

1 AZƏRBAYCAN RESPUBLİKASI ELM TƏHSİL NAZİRLİYİ
AZƏRBAYCAN TEXNİKİ UNİVERSİTETİ

Əlyazması hüququnda
Durmuşov Vüsal Mehman oğlu
İsayev Orxan Elman oğlu

**Javascript proqramının tətbiqlə informasiya təhlükəsizliyi üsulunun işlənməsi
mövzusunda**

MAGİSTRİK DİSSERTASİYASI

**İxtisas: 060627 - "Elektronika , telekommunikasiya və radiotexnika
mühəndisliyi"**

İxtisaslaşma: "Telekommunikasiya sistemlərinin informasiya təhlükəsizliyi"

Elmi rəhbər:

t.e.n., dosent Məmmədov Elçin Sabir oğlu

MÜNDƏRİCAT

Giriş	3
I FƏSİL. Javascript proqramlaşdırma dilinin tətbiq sahələri , digər dillərlə müqayisəsi və günümüzdəki aktuallığı	6
1.1 Proqramlaşdırma dedikdə nə başa düşülür?	6
1.2 Proqramlaşdırma dilləri nəyə görə vacibdir	7
1.3 Proqramlaşdırma dillərinin tətbiq sahələri.....	8
1.4 Javascript-in texniki xarakteristikaları.....	10
1.5 HTML-in texniki xarakteristikaları.....	12
1.6 CSS-in texniki xarakteristikaları.....	14
1.7 Javascript in tətbiq sahələri.....	20
1.8 Javascript ilə işləmək təhlükəsizlik cəhətdən məqsədə uyğundurmu	24
1.9 Javascript proqramlaşdırma dillərinin digər dillər ilə müqayisəsi	27
II FƏSİL. İnformasiya təhlükəsizliyinin tətbiq sahələri, metodları və günümüzdəki aktuallığı	32
2.1 İnformasiya təhlükəsizliyi nədir	32
2.2 Simmetrik şifrələmə nədir.....	33
2.3 Assimmetrik şifrələmə.....	33
2.4 Rəqəmsal imza	35
2.5 Şifrələmə metodlarının fərqli növləri.....	37
2.6 Sezar şifrələmə nədir ?	42
2.7 Kodlaşdırılma əməliyyatının alqoritmik blok sxeminin qurulması.....	45
III FƏSİL. Kodlama əməliyyatının Visual Studio Code proqramı vasitəsi ilə yazılması və alınan nəticələr	48
3.1 Kodlaşdırma yoxsa dekodlaşdırma prosesinin öyrənilməsi.	48
3.2 Kodlamada addım istiqamətinin öyrənilməsi mərhələsi.....	49
3.3 Düzgünlüyün yoxlanılması prosesi:.....	51
3.4 Dekodlaşdırma mərhələsi	51
3.5 Doğruluğun yoxlanılması mərhələsi.....	53
Nəticə	54
Ədəbiyyat	55
İxtisarlarmın siyahısı	57

GİRİŞ

Mövzunun aktuallığı. Fərdlərin və təşkilatların texnologiyadan artan istifadə ilə qarşılaşdıqları çoxsaylı çətinlikləri nəzərə alaraq, informasiya təhlükəsizliyi müasir dünyanın vacib aspektinə çevrilmişdir. İnformasiya təhlükəsizliyinin aktuallığı həssas məlumatları icazəsiz daxil olmaqdan, dəyişdirilməkdən və ya məhv olmaqdan qorumaqdır. İnformasiya təhlükəsizliyi məlumatların yalnız səlahiyyətli işçilər üçün əlçatan olmasını təmin edir və xarici və ya daxili mənbələrdən gələ biləcək hücumlardan qorunur.

Bu dissertasiya işində biz müxtəlif kontekstlərdə, o cümlədən şəxsi, biznes və hökumət mühitlərində informasiya təhlükəsizliyinin əhəmiyyətini müzakirə edəcəyik. Şifrələmə üsulları məlumatların təhlükəsizliyini təmin etmək üçün son dərəcə vacibdir. Fərqli metodların üstünlükləri və mənfi cəhətləri var və hər bir üsul istifadə ediləcəyi vəziyyətə uyğun seçilməlidir. Yaxşı şifrələmə üsulu məlumatların məxfiliyini qoruya, məlumatların bütövlüyünü təmin edə və icazəsiz girişdən qoruya bilər. Bununla belə, şifrələmə üsulları texnologiyanın inkişafı ilə daim yenilənir və yenilənir. Buna görə şifrələmə üsulları haqqında məlumatlı olmaq və yeniləmələri izləmək vacibdir. İnformasiya təhlükəsizliyinin ən mühüm üstünlüklərindən biri şəxsi məlumatların qorunmasıdır. Şəxsi məlumatlara sosial təminat nömrələri, kredit kartı məlumatları və tibbi qeydlər kimi fərdi şəxsiyyəti müəyyənləşdirən hər hansı məlumat daxildir. Texnologiyanın artan istifadəsi ilə fərdi məlumatlar indi elektron şəkildə saxlanılır və ötürülür ki, bu da onu kiberhücumlara qarşı həssas edir. İnformasiya təhlükəsizliyi şəxsi məlumatların şifrələnməsini və icazəsiz girişdən qorunmasını təmin edir, şəxsiyyət oğurluğu və digər saxtakarlıq formalarının qarşısını alır. Kiberhücumlar bu gün fərdlərin və təşkilatların üzləşdiyi ən mühüm təhlükələrdən biridir.

Tədqiqatın məqsədi. Qarşıya qoyulan əsas məqsəd isə informasiya təhlükəsizliyi artırmaqdır. İnformasiyanın yalnız səlahiyyətli işçilər üçün əlçatan olunmasının təmin edilməsidir. Proqramlaşdırma dili olan Javascript (JS) ilə əsas sahələrdən biri olan informasiya təhlükəsizliyini bir vəhdət şəklində ifadə edərək informasiya təhlükəsizliyini artırmaqdır.

Tədqiqatın obyektı. Tədqiqatın obyektı kodlaşdırılan və dekodlaşdırma əməliyyatı aparılan informasiyadır.

Tədqiqatın predmeti. Qəbul olunmuş şifrələnmiş məlumat və göndərilən məlumat.

Tədqiqatın həqiqiliyi. Dissertasiya işində aparılmış tədqiqatlar, həmçinin alınan nəticələr mövcud ədəbiyyatlarda verilənlərlə tamamilə uzlaşır.

İşin elmi yeniliyi. Proqramlaşdırma dili olan Javascriptdən istifadə edərək Sezar şifrələmə metodunun alqoritmik blok sistemi yenidən qurulmuş, kodlaşdırılmış və informasiya təhlükəsizliyi artırılmışdır.

Müdafiəyə çıxarılan əsas müddəalar.

Müdafiəyə əsasən aşağıdakılar çıxarılır:

- Javascript proqramlaşdırma dili ilə informasiyanın kodlaşdırılması və dekodlaşdırılması
- İnformasiya kodlaşdırılan zaman hansı alqoritmik addımlardan istifadə olunur.
- İnformasiya təhlükəsizliyinin artırılması metodları.

Dissertasiya işinin təcrübü əhəmiyyəti.

Dissertasiya işində təcrübü əhəmiyyət ondan ibarətdir ki, Javascript proqramlaşdırma dili ilə kodların alqoritməyə uyğun yazılışı və təhlükəsizliyin bir üst səviyyəyə qaldırılmasıdır. Dissertasiya işində tədqiq və təklif olunan proqramlaşdırma dilinin tətbiqi ilə informasiya təhlükəsizlik üsulunun işlənməsi metodu bu sahəni yeni öyrənən şəxslərdə, təhlükəsizlik metodunun hansı ardıcılıq ilə və hansı formada getdiyini və hansı alqoritmik addımlardan istifadə edərək davam etdirildiyini görmək üçün gözəl təcrübü yoldur. Həmçinin proqramlaşdırma dili ilə informasiya təhlükəsizliyi arasında hansı əlaqənin olmasını necə bir kod ardıcılığının istifadə olunmasının öyrənilməsinə imkan verir.

Dissertasiya işinin strukturu və həcmi.

Dissertasiya işi girişdən, 3 fəsildən, nəticədən, 21 sayda ədəbiyyat siyahısından ibarətdir. İşin əsas hissəsi 50 səhifə mətndən, 22 şəkildən və 3 cədvəldən ibarətdir. İşin ümumi həcmi 59 səhifədir.

Girişdə mövzunun aktuallığı, işin məqsədi, onun elmi yeniliyi və təcrübi əhəmiyyəti əsaslandırılmışdır.

Birinci fəsildə Javascript proqramlaşma dili haqqında və onun başqa proqramlaşdırma dilləri ilə müqayisəsinin aparılmasını və javascript proqramlaşdırma dili hansı sahələrdə nə məqsədlə istifadə olunmasının araşdırılması aparılmışdır.

İkinci fəsildə informasiya təhlükəsizliyinin nə olduğunu, nə məqsədlə aparılmasını, hansı informasiya təhlükəsizlik metodlarının olmasını, keçmişdən bu günə nə kimi dəyişikliklərin olmasının araşdırılması aparılmışdır.

Üçüncü fəsildə Kodlama əməliyyatının Visual Studio Code proqramı vasitəsi ilə yazılmasını, alqoritmin qurulmasını və izahını bununla yanaşı alınan nəticələri ekranda göstərməklə alqoritmin hansı ardıcılıqda getməsini vizual olaraq izlədik. Kodlaşdırılma məntiqini və ardıcılığını araşdırdıq.

I FƏSİL.

JS proqramlaşdırma dilinin tətbiq sahələri digər dillər ilə müqayisəsi və günümüzdəki aktualığı.

1.1 Proqramlaşdırma dedikdə nə başa düşülür ?

Proqramlaşdırma dilləri müasir texnologiyanın təməl daşındır. Onlar bizə gündəlik həyatımızda vacib olan proqram təminatı, veb-saytlar və digər rəqəmsal alətlər yaratmağa imkan verir. Proqramlaşdırma dili proqramçıya müəyyən tapşırıqları yerinə yetirmək üçün kompüterlə əlaqə saxlamağa imkan verən qaydalar toplusudur. Bu yazıda proqramlaşdırma dillərinin nə olduğunu, necə işlədiyini və müasir dünyada əhəmiyyətini dərinlən öyrənəcəyik.

Proqramlaşdırma dili, kompüterə xüsusi tapşırıqları yerinə yetirmək üçün göstəriş verən kodu yazmaq üçün istifadə olunan rəsmi dildir. Proqramı yerinə yetirmək üçün kompüter tərəfindən şərh edilə bilən qaydalar və sintaksis toplusudur. Çoxlu proqramlaşdırma dilləri var, hər birinin öz sintaksisi və qaydaları var. Proqramlaşdırma dilləri proqram təminatı, veb-saytlar və digər rəqəmsal alətlər yaratmaq üçün istifadə olunur. Onlar həmçinin alqoritmlər hazırlamaq və elmi hesablamaları yerinə yetirmək üçün istifadə olunur. Yüksək səviyyəli, aşağı səviyyəli və skript dilləri də daxil olmaqla müxtəlif növ proqramlaşdırma dilləri var. Yüksək səviyyəli proqramlaşdırma dilləri insanların oxuması və yazması asan olması üçün nəzərdə tutulmuşdur. Onlar tez-tez proqramlar və veb saytlar yaratmaq üçün istifadə olunur. Digər tərəfdən aşağı səviyyəli proqramlaşdırma dilləri maşınların oxuması və icrası asan olması üçün nəzərdə tutulmuşdur. Onlar əməliyyat sistemləri və proqram təminatı yaratmaq üçün istifadə olunur. Skript dilləri veb-saytın saxlanması və ya məlumatların təhlili kimi tapşırıqları avtomatlaşdıran skriptlər yaratmaq üçün istifadə olunur.

Proqramlaşdırma dilləri necə işləyir?

Proqramlaşdırma dilləri həmin dilə xas olan bir sıra qaydalar və sintaksisdən istifadə etməklə yazılır. Bu qaydalar və sintaksis həm insanlar, həm də kompüterlər

tərəfindən asanlıqla başa düşülməsi üçün nəzərdə tutulmuşdur. Proqramçı proqramlaşdırma dilindən istifadə edərək kod yazdıqda, kod JavaScript üçün .js və ya Python üçün .py kimi xüsusi genişlənmə ilə faylda saxlanılır. Daha sonra kod kompilyator və ya tərcüməçi adlanan proqram proqramı tərəfindən tərtib edilir və ya şərh edilir. Kompilyator və ya tərcüməçi kodu oxuyur və onu kompüter tərəfindən yerinə yetirilə bilən maşın koduna çevirir. Bu maşın kodu kompüterin başa düşə və icra edə biləcəyi bir və sıfırlardan ibarət ikili dildir.

1.2 Proqramlaşdırma dilləri nəyə görə vacibdir?

Proqramlaşdırma dilləri müasir dünyada vacibdir, çünki onlar bizə hər gün milyonlarla insanın istifadə etdiyi proqram təminatı və veb-saytlar yaratmağa imkan verir. Proqramlaşdırma dilləri olmasaydı, Windows və ya macOS kimi əməliyyat sistemlərimiz və ya Microsoft Office və ya Google Chrome kimi proqramlarımız olmazdı. Proqramlaşdırma dilləri elmi tədqiqatlarda və məlumatların təhlilində də istifadə olunur. Onlar alimlərə alqoritmlər hazırlamağa və onlarsız mümkün olmayan simulyasiyalar həyata keçirməyə imkan verir. Bundan əlavə, proqramlaşdırma dilləri səhiyyə, maliyyə və nəqliyyat kimi sahələrdə istifadə olunan maşın öyrənmə modellərini yaratmaq üçün istifadə olunur.

Yaşadığımız əsrdə proqramlaşdırma dilləri həyatımızın ayrılmaz hissəsinə çevrilib. Kompüter və smartfonlarımızda istifadə etdiyimiz proqram təminatından onlayn ziyarət etdiyimiz veb-saytlara qədər hər gün qarşılıqlı əlaqədə olduğumuz texnologiyaları yaratmaq üçün proqramlaşdırma dillərindən istifadə edilir. Bəs proqramlaşdırma dili tam olaraq nədir?

Özündə proqramlaşdırma dili kompüterə və ya digər maşına göstərişlər vermək üçün istifadə edilən rəsmi dildir. Bu təlimatlar kodla yazılır, daha sonra kompüter tərəfindən yerinə yetirilə bilən maşın tərəfindən oxuna bilən təlimatlara çevrilir. Proqramlaşdırma dilləri proqram proqramlarının işlənilməsi, hazırlanması, veb-saytların və veb proqramların yaradılması, mobil proqramların yaradılması, oyunların dizaynı və məlumatların təhlili daxil olmaqla geniş çeşidli proqramlar üçün istifadə

olunur. Bəzi populyar proqramlaşdırma dillərinə Java, Python, C++, JavaScript və Ruby və digərləri daxildir. Proqramlaşdırma dillərinin əsas xüsusiyyətlərindən biri onların sintaksisidir ki, bu da kod yaratmaq üçün istifadə olunan qaydalar və simvollar toplusuna istinad edir. Sintaksis proqramçılara həm digər proqramçılar, həm də maşınlar üçün oxuna bilən və başa düşülən kod yazmağa imkan verən şeydir. Proqramlaşdırma dillərini iki əsas kateqoriyaya bölmək olar: aşağı səviyyəli və yüksək səviyyəli dillər. Assembly dili kimi aşağı səviyyəli dillər aparat resursları üzərində daha birbaşa nəzarəti təmin edir, lakin ümumiyyətlə öyrənmək və istifadə etmək daha çətinidir. Python və Java kimi yüksək səviyyəli dillər daha mücərrəddir və mürəkkəb proqramların yaradılması üçün daha güclü alətlər təqdim edir. Proqramlaşdırma dilləri sintaksisi və abstraksiya səviyyəsi ilə yanaşı, təyinatına görə də fərqli ola bilər. Bəzi dillər ümumi məqsədli proqramlaşdırma üçün nəzərdə tutulmuşdur, digərləri isə veb inkişafı və ya məlumatların təhlili kimi xüsusi proqramlar üçün ixtisaslaşmışdır. Ümumiyyətlə, proqramlaşdırma dilləri proqram təminatının inkişafı və rəqəmsal innovasiyalar üçün vacib vasitədir. Texnologiya inkişaf etməyə davam etdikcə, proqramlaşdırma dilləri, şübhəsiz ki, ətrafımızdakı dünyanın formalaşmasında getdikcə daha mühüm rol oynayacaq.

1.3 Proqramlaşdırma dillərinin tətbiq sahələri.

Proqramlaşdırma dilləri kompyuterin dilidir. Proqramçılar proqram təminatını yaratmaq və təkmilləşdirmək üçün scriptlərdən, təlimatlardan bir sözlə kodlardan istifadə edir. Baxmayaraq ki proqramlaşdırma dilləri bir birinə bənzəyir hər birinin özünə xas sintaksisi vardır. Bir proqramçı proqram dilinin qaydalarını, sintaksisini, əsasını öyrənir və text editorunda tətbiq edir. Text editorunda kodlar yazılır və bu kodlar kompyuterin başa düşüləcəyi dilə çevrilir. Bu gün yüzlərlə proqramlaşdırma dili mövcüddür.

- Tətbiq və proqram inkişafı

Tətbiq və proqram inkişafı gündəlik işlədiyiniz proqramları əhatə edir. Məsələn, bu veb-səhifəyə baxmaq üçün istifadə etdiyiniz İnternet brauzeri proqram hesab olunur.

Proqram hazırlamaqda maraqlısınızsa, aşağıdakı dilləri nəzərdən keçirin: C , C#, C++, D, Java, Swift, Visual Basic

- Süni intellektin inkişafı

Süni intellekt və ya əlaqəli sahələr kompüter oyunlarında, qərar qəbul edən proqramların hissələrində, chatbotlarda və s. xarakterlərin qarşılıqlı əlaqəsini yaratmağı əhatə edir. Əgər süni intellekt inkişaf etdirməkdə maraqlısınızsa, aşağıdakı dilləri nəzərdən keçirin: C , C#, C++, Prolog, Phyton.

- Verilənlər bazasının inkişafı

Verilənlər bazası tərtibatçıları verilənlər bazası yaradır və saxlayır. Əgər verilənlər bazası yaratmaq və ya saxlamaqla maraqlanırsınızsa, aşağıdakı dillərdən birini nəzərdən keçirin: Dbase, FoxPro, MySql, Sql, Visual FoxPro

- Oyun inkişafı

Oyunların hazırlanması kompüter oyunları və ya digər əyləncə proqramlarının yaradılmasını nəzərdə tutur. Bir oyun hazırlamaqda maraqlısınızsa, aşağıdakı dilləri nəzərdən keçirin: C , C#, C++, DarkBasic, Java

- Kompüter sürücüləri və ya digər hardware inkişafı

Kompüter sürücüləri və proqramlaşdırma aparat interfeysi dəstəyi hardware funksionallığı üçün zəruridir. Avadanlıq cihazları üçün sürücülər və ya proqram interfeysləri hazırlamaqda maraqlısınızsa, aşağıdakı dilləri nəzərdən keçirin: Assembly, C

- İnternet və veb sahifənin inkişafı

İnternet və veb sahifənin inkişafı İnternetin mahiyyətini təşkil edir. Tərtibatçılar olmasaydı, İnternet mövcud olmazdı. Veb sahifələr, İnternet proqramları və ya İnternetlə əlaqəli digər tapşırıqlar yaratmaqla maraqlanırsınızsa, aşağıdakı dilləri nəzərdən keçirin: Html, Java, Javascript, Perl, Php, Phyton, Xml

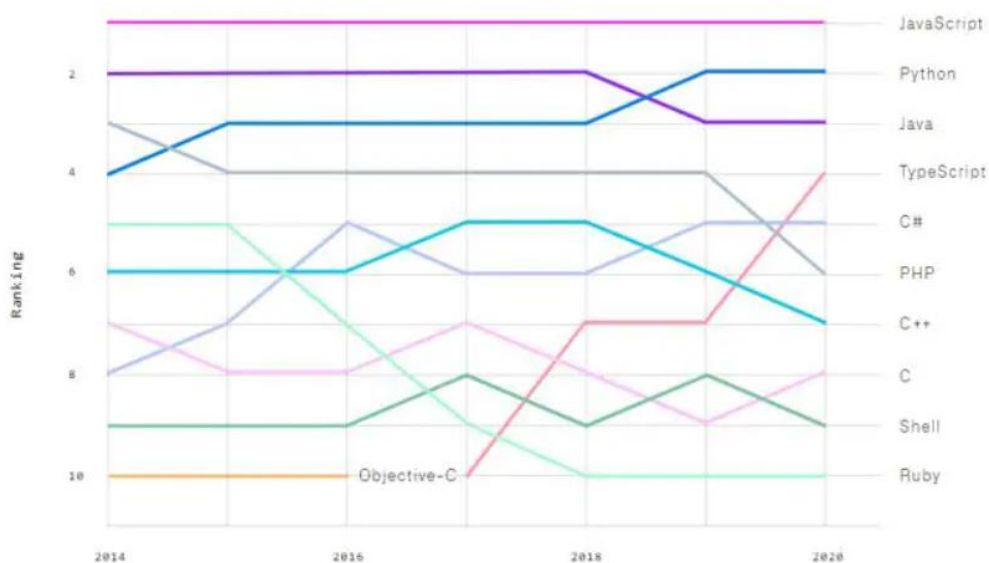
- Skript inkişafı

Karyeraya çevrilmə ehtimalı olmasa da, skriptləri necə yaratmağı və inkişaf etdirməyi bilmək sizin və ya şirkətiniz üçün məhsuldarlığı artıraraq, saysız-hesabsız saatlara qənaət edə bilər. Skriptlər hazırlamaqda maraqlısınızsa, aşağıdakı dilləri nəzərdən keçirin: AutoHotkey, awk, bash, Batch file, Perl, Phyton, Tcl

Neçə proqramlaşdırma dili var ?

Kompüterlər inkişaf etdikcə, müxtəlif inkişaf növləri üçün bir çox fərqli kompüter proqramlaşdırma dilləri yaradıldı. Baxmayaraq ki, saytımızda bir neçə onlarla dil sadalanıb, yüzlərlə dil var. Fərqli proqramlaşdırma dilləri haqqında fikir əldə etmək üçün siz 591-dən çox müxtəlif proqramlaşdırma dillərində "Salam Dünya"nın necə yazılacağını göstərən demo təqdim edən The Hello World Collection-a baş çəkə bilərsiniz.

Şəkil 1-də siz ilk 10 dili görə bilərsiniz: JavaScript, Java, Python, PHP, C++, C#, TypeScript, Shell, C və Ruby.



Şəkil 1.3.1 Müxtəlif proqramlaşdırma dillərinin illərə görə populyarlığı qrafiki

1.4 JavaScript-in texniki xarakteristikaları.

Brendan Eich tərəfindən hazırlanmış və ilk olaraq LiveScript kimi tanınan proqramlaşdırma dili JavaScript 1995-ci ildə dəyişdirilmiş və indi Oracle-ın ticarət nişanıdır.

JavaScript veb-dizaynerə öz veb səhifəsinə kod daxil etmək imkanı verən şərh edilmiş müştəri tərəfi skript dilidir. JavaScript adətən HTML və ya ASP faylına yerləşdirilir və birbaşa veb-səhifədən işləyir və bu gün ən populyar proqramlaşdırma dilidir. JavaScript vaxt və tarixi çap etmək, təqvim yaratmaq və ya HTML-də mümkün olmayan digər tapşırıqları yerinə yetirə bilər.

2020 Stack Overflow məlumatlarına görə Javascript-in son səkkiz ilin ən çox istifadə olunan proqramlaşdırma dili olduğunu bilirsinizmi? Buna görə də bugünkü

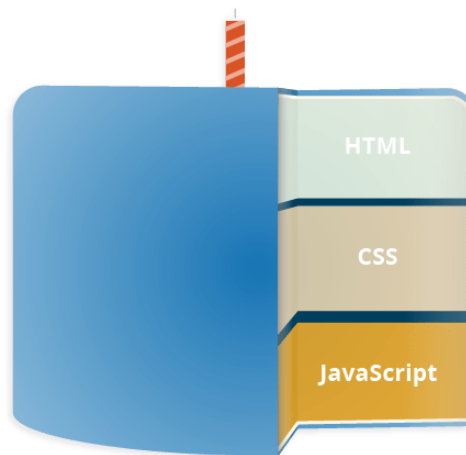
“JavaScript nədir?” sualına cavab vermək və bu dilə Google, Facebook, Alibaba, Microsoft kimi texnologiya nəhəngləri tərəfindən üstünlük verilməsinin və uzun müddətdir məşhur olmasının səbəblərinə baxaq.

1995-ci ildən bəri bir çox proqramlaşdırma dillərinə təsir edən JavaScript əslində "LiveScript" kimi səhnəyə çıxdı. Bununla belə, Java o dövrdə çox populyar olduğundan, Java-nın "kiçik qardaşı" kimi yerləşdirildi və JavaScript adlandırıldı. Zamanla ECMAScript ilə tamamilə müstəqil bir dil oldu. JavaScript kimi çox paradiqma dilinin ən böyük üstünlüyü ondan ibarətdir ki, o, həm müştəri tərəfində, həm də server tərəfində və ya JavaScript mühərriki olan istənilən cihazda icra oluna bilər. Hər dəfə veb sayta daxil olanda brauzer JavaScript mühərriki sayəsində qarşılaşdığınız istənilən JavaScript məzmununu emal edə bilər. JavaScript veb səhifələri canlandıran və onları interaktiv etmək üçün istifadə edilən güclü proqramlaşdırma dilidir. JavaScript olmadan veb-saytlar sadə mətn və şəkillər olaraq qalır: Google Xəritələr sadəcə sadə xəritədir və Facebook qəzetə bənzəyir. O, JavaScript kimi sadə bir vasitə ilə yazıla bilən kodlardan ibarətdir, mətn redaktoru, əvvəlcə veb səhifələri canlandırmaq üçün yaradılmış və bu gün də oyun və mobil proqramların inkişafı üçün tez-tez istifadə olunur. Öyrənilməsi ən asan proqramlaşdırma dillərindən biridir. JavaScript internetin dili hesab olunur, çünki o, internetdə mürəkkəb funksiyaları asanlıqla həyata keçirməyə imkan verir, serverin qarşılıqlı əlaqəsini və cavab vaxtlarını azaldır və istifadəçi təcrübəsini yaxşılaşdırır. Bu, onu istənilən veb tətbiqi üçün vacib bir dil edir. HTML və CSS ilə tam veb sayt yaratmaq üçün istifadə edilə bilər. JavaScript internet səhifənizi təkmilləşdirməyə və mürəkkəb xüsusiyyətlər yerləşdirməyə imkan yaradan script və ya proqramlaşdırma dilidir. Bu proqramlaşdırma dili sizin düşündüyünüzdən daha çox şeylər edir və baxmağınız üçün statik məlumatlar verir-vaxtında kontent dəyişdirməyə şərait yaradır-interaktiv xəritələr, cizgi 2D/3D qrafika, sürüşən video musiqi qutuları və s. JavaScript, tərtibatçıların interaktiv veb səhifələr yaratmaq üçün istifadə etdiyi proqramlaşdırma dilidir. Yenilənən sosial media lentlərindən tutmuş animasiyalar və interaktiv xəritələrin nümayişinə qədər JavaScript funksiyaları veb sayt istifadəçisinin təcrübəsini yaxşılaşdırma bilər. Müştəri tərəfi skript dili olaraq, Ümumdünya

Şəbəkəsinin əsas texnologiyalarından biridir. Məsələn, siz internetə baxarkən şəkil döngəsini, klikləmək üçün açılan menyunu və ya veb səhifəsində dinamik olaraq dəyişən element rənglərini görəndə JavaScript effektlərini görürsünüz. Təkcə bunlar JavaScript vasitəsilə ediləmir. Bu işin arxasında HTML və CSS durur.

JavaScript mühərrikləri müxtəlif veb brauzerlərdə fərqli adlar alır. Misal üçün;

- Google Chrome-da V8,
- Firefox-da SpiderMonkey,
- Microsoft Edge-də ChakraCore,
- Safari-də Nitro və SquirrelFish.



Şəkil 1.4.1 Web-Developmentin sadə formada izahı

1.5 Html və onun texniki xarakteristikaları.

Əsas məqsədi veb saytın strukturunu qutmaq və məna verməkdir. Misal olaraq Veb saytımıza müxtəlif paragraflar, başlıqlar və məlumatlar cədvəli şəkillər və videolar yerləşdirmək.

Proqram təminatını quraşdırdıqdan sonra JavaScript kodunu yazmağa başlaya bilərsiniz. HTML faylına JavaScript kodu əlavə etmək üçün mətn/HTML redaktorunuzla HTML faylı yaradın və ya açın. Əsas HTML faylında docType və bəzi əsas HTML teqləri var, məsələn, <html>, <head> və <body>. Məsələn, əsas HTML5 faylı aşağıda göstərilən kimi görünə bilər.

HTML (Hypertext Markup Language) veb səhifələr və veb proqramlar yaratmaq üçün istifadə olunan standart işarələmə dilidir. Bu mətn, şəkillər və multimedia daxil olmaqla veb-səhifənin strukturunu və məzmununu təsvir etmək üçün istifadə olunan

dildir. HTML internetin əsasını təşkil edir və onsuz bildiyimiz kimi Ümumdünya Şəbəkəsi mövcud olmazdı.

Bu yazıda biz HTML-ni onun tarixi, sintaksisi, xüsusiyyətləri və tətbiqləri daxil olmaqla ətraflı müzakirə edəcəyik.

HTML tarixi

HTML ilk dəfə 1991-ci ildə Ümumdünya Şəbəkəsini icad edən İngilis kompüter alimi Tim Berners-Li tərəfindən təqdim edilmişdir. HTML-nin ilk versiyası olan HTML 1.0, istifadəçilərə başlıqlar, paraqraflar və siyahılarla əsas veb səhifələr yaratmağa imkan verən sadə bir dil idi. İllər keçdikcə HTML daha mürəkkəb bir dilə çevrildi və hər yeni versiya ilə yeni xüsusiyyətlər və imkanlar əlavə edildi. HTML-nin ən son versiyası HTML5 2014-cü ildə buraxıldı və bu günə qədər dilin ən əhatəli versiyasıdır.

HTML sintaksisi. HTML veb-səhifənin məzmununu təsvir etmək üçün bir sıra teqlərdən istifadə edir. Teqlər bucaqlı mötərizə (<>) içərisindədir və məzmun haqqında əlavə məlumat verən atributları ehtiva edə bilər. Budur HTML etiketinə bir nümunə:

<p>User 1: Vusal Durmushov</p> Bu sadə bir html kodudur.

Bu misalda paraqrafı təyin etmək üçün <p> teqindən istifadə edilir və paraqrafın məzmununu açılış və bağlanış teqləri arasında yerləşdirilir.

HTML xüsusiyyətləri

Struktur: HTML, başlıqlar, paraqraflar və siyahılar daxil olmaqla veb-səhifədəki məzmunu təşkil etmək üçün strukturlaşdırılmış üsul təqdim edir.

Media dəstəyi: HTML şəkillər, audio və video kimi multimedia məzmununu daxil etməyə imkan verir.

Əlçatanlıq: HTML məzmunu əlilliyi olan istifadəçilər üçün əlçatan etmək üçün funksiyaları ehtiva edir, məsələn, ekran oxuyanlar üçün şəkillərə alternativ mətn əlavə etmək imkanı.

Çarpaz Platforma Uyğunluğu: HTML bütün əsas veb-brauzerlərlə uyğun gəlir və geniş çeşidli cihazlarda baxıla bilər.

HTML proqramları

Veb İnkişafı: HTML veb inkişafının əsasını təşkil edir və veb saytlar, veb proqramlar və mobil proqramlar yaratmaq üçün istifadə olunur.

E-poçt Şablonları: HTML e-poçt şablonları yaratmaq üçün istifadə edilə bilər ki, bu da marketoloqlara oxunma ehtimalı daha çox olan vizual cəlbedici e-poçtlar yaratmağa imkan verir.

Elektron ticarət: HTML pərakəndə satıcılara onlayn mağazalar və məhsul səhifələri yaratmağa imkan verən e-ticarətdə geniş şəkildə istifadə olunur.

Təhsil: HTML təhsildə dünyanın istənilən yerindən tələbələrin əldə edə biləcəyi onlayn kurslar və təhsil resursları yaratmaq üçün istifadə olunur.

Nəticə

HTML internetin əsas komponentidir və veb səhifələr və veb proqramlar yaratmaq üçün vacibdir. O, məzmunu təşkil etmək üçün strukturlaşdırılmış bir üsul təqdim edir və tərtibatçılara geniş çeşiddə cihazlarda baxıla bilən vizual cəlbedici və əlçatan veb saytlar yaratmağa imkan verir. HTML təkamül etməkdə davam edir və hər yeni versiya ilə yeni funksiyalar əlavə olunur, bu da onun veb tərtibatçıları üçün müvafiq və faydalı dil olaraq qalmasını təmin edir.

1.6 CSS və onun texniki xarakteristikaları.

Css - Bu kodları vasitəsilə biz saytımıza müxtəlif dizaynlar veririk, səlhiyəyə salırıq, daha gözə yatımlı formaya salırıq yazdığımız Html kodları üzərində işləyirik. Misal olaraq arxa planları rəngləmək, animasiyalar vermək, ölçüləri böyütmək və ya kiçiltmək üçün istifadə etmək olar. Veb səhifənin daha yaxşı görünməsi sizin xəyal gücünüzdən asılıdır.

CSS (Cascading Style Sheets) HTML və ya XML-də yazılmış sənədin təqdimatını təsvir etmək üçün istifadə edilən üslub cədvəli dilidir. CSS tərtibatçılara veb səhifənin tərtibatını, tipografiyasını, rəngini və digər vizual aspektlərini idarə etməyə imkan verir. O, müasir veb inkişafının vacib komponentidir və dinamik və vizual cəlbedici veb səhifələr yaratmaq üçün HTML və JavaScript ilə birlikdə istifadə olunur.

Bu yazıda biz CSS-ni onun tarixi, sintaksisi, xüsusiyyətləri və tətbiqləri daxil olmaqla ətraflı müzakirə edəcəyik.

CSS tarixi

CSS ilk dəfə 1994-cü ildə norveçli veb-pioner Håkon Wium Lie və hollandiyalı kompüter alimi Bert Bos tərəfindən təklif edilmişdir. CSS-in ilk versiyası, CSS1, 1996-cı ildə nəşr olundu və veb səhifələrin üslubu üçün əsas funksiyalar dəstini təmin etdi. İllər keçdikcə CSS daha güclü bir dilə çevrildi və hər yeni versiya ilə yeni funksiyalar və imkanlar əlavə edildi. CSS-in ən son versiyası olan CSS3 2011-ci ildə buraxılıb və bu günə qədər dilin ən əhatəli versiyasıdır.

CSS sintaksisi

CSS HTML sənədindəki elementlərin necə göstəriləcəyini müəyyən etmək üçün bir sıra qaydalardan istifadə edir. Hər bir qayda seçicidən, xassələr dəstindən və dəyərlərdən ibarətdir. Budur bir CSS qaydası nümunəsi:

```
css
```

```
h1 {
    şrift ölçüsü: 32px;
    rəng: mavi;
}
```

Bu misalda h1 seçicisi səhifədəki bütün başlıq elementlərini seçir və şrift ölçüsü və rəng xüsusiyyətləri mətnin necə göstəriləcəyini müəyyənləşdirir.

CSS-in xüsusiyyətləri

Layout: CSS tərtibatçılara veb-səhifənin tərtibatına, o cümlədən elementlərin yerləşdirilməsi, elementlərin ölçüsü və elementlər arasındakı məsafəyə nəzarət etməyə imkan verir.**Tipoqrafiya:** CSS tərtibatçılara şrift ailəsi, şrift ölçüsü və şriftin çəkisi daxil olmaqla veb-səhifənin tipoqrafiyasına nəzarət etməyə imkan verir.

Rəng: CSS tərtibatçılara veb-səhifədəki elementlərin rəngini, o cümlədən fon rəngini, mətn rəngini və haşiyə rəngini idarə etməyə imkan verir.

Cavab verən dizayn: CSS tərtibatçılara müxtəlif ekran ölçülərinə uyğunlaşan həssas dizaynlar yaratmağa imkan verir ki, bu da veb səhifələri mobil cihazlarda daha əlçatan edir.

CSS tətbiqləri

Veb İnkişafı: CSS müasir veb inkişafının vacib komponentidir və vizual olaraq cəlbedici və həssas veb səhifələr yaratmaq üçün istifadə olunur.

İstifadəçi interfeysi dizaynı: CSS istifadəçi interfeysi dizaynında veb və mobil proqramların tərtibatını, tipografiyasını və rəngini idarə etmək üçün istifadə olunur.

Çap Dizaynı: CSS çap sənədlərinin tərtibatına və tipografiyasına nəzarət etmək üçün istifadə edilə bilər ki, bu da tərtibatçılara veb səhifələrinin üslubuna uyğun çap materialları yaratmağa imkan verir.

Brendləşdirmə: CSS veb sayt və ya tətbiqdə ardıcıl vizual üslub yaratmaq üçün istifadə oluna bilər və brend kimliyini yaratmağa kömək edə bilər.

Nəticə

CSS vizual cəlbedici və həssas veb səhifələr yaratmaq üçün vacib olan güclü bir dildir. O, tərtibatçılara veb-səhifədəki elementlərin quruluşunu, tipografiyasını və rəngini idarə etməyə imkan verir və dinamik və interaktiv veb proqramları yaratmaq üçün HTML və JavaScript ilə birlikdə istifadə olunur. CSS inkişaf etməyə davam edir və hər yeni versiya ilə yeni funksiyalar əlavə olunur, bu da onun veb tərtibatçıları üçün müvafiq və faydalı dil olaraq qalmasını təmin edir.

CSS (Cascading Style Sheets) veb səhifələrin üslubu üçün istifadə olunan güclü bir dildir. Bununla belə, CSS-nin yazılması, xüsusən daha böyük layihələr üçün vaxt aparan və səhvlərə meyilli ola bilər. CSS işlənməsini daha səmərəli etmək üçün CSS preprocessorları təqdim edildi. CSS preprocessorları CSS-in imkanlarını genişləndirən, dəyişənlər, yuvalama və funksiyalar kimi xüsusiyyətləri təmin edən, CSS kodunu yazmağı və saxlamağı asanlaşdıran dillərdir.

Bu yazıda biz CSS preprocessorlarını onların tarixi, sintaksisi, xüsusiyyətləri və üstünlükləri daxil olmaqla ətraflı müzakirə edəcəyik.

CSS Preprocessors tarixi

İlk CSS preprocessoru 2006-cı ildə buraxılmış Sass (Syntactically Awesome Style Sheets) idi. Sass Hampton Catlin tərəfindən yaradılmış və Ruby dilində yazılmışdır. Sass, CSS kodunu yazmağı və saxlamağı asanlaşdıran dəyişənlər, yuvalar və miksinlər kimi xüsusiyyətlər təqdim etdi.

2009-cu ildə Less (Leaner Style Sheets) buraxıldı. Less Alexis Sellier tərəfindən yaradılmış və JavaScript-də yazılmışdır. Less Sass-a bənzəyir, dəyişənlər, yuvalar və miksinlər kimi xüsusiyyətləri təmin edir, eyni zamanda rəng funksiyaları və riyazi əməliyyatlar kimi özünəməxsus xüsusiyyətlərinə malikdir.

2014-cü ildə PostCSS təqdim edildi. PostCSS bir ön prosessor deyil, pluginlərdən istifadə edərək CSS-i çevirən bir vasitədir. Bu, tərtibatçılara xüsusi xüsusiyyətlər CSS Grid kimi müasir CSS sintaksisindən istifadə etməyə imkan verir, eyni zamanda köhnə brauzerlərlə geriye uyğunluğu təmin edir.

CSS preprocessorlarının sintaksisi

CSS preprocessorları dəyişənlər, yuvalama və funksiyalar kimi əlavə funksiyalar təmin edərək CSS sintaksisini genişləndirir. Sass dəyişəninə bir nümunə:

```
$primary-color: #007bff;
.button {
background-color: $primary-color;
}
```

Bu misalda, \$primary-color dəyişəni faylın yuxarı hissəsində müəyyən edilir və sonra fon rəngini təyin etmək üçün .düymə seçicisində istifadə olunur.

CSS Preprocessors xüsusiyyətləri

Dəyişənlər: CSS preprocessorları tərtibatçılara dəyişənləri müəyyən etməyə imkan verir, rənglər, şrift ölçüləri və boşluq kimi dəyərlərin təkrar istifadəsini asanlaşdırır.

Yerləşdirmə: CSS preprocessorları tərtibatçılara seçiciləri yerləşdirməyə imkan verir, CSS kodunu yazmağı və saxlamağı asanlaşdırır.

Funksiyalar: CSS preprocessorları hesablamaları yerinə yetirmək və ya dəyərləri manipulyasiya etmək üçün istifadə edilə bilən funksiyaları təmin edir və mürəkkəb üslublar yaratmağı asanlaşdırır.

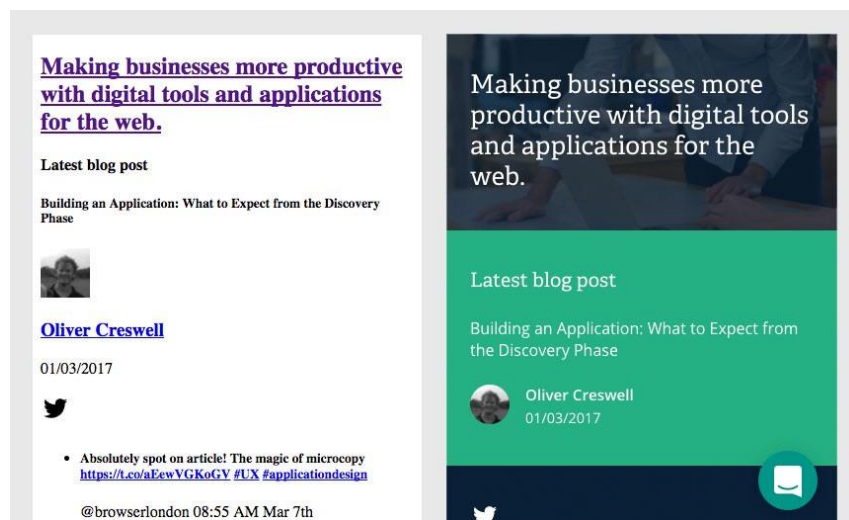
Qarışıqlar: CSS preprocessorları tərtibatçılara üslub qruplarını müəyyənləşdirməyə və təkrar istifadə etməyə imkan verir ki, bu da layihə üzrə ardıcılığı saxlamağı asanlaşdırır.

CSS Preprocessors üstünlükləri

Səmərəlilik: CSS preprocessorları CSS kodunu yazmağı və saxlamağı asanlaşdırır, veb səhifəni tərtib etmək üçün tələb olunan vaxtı və səyi azaldır. **Ardıcılıq:** CSS preprocessorları üslubların bütün səhifələr və elementlər arasında ardıcıl şəkildə tətbiq olunmasını təmin edərək, layihə üzrə ardıcılığı saxlamağı asanlaşdırır.

Çeviklik: CSS preprocessorları əlavə funksiyalar və sintaksis təmin edərək daha mürəkkəb və dinamik üslublar yaratmağa imkan verir.

Uyğunluq: PostCSS kimi CSS preprocessorları köhnə brauzerlərlə geriye uyğunluğu təmin edərək üslubların bütün cihazlarda düzgün göstərilməsini təmin edir.



Şəkil 1.6.1 Html və CSS birgə istifadəsi zamanı saytın dəyişməsi

CSS preprocessorları CSS-in imkanlarını genişləndirən, CSS kodunu yazmağı və saxlamağı asanlaşdıran güclü alətlərdir. Onlar daha mürəkkəb və dinamik üslublar yaratmağa imkan verən dəyişənlər, yuvalama və funksiyalar kimi əlavə funksiyalar təqdim edir. Sass, Less və PostCSS kimi CSS preprocessorları müasir veb tərtibatında geniş istifadə olunur və veb səhifələri üçün səmərəli, ardıcıl və çevik üslublar yaratmaq istəyən tərtibatçılar üçün vacib alətə çevrilib. Şəkil 1.3də Html və Css in birgə istifadəsi zamanı veb saytın görünüşünün dəyişimi göstərilmişdir.

```

p {
  font-family: "Roboto", sans-serif;
  letter-spacing: 1px;
  text-transform: uppercase;
  text-align: center;
  border: 2px solid rgba(0, 0, 200, 0.6);
  background: rgba(0, 0, 200, 0.3);
  color: rgba(0, 0, 200, 0.6);
  box-shadow: 1px 1px 2px rgba(0, 0, 200, 0.4);
  border-radius: 10px;
  padding: 3px 10px;
  display: inline-block;
  cursor: pointer;
}

```

USER 1 :VUSAL DURMUSHOV

VUSAL DURMUSHOV

127.0.0.1:5500 says

Vusal Durmushov

OK

Şəkil 1.6.2 Yazılmış koda əsasən ekranda alınmış nəticə

```

<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Document</title>
  </head>
  <style>
    h2 {
      font-family: "helvetica neue", helvetica, sans-serif;
      letter-spacing: 1px;
      text-transform: uppercase;
      text-align: center;
    }
  </style>

```

```

border: 2px solid rgba(0, 0, 200, 0.6);
background: rgba(0, 0, 200, 0.3);
color: rgba(0, 0, 200, 0.6);
box-shadow: 1px 1px 2px rgba(0, 0, 200, 0.4);
border-radius: 10px;
padding: 3px 10px;
display: inline-block;
cursor: pointer;
}
</style>
<body>
  <h2 onclick="myFunction()">Vusal Durmushov</h2>

  <script>
    function myFunction() {
      alert("Vusal Durmushov");
    }
  </script>
</body>
</html>

```

Gəlin bu yazdığımız kodu analiz edək. Əvvəlcə Html vasitəsilə veb səhifəmizdə yazmaq istədiyimiz mətni yerləşdirdik və sonra isə Css vasitəsilə rəngini, yazılış stilini, arxa plan rəngini dəyişdirdik və müxtəlik diazyn işləri apardır. Ən sonda isə JavaScript vasitəsilə funksiya əlavə etdik. Yazılan adı kliklədikdə alert yəni məlumaq qutusu açılsın.

Bu işimizi daha aydın başa salmaq üçün belə bir misal da, çəkə bilərik. Tutaq ki biz bir bina tikirik onun bünövrəsi, strukturunu Html vasitəsilə qurulur, Css vasitəsilə biz onu daha səliqəli insan yaşaya biləcək formaya salırıq. Js vasitəsilə isə funksionallıq qatırıq misal olaraq lift sistemi gözəl nümunə olardı

1.7 Javascript-in tətbiq sahələri ?

JavaScript-in ilk versiyası cəmi 10 gün ərzində yazılmışdır, lakin onun bu günə qədər sağ qalmasına kömək edən bir çox irəli düşünən yenilikləri var. JavaScript-i unikal edən xüsusiyyətlərdən biri onun HTML/CSS ilə tam inteqrasiyasıdır. JavaScript-in unikallığını vurğulayan bəzi digər xüsusiyyətlər bunlardır: Onun icrası brauzer daxilində baş verdiyi üçün heç bir media quraşdırma tələb etmir. Bütün əsas brauzerlər tərəfindən dəstəklənir.

Yüngül, sürətli və platformalararası bir dil olmaqla, geniş şəkildə mənimsənilmiş və mobil, masaüstü və oyun proqramları kimi sahələrdə yayılmışdır. Sənayedə istifadə edilən ən məşhur skript dilidir.

Onun geniş və yetkin icma dəstəyi var. Ən təhlükəsiz proqramlaşdırma dillərindən biridir. Əsas “core client-side ”olan Javascript proqramlaşdırma dilinə bunları da əlavə edə bilərik

1) Əsas dəyərləri yəni bizə lazım olubları dəyər şəklində funksiyada yadda saxlaya bilərik. Sonra isə bizə lazım olduqda həmi dəyəri asand formada çağıra bilərik. Misal olaraq let name = Azer, Artıq funksiyada name dəyəri Azər-ə bərabər olur.

2) Html-də yazdığımız mətn parçaları üzərində əməliyyatlar apara bilərik. Misal üçün Server tərəfdən gələn məlumatları ekrana yazdırmaq istəyiriksə bu çox gözəl bir metoddur.

3) Veb səhifədə baş verən müəyyən hərəkətlərə cavab olaraq lazımı hadisələr baş verir. Misal olaraq istifadəçi özü haqqında məlumat almaq istədikdə info sözünü klikləndikdə modul açılır və əlavə məlumatlar görsənir.

Bu populyar və güclü proqramlaşdırma dilinin imkanlarını sadalayarkən qeyd etmək lazım olan ilk şey JavaScript-in Html səhifələrinin məzmununu və ya üslublarını dinamik şəkildə əlavə etməyə və dəyişməyə imkan verməsidir. JavaScript işləmə zamanı veb səhifəni və ya CSS elementlərini göstərə və dəyişdirə və veb səhifələrin görünüşünü yeniləyə bilən bir neçə üsul təqdim edir.

JavaScript açılan menyular kimi naviqasiya vasitələri ilə veb saytların dizaynını təkmilləşdirə bilər. JavaScript-in köməyi ilə veb səhifələrdə siçan və klaviatura hərəkətlərini fərdiləşdirə bilərsiniz.

Veb səhifəyə JavaScript (JS) əlavə etmək istənilən HTML məzmununu əlavə etməyə çox bənzəyir. <script> teqi adətən HTML daxilindən <head> teqləri arasında yerləşdirilməsi tövsiyə olunan JS kodunu çağırmaq üçün istifadə olunur. Əgər səhifənizə əlavə etmək istədiyiniz funksiyanın səhifənin mətnində olmasını istəyirsinizsə, JS kodunu <body> bölməsinə əlavə edə bilərsiniz. JavaScript kodunu HTML faylına əlavə etmək əvəzinə onu başqa fayla əlavə etmək və HTML daxilində bu fayla istinad etmək; Səhifələrin yüklənmə müddətini qısaltmaq kimi üstünlükləri

var. JS kodunu WordPress veb sayınıza əlavə etmək istəyirsinizsə, veb sayınızda istifadə etdiyiniz mövzuda JS kodu əlavə etmək üçün sahənin olub olmadığını yoxlayın. Əgər belə bir seçim varsa, JS kodunu göstərilən yerə əlavə edə bilərsiniz. Yadda saxlayın ki, kodun səhv yerləşdirilməsi saytın işinə zərər verə bilər. WordPress veb sayınıza JavaScript kodu əlavə etməyin başqa bir yolu plaginlərdən istifadə etməkdir. Əksər plaginlər JS kodu əlavə etmək üçün seçimlər təqdim edir. Plagindən asılı olaraq, bu kod sayt daxilində və ya xüsusi səhifə üçün tətbiq oluna bilər. Bu metoddan istifadə edərkən daha kiçik JS kodunu daxil etmək tövsiyə olunur. Əks halda səhifə yükləmə sürəti azala bilər. Daha böyük JS kodu əlavə etmək istəyirsinizsə, fərqli bir üsuldən istifadə etməyi düşünməlisiniz. Kompüter və ya smartfondan istifadə edən hər kəs JavaScript ilə təcrübə keçir, demək olar ki, bütün smartfonlar bu proqramlaşdırma dili ilə yaradılmış proqramlardan istifadə edir. Məsələn, Facebook tərəfindən yaradılmış React Native kimi texnologiyaların köməyi ilə siz mobil proqramlar yaratmaq üçün JavaScript-dən istifadə edə bilərsiniz. JavaScript faylları yükləmək və yükləmək və uzaq serverlərə sorğu göndərmək kimi funksiyalara malikdir. O, istifadəçinin serverində əvvəlki sayt fəaliyyəti və veb səhifə seçimləri kimi məlumatları saxlaya bilər. Kukilər hər dəfə məlumat tələb olunduqda veb-brauzerin veb serverlə əlaqə saxlamamasını təmin edir. JavaScript bu kukilərin dəyərlərini təyin etmək və əldə etmək üçün üsul və xassələri təmin edir. JavaScript istifadəçi hərəkətləri üçün fərdiləşdirilmiş pop-upları və mesajları göstərməyə imkan verir. O, qeydiyyat formaları və giriş etimadnaməsi kimi onlayn formalar üçün əsas təsdiqləmə və məlumat formatının təsdiqini təmin edə bilər. JavaScript istifadəçinin aktiv və ya qeyri-aktiv olduğunu toxunma ekranı, cursor hərəkətləri, siçan klikləri, sürüşdürmə və klaviatura daxiletmələri vasitəsilə aşkar edə bilər. Daha maraqlısı isə müştəri tərəfində JavaScript dilinin üzərində qurulmuş funksionallıqdır.

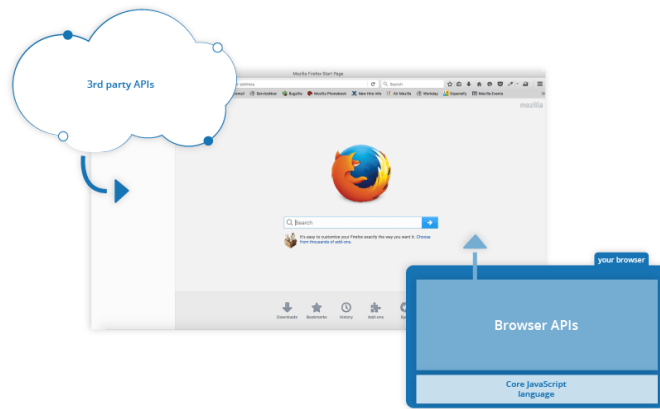
Tətbiq Proqramlaşdırma İnterfeysləri (API) sizə JavaScript kodunuzda istifadə etmək üçün əlavə super güclər təqdim edir. Xam JavaScript dilini tərtibatçılar “vanilla JavaScript” adlandırırlar. JavaScript funksiyalarını vanil JavaScript-dən başqa heç bir alətdən istifadə etməklə kodlaşdırmaq mümkündür, lakin JavaScript; O, tərtibatçıların həyatını əhəmiyyətli dərəcədə asanlaşdıran JavaScript kitabxanalarına

və çərçivələrə malikdir. Bu alətlərə jQuery kimi JavaScript kitabxanaları və AngularJS və ReactJS kimi JavaScript çərçivələri daxildir; Tərtibatçılara əsas JavaScript funksiyaları üçün əvvəlcədən yazılmış JavaScript kodları verməklə kodlaşdırma prosesini sürətləndirir. JavaScript kitabxanaları adi JavaScript tapşırıqlarını yerinə yetirmək üçün istifadə edilə bilən JS kod parçalarının toplusudur. JavaScript çərçivələri veb saytlarda JavaScript kodunu redaktə etmək üçün şablonlar təqdim edən JS kitabxanalarının kolleksiyalarıdır. JavaScript Tətbiq Proqramlaşdırma Interfeysləri (API) ilə tamamlanır. Bu əvvəlcədən qurulmuş kod blokları, məlumatların ötürülməsini asanlaşdıraraq, öz-özünə kodlanması çətin olan proqramları daxil etməyə və ya həyata keçirməyə imkan verir. Misal üçün:

- Geolocation API
- O, əsasən Google Xəritədə görünən coğrafi məlumatlara nəzarət edir.
- Əlfəcinlər API
- Əlfəcin edilmiş səhifələrin yaradılması, dəyişdirilməsi və redaktə edilməsinə imkan verir.
- TopSites API
- İstifadəçinin ən çox ziyarət etdiyi saytlara girişi təmin edir.
- Google Maps API
- Naviqasiyanı asanlaşdırmaq üçün xüsusi xəritələr və istiqamətlər daxil edir.
- Facebook Giriş API
- Hər girişdə istifadəçi identifikasiyasına imkan verir.

API-lar tərtibatçıya yerinə yetirilməsi çətin və ya qeyri mümkün olan proqramları həyata keçirməyə imkan verən verən hazır kod blokları dəstidir.

Necə ev tikmək üçün mebellərdən istifadə ediriksə Veb səhifəmizi düzəldən zaman API lardan istifadə edirik. Şəkil 1.4 Js proqramlaşdırma dilinin API ilə birgə istifadəsi göstərilmişdir. Misal olaraq bir mağazanız var məhsullarınız isə həddindən çoxdur. API lara bu məlumatları yığaraq serverinizdə saxlayırsınız və lazım olduqda məlumatlar oradan çəkirsiniz. Kodunuzda çox sətirlik kod yazaraq məlumatları orada saxlayıb əlavə zaman itirməkdənsə bu metod daha məqsədə uyğundur.



Şəkil 1.7.1 Js proqramlaşdırma dilinin API ilə birgə istifadəsi

JavaScript-i işə salmaq üçün nə tələb olunur?

JavaScript yalnız JavaScript aktivləşdirilmiş İnternet brauzerini tələb edir, bütün brauzerlərdə defolt olaraq bu brauzer var.

1.8 JavaScript ilə işləmək təhlükəsizlik cəhətdən məqsədə uyğundurmu ?

JavaScript müxtəlif məbələrə veb brauzerlərə yükləndiyi üçün təhlükəsizlik cəhətdən zəifdir. Lakin bu təhlükəsizliyi azaltmaq üçün birinci istifadəçi diqqətli olmalıdır ikincisi isə bunun üçün Sun inkişafın əvvəlki mərhələlərindəki möhkəm təhlükəsizlik xüsusiyyətlərini tətbiq etdi. Bu xüsusiyyətlərə aşağıdakılar daxildir:

- Brauzer JavaScript-i işlədirsə, o, brauzer prosesindən təcrid olunur. Chrome və Edge kimi brauzerlər əlavə təhlükəsizlik tədbiri kimi JavaScript sandboxdan istifadə edilir.
- JavaScript sərt diskdən və ya kompüterə qoşulmuş digər yaddaş qurğusundan oxuya və ya ona yazma bilməz.

Bununla belə, yuxarıda göstərilən təhlükəsizlik tədbirləri ilə belə, İnternetə qoşulmuş hər şey heç vaxt 100% təhlükəsiz deyil və kompüter JavaScript-dən istifadə etməklə təhlükə altına düşə bilər. Mümkün təhlükəsizlik pozuntularına kömək etmək üçün JavaScript deaktiv edilə bilər. Bununla belə, bu gün əksər saytlar JavaScript-in tam işləməsini tələb edir. Alternativ olaraq, təhlükəsizliyinizi qorumaq üçün brauzerinizin ən son versiyasını işlətdiyinizə həmişə əmin ola bilərsiniz. Veb saytlarda JavaScript

və ya onun xüsusiyyətlərini idarə etməyə kömək etmək üçün quraşdırıla bilən brauzer əlavələri də var.

Java və JavaScript eyni şeydirmi?

Xeyr. Java və JavaScript bir neçə oxşarlığı olan iki fərqli kompüter proqramlaşdırma dilidir. Əgər dillərdən birini öyrənmək və İnternet və veb saytlar üçün funksiyalar inkişaf etdirmək istəyirsinizsə, JavaScript öyrənin. Tətbiqlər (məsələn, brauzerə ehtiyac duymadan kompüterdə və ya smartfonda işləyən oyun) inkişaf etdirmək istəyirsinizsə, Java öyrənin.

Java və JavaScript müxtəlif məqsədlər üçün istifadə olunan iki məşhur proqramlaşdırma dilidir. Java müxtəlif platformalarda işləyən proqramlar yaratmaq üçün istifadə edilən yüksək səviyyəli proqramlaşdırma dilidir, JavaScript isə interaktiv veb səhifələr və veb proqramlar yaratmaq üçün istifadə edilən skript dilidir. Bu yazıda biz hər iki dilin tarixi, sintaksisi, xüsusiyyətləri və tətbiqləri daxil olmaqla ətraflı müzakirə edəcəyik.

Java 1995-ci ildə Sun Microsystems-də James Gosling tərəfindən hazırlanmış yüksək səviyyəli proqramlaşdırma dilidir. Bu, masaüstü, veb və mobil proqramlar yaratmaq üçün istifadə edilə bilən ümumi təyinatlı dildir. Java obyekt yönümlü bir dildir, yəni tərtibatçılara məlumatları və funksionallığı əhatə edən siniflər və obyektlər yaratmağa imkan verir. Java öyrənmək üçün asan olan sadə sintaksisə malikdir və onu yeni başlayanlar üçün məşhur seçim edir. O, həmçinin platformada müstəqildir, yəni Java kodu bir dəfə tərtib oluna və Java-nı dəstəkləyən istənilən platformada işləyə bilər. Bu xüsusiyyət Java kodunu JVM quraşdırılmış istənilən platformada işləyə bilən baykoda çevirən Java Virtual Maşın (JVM) sayəsində mümkün olmuşdur.

Java-nın xüsusiyyətləri

Obyekt yönümlü proqramlaşdırma: Java obyekt yönümlü bir dildir, yəni o, tərtibatçılara məlumatları və funksionallığı əhatə edən siniflər və obyektlər yaratmağa imkan verir. Bu xüsusiyyət böyük və mürəkkəb proqramları idarə etməyi asanlaşdırır. Platformadan Müstəqil: Java kodu bir dəfə tərtib edilə və Java-nı dəstəkləyən istənilən platformada işlədilə bilər. Bu xüsusiyyət müxtəlif platformalarda istifadə oluna bilən proqramların hazırlanmasını asanlaşdırır. Yaddaşın İdarə Edilməsi: Java

avtomatik yaddaş idarəetməsinə malikdir, yəni yaddaşın ayrılması və boşaldılmasını avtomatik idarə edir və yaddaş sızmasının qarşısını almağı asanlaşdırır. Təhlükəsizlik: Java sistemi zərərli kodlardan qoruyan güclü təhlükəsizlik modelinə malikdir.

Java proqramları

- 1) Masaüstü Proqramlar: Java, media pleyerlər, veb brauzerlər və ofis proqramları kimi masaüstü proqramlar yaratmaq üçün istifadə edilə bilər.
- 2) Veb Tətbiqləri: Java, e-ticarət saytları, sosial şəbəkə saytları və məzmun idarəetmə sistemləri daxil olmaqla veb proqramların yaradılması üçün geniş istifadə olunur.
- 3) Mobil Tətbiqlər: Java ən populyar mobil əməliyyat sistemlərindən biri olan Android üçün mobil proqramlar yaratmaq üçün istifadə edilə bilər.
- 4) Müəssisə Tətbiqləri: Java tez-tez müştəri münasibətlərinin idarə edilməsi (CRM) sistemləri, inventar idarəetmə sistemləri və təchizat zəncirinin idarəetmə sistemləri daxil olmaqla, müəssisə proqramlarının yaradılması üçün istifadə olunur.

JavaScript

JavaScript interaktiv veb səhifələr və veb proqramlar yaratmaq üçün istifadə edilən skript dilidir. O, 1995-ci ildə Netscape Communications-da Brendan Eich tərəfindən yaradılmışdır. JavaScript müştəri tərəfi dilidir, yəni serverdə deyil, brauzerdə işləyir. JavaScript öyrənmək asan olan sadə və çevik sintaksisə malikdir və onu yeni başlayanlar üçün məşhur seçim edir. O, dinamik veb səhifələrin yaradılması, formaların təsdiqlənməsi və animasiyaların yaradılması daxil olmaqla, geniş çeşidli proqramlar üçün istifadə olunur.

JavaScript xüsusiyyətləri İnteraktivlik: JavaScript düymələri basmaq və ya formaları doldurmaq kimi istifadəçi hərəkətlərinə cavab verən interaktiv veb səhifələr yaratmaq üçün istifadə olunur. Hadisəyə əsaslanan: JavaScript hadisə ilə idarə olunur, yəni siçan klikləri və ya düymələrin basılması kimi hadisələrə cavab verir. Çarpaz Platforma: JavaScript bütün müasir veb-brauzerlərdə işləyir və onu veb tərtibatçıları üçün məşhur seçim edir. Yüngül: JavaScript kodu yüngüldür və brauzerdə tez işləyir, bu da onu dinamik veb səhifələr yaratmaq üçün ideal edir. JavaScript tətbiqləri

- 1) Veb Tətbiqləri: JavaScript sosial şəbəkə saytları, e-ticarət saytları və məzmun idarəetmə sistemləri kimi veb proqramlar yaratmaq üçün istifadə olunur.
- 2) Oyunlar: JavaScript tapmaca oyunları və macərə oyunları kimi brauzer əsaslı oyunlar yaratmaq üçün istifadə olunur.
- 3) Animasiya: JavaScript animasiya menyuları və şəkil sürgüləri kimi animasiyalar yaratmaq üçün istifadə olunur.
- 4) Form Validasiyası: JavaScript istifadəçilərin düzgün məlumatı daxil etmələrini təmin edərək forma məlumatlarını yoxlamaq üçün istifadə olunur.

1.9 JavaScript digər proqram dillərilə müqayisəsi?

JavaScript ilk dəfə peyda olanda bu, Netscape brauzerinə əlavə edilmiş Turing-tamamlanmış əlavə bir seçim idi. Bir çox proqramçı buna məhəl qoymasa da, Lambdas kimi irəli düşünən xüsusiyyətlərin dəyərini tanıyanlar da var idi. Bu gün proqramçılar daha yüksək sürət və eyni kodu server və müştəri üzərində işlətmək üçün Node.js-ə müraciət edirlər və biz JavaScript-də yazılmış server proqramlarını görürük. Beləliklə, JavaScript onu bəzi əsas proqramlaşdırma dilləri ilə müqayisə etdikdə nə təklif edir?

C və JavaScript

C mövzulara açıq nəzarəti təmin edir, JavaScript isə tapşırıqları verilənlər hazır olduqda çağırılan asinxron funksiyalara bölür və istifadəçilərə çoxlu tapşırıqları yerinə yetirməyə imkan verir. C əvvəlcədən tərtib edilmiş və statik olaraq yazılmışdır. JavaScript şərh olunur və dinamik şəkildə yazılır. C proqramçılardan yaddaş bloklarını ayırmağı və geri qaytarmağı tələb edir. JavaScript avtomatik olaraq bununla məşğul olur. C kodu başqa bir prosessorla köçürüldükdə yenidən tərtib edilməlidir. JavaScript-də belə olmaq lazım deyil. C göstəricilər vasitəsilə birbaşa kompüterin yaddaşı ilə işləmək üçün nəzərdə tutulmuşdur və adətən əməliyyat sistemləri kimi yüksək məhsuldar proqramlar üçün istifadə olunur. JavaScript əvvəlcə yalnız veb səhifələri hədəfləsə də, Node.js vasitəsilə hazırlanmış server proqramlarında da rol oynamağa başlamışdır.

Java və JavaScript

1. Java obyekt yönümlü proqramlaşdırma dilidir və demək olar ki, istənilən platformada işləyən tərtib edilmiş proqramlar yaratmağa imkan verən virtual maşın platformasına malikdir. JavaScript obyekt yönümlü skript dilidir.
2. Java statik olaraq, JavaScript isə dinamik şəkildə yazılır.
3. Java böyük proqramları dəstəkləmək üçün nəzərdə tutulmuşdur. JavaScript bunu birbaşa təklif etmir, ona görə də proqramçılar onu sintez ediblər.
4. Java proqramları istənilən virtual maşında (JVM) və ya brauzerdə işləyə bilər. Java əvvəllər brauzerlərdə və serverlərdə işləyirdi, lakin bu gün o, əsasən serverlərlə məhdudlaşır. JavaScript, digər tərəfdən, bir vaxtlar yalnız brauzerlərdə işləyirdi, lakin indi server tərəfində artan istifadənin şahidi oluruq.
5. Həm Java, həm də JavaScript bir çox başqa dilləri dəstəkləyir.
6. Java obyektləri sinifə əsaslanır, hətta biz Java-da sinif yaratmadan heç bir proqramı edə bilmərik. JavaScript obyektləri isə prototipə əsaslanır.
7. Java proqramında “.java” fayl uzantısı var və mənbə kodu JVM (Java Virtual Maşın) tərəfindən idarə olunan baytkodlara tərcümə edir. JavaScript faylı “.js” fayl uzantısına malikdir və şərh edilir, lakin tərtib edilmir, hər bir brauzerdə JS kodunu yerinə yetirmək üçün Javascript tərcüməçisi var.
8. Java 8 JavaScript-in yerləşdirilməsi mexanizmini ehtiva edir, buna görə də Java JavaScript-in bir çox üstünlüklərindən istifadə edə bilər.
9. Java daha çox yaddaş istifadə edir. Buna görə də veb səhifələr üçün daha az yaddaş tələb edən JavaScript-ə üstünlük verilir.
10. Java paralellik üçün mövzu əsaslı bir yanaşmaya malikdir. Javascript paralellik üçün hadisəyə əsaslanan yanaşmaya malikdir.

C# və JavaScript

1. C# çox müxtəlif proqramlarda tapıla bilər. JavaScript-in çeşidi məhduddur və onun ən mühüm funksiyası veb proqramları intuitiv şəkildə formalaşdırmaqdır.
2. C# obyekt yönümlü proqramlaşdırma dilidir. JavaScript bir skript dilidir.
3. C#-da güclü .NET komponenti olan LINQ var, lakin JavaScript-in bu funksiyanı əhatə edə bilən Underscore.js kimi ayrıca kitabxanaları var.
4. C# JIT kompilyatoru tərəfindən idarə olunaq aralıq forma olan bayt koduna tərtib edilmişdir. JavaScript adətən JIT kompilyatoru vasitəsilə idarə olunur.

5. C# statik olaraq, JavaScript isə dinamik şəkildə yazılır.
6. C# menecer və dönüşüm üçün həddindən artıq yükləmə qabiliyyətinə malikdir. JavaScript həddən artıq yüklənmələri dəyişdirir.
7. C# plugin tələb edir. JavaScript heç bir plug-in tələbi olmayan əksər brauzerlərdə işləyir.

Python və JavaScript

1. JavaScript brauzerdə HTML səhifələrini dəstəkləmək üçün hazırlanmışdır. Python, digər tərəfdən, serverləri və masa üstü kompüterləri dəstəkləmək üçün əmr satırından idarə olunmaq üçün hazırlanmışdır.
2. Python sosial elmlərdə məlumatların manipulyasiya dili kimi məşhurdur, JavaScript isə serverlərdə proqramlar tapır və veb sayt və ya yerli proqram yaratmağa kömək edir. Python əsasən məlumatların təhlili, maşın öyrənməsi və riyazi intensiv tapşırıqlar üçün istifadə olunur.
3. JavaScript kod bloklarını işarələmək üçün əyri mötərizələrdən istifadə edir. Python girinti və boşluqlardan istifadə edir.
4. Python metodlar üçün çevik parametr blokları təklif edir, JavaScript isə bütün parametrlərin göstərilməli olduğunu israr edir.
5. JavaScript kodu tez-tez siçan klikləri və ya düymə vuruşları kimi hadisələrlə idarə olunur. Python funksiya çağırışlarına cavab verir.
6. Əksər Python kitabxanaları məlumatların manipulyasiyasına və təhlilinə diqqət yetirsə də, bir çox JavaScript kitabxanaları brauzerdə HTML ilə işləmək üçün nəzərdə tutulub.
7. JavaScript-də dəyişən və dəyişməz dəyişilməz anlayışı yoxdur, lakin Python dəyişən və dəyişməz məlumat növlərinə malikdir.
8. Python mənbə kodu hər hansı kodlaşdırma formatını təyin etmədiyiniz halda ASCII-yə defolt olaraq verilir, JavaScript isə UTF-16 kimi kodlaşdırılmalıdır, çünki o, xam baytların idarə edilməsi üçün daxili dəstək təklif etmir.

PHP və JavaScript

1. Hər ikisi adətən JIT tərtibçisi tərəfindən çevrilən skript dilləridir. Hər ikisi dinamik tipli dəyişənlər təklif edir.

2. Əvvəlcə PHP serverlər üçün HTML faylları yığmaq üçün nəzərdə tutulmuşdu, yuxarıda qeyd etdik ki, JavaScript indi də serverdə HTML faylları yaradır.
3. PHP işləmək üçün server tələb edir. Onu serversiz idarə etmək olmaz. JavaScript həm frontend, həm də backend üçün istifadə olunur. PHP əsasən backend məqsədləri üçün istifadə olunur.
4. PHP HTML tərtib etməkdən başqa nadir hallarda istifadə olunur. JavaScript biznes məntiqi və verilənlər bazalarını birləşdirən brauzer və server tərəfi yığınlarında getdikcə daha mürəkkəb interfeyslər üçün əsas təşkil edir.
5. JavaScript; HTML, AJAX və XML ilə birləşdirilə bilər. PHP yalnız HTML ilə birləşdirilə bilər.
6. Javascript asinxronudur, giriş/çıxış əməliyyatlarını gözləmir. PHP sinxronudur, IO əməliyyatlarının icrasını gözləyir.
7. PHP nisbətən sadə dildir. JavaScript isə həm brauzerin qarşılıqlı əlaqəsi, həm də daha ümumi həllər üçün istifadə edilən daha ümumi dilə çevrilir.
8. PHP-nin gələcəyi ilə bağlı müzakirələr davam etsə də, PHP sadəliyinə və ən sürətli proqramlaşdırma dillərindən biri olmasına görə nədir? Məqaləmizi oxumağınızı tövsiyə edirik.

Ruby və JavaScript

1. Ruby adətən server tərəfində inkişaf üçün istifadə olunan obyekt yönümlü bir dildir. JavaScript obyekt yönümlüdür və tez-tez müştəri tərəfi proqramlar üçün istifadə olunur.
2. Həm Ruby, həm də JavaScript dinamik şəkildə yazılır.
3. JavaScript blokları təsvir etmək üçün mötərizələrdən istifadə edərkən Ruby “son” sözündən istifadə edir.
4. JavaScript sintaksisinin çox hissəsini Java və C, Ruby-dən götürür; Perl və Smalltalk-dan.
5. Ruby, ümumiyyətlə, Rails çərçivəsini istifadə edən server proqramları ilə məhdudlaşır, baxmayaraq ki, bəzi idarəçilər server tərəfindən dəstəklənən komanda xətti proqramları üçün Ruby-yə üstünlük verirlər. JavaScript getdikcə ümumi məqsədli dilə çevrilir.

6. Ruby siniflərlə obyekt yönümlüdür, Ruby-də demək olar ki, hər şey bir obyektidir. JavaScript obyekt yönümlüdür, lakin sinifsizdir.
7. Ruby JavaScript-dən daha yavaşıdır.
8. JavaScript yazmaq və öyrənmək Ruby-dən daha çətinidir.

Xülasə

JavaScript-in istifadə sahələri təbii ki, burada götürdüyümüz nümunələrlə məhdudlaşmır. Əməliyyat sistemə birbaşa çıxışın olmaması və ondan çox yivli proqramların hazırlanmasında istifadə edə bilməməsi kimi məhdudiyyətlərinə baxmayaraq, milyonlarla proqram tərtibatçısı öz veb-saytları və proqram təminatı üçün müxtəlif sahələrdə Javascript-dən istifadə etməyə davam edir.

JavaScript nədir? Suala bütün brauzerlərin başa düşdüyü dil olaraq qaldığı müddətcə populyarlığını qoruyan bir proqramlaşdırma dili kimi cavab vermək mümkündür.

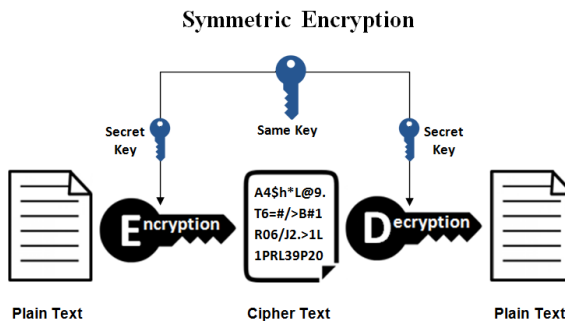
II FƏSİL. İnformasiya təhlükəsizliyinin tətbiq sahələri, metodları və günümüzdəki aktuallığı mövzusunda

2.1 İnformasiya təhlükəsizliyi nədir ?

Kriptoqrafiyada bir çox şifrələmə alqoritmləri mövcuddur. Ümumiyyətlə, onları üç növə bölmək olar: simmetrik şifrələmə alqoritmi, asimmetrik şifrələmə alqoritmi və tək girişli şifrələmə alqoritmi. Fərqli şifrələmə alqoritmləri təhlükəsizlik, şifrələmə səmərəliliyi, tətbiqin mürəkkəbliyində istifadə olunur, istifadə etmək üçün ən yaxşı yerlərdə böyük fərqlər var. Onların arasında simmetrik şifrələmə alqoritmi şifrələmə və deşifrə üçün eyni açardan istifadə edir. Alqoritm geri çevrilə bilən və ya deşifrə edilə biləndir. Ümumi alqoritmlərə IDEA, DESX, RC4, RC5, RC6, DES, 3DES və AES daxildir. Simmetrik şifrələmə alqoritmləri çox vaxt böyük miqdarda məlumatların şifrələndiyi və ya məlumatların tez-tez göndərildiyi hallarda istifadə olunur. Asimmetrik şifrələmə alqoritmi o deməkdir ki, şifrələmə və deşifrə üçün müxtəlif açıq və özəl açarlardan istifadə edilir. Bəzən asimmetrik şifrələmə açıq açar şifrələməsi də adlanır. Alqoritm də geri çevrilə bilər. Ümumi alqoritmlərə RSA, DSA, ECC, Diffie-Hellman və El daxildir. Şifrələmə üsulları düz mətni məxfi şəkildə ötürmək və ya saxlamaq üçün istifadə olunan riyazi üsullardır. Bu üsullar həssas məlumatların qorunması üçün vacibdir və kibertəhlükəsizlikdə böyük rol oynayır.

2.2 Simmetrik Açar Şifrələmə

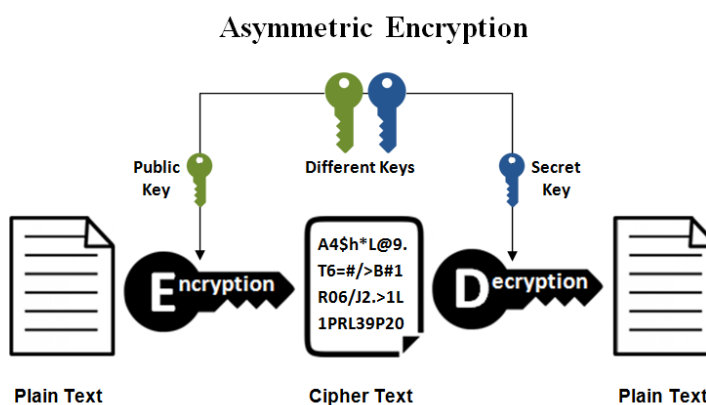
Simmetrik açar şifrələməsi həm şifrələmə, həm də deşifrə əməliyyatları üçün eyni açarın istifadə edildiyi şifrələmə üsuludur. Bu üsulda açıq mətn əvvəlcə açarla şifrələnir və şifrəli mətn formalaşır. Şifrə mətni eyni açardan istifadə edərək deşifrə edilir. Simmetrik açar şifrələməsi sürətli və effektivdir, lakin təhlükəsizlik baxımından zəif ola bilər. Açarın məxfiliyi qorunmursa, təcavüzkarlar açıq mətnə giriş əldə edə bilərlər. Simmetrik şifrələmə məlumatları şifrələmək üçün istifadə edilən şifrələmə üsuludur. Bu üsulda eyni açar şifrələmə və deşifrə üçün istifadə olunur.



Şəkil 2.2.1 Symmetric şifrələmənin sadə formada təsviri

Bu açar verilənləri şifrələmək və deşifrə etmək üçün istifadə edilən şifrələmə alqoritmi ilə müəyyən edilir. Simmetrik şifrələmə alqoritmləri ümumiyyətlə daha sürətli və daha az resurs istehlak edir. Bununla belə, açarın paylaşılması təhlükəsiz şəkildə həyata keçirilmədikdə, şifrələnmiş məlumatlar oğurlana bilər. Ən çox istifadə edilən simmetrik şifrələmə alqoritmlərinə DES (Data Encryption Standard), AES (Advanced Encryption Standard) və Blowfish daxildir. Bu alqoritmlər bank işindən tutmuş hərbi rabitəyə qədər müxtəlif sahələrdə istifadə olunur və təhlükəsizliyin təmin edilməsi üçün etibarlı variant hesab olunur. Şəkil 2.1-də Symmetric şifrələmənin sadə formada təsviri göstərilmişdir.

2.3 Asimmetrik Açar Şifrələmə



Şəkil 2.3.1 Asymmetric şifrələmənin sadə formada təsviri

Asimmetrik açar şifrələməsi iki fərqli açıardan istifadə edən şifrələmə üsuludur. Bir açar açıq mətni şifrələmək üçün, digər açar isə şifrəli mətnin şifrəsini açmaq üçün istifadə olunur. Bu üsul simmetrik açar şifrələmə metodundan daha təhlükəsiz ola bilər,

lakin daha yavaşıdır və daha mürəkkəb infrastruktur tələb edir. Bu iki açar riyazi cəhətdən bir-biri ilə əlaqəlidir. Bu açarlardan biri “Public key”-yəni hamı tərəfdən əlçatanlığı olan bir açar, digəri isə “Private key”-yalnız onu yaradan şəxsə məlum olan açardır. Onların bağlı olduğu şəxslər ayrı olsa da, ayrı-ayrılıqda işləyə bilmirlər. Gəlin gündəlik həyatımızdan assimetrik şifrələməyə bir nümunə deyək. Şəkil 2.3.1 Asymmetric şifrələmənin sadə formada təsviri göstərilmişdir.

Tom və Mary bir- biriləri ilə əlaqə saxlayacaqlar və Mary isə bunu təhlükəsizlik cəhətdən yüksək olduğu üçün assimetrik şifrələmə ilə etmək istəyir. Əvvəlcə Mary 1 cüt açar düzəldir. “Public key” və “Private key” Sonra isə “Private key”-i özündə saxlayır və “Public key”-i Toma göndərir. Tom mesajı yazır və onu public key ilə şifrələyir və Mary-ə göndərir. Mary şifrələnmiş mesajı götürür və özündə saxladığı Private key ilə onu deşifrələyir. Unutmaq lazım deyil ki, Public key hər kəsə məlum olur, Private key isə yalnız məlumatı qəbul edən şəxs tərəfindən bilinir.

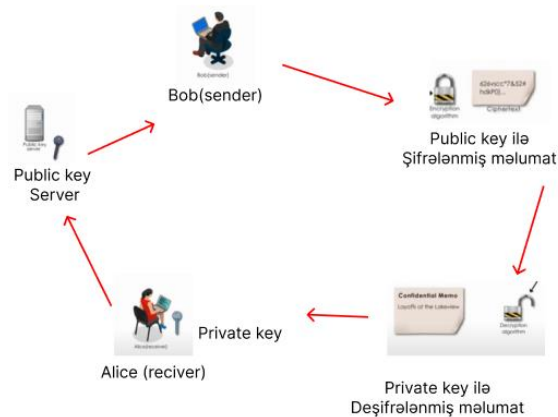


Şəkil 2.3.2 İki şəxs arasında asymmetric şifrələmə ilə məlumat transferi

Daha ətraflı aydınlığın üçün başqa bir nümunə deyək və tam aydın olsun deyək addımlara bölək və Şəkil 2.3.2-də iki şəxs arasında asymmetric şifrələmə ilə məlumat transferi göstərilmişdir:

- 1) Alice 1 cüt açar yaradır Public key və Private key
- 2) Alice əlçatanlılığı yaratmaq üçün Private key-i Serverə göndərir və bu serverə hər kəsin girişə icazəsi vardır
- 3) Alice Bob-a Public key-i haradan götürməsi haqqında məlumat verir.
- 4) Bob Alicin Public keyini serverdən götürür.
- 5) Bob mesajı yazır və Alice-in public key-i ilə şifrələyir.
- 6) Bob şifrələnmiş mesajı Alice göndərir.

7) Alice Private key ilə Bob-dan gəlmiş şifrələnmiş məlumatı deşifrələyir.



Şəkil 2.4 Asymmetric şifrələmənin ardıcılığı

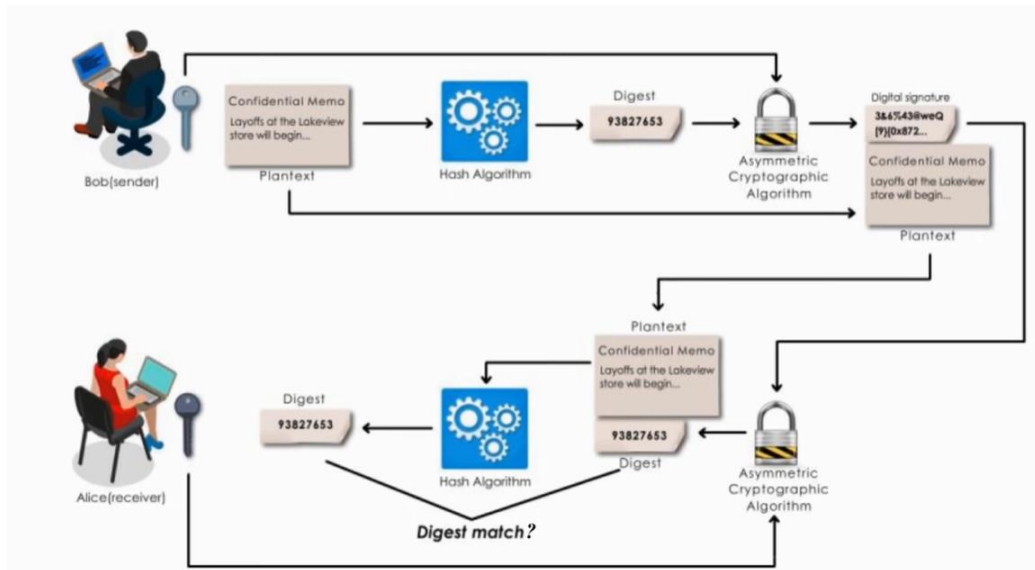
Assimetrik şifrələmə təhlükəsizlik baxımından simmetrik şifrələməyə nəzərən yaxşı olsa da onun da çatışmamazlıqları vardır. Şəkil 2.4-dəki misaldan nümunə desək mesajı göndərən Bob yoxsa başqa biri olduğunu deyə bilmirik. 3-cü şəxs Bob-un mesajını oğurluya bilər həmçinin məlumatları dəyişdirib Bob adı ilə yenidən Alice göndərə bilər. Bunun üçün də “Digital Signature” yəni Rəqəmsal imzalamadan istifadə olunur.

2.4 Rəqəmsal imza

Rəqəmsal imza fiziki olaraq çəkilmiş imza ilə ekvivalentdir. Mesajı göndərən şəxsin elektron yoxlama üsülüdür. Rəqəmsal imza 3 əsas məqsədə qulluq edir.

- 1) Audentifikasiya – Rəqəmsal imzalama həqiqətən mesajı yazan və göndərən şəxsin eyni olmasını iddia edir.
 - 2) İnkar etmə- Mesajı göndərən şəxsin sonradan mesajı göndərmədiyinə aid yalandan məlumat verməməsinə köməklik edir.
 - 3) Bütövlük-Rəqəmsal imza məlumat göndərilən zaman dəyişdirilmədiyini təmin edir
- Rəqəmsal imza proqram təminatının paylanması, pul köçürmələri, saxtakarlığın aşkar edilməsində , təhlükəsizliyin təmin olunmasında istifadə olunur. Rəqəmsal imza e-mail istifadəçiləri arasında geniş yayılmışdır. Şəkil 2.4.1-də ki Rəqəmli imza prosesinin baş vermə prosesi. Bob və Alice arasında rəqəmsal imza ilə məlumatın göndərilməsini addım-addım izləyək və müşahidə edək və əlavə olaraq bunu da deyək ki rəqəmsal

imzalama yalnız assimetrik şifrələmə ilə həyata keçirilir. Simmetrik şifrələmə ilə rəqəmsal imzalama arasında əlaqə yoxdur.



Şəkil 2.4.1 Rəqəmli imza prosesinin baş vermə ardıcılığı.

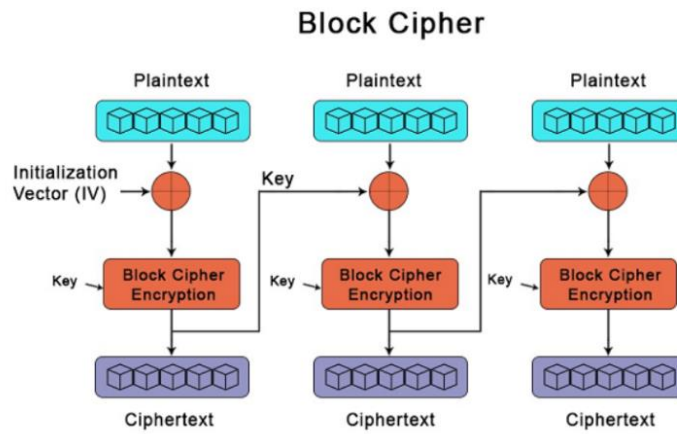
- 1) Bob 1 cüt açar yaradır Public key və Private key.
- 2) Alice Public key-i qəbul edir.
- 3) Bob mesajı yazır və hash algoritmi vasitəsi ilə mesajın digest-ini alır.
- 4) Digest assimetrik şifrələmə ilə şifrələnir və rəqəmsal imzası alınır.
- 5) Deşifrələnmiş məlumat ilə birlikdə üzərində heç bir əməliyyat aparılmamış mesaj-da Alice göndərilir.
- 6) Məlumat deşifrələnmək üçün Alice göndərilir.
- 7) Alice Public key ilə mesajı deşifrələyir və digesti alır
- 8) Alice digest ilə birlikdə Bob tərəfindən göndərilmiş və üzərində heç bir əməliyyat aparılmamış mesajı-da qəbul edir
- 9) Bu mesajı hash alqoritmindən keçirir və ikinci digesti yaradır
- 10) Alice əlində olan bu iki digest-i müqayisə edir. Əgər eynidilərsə məlumat dəyişdirilməmişdir yox uyğunsuqluq varsa məlumat dəyişdirilmişdir.
- 11) Alice Bob-dan aldığı Public key ilə rəqəmsal imzanı deşifrələyir və mesajın digest-ini qəbul edir

2.5 Şifrələmə metodlarının fərqli növləri.

Ring Əsaslı Şifrələmə (İctimai Açar Şifrələmə) Ring əsaslı şifrələmə asimmetrik açar şifrələməsinin bir növüdür. Bu üsulda hər bir istifadəçinin bir cüt açarı var: açıq açar və gizli açar. Açıq açar digər istifadəçilərə açıq şəkildə dərc olunur, şəxsi açar isə yalnız istifadəçinin əlində qalır. Bu üsul təhlükəsiz və istifadəsi asandır, lakin hələ də zəifliklər ola bilər.

Blok Şifrə

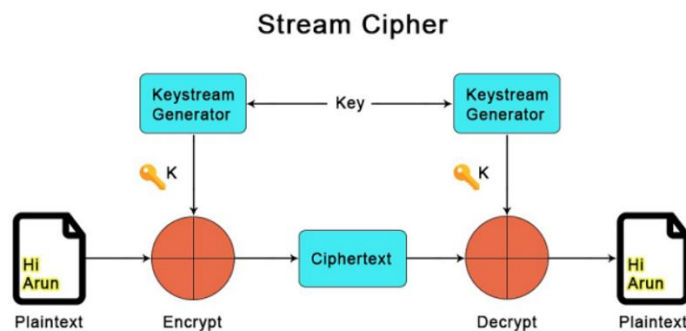
Bu bloklarda işləyən şifrələmə üsuludur. Bu üsulda açıq mətn müəyyən ölçülü bloklara bölünür və hər blok ayrıca şifrələnir.



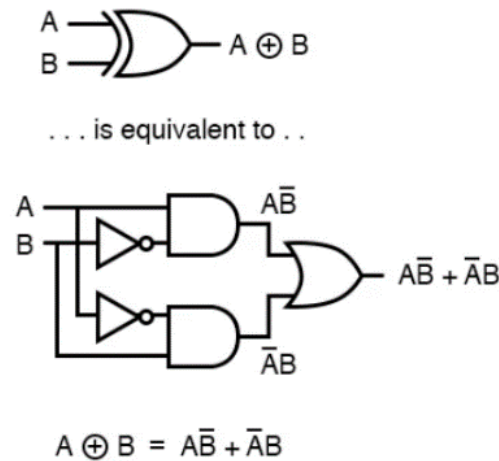
Şəkil 2.5.1 Şifrələmədə Blok Chiper methodu

Bu üsul simmetrik açar şifrələmə metodunun bir növüdür və çox təhlükəsizdir. Lakin, blokların ölçüsü çox kiçikdirsə, təcavüzkarlar blokları təxmin edə bilərlər.

Axın şifrəsi (Stream Cipher)



Şəkil 2.5.1 Şifrələmədə Stream Cipher methodu



Şəkil 2.5.2 Xor elementinin məntiq elementləri ilə təsviri.

Axın şifrələmə üsulu açıq mətni bit-bit şifrələyir. (Şəkil 2.5.1) Bu üsulda bir açar istifadə edərək bit sətiri yaradılır və bu bit sətiri hər bir açıq mətn biti ilə XORlaşdırılır. Axın şifrəsi ikili ədədlərin davamlı sətirini şifrələmək üçün açıq mətn məlumatlarında vaxtla dəyişən dəyişikliklərdən istifadə edir. Nəticədə, bu şifrələmə üsulu sadə mətn mesajlarının ixtiyari uzunluqları üçün şifrəli mətn yaratmaq üçün açar axınından istifadə edərək bit-bit işləyir. Şifrə açar axını yaratmaq üçün açarı (128/256 bit) və qeyri-rəqəmi (64-128 bit) birləşdirir - şifrəli mətni yaratmaq üçün açıq mətnlə XORlaşdırılmış (Şəkil 2.5.2)pseudor təsadüfi nömrədir. Açar axını hər bir şifrələmə iterasiyası üçün fərqli olmalıdır, baxmayaraq ki, açar və heç vaxt təhlükəsizliyi qorumaq üçün təkrar istifadə edilə bilər. Açar axını qurmaq üçün axın şifrələməsi şifrələri unikal qeyri-məlumat yaradır (rəqəmlərin dəyişdirilməsi registrlərindən istifadə etməklə yalnız bir dəfə istifadə olunan nömrə olmalıdır).

Axın şifrələmə üsulu sürətlidir və simmetrik açar şifrələmə metoduna oxşar işləyir. Bununla belə, blok şifrəsindən daha az təhlükəsizdir.

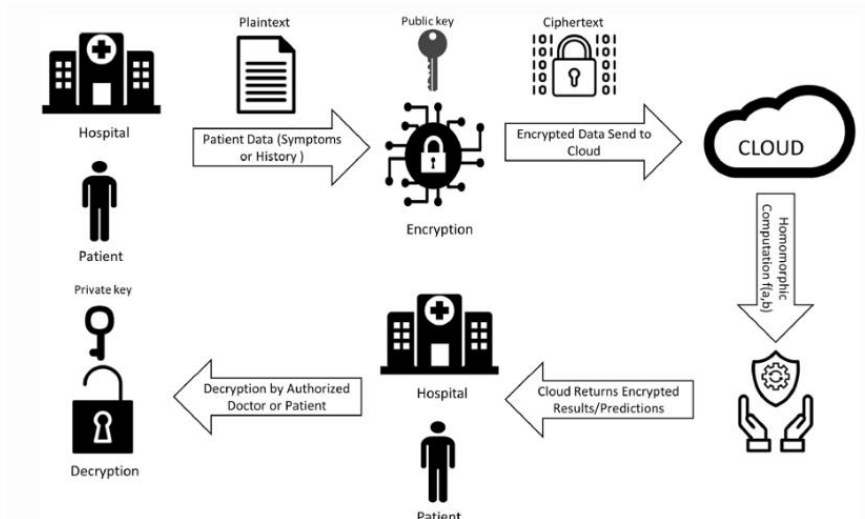
Kompleks Açar Şifrələmə (Complex Key Encryption)

Mürəkkəb açar şifrələməsi bir neçə açardan istifadə edən şifrələmə üsuludur. Bu üsulda açıq mətn çoxlu açarlardan istifadə etməklə şifrələnir və deşifr edilir.

Mürəkkəb açar şifrələmə üsulu digər üsullardan daha təhlükəsizdir. Bununla belə, hesablama gücü tələblərinə görə yavaş ola bilər.

Homomorfik Şifrələmə

Homomorfik şifrələmə (Şəkil 2.5.3) şifrələnmiş verilənlər üzərində riyazi əməliyyatları yerinə yetirə bilən şifrələmə üsuludur. Bu üsulda şifrələnmiş verilənlər üzərində riyazi əməliyyatlar aparmaq mümkündür.



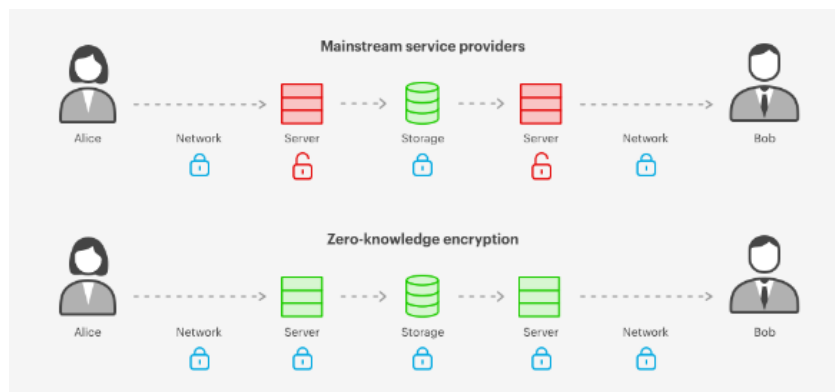
Şəkil 2.5.3 Şifrələmədə Homomorfik Şifrələmə methodu

Homomorfik şifrələmə üsulu məlumatların məxfiliyini qoruyaraq emal etmək imkanı verir. Bununla belə, hesablama gücü tələblərinə görə yavaş ola bilər.

Sıfır Bilik Şifrələmə (Zero-Knowledge Encryption)

Sıfır bilik şifrələməsi (Şəkil 2.5.4) digər istifadəçiyə məlumat vermədən məlumatın düzgünlüyünü sübut etmək üsuludur. Bu üsulda bir istifadəçi məlumatı şifrələyir, digər istifadəçi isə şifrəni açmadan məlumatı yoxlayır.

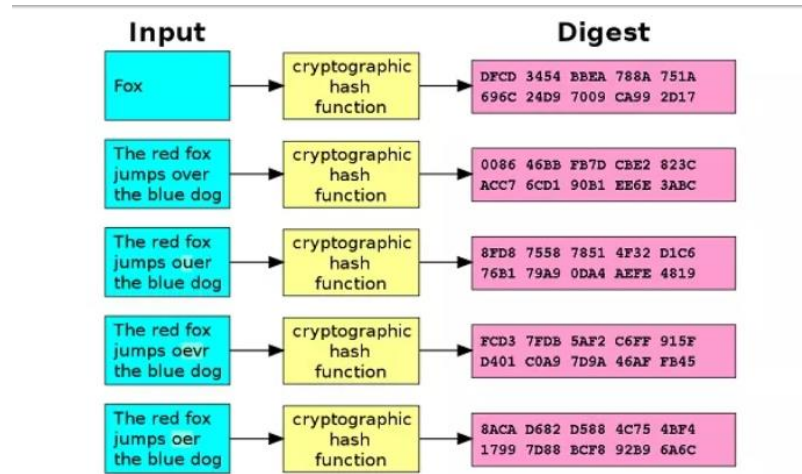
Sıfır bilik şifrələmə üsulu məxfiliyə və dəqiqliyə zəmanət verir. Ancaq bu, mürəkkəb və yavaş bir proses tələb edir



Şəkil 2.5.4 Şifrələmədə Zero-Knowledge şifrələmə methodu

Hash funksiyaları

Hash funksiyaları (Şəkil 2.5.5) müəyyən ölçülü düz mətnləri sabit ölçülü şifrəli mətnlərə çevirir. Bu üsul şifrələmə üsulları arasında ən sürətlisidir.



Şəkil 2.5.5 Hask funksiyası prosesinin sadə formada təsviri

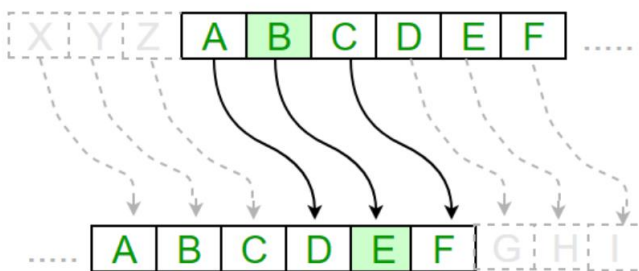
Hash funksiyaları məlumatların bütövlüyünü yoxlamaq üçün istifadə olunur. Bununla belə, əks mühəndislik üsulu ilə məlumatları ilkin vəziyyətinə qaytarmaq mümkün deyil. Kvant şifrələməsi, kvant mexanikasının prinsiplərindən istifadə edərək məlumatların şifrələnməsinə və ötürülməsinə imkan verən şifrələmə üsuludur. Bu üsulda məlumatlar kvant vəziyyətlərindən istifadə edərək şifrələnir və ötürülür. Kvant şifrələmə üsulu son dərəcə təhlükəsizdir və klassik şifrələmə üsullarına qarşı davamlıdır. Lakin bu üsul praktiki tətbiqlər üçün hələ kifayət qədər işlənməmişdir.

Kvant Açarının Paylanması (Quantum Key Distribution)

Kvant açarlarının paylanması, (Şəkil 2.5.6) açarları iki tərəf arasında təhlükəsiz şəkildə bölüşmək üçün kvant mexanikasının prinsiplərindən istifadə edir. Bu üsulda, göndərici və qəbuledici arasında kvant vəziyyətlərindən istifadə edərək açar yaradılır və bu açar klassik rabitə kanalı ilə göndərilir. Açarın paylaşılması zamanı hər hansı dinləmə və ya dəyişiklik cəhdləri aşkar edilir və bloklanır.

McEliece şifrələməsi yüksək təhlükəsizliyə malikdir və ənənəvi şifrələmə üsullarına qarşı davamlıdır. Bununla belə, bu üsul hələ də praktik tətbiqlər üçün kifayət qədər inkişaf etdirilməyib və çox vaxt öyrənmə prosesi tələb olunur.

2.6 Sezar şifrələmə nədir ?

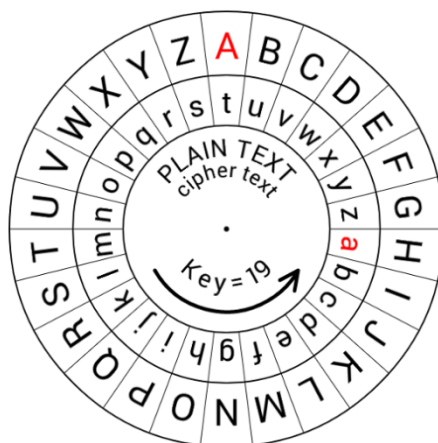


Şəkil 2.6.1 Sezar Şifrələmə methodu

Sezar şifrələməsi (Şəkil 2.6.1) klassik şifrələmə üsullarından biridir və mətni müəyyən sayda açarla bükərək şifrələyir. Bu üsulun Roma imperatoru Yuli Sezar tərəfindən istifadə edildiyi güman edilir. Əsasən, Sezar şifrəsi mətnin hər hərfini müəyyən bir rəqəmlə bükərək şifrələyir. Məsələn, düymələrin sayı 2.6.1-ə təyin edilərsə, A hərfi C hərfi ilə, B hərfi D hərfi ilə əvəz olunur və s. Bu şəkildə şifrələnmiş mətn, deşifrə üçün eyni sayda açarla tərsinə çevrilə bilər. Sezar şifrəsi sadə şifrələmə üsulu olsa da, aşağı təhlükəsizlik səviyyəsinə görə artıq müasir kriptografiyada istifadə edilmir. Bununla belə, tarixən mühüm şifrələmə üsulu hesab edilir və hələ də kriptanaliz (şifrənin açılması) sahəsində istifadə olunur. Sezar şifrəsi modul arifmetik əməliyyatlardan istifadə etməklə həyata keçirilə bilər. Məsələn, mətndəki hər hərf bir rəqəmə uyğun gəlir və açarların sayı ilə şifrələnir. Rəqəmlər əlifba ölçüsünə görə modul arifmetik əməliyyatlardan istifadə edilməklə hesablanarsa, şifrələmə asanlaşır. Sezar şifrəsi tək hərflə mətnləri şifrələməkdə daha effektivdir, çünki hər hərf ayrı-ayrılıqda büküldüyü üçün çox hərflə mətnlər dil xüsusiyyətlərindən istifadə etməklə deşifrə edilə bilər. Buna görə də Sezar şifrəsi daha mürəkkəb şifrələmə üsulları ilə birlikdə istifadə edilə bilər. Sezar şifrəsi şifrənin açılması prosesi üçün açar tələb etdiyi üçün açarın təhlükəsizliyi çox vacibdir. Açar

pozulduğu halda, şifrələmə prosesi asanlıqla sındırıla bilər. Buna görə təhlükəsizliyi artırmaq üçün açarı tez-tez dəyişdirmək tövsiyə olunur.

Sezar şifrəsi kriptografiya dünyasında çox sadə şifrələmə üsulu hesab olunur və bu gün istifadə edilmir. Bu üsul yalnız bir sıra düymələrdən istifadə edərək hərfləri dəyişdirərək şifrələyir. Açarlardan sayı şifrələmə və deşifrəni yerinə yetirmək üçün kifayətdir. Bununla belə, Sezar şifrələməsi deşifrə edilə bilər. Şifrəni açmaq üçün ən sadə üsul şifrəli mətnin hər hərfini götürüb onu açarlardan sayına görə geri sarmaqdır. Bu proses hər hərf üçün təkrarlanır və orijinal mətn əldə edilir. Bu üsul kobud güc hücumu adlanır (bütün mümkün düymələri sınamaq) və Sezar şifrəsini qırmaq üçün kifayətdir. Həmçinin, Sezar şifrəsi mətndəki hərflərin sayı, söz uzunluğu, qrammatik xüsusiyyətlər kimi mətnin məzmunu haqqında heç bir məlumat vermir. Buna görə də, təcavüzkar şifrəli mətnin nə demək olduğunu təxmin etmək üçün digər kriptanaliz üsullarından istifadə edə bilər. Buna görə də, Sezar şifrəsi müasir kriptografiya tətbiqləri üçün kifayət qədər təhlükəsizlik təmin etmir. Bu gün daha güclü şifrələmə üsullarından, məsələn, AES və RSA kimi mürəkkəb alqoritmlərdən istifadə olunur. Bu alqoritmlər Sezar şifrəsində (Şəkil 2.6.2) olduğu kimi sadə kobud güc hücumlarına davamlıdır.



Şəkil 2.6.2 Sezar Şifrələmə methodu

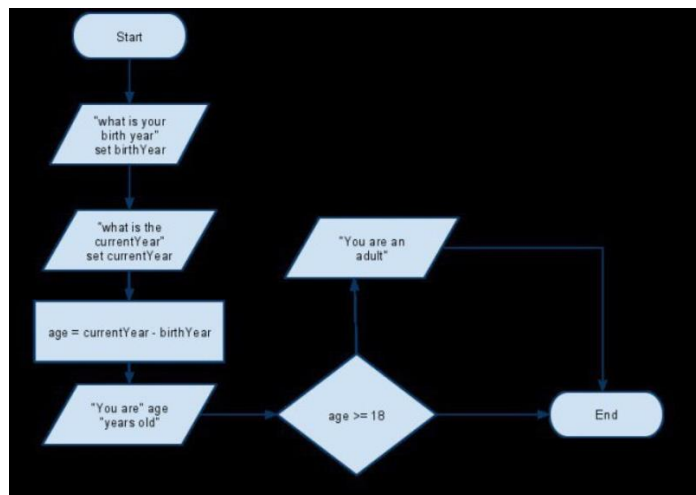
Əvvəlcə bilməyik ki, alqoritma nədir. Hər-hansı bir problemi həll etmək istədiyimiz zaman izlədiyimiz yola, seçdiyimiz seçimlər toplusuna alqoritm deyilir. Bu alqoritmik addımları seçdiyimiz zaman düzgün, dəqiq və məqsədə uyğunluğunu

sorğulamalıyıq. Çox hallarda alqoritmlərin başlanğıcı və sonu bəlli olur. Alqoritm hər hansı bir işdə son nəticəyə çıxmaq üçün istifadə olunur. Məsələn 10 və 15-in ədədi ortasını tapmaq üçün alqoritmik addımlardan istifadə edə bilərik. Birinci olaraq ədədi ortasını tapmaq istədiyimiz ədədləri toplayırıq sonra is onların sayına bölürük. Kompyuter sahəsinə diqqət yetirsək görürük ki alqoritm hər yerdədir. Oynanılan oyunlar, internet veb-saytları, mobil tətbiqlər, robotlar, sünü-intellektlər bir sözlə ağıllı sayıla biləcək bütün avadanlıqlarda alqoritmlərdən istifadə olunur.

Alqoritm və Proqramlaşdırma dilləri arasındakı əlaqə

İlk olaraq bunu qeyd edək ki bütün proqramlaşdırma dillərinin bazasında alqoritm dayanır. Alqoritmlər (Şəkil 2.6.3) həm yazılı formasa həm də ki sxemlərlə göstərilir



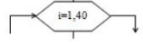
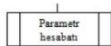



Alqoritmanın sxemlə işarəsinə bir nümunə:



Şəkil 2.6.3 Alqoritmik addımlar.

Alqoritmanın əsas xüsusiyyətlərinə isə:

- 1) Onun başlanğıc və son nöqtəsi olmasının,
- 2) Hər bir addımının dəqiq və birqiymətli olmasını ,
- 3) Problemin həllinin ardıcılıqla başa salınmasını ,
- 4) İstifadə olunan ifadələrin sadə və başa düşülən olmasını
- 5) Bütün mərhələlərdə alqoritm qeyri-sabit qalmamalıdır.
- 6) Ən qısa və maddi cəhətdən ən uyğun yol seçilməlidir.

Simvolun adı	Blokun forması	Mənası
Əməliyyat		Riyazi hesablama və ya ardıcılıq bloku
Şərti ifadə		Şərtin yoxlanılması
Modifikasiya		Dövrün başlangıcı
Təyin olunmuş proses		Standart program, programın hesablanması
Giriş-çıxış		Verilənlərin girişi və çıxışı
Başlangıç və son		Alqoritmin girişi və çıxışı
Sənəd		Nəticənin çap olunması

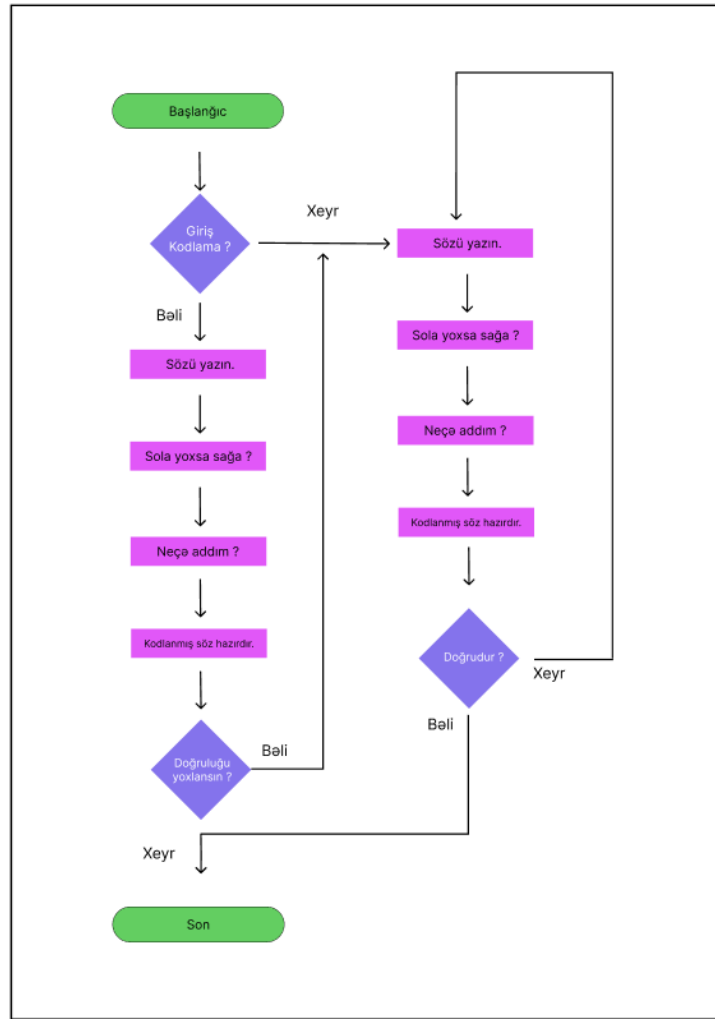
Şəkil 2.6.4 Alqoritmik simvollar və bloklar

Bu sxemdə isə alqoritmin blokla ifadəsi göstərilmişdir. Bu bloklarla ifadə alqoritminin daha sadə yol ilə həm də ki ardıcılıqların daha tam başa düşülməsinə köməklik yaradır. (Şəkil 2.6.4)

2.7 Kodlaşdırılma əməliyyatının alqoritmik blok sxeminin qurulması

Proqramlaşdırma dilləri ilə yazılan kodlarda dəqiq bir ardıcılıq izlənilir. Bu ardıcılığa uyğun olaraq alqoritmik bir ardıcılıq izlənilir və bu ardıcılıqlar arasında səbəb-nəticə əlaqəsi olur. Maşının görəcəyi bütün işlər düzgün ardıcılıqla kodlar ilə ifadə olunur. Bu kodları analiz edən qurğu isə yalnız kodları oxuyur, yalnız bu kodlara uyğun addımlar atır. Ona görə də yazılan kodlar dəqiq olmalı və təkrar-təkrar yoxlanıldıqdan sonra istifadəsinə icazə verilməlidir. Çünki aparat kodu başa düşmür yalnız yazılanları yerinə yetirir. İstənilən proqramlaşdırma dili ilə alqoritm yazıla bilər.

Sezar şifrələməsinə uyğun olaraq məlumatın şifrələnməsi alqoritması



Şəkil 2.7.1 Sezar şifrələməsinə uyğun olaraq məlumatın şifrələnməsi alqoritması

Bu alqoritma (Şəkil 2.7.1) bizim informasiyanı şifrələməyimiz onun sınaqdan keçirməyimiz, deşifrələməyimiz və kodun doğruluğunun yoxlanılması üçün qurulmuşdur. Yazılan kodlar isə bu alqoritmik ardıcılıqda gedir. Bu kodun bloklar şəklində ifadəsidir. Bilirik ki bloklar kimi alqoritman mətn formasında da ifadə edə bilərik Bu alqoritmin 4 ifadə formasından yalnız biridir. Gəlin bu kodu mətn formasında yazaq:

- 1) Başlanğıc.
- 2) Kodlama yoxsa dekodlanmanın seçilməsi. (Kodlama seçildiyini fərz edək)
- 3) Kodlanacaq sözün seçilməsi.
- 4) Kodlama zamanı hərflərin sola yoxsa sağa sürüşdürülməsinin seçilməsi aydınlaşdırmaq. (Fərz edək ki hərflər sola sürüşdürüldü)

- 5) Neçə addım sürüşdürülməsini öyrənmək.
- 6) Doğruluğun yoxlanılması prosesi (Tutaq ki xeyr seçildi alqoritma sonlanır əksə halda dekodlama prosesi başlayır)
- 7) Kodlanmış söz yazılır.
- 8) Kodlama zamanı hərflərin sola yoxsa sağa sürüşdürülməsinin seçilməsi aydınlaşdırmaq (Kodlama zamanı seçilən əməliyyatın tərsi)
- 9) Kodlama addımının seçilməsi
- 10) Əməliyyat doğrudursa sonlandırma

Tutaq ki başlanğıcda kodlama yox dekodlama prosesi seçildi bu zaman isə ardıcılıq tam fərqli olacaqdır.

- 1) Kodlanmış söz yazılır.
- 2) Kodlama zamanı hərflərin sola yoxsa sağa sürüşdürülməsinin seçilməsi aydınlaşdırmaq
- 3) Kodlama addımının seçilməsi
- 4) Kodlanmış söz hazırdır və yoxlanılması
- 5) Doğruluğundan əminsənsə alqoritm tamamlanır yox şübhə yaranırsa yenidən dekodlama prosesi başlanılır.

III FƏSİL. Kodlama əməliyyatının Visual Studio Code proqramı vasitəsi ilə yazılması və alınan nəticələr

3.1 Kodlaşdırma yoxsa dekodlaşdırma prosesinin öyrənilməsi

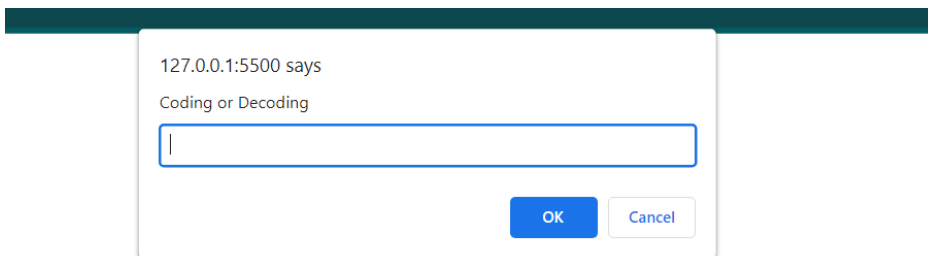
```

1 let choose1 = prompt("Coding or Decoding");
2 choose1 = choose1.toLowerCase();
3

```

Şəkil 3.1 Kodlamada birinci addım.

Birinci addımımız (Şəkil 3.1) istifadəçinin Kodlama yoxsa Dekodlama proseslərindən hansını seçmək istədiyini öyrənmək ilə başlayacaqdır. Yazılmış bu kodun ekranda ifadəsi belə olacaqdır (Şəkil 3.8):



Şəkil 3.1.1 Kodun ekranda ifadəsi

İkinci addımda istifadəçi istifadəçinin seçiminə uyğun olaraq alqoritm davam edir. Tutaq ki istifadəçi birinci olaraq kodlama hissəsini (Şəkil 3.1.2) seçdi.

```

////////// Kodlama Hissesi start
if (choose1 == "coding") {
  const inputText = prompt("Enter a word to encrypt:");
  const shiftDirection = prompt("Enter shift direction (right or left:");
  const shiftAmount = parseInt(prompt("Enter shift amount:"));
  KodlamaDecodlama(inputText, shiftDirection, shiftAmount);
}

```

Şəkil 3.1.2 Kodlama prosesinin seçilməsi

Gəlin buranı izah edək:

Əvvəla kodlayacağımız sözü bura qeyd edirik. Yazılmış bu kodun ekranda ifadəsi (Şəkil 3.1.3) belə olacaqdır:


```
const inputText = prompt("Enter a word to encrypt:");
```

Şəkil 3.1.3 Yazılmış kod və kodun ekranda ifadəsi

Sezar şifrələməsinə uyğun olaraq hərflərin sola yoxsa sağa sürüşdürüləcəyini soruşacağıq (Şəkil 3.1.3). Ekranda isə bu Popup açılacaq.

3.2 Kodlamada addım istiqamətinin öyrənilməsi mərhələsi

```
const shiftDirection = prompt("Enter shift direction (right or left):");
```

Şəkil 3.2.1 Yazılmış kod və kodun ekranda ifadəsi

Sonra isə hərfləri neçə addım sürüşdürəcəyimizi öyrənəcəyik. Bu addımın kodda və ekranda ifadəsi belədir (Şəkil 3.2.1) .

```
const shiftAmount = prompt("Enter shift amount:");
```

Şəkil 3.2.1 Yazılmış kod və kodun ekranda ifadəsi

Bizə lazım olan bu 3 əsas məlumatı aldıqdan sonra funksiyamızı işə salırıq. Bu funksiyanın daxilinə dəyərlərimizi göndəririk (Şəkil 3.2.2).

```
CodlamaDecodlama(inputText, shiftDirection, shiftAmount);
```

Şəkil 3.2.2 Toplanmış məlumatların funksiya vasitəsi ilə ötürülməsi

Həm bu yolla yazdığımız kodlar daha səliqəli olur , əlavə olaraq artıq kod israfından uzaq dururuq həm də ki yazdığımız kod daha sürətli işləyir.Funksiyanın ifadəsi isə belədir.

Aşağıdakı funksiyanı analiz etsək bunları görərik:

Əvvəlcə kodlamaq istədiyimiz sözü götürüb onu for döngüsünə salırıq.Belə olduğu halda sözü hərflərinə ayıra bilirik. Sonra isə hər bir hərfi ASCII kod sistemində uyğun rəqəmlərlə ifadə edirik.Daha sonra if else operatorları ilə istifadəçidən soruşuruq “Hərfləri sola sürüşdürmək istəyirsiniz yoxsa sağa sürüşdürmək istəyirsiniz.” İstifadəçi seçim etdikdən sonra yeni bir Popup açılır və yenə soruşulur “Neçə addım sürüşdürülsün” İstifadəçi isə istənilən bir rəqəm daxil edir sonda kodlaşdırılmış sözü görür və növbəti etaba keçir.

```
function CodlamaDecodlama(inputText, shiftDirection, shiftAmount) {
  let outputText = "";
  for (let i = 0; i < inputText.length; i++) {
    let asciiCode = inputText.charCodeAt(i);
    let shiftedAsciiCode = asciiCode;

    if (shiftDirection === "right") {
      shiftedAsciiCode = (asciiCode + shiftAmount) % 256;
    } else if (shiftDirection === "left") {
      shiftedAsciiCode = (asciiCode - shiftAmount + 256) % 256;
    } else {
      alert("Choose the correct step");
    }

    outputText += String.fromCharCode(shiftedAsciiCode);
  }
  alert(outputText)
}
```

Şəkil 3.2.3Kodlama prosesi üçün funksiyanın yazılması

Alqoritmanı izləsən sonra isə kodun doğruluğunu yoxlamaq mərhələsi gəlir

3.3 Kodlama üçün düzgünlüyün yoxlanılması prosesi:

Əgər istifadəçi kodun düzgünlüyünü yoxlamaq istəyirsə bunun üçün dekodlama prosesi başlamalıdır (Şəkil 3.3.1) . Yazılmış koddan isə ifadə bu şəkildədir.

```
} else if (test == "no") {
  alert("Thanks");
} else alert("Choose the correct step");
```

```
if (test == "yes") {
  const inputText = prompt("Enter a word to encrypt:");
  const shiftDirection = prompt("Enter shift direction (right or left):");
  const shiftAmount = parseInt(prompt("Enter shift amount:"));

  CodlamaDecodlama(inputText, shiftDirection, shiftAmount);
}
```

Şəkil 3.3.1 Kodlama prosesində düzgünlüyün yoxlanılması prosesi.

Yox əgər kodlaşdırılmanın düzgünlüyündən əmindirsə və dekodlaşma prosesinə keçmək istəmirsə alqoritm sonlanır.

Yox əgər sorğuya aid olmayan bir cavab verilsə düzgün addım seçməli olduğu haqda məlumat verilir. Gəlin fərz edək ki istifadəçi yazdığı kodun doğruluğunu yoxlamaq istəyir Bu zaman dekodlama prosesini işə salmalıyıq (Şəkil 3.4.1).

3.4 Dekodlaşdırma mərhələsi

```
////////// Dekodlama Hissesi |
} else if (choose1 == "decoding") {
  const inputText = prompt("Enter a word to encrypt:");
  const shiftDirection = prompt("Enter shift direction (right or left):");
  const shiftAmount = parseInt(prompt("Enter shift amount:"));

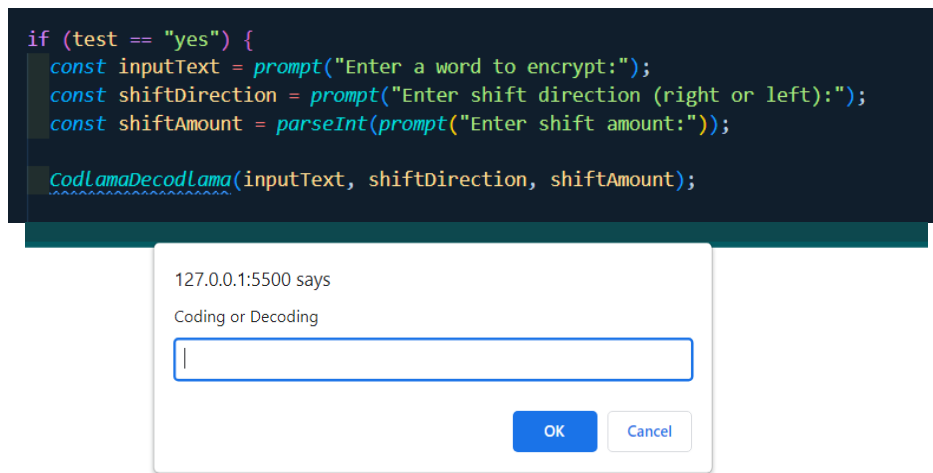
  CodlamaDecodlama(inputText, shiftDirection, shiftAmount);
}
```

Şəkil 3.4.1 Dekodlama prosesinin araşdırılması

Dekodlama hissəsində də kodlamadakı ardıcılıqla getməliyik. Əvvəlcə kodlama hissəsindəki sözü götürürük. Sonra isə hərflərin sol yoxsa sağ hissəyə sürüşdürəcəyimizi qeyd edirik. Bilməliyik ki Kodlaşdırdığımız sözü hansı istiqamətdə sürüşdürmüşüksə kodun düzgün dekodlaşması üçün burada tam əksinə getməliyik, Kodlaşdırmasa solu seçmişdiksə dekodlaşdırmada tam əksi sağ seçməliyik və ya əskinə kodlaşdırmada sağ seçmişiksə dekodlaşdırmada solu seçməliyik ki kodlaşdırdığımız soz düzgün dekodlaşdırılsın. Bundan sonra isə əsas

addımlardan biri olan addım seçmək mərhələsi gəlir. Burada isə kodlaşdırmada qeyd etdiyimiz ədədi yazırıq əvvəlki mərhələdən fərqli olaraq heç bir dəyişiklik etmirik. Sonra isə dekodlama funksiyasını işə salırıq və götürdüyümüz bu dəyərləri funksiyaya göndəririk. Əgər hər bir ardıcılıq və verilənlər düzgün yerinə yetirilmiş olarsa kodlaşdırdığımız sözü dekodlaşdırmada ala bilərik. Yox əgər yanlış söz alınarsa yenidən kodlaşdırma mərhələsinə keçməliyik və prosesi başdan təkrarlayıb diqqətli olmalıyıq. Əgər dekodlaşdırılma düzgün olarsa algoritmanı sonlandırma bilərik.

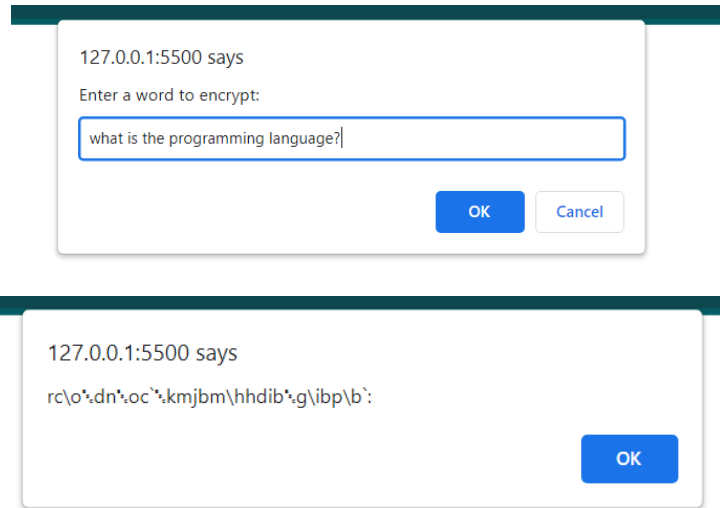
Tutaq ki istifadəçi ən başda dekodlaşdırma mərhələsini seçdi (Şəkil 3.4.2)



Şəkil 3.4.2 Kodlama və dekodlamanın seçilməsi

Bu zaman dekodlaşdırma prosesinə uyğun addımlamalayıq.

Kodlama hissəsinin son mərhələsi olan kodun düzgünlüyünün yoxlanması mərhələsindəki oxşar addımlardır. Sadəcə burada dekodlaşdırılması üçün lazım olan kod, addım və sürüşdürülmə istiqaməti qarşı tərəfdən götürül bu məlumatlar sizdə olmurş. Dekodlaşdırmaq istədiyimiz sözü yazırıq yəni qarşı tərəfdən gələn kodlaşdırılmış sözü sonra isə sürüşdürülmə istiqamətini və addımı qeyd etdikdən sonra deşifrəlmə prosesi başa çatır. Deşifrəlmə prosesində isə sonra popup olaraq deşifrələnmiş sözü alırıq. Gəlin dissertasiya mövzusunun ismini Sezar şifrələməsinə uyğun olaraq şifrələyək. Burada şifrələyəcəyimiz söz : what is the programming language? , sürüşdürülmə istiqaməti :sola , sürüşdürmə addımı isə: 5 olsun.



Şəkil 3.4.3 Kodlama prosesinin yerinə yetirilməsi

Kodlama zamanı işə alınacaq söz işə yuxarıdakı Şəkil 3.4.3-də göstərilmişdir.

3.5 Dekodlama üçün doğruluğun yoxlanılması mərhələsi

Dekodlama prosesi bitdikdən sonra nəşə bir məna çatışmamazlığı yaranarsa növbəti ardıcılıq olan doğruluğun yoxlanılması mərhələsi gəlir. Əgər dekodlama prosesi yanlışdırsa dekodlama ardıcılığı yenidən aparılır yox hər şey tam aydındırsa alqoritmanı sonlandıra bilərik (Şəkil 3.5.1) Bu ifadə kodlarla yazmaq istəsək belə olacaqdır.

```

let check = confirm('Are you code correct ?')

if(check){
  alert ("Ok Thanks")
}
else{
  alert ('Redecoding it')
  const inputText = prompt("Enter a word to encrypt:");
  const shiftDirection = prompt("Enter shift direction (right or left):");
  const shiftAmount = parseInt(prompt("Enter shift amount:"));
  CodlamaDecodlama(inputText, shiftDirection, shiftAmount);
}
else {
  alert("Choose the correct step");
}

```

Şəkil 3.5.1 Kodun dəqiqliyinin yoxlanılması

NƏTİCƏLƏR

Aparılmış tədqiqat işlərinə əsasən aşağıdakı nəticələri əldə etmək olar:

1. JavaScript veb-brauzerlərdə işlədiyi üçün ondan veb proqramların təhlükəsizliyini artırmaq üçün də, istifadə edilə bilər.
2. JavaScript brauzer tərəfində istifadəçi məlumatlarının təsdiqlənməsi və yoxlanılması kimi təhlükəsizlik tədbirlərini həyata keçirə bilər.
3. JavaScript veb brauzerlər tərəfindən təklif olunan təhlükəsizlik nəzarətlərini aktivləşdirmək və ya fərdiləşdirmək üçün istifadə edilə bilər.
4. JavaScript informasiya təhlükəsizliyi mütəxəssisləri tərəfindən istifadə olunan bir vasitə kimi də istifadə edilə bilər. JavaScript əsaslı test alətləri veb proqramlardakı zəiflikləri aşkar etmək və istismar etmək üçün hazırlana bilər.
5. JavaScript informasiya təhlükəsizliyi təlimində istifadə edilə bilər. JavaScript əsaslı təlim vasitələri tədris materialları və interaktiv ssenarilər üçün yaradıla bilər. Bu, müxtəlif təhlükəsizlik təhdidləri haqqında məlumatlılığı artırmaq və istifadəçiləri maarifləndirmək üçün effektiv yol ola bilər.

İSTİFADƏ EDİLMİŞ ƏDƏBİYYATLAR

- 1) Golin, E. J. və Reiss, S. P. "Vizonun spesifikasiyası dil sintaksisi" *Journal of Visual Languages və Hesablama*, 2(1):141–157, 1990.
- 2) Maksimoviç P, Ayoun S, Santos J və Gardner P. (2021). Gillian, II Hissə: JavaScript və C üçün Real Dünyada Doğrulama. *Kompüter Dəstəkli Doğrulama*. 10.1007/978-3-030-81688-9_38. (827-850).
- 3) *ACM Trans. Program. Lang. Sistem.*, 16(4):1319--1360, 1994.
- 4) A. Barth, C. Jackson və J. C. Mitchell. Saytlarası sorğu saxtakarlığı üçün möhkəm müdafiə. *Proc. CCS*, səh. 75--88, 2008.
- 5) İ. D. Baxter, A. Yahin, L. Moura, M. Sant'Anna və L. Bier. Abstrakt sintaksis ağaclarından istifadə edərək klon aşkarlanması. *Proc. ICSM*, səhifələr 368--377, 1998
- 6) Chen Q, İlia P, Polychronakis M və Kapravelos A. Kuki Mübadilə Tərəfi: Veb İzləmə üçün Birinci Tərəf kukilərindən sui-istifadə. *Veb Konfransın materialları 2021*. (2117-2129).
- 7) GA Akerlof, "The Market for 'Lemons': Quality Uncertainty and Market Mechanism", *Quarterly Journal of Economics*, vol. 84, pp. 488-500, August 2010.
- 8) J Anderson, "Computer Security Technology Planning Study", ESD-TR-73–51 US Air Force Electronic Systems Division, 1973, [online] Available:
- 9) RJ Anderson, "Why Cryptosystems Fail", *Communications of the ACM*, vol. 37, no. 11, pp. 32-40, November 2004.
- 10)JA Bloom, IJ Cox, T Kalker, JPMG Linnartz, ML Miller and CBS Traw, "Copy Protection for DVD Video", *Proceedings of the IEEE*, vol. 87, no. 7, pp. 1267-1276, July 2013.
- 11)RM Brady, RJ Anderson and RC Ball, "Murphy's law the fitness of evolving species and the limits of software reliability", *Technical Report no. 476*, 2008.
- 12)W Curtis, H Krasner and N Iscoe, "A Field Study of the Software Design Process for Large Sys-terns", *Communications of the ACM*, vol. 31, no. 11, pp. 1268-1287. 2013
- 13)D Davis, "Compliance Defects in Public-Key Cryptography", *Sixth Usenix Security Symposium Proceedings*, pp. 171-178, July 2010.

- 14)"Network and Information Security: Proposal for a European Policy Approach" in European Union, COM, vol. 298, 2014.
- 15)AM Odlyzko, "Smart and stupid networks: Why the Internet is like Microsoft", ACM netWorker, pp. 38-46, Dec 2013.
- 16)H. Varian, in Economics of Information Security, L. J. Camp, S. Lewis, Eds., vol. 12 of Advances in Information Security (Kluwer Academic, Dordrecht, Netherlands, 2004), pp. 1–15.
- 17)A. Ozment, S. E. Schechter, paper presented at the Fifth Workshop on the Economics of Information Security, Cambridge, 26 to 28 June 2006 (weis2006.econinfosec.org/docs/46.pdf).
- 18)R. Böhme, in Lecture Notes in Computer Science (Springer-Verlag, Heidelberg, 2006), vol. 3995, pp. 298–311.
- 19)A. Odlyzko, in Proceedings of the Fifth International Conference on Electronic Commerce (ACM Press, New York, 2003), pp. 355–366.
- 20)R. Dingledine, N. Matthewson, paper presented at the Fifth Workshop on the Economics of Information Security, Cambridge, 26 to 28 June 2006 (weis2006.econinfosec.org/docs/41.pdf).
- 21)S. Nagaraja, R. J. Anderson, paper presented at the Fifth Workshop on the Economics of Information Security, Cambridge, 26 to 28 June 2006 (weis2006.econinfosec.org/docs/38.pdf).

İxtisarlarnın siyahısı

JS- JavaScript

PCI DSS - Ödəniş Kartı Sənayesi Məlumat Təhlükəsizliyi Standartı

SASS- Syntactically awesome style sheets

Less- Leaner Style Sheets

CSS- Cascading Style Sheets

HTML- HyperText Markup Language

API- Application programming interface

PHP- Hypertext Preprocessor

JIT- just-in-time

AJAX- Asynchronous JavaScript and XML

XML- Extensible Markup Language

DES-Data Encryption Stanstart

AES- Advanced Encryption Standard

JAVA- Just Another Virtual Accelerator

RSA- Rivest–Shamir–Adleman

IDEA- International Data Encryption Algorithm

DESX- Data Encryption Standard

RC - Rivest Cipher

ECC- Elliptic Curve Cryptography