

- JavaScript nedir?

- ✓ JavaScript, veb proqramlaşdırmada istifadə olunan proqramlaşdırma dilidir. Bu dil, həm sadə interaktiv xüsusiyyətlər, həm də kompleks veb tətbiqləri yaratmaq üçün istifadə edilə bilər. JavaScript Veb tətbiqləri, Mobil tətbiqlər, Server tərəfli proqramlaşdırma və brauzer əsaslı oyunlar üçün geniş istifadə olunur.

- Javascript browserde neleri ede bilir, neleri ede bilmir?

- ✓ Səhifələrin dinamik olmasını təmin edir. Klik, sürüşdürmə, klaviatura daxil etmələri kimi hadisələrə reaksiya vermək üçün istifadə olunur. Məsələn, düymə klikləndikdən sonra məzmunun dəyişdirilməsi və ya yeni elementlərin əlavə edilməsi mümkündür. JavaScript, brauzerlər tərəfindən şərh olunur və icra edilir, kompilyasiya etməyə ehtiyac yoxdur. CSS ilə birlikdə istifadə edilərək, səhifədə animasiyalar yaratmaq və ya görünüşünü dəyişdirmək üçün JavaScript-dən istifadə oluna bilər.
- ✓ JavaScript, brauzer daxilində işləyərkən kompüterin fayl sisteminə birbaşa giriş imkanı yoxdur. Bu, təhlükəsizliyi qorumaq üçün tətbiq edilir. Fərqli domenlərdən məlumat əldə etmə məhdudiyyəti tətbiq olunur ki bu da təhlükəsizliyi təmin etmək üçündür. JavaScript, birbaşa olaraq çoxlu tapşırıqların icrasını dəstəkləmir

- Javascript variables

- ✓ JavaScript-də Variables məlumatları saxlamaq üçün istifadə olunur. Dəyişənlər proqram boyunca müxtəlif dəyərləri yadda saxlamağa və onlara istinad etməyə imkan verir. Dəyişənlərin istifadəsi proqramlaşdırmanın əsasını təşkil edir, çünki onlar məlumatları saxlamaq, hesablama aparmaq və məntiqi əməliyyatları yerinə yetirmək üçün istifadə olunur.

- Variable type

- ✓ String- metn üçün istifadə olunur
`let name = "Turan";`
`let location = 'Nesimi rayonu';`
tek və ya iki dırnaq içərisində yazılır.
- ✓ Number- Rəqəmləri təmsil edir. JavaScript-də bütün rəqəmlər (tam ədədlər və onluq kəsrlər) eyni məlumat tipinə aiddir.
`let year = 25;`
`let price = 3.14159;`
- ✓ Boolean- yalnız iki mümkün dəyəri var, true və false.
`let oxunub = true;`
`let oxunmayıb = false;`

- ✓ Object- Daha mürəkkəb məlumat strukturlarını saxlamaq üçün istifadə olunan tiptir. Obyektlər daxilində başqa məlumat tiplərini və ya funksiyaları saxlaya bilər.
let player = { ad: "John", yas: 30 };
- ✓ Array- Bir neçə dəyəri tək bir dəyişəndə saxlamaq üçün istifadə olunan xüsusi bir obyekt tipidir. Array-lar sıralanmış siyahı kimi işləyir və müxtəlif növ məlumatları saxlaya bilər.
let numbers = [1, 2, 3, 4, 5];
let names = ["Alice", "Bob", "Charlie"];
- ✓ Undefined- Dəyişən elan edilib, amma ona heç bir dəyər təyin edilməyibsə, onun tipi undefined olur.
let teyinOlunmayan; console.log(teyinOlunmayan);
- ✓ Null- null dəyəri obyektin bilərəklə olmamasını təmsil edir. Bu, dəyişənin heç bir obyektə işarə etmədiyini göstərir.
let emty = null;
- ✓ Symbol- Unikaldır və dəyişdirilə bilməyendir.
- ✓ BigInt- JavaScript-də çox böyük rəqəmləri (Number tipinə sığmayan dəyərləri) təmsil etmək üçün istifadə olunur.
let bigNumber = 1234567890123456789012345678901234567890n;

- Type çevrilməsi

- ✓ Type çevrilməsi proqramlaşdırmada bir məlumat tipini digər bir məlumata çevirmək üçün istifadə olunan prosesdir. JavaScript-də bu çevirmə iki əsas formada baş verir: avtomatik (implicit) çevirmə və əl ilə (explicit) çevirmə.

Avtomatik (Implicit) Type Çevrilməsi

Əgər bir sətir və bir ədəd toplanarsa, JavaScript avtomatik olaraq ədədi sətirə çevirir və onları birləşdirir.

```
let number = 5;
let name = "10";
let name = number + name;
```

Əl ilə (Explicit) Type Çevrilməsi

Əl ilə type çevrilməsi, proqramçı tərəfindən məlumat tipinin dəyişdirilməsi prosesidir. Bu çevirmə JavaScript-də xüsusi funksiyalar və ya metodlar vasitəsilə həyata keçirilir:

```
let number = 123;
let name = String(number);
```

- Primitive vs Reference(Stack ve Heap nedir? , ferqi nedir? Ram nedir?) Niye rama ehtiyac var?

- ✓ JavaScript-də dəyişənlər iki əsas kateqoriyadan birinə aid ola bilər: Primitive tiplər və Reference tiplər.

Primitiv tiplər JavaScript-də sadə və dəyər əsaslı məlumatları təmsil edir. Bu tiplər birbaşa dəyişən tərəfindən saxlanılır və birbaşa dəyər kimi ötürülür. Primitiv dəyərlər kopyalanarkən, dəyişənin öz dəyəri kopyalanır. Bu o deməkdir ki, bir primitiv tipin digərinə kopyalanması zamanı hər iki dəyişən öz müstəqil dəyərlərini saxlayır.

Referans tiplər isə mürəkkəb məlumat strukturlarını təmsil edir. Referans tiplər birbaşa dəyəri deyil, həmin dəyərin yaddaşdakı yerini saxlayır.

Stack ve Heap nedir?

- JavaScript-də yaddaşın necə idarə olunduğunu başa düşmək üçün vacib anlayışlardır. Onlar fərqli yaddaş bölgələridir və müxtəlif məqsədlər üçün istifadə olunur.

Stack proqramın funksional və lokal məlumatlarını saxlamaq üçün istifadə olunur. Stack yaddaşı "Last In, First Out" (LIFO) prinsipi ilə idarə edilir, yəni sonuncu daxil olan ilk çıxarılır. yaddaşın idarə edilməsi çox sürətlidir və proqramın işini asanlaşdırır, çünki dəyişənlər avtomatik olaraq işə salınır və işin bitməsi ilə silinir.

Heap mürəkkəb məlumat strukturlarını saxlamaq üçün istifadə olunur. Heap yaddaşı dinamik yaddaş idarəsi üçün istifadə olunur.

Ferqi nedir?

- Stack avtomatik olaraq dəyişənlərin həyat dövrünü idarə edir, funksiya daxilində yaradılan dəyişənlər avtomatik olaraq silinir. Heap isə dinamik olaraq yaddaşı ayırır və bu yaddaşın təmizlənməsi Garbage Collection vasitəsilə həyata keçirilir. Stack daha sürətlidir, Heap daha yavaş ola bilər, çünki burada dinamik yaddaşın idarəsi daha mürəkkəbdir və təmizləmə prosesi əlavə vaxt tələb edir. Stack daha məhdud yaddaş həcminə malikdir, Heap isə daha geniş yaddaş həcmində məlumat saxlaya bilər və dinamik olaraq yaddaşı genişləndirmək imkanı təqdim edir.

Ram nedir?

- RAM (Random Access Memory) kompüterlərin və digər elektron cihazların yaddaş növüdür. RAM müvəqqəti yaddaşdır və proqramların və əməliyyatların işləməsi üçün lazım olan məlumatları saxlayır.

Niye RAM-a ehtiyac var?

- Prosesorun məlumatlara tez bir şəkildə çıxış etməsi üçün istifadə olunur, çünki RAM-dan məlumat oxumaq və yazmaq, sabit disklərdən və digər yaddaş növlərindən daha sürətlidir. Bu, sistemin sürətli işləməsi, çoxsaylı

proqramların eyni anda icra olunması və müxtəlif əməliyyatların effektiv yerinə yetirilməsi üçün vacibdir.