* **1. Javsript nedir?**
* JavaSript dinamik websaytlar hazırlamaq üçün istifadə olunan proqramlaşdırma və ya script dilidir. JavaScript təkcə brauzerdə yox, həmdə server tərəfdə işləyə bilir. Bunu təmin edən proqram JavaSript engine adlanır. JavaSript bu proqramın olduğu istənilən cihazda işləyə bilir.
* **2. Javascript browserde neleri ede bilir?**
* Səhifəyə yeni HTML əlavə edə, mövcud  HTML - i dəyişdirə bilər.
* İstifadəçi hərəkətlərinə reaksiya verir. Meselen: Maus klikləri, scrollu herketi, her hansi duymeye klikleyende ne bas versin ve s.
* Sebeke uzerinden uzaqdaki serverlere sorgu gondere ve ya fayl endire yada yukeye bilir.
* Brauzerin yaddasinda her hansi muellimati saxlaya bilir
* **Javascript browserde neleri ede bilmir?**
* Veb səhifədəki JavaScript sabit diskdəki ixtiyari faylları oxuya,yaza, onları kopyalaya və ya proqramları icra edə bilməz.
* Müasir brauzerlər JavaScripte fayllarla işləməyə imkan verir, lakin giriş məhduddur. Bir <input> teqi yaranaibki onun vasitesi ile biz emeliyyat sisteminden fayillari yukleye bilerik. <input> teqi tehlukesizlik cehetden yazilib.
* Kamera,mikrofon və digər cihazlarla qarşılıqlı əlaqə qurmağın yolları var, lakin onlar istifadəçinin icazəsini tələb edir.
* JavaScript şəbəkə üzərindən Xarici servislere sorgu ata bilmir. Ancaq eyni domen adi altindaki daxili servislere sorgu gondere bilir
* 3. Javscript variables
* JavaScript de muelimatlari yaddasda saxlamaq ucun deyisenlerden istifade edirik. Deyisenler “var” ve ya “let” acar sozu vasitesi ile yazilir, sonra deyisenin adi yazilir, daha sonra menimsetme operatoru ve deyisene bir deyer menimsedilir.
* Meselen :
* var userName = "Ilkin";
* console.log(userName)
* Belede yaza bilerik :
* var userName;
* userName = "Ilkin"
* console.log(userName)
* Deyisenin muxtelif tipleri olur

var user = 'Taleh';

var age = 25;

var message = 'Salam';

Deyisenin adlandirilmasinda muxtelif qaydalar var. Deyisenin adi reqemle ve simvolla baslaya bilmez. Istisna olaraq \_ ve $ olar;

Deyisenin adinda iki soz arasinda bosluq olmaz.if, else , return ve s proqramlasmada istifade olunan sozler olmaz

* 4. Variable type

JavaScript-də dəyər həmişə müəyyən tipdə olur. Məsələn, sətir və ya rəqəm.

JavaScript-də 8 deyer tipi var.

String : Her hansi text yazi yazilir  
Number : Reqem yazilir  
Bigint : Daha boyuk reqemleri yazmaq ucundur  
Boolean : “true” ve “false” deyerleri ucundur  
Undefined : Deyisene deyer teyin olunmadiqda  
Null : Ozu var amma icinde hec bir deyer yoxdur  
Symbol : Yunik deyerler ucun isdifade olunur  
Object : birden cox datani ozunde saxlayir

String : var userName = "Vuqar";

Number: var age = 25;

Bigint: var x = 999999999999999;

Boolean: bool isProgrammingFun = true;

bool isProgrammingFun = false;

Undefined: var variable = undefined;

Ve ya : var variable;

Null: var maybeNull = null

Object:

var user = {

    fullName : "Ilkin",

    age : 24,

    Pl : "PHP"

};

Ve ya array :

var users = ["Ilkin", "Vuqar", "Taleh"];

* 5. Type cevrilmesi

Çox vaxt operatorlar və funksiyalar onlara verilən dəyərləri avtomatik olaraq düzgün tipə çevirir.

Məsələn, alert istənilən dəyəri göstərmək üçün onu avtomatik olaraq sətirə çevirir.

Bezen biz deyerin tipini cevirmeli oluruq. Boolean-dan Strinqe ve ya Strinqden Numbera ve s.

var value = true;

alert(typeof value); *// boolean*

value = String(value);

alert(typeof value); *//String*

Meslen stringi numbere ceviririk

var str = "321";

alert(typeof str); *// string*

var num = Number(str);

alert(typeof num); *// number*

String i Boolean cevirmek

var str = "12";

alert(typeof str); *// String*

var str = Boolean(str);

alert(typeof str ) *// Boolean*

* 6. Primitve vs Referance( Stack ve Heap nedir? , ferqi nedir? Ram nedir? ) Niye rama ehtiyac var?

JavaScript-də dəyişən yaratdığınız zaman həmin dəyişən iki növ məlumatdan birini, **primitiv** dəyər və ya **Referance** dəyərini saxlaya bilər. Əgər dəyər **number**, **string**, **boolean**, **undefined**, **null**və ya olarsa **symbol**, bu, **primitiv** dəyərdir. Əgər deyer typeof **object** , bu, **Referance**  dəyəridir.

Komputerin Rami muelimatlari muveqqeti saxlamaq ucun isdifade olunur. Ram vasitesi ile biz eyni anda bir nece proqramda isleye bilirk.

Ancaq computer sondukde Ramda olan muelimatlar silinir cunki ram enerjiden asli yaddasdir.

Ram 2 hisseden ibaret olur. Stack ve Heap

Biz bir **primitiv** dəyərli deyisen yaratdiqda o gelib Stack yaddasa dusur.

Sonra biz hemin deyiseni isdifade etmek isteyende adini yaziriq o gedib Ramdan (Stackdan) hemin datani cekib getirir.

Biz object yeni Referance deyerli deyisen yaratdiqda ise deyisenin ozu Stackde saxlanilir ancaq deyisenin deyeri ise Heap de saxlanilir.

Stackin hecmi heape nisbeten azdir. Stack heape nisbeten daha suretlidir. Stack yaddasin omru qisadir. Obuektler stack yaddasda saxlanila bilmez. Stack esasen thread yonumludur.

Heap yaddas daha boyuk hecme malikdir. Heap yaddasin omru uzundur. Obyektler heap yaddasda saxlanilir.