# **JavaScript Nəzəri Təlim Sənədi**

## 1. JavaScript nədir?

JavaScript veb proqramlaşdırma sahəsində istifadə olunan əsas dillərdən biridir. Əsas məqsədi veb səhifələrə interaktivlik əlavə etməkdir. JavaScript vasitəsilə istifadəçi ilə qarşılıqlı əlaqə, form məlumatlarının yoxlanması, dinamik məzmunun göstərilməsi, animasiyalar, hadisələrə reaksiya və daha çoxunu etmək mümkündür. Əvvəllər yalnız brauzerlərdə işləyən bir dildi, lakin Node.js texnologiyası ilə artıq server tərəfli proqramlaşdırmada da geniş istifadə olunur. JavaScript müasir vebin əsas sütunlarından biridir və HTML, CSS ilə birlikdə istifadə olunur.

## 2. JavaScript-də məlumat tipləri

JavaScript-də məlumat tipləri iki əsas qrupa bölünür: sadə (primitive) və mürəkkəb (reference) tiplər. Sadə tiplərə mətnlər, ədədlər, məntiqi dəyərlər, null, undefined kimi tiplər daxildir. Bu tiplər birbaşa dəyəri saxlayır və yaddaşda birbaşa dəyər olaraq işlənir. Mürəkkəb tiplərə isə massivlər, obyektlər və funksiyalar daxildir. Bunlar istinadla işlənir, yəni yaddaşda saxlanan yerə işarə edirlər. Proqramçı bu tipləri tanımağı və onlarla düzgün işləməyi öyrənməlidir.

## 3. var, let və const dəyişənləri

JavaScript-də dəyişənləri elan etmək üçün əsas üç açar söz var: var, let və const. var əvvəllər geniş istifadə olunsa da, bəzi problemlərə səbəb olduğu üçün müasir yazılışda artıq let və const üstünlük təşkil edir. let dəyişənlərin dəyərinin sonradan dəyişdirilməsinə imkan verir və blok səviyyəsində tanınır. const isə sabit dəyərləri saxlamaq üçün istifadə olunur və bir dəfə təyin edildikdən sonra dəyəri dəyişdirilə bilməz. Hər ikisi daha təhlükəsiz və proqnozlaşdırıla bilən davranışa malikdir. var isə funksion səviyyəli olduğu üçün bəzən gözlənilməyən nəticələrə səbəb ola bilər.

## 4. Operatorlar

JavaScript-də operatorlar proqramın məntiqini qurmaq və müxtəlif əməliyyatları icra etmək üçün istifadə olunur. Riyazi operatorlar ədədlər üzərində hesablamalar aparmaq üçün, təyinat operatorları dəyişənlərə dəyər mənimsətmək üçün, müqayisə operatorları isə dəyərləri bir-biri ilə müqayisə etmək üçün istifadə edilir. Məntiqi operatorlar daha çox şərt bloklarında istifadə olunur və birdən çox şərtin eyni anda qiymətləndirilməsinə imkan verir. Bunlardan əlavə typeof, instanceof kimi xüsusi operatorlar da var ki, onlar da məlumat tipi və obyekt tiplərinin yoxlanılmasında əhəmiyyətlidir.

## 5. Adlandırma qaydaları

JavaScript-də adlandırma qaydaları kodun oxunaqlığını və saxlanmasını asanlaşdırır. Funksiya və dəyişən adları mənalı olmalı və nə etdiyini açıqlamalıdır. Müxtəlif stil yanaşmaları mövcuddur: camelCase – yəni ilk söz kiçik, sonrakılar böyük hərflə başlayır; PascalCase – bütün sözlər böyük hərflə başlayır, daha çox class adlarında istifadə olunur; və snake\_case – sözlər alt xətlə ayrılır, bəzi hallarda istifadə olunur. Dəyişən adları rəqəmlə başlamamalıdır və xüsusi simvollar ehtiva etməməlidir. Proqramda dəyişən və funksiyaların adlarını qısa, ancaq aydın saxlamaq tövsiyə olunur.

## 6. if/else, switch və ternary operatorları

Şərt operatorları proqramda qərar vermə məntiqini təşkil edir. if/else operatoru müəyyən bir şərtin doğru olub-olmadığını yoxlayır və nəticəyə görə fərqli əmrləri icra edir. Bu struktur çox geniş istifadə olunur. switch operatoru isə birdən çox mümkün dəyəri yoxlamaq üçün daha əlverişlidir, xüsusilə eyni dəyişənin bir neçə mümkün dəyəri varsa. Ternary operatoru isə daha qısa formada if/else funksionallığını təmin edir və əsasən bir sətrə sığacaq qədər sadə məntiq üçün nəzərdə tutulub. Bu üsullar şərtlərin daha effektiv şəkildə yazılmasına və oxunmasına imkan yaradır.

## 7. Dövrlər: for, while, do...while, break, continue

Dövrlər (loop) proqramda müəyyən əməliyyatları təkrar-təkrar icra etmək üçün istifadə olunur. for dövrü əvvəlcədən məlum olan təkrar sayı üçün uyğundur. while dövrü isə şərt doğru olduğu müddətdə işləyir və əvvəlcədən təkrar sayı məlum olmayan hallarda istifadə olunur. do...while dövrü isə əvvəlcə əməliyyatı icra edir, sonra şərti yoxlayır, yəni ən azı bir dəfə işləyir. break əmri dövrü vaxtından əvvəl dayandırmaq, continue isə dövrün növbəti iterasiyasına keçmək üçün istifadə olunur. Bu vasitələr proqram axınına nəzarəti artırır.

## 8. Massivlər (arraylar)

Massiv – birdən çox dəyəri bir dəyişəndə saxlamaq üçün istifadə olunan məlumat strukturudur. Massivlər ardıcıl indekslərə malikdir və sıralı şəkildə işləyirlər. Massivin elementlərinə indeks vasitəsilə müraciət edilir. Massivlərdə məlumat əlavə etmək, silmək, dəyişmək üçün müxtəlif metodlar mövcuddur. Bunlara push, pop, shift, unshift, splice, map, filter və başqaları daxildir. Massivlər həm sadə, həm də mürəkkəb verilənləri saxlaya bilir və praktiki proqramlaşdırmada çox geniş istifadə olunur.

## 9. Obyektlər

Obyektlər JavaScript-də daha mürəkkəb məlumatların saxlanması üçün əsas struktur hesab olunur. Obyektlər açar-dəyər (key-value) cütlərindən ibarətdir. Onlar real dünya varlıqlarını modelləşdirmək üçün çox əlverişlidir. Məsələn, bir istifadəçi obyektində onun adı, yaşı, email ünvanı və digər məlumatları saxlanıla bilər. Obyektlərdə həm dəyərlər, həm də funksiyalar (metodlar) saxlanıla bilər. Obyektlərin strukturu çox elastikdir və dinamik şəkildə yeni sahələr əlavə edilə və ya mövcud sahələr dəyişdirilə bilər.

## 10. Funksiyalar

Funksiya – müəyyən əməliyyatı yerinə yetirmək üçün yazılmış kod blokudur. JavaScript-də funksiyalar proqramın təkrar-təkrar işləyəcək hissələrini daha rahat və strukturlu şəkildə idarə etməyə imkan verir. Funksiya yazmağın bir neçə forması var: ənənəvi funksiyalar, funksiya ifadələri və ox funksiyaları (arrow functions). Hər birinin öz sintaksisi və istifadəsi mövcuddur. Funksiyalar parametr qəbul edə və nəticə qaytara bilər. JavaScript funksiyaları birinci səviyyəli obyektlərdir və başqa funksiyalara ötürülə və geri qaytarıla bilər.

## 11. Template literal (şablon ifadələr)

Şablon ifadələr JavaScript-də mətn birləşdirmə və dəyişənlərin mətndə istifadəsini daha rahat və oxunaqlı hala gətirmək üçün nəzərdə tutulub. Əvvəllər mətn birləşdirmə + işarəsi ilə edilirdisə, indi backtick (əks dırnaq) işarələri ilə daha təmiz yazılır. Bu sintaksis dəyişənləri ${} içində yaza bilməyə imkan verir. Həmçinin birdən çox sətirlik mətnlər yazmaq üçün də çox uyğundur. Bu, xüsusilə istifadəçiyə yönəlik mesajlar, şablonlar və HTML ilə işləyərkən faydalıdır.

## 12. Sətir (string) metodları

JavaScript-də mətnlər (string) ilə işləmək üçün geniş metod dəsti mövcuddur. Bu metodlar vasitəsilə mətn uzunluğunu öyrənmək, böyük və ya kiçik hərflərə çevirmək, hissə hissə bölmək, daxilində müəyyən sözün olub olmadığını yoxlamaq və digər əməliyyatlar aparmaq mümkündür. String metodları proqramda istifadəçid

ən gələn məlumatların yoxlanması və formatlaşdırılması zamanı çox önəmlidir.

## 13. Hoisting anlayışı

Hoisting – JavaScript-in daxili davranış qaydalarından biridir. Bu anlayışa görə, dəyişən və funksiyaların elan hissəsi avtomatik olaraq onların yuxarı hissəyə “qaldırılması” kimi başa düşülür. var ilə elan olunan dəyişənlər hoist olunur, lakin onların dəyəri undefined olur. let və const ilə elan olunan dəyişənlər də hoist olunur, amma onlar istifadə edilənədək aktiv vəziyyətdə olmurlar və istifadə olunarsa səhv (ReferenceError) baş verir. Bu səbəbdən proqramda dəyişənləri həmişə əvvəlcədən elan etmək yaxşı təcrübə sayılır.