

  return age \* 12;

};

// console.log(sumExpressionway(4));

Arrow function

/const sumarrowfunction = (age) => {

return age \* 12;

};

console.log(sumarrowfunction(5));

Arrow qisa yazilis

const sumarrowfunction1 = (age) => age \* 12;

console.log(sumarrowfunction1(5));

11 js de string templete literal orneyi

JavaScript-də **template literal** çox rahat şəkildə dinamik olaraq stringləri yaratmağa imkan verir. Bu xüsusiyyət **backtick (`) işarəsi ilə yazılır və ${}`** sintaksisi istifadə edərək dəyişənləri və ifadələri daxil etməyə imkan verir.

let name = "Aygün";

let age = 20;

let message = `Mənim adım ${name} və mən ${age} yaşımdayam.`;

console.log(message);

${a + b} toplama əməliyyatı vəya backtick də sətirdən sətirə keçmək olur

12 js de strring metodlari

Length- Bir stringin uzunluğunu verir

toUpperCase()-Stringin bütün hərflərini böyük hərflərə çevirir.

toLowerCase()-Stringin bütün hərflərini kiçik hərflərə çevirir.

charAt()- Stringin müəyyən bir indeksindəki hərfi qaytarır.

trim()-Stringin əvvəlində və sonunda olan boşluqları silir.

repeat()-Stringi müəyyən bir sayda təkrarlayır.

12 js de hoisting nedir

Funksiyanı təyin etməzdən əvvəl çağırıla bilir