JavaScript nədir?

Javascript, webdə çox geniş istifadə edilən proqramlaşdırma dilidir. HTML və CSS ilə yaradılan web səhifələrdə yerləşdirilən elementlərin dinamikliyini təmin edən əsas texnologiyalardan biridir. Javascript, 1995-ci ildə brendan eich tərəfindən yaradılmışdır. Əsasən veb brauzerlərdə işləyir, bəzi mühitlərdə server tərəfində də istifadə oluna bilir. Ümumilikdə desək developerlərin veb səhifələri, tətbiqləri, serverləri və hətta oyunları inkişaf etdirərkən daha dinamik qarşılıqlı əlaqə yaratmaq üçün istifadə etdiyi sinxron və single-thread proqramlaşdırma (skript) dilidir. Yəni sətir-sətir işləyir və bir sətir yerinə yetirilənə və əməliyyat başa çatana qədər digərinə keçmir.

JAVASCRİPT BRAUZERDƏ ÇOXSAYLI FUNKSİYA VƏ İMKANLARI VAR

- **HTML** və **CSS** ilə qarşılıqlı əlaqə qurmağa imkan verir. **DOM** (Document Object Model) vasitəsilə səhifədəki elementləri daha dinamik, interaktiv hala gətirmək, dəyişdirmək olar.
- **Canvas API**: Dinamik qrafiklər və animasiyalar, təsvirlər, şəkillər və digər qrafik elementləri yaratmağa imkan verir. **WebGL:** Həmçinin 3D qrafika yaratmaq üçün geniş imkanlar təqdim edir.
- **Audio və Video**: Dinamik olaraq media fayllarını oynatma və idarə etməyə imkan verir. **WebRTC**: Canlı video və səsləri yaratmaq üçün istifadə olunur.
- **Geolocation API:** İstifadəçinin cari yerini əldə etməyə və bu məlumatı tətbiqdə istifadə etməyə imkan verir.
- **Formaların İdarə Edilməsi**
- **Məlumatların saxlanması** və s.

AMMA EYNİ ZAMANDA BƏZİ MƏHDUDİYYƏTLƏRƏ DƏ MALİKDİR.

- **Fayl Sistemi Girişi:** Brauzerdə JavaScript birbaşa fayl sisteminə daxil ola bilmir. Faylları oxumaq və ya yazmaq üçün istifadəçi icazəsi və ya xüsusi API-lər tələb olunur. **Server Tərəfi proseslər**: Məsələn, serverdəki məlumat bazasına birbaşa daxil olmaq. Bunun üçün Node.js kimi server tərəfi mühitlərdən istifadə edilir.
- **Təhlükəsizlik Məhdudiyyətləri :** Brauzer təhlükəsizlik səbəbi ilə eyni domenə məxsus olmayan resurslardan məlumat çəkməkdə məhdudiyyətlər tətbiq edir. CORS (Cross-Origin Resource Sharing) ilə həll oluna bilər. **Sistem Resurslarına Giriş**: JavaScript brauzer içində istifadəçi sisteminin hardware resurslarına (məsələn, kamera, mikrofon, fayl sistemi) birbaşa giriş əldə edə bilmir. Bu cür funksiyalar istifadəçi icazəsi ilə həyata keçirilir.
- **Performans Məhdudiyyətləri**: Brauzer mühitində bəzi ağır hesablama işləri üçün məhdud ola bilər. Belə hallarda, WebAssembly kimi əlavə texnologiyalardan istifadə etmək lazım ola bilər.
- **Dəstəklənməyən Köhnə Brauzerlər** və s

JAVASCRİPT-DƏ DƏYİŞƏNLƏR (VARİABLES) PROQRAM KODUNDAKI DƏYƏRLƏRİ SAXLAMAQ VƏ ONLARLA İŞLƏMƏK ÜÇÜN BİR VASİTƏDİR, STRUKTURDUR.

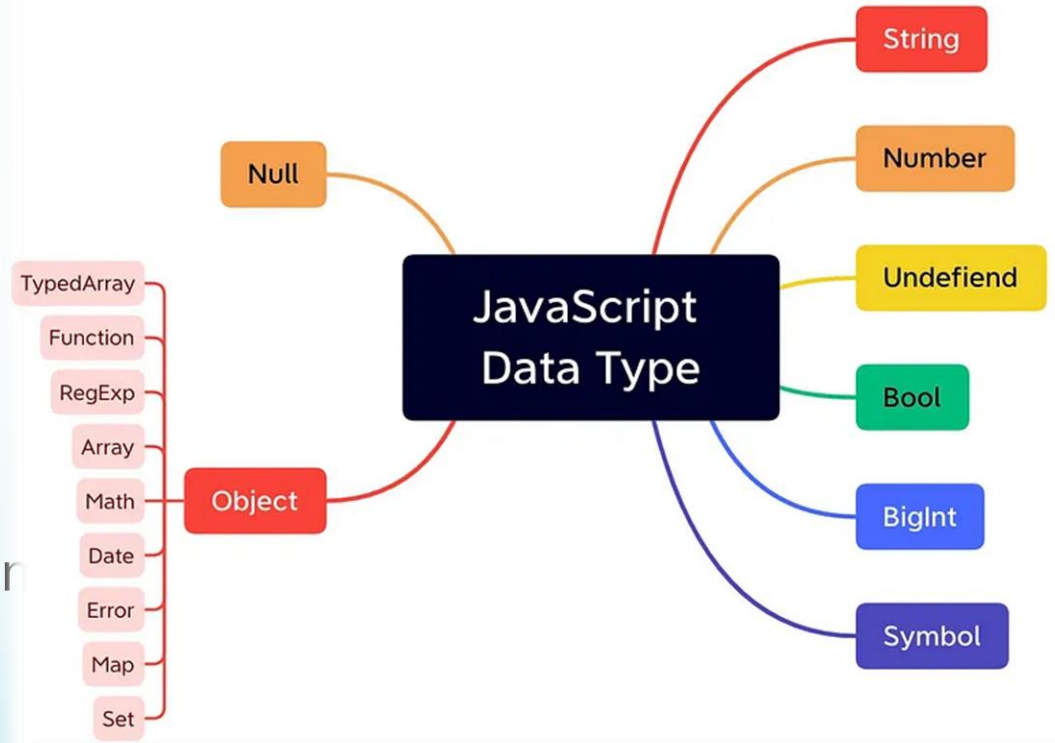
JavaScript-də dəyişənlər var, let, və const açar sözləri ilə təyin edilir. Hər birinin öz spesifik xüsusiyyətləri var:

- **var**: Köhnə JavaScript versiyalarında istifadə olunur. blok-səviyyəli deyil, yəni bir blok (məsələn, döngü və ya şərt bloku) daxilində təyin edilən **var** dəyişəni blokdan xaricdə də görünə bilər.
- **let**: let ilə təyin edilmiş dəyişənlər blok-səviyyəlidir, yəni bir blok daxilində təyin edilən let dəyişəni yalnız həmin blok daxilində görünür.
- **const**: Dəyişməz (immutable) dəyərləri saxlamaq üçün istifadə olunur. const ilə təyin edilən dəyişənin dəyəri təyin edildikdən sonra dəyişdirilə bilmir. Ancaq, obyektlər və massivlər kimi kompleks dəyərlərdə, onların özlükləri dəyişdirilə bilər.

JAVASCRİPT-DƏ DƏYİŞƏN TİPLƏRİ (DATA TYPES) MƏLUMATLARIN MÜƏYYƏN EDİR. DƏYİŞƏNLƏRİN TİPİ JAVASCRİPT AVTOMATİK OLARAQ TƏYİN EDİR VƏ DƏYİŞƏ BİLƏR. JAVASCRİPT-DƏ ƏSASƏN İKİ NÖV DƏYİŞƏN TİPİ MÖVCUDDUR:

- **Əsas (primitiv) tip** birbaşa dəyəri saxlayır və dəyişməz (immutable) xüsusiyyətə malikdir. Onlar birbaşa dəyişdirmək mümkün deyil və bir dəyişənin dəyəri digər dəyişənə təyin edildikdə, yeni dəyər təyin olunur: **number**, **string**, **boolean**, **undefined, null, symbol, bigint**
- **Obyekt (referans) tip** daha mürəkkəb məlumat strukturlarını təmsil edir. Obyektlər dinamik, dəyişkən (mutable) və ya bir neçə dəyərdən ibarət ola bilər: **object**, **array**, **function**, **date**, **regExp**, **map**, **set**

- Number let num1 = 103;
  let num2 = 20.5;
- String let str1 = “Aygun”
  let str2 = “My daughter name is Nazli”;
- Boolean let bool = true;
  let bool = false;
- Undefined let str ; console.log(str); → undefi
- Null let a = null; console.log(a); → 0
- Symbol let sym = Symbol('id’);
- BigInt let big_num = BigInt(99999999999999999999452456251);
- Array let array = [10, 20, 30, 40]; let names =[“Hecer”, “Zinyat”, “Tamilla”, “Aygun”]
- Object




```
let person = {
  name: "Aygun",
  age: 28,
  displayInfo: function() {
    console.log("CodeMarketing student " + this.name);
  }
};
```

JAVASCRİPT-DƏ TİPLƏRİN ÇEVRİLMƏSİ (TYPE CONVERSİON) MƏLUMATLARIN BİR TİPDƏN DİGƏRİNƏ ÇEVRİLMƏSİ PROSESİDİR.

- **String().** Stringə çevrilmə → **toString()**
- **Number().** Ədədi dəyərə çevrilmə → **parseInt()** və **parseFloat()**
- **Boolean().** Boolean dəyərə çevrilmə →JavaScript-də aşağıdakı dəyərlər **false** olaraq qiymətləndirilir: false, 0, -0, 0n, null, undefined, NaN, və boş string (""). Bütün digər dəyərlər **true** olaraq qiymətləndirilir.
- **valueOf().** Əsas dəyəri qaytarır.

Avtomatik çevrilməni minimuma endirmək və tiplərin də bərabərliyini yoxlamaq üçün === istifadə etmək tövsiyə olunur.

RAM (RANDOM ACCESS MEMORY) VƏ JAVASCRIPT

- **RAM** kompüter sistemində müvəqqəti məlumatların saxlanıldığı yaddaş növüdür. Bu yaddaş məlumatları sürətli şəkildə oxumaq və yazmaq üçün istifadə olunur və proqramların işləməsi üçün əhəmiyyətli rol oynayır. JavaScript proqramları RAM-də müxtəlif məqsədlər üçün yaddaş istifadə edir, o cümlədən:

✓ **Kod Yüklənməsi və İcra Edilməsi** JavaScript kodu RAM-ə yüklənir və burada icra olunur. JavaScript-də lokal dəyişənlər və funksiyalar stack yaddaşda saxlanılır. Funksiya çağırıldığında, bu funksiyanın dəyişənləri və məlumatlar stacka əlavə olunur və funksiyanın icrası bitdikdə, bu məlumatlar stackdən çıxarılır. Obyektlər və massivlər heap yaddaşında saxlanılır. Heap-də məlumatlar dinamik olaraq yaradılır və müstəqil olaraq idarə olunur. Obyektlər və massivlər JavaScript proqramında uzun müddət istifadə oluna bilər və bu səbəbdən heap-də saxlanılır.

STACK VƏ HEAP JAVASCRİPT-DƏ YADDAŞIN TƏŞKİLİNİ VƏ İDARƏ EDİLMƏSİNİ TƏMİN EDİR

- **Stack**: LIFO prinsipinə uyğun təşkil edilmişdir. Buradakı məlumatlar ardıcıllıqla əlavə edilir və çıxarılır. Stack-da Primitiv tiplər və funksiyaların lokal dəyişənləri saxlanılır. Yaddaş avtomatik idarə olunur. Funksiya çağırışları bitdikdə yaddaş avtomatik olaraq təmizlənir. Daha sürətli giriş və çıxış əməliyyatları təklif edir, çünki məlumatlar ardıcıllıqla idarə olunur.

- **Heap**: Dinamik və qeyri-strukturlaşdırılmış şəkildə təşkil edilmişdir. Məlumatlar sərbəst şəkildə yerləşdirilir və idarə olunur. Obyektlər, massivlər və digər mürəkkəb məlumat strukturları saxlanılır. Yaddaşın idarə edilməsi daha mürəkkəbdir. Garbage collection tərəfindən istifadə olunmayan yaddaşın təmizlənməsi həyata keçirilir. Daha mürəkkəb idarəetmə tələb edir və daha yavaş ola bilər, amma daha çox məlumatın saxlanmasına imkan verir.