Mündəricat

[1. Github versiya nəzarət sistemi 3](#_Toc169788611)

[2. Git-də repository nədir? 3](#_Toc169788612)

[3. Git və Github fərqi 4](#_Toc169788613)

[4. Git-də branch anlayışı 5](#_Toc169788614)

[5. Git-də commitin mahiyyəti 5](#_Toc169788615)

[6. Git repository-ni local kompüterinizə necə klon edə bilərsiz? 6](#_Toc169788616)

[7. git add . əmri nədir? 7](#_Toc169788617)

[8. git commit -m əmri nədir? 8](#_Toc169788618)

[9. git push əmri nədir? 8](#_Toc169788619)

[10. git pull əmri nədir? 9](#_Toc169788620)

[11. git status və git status -s fərqləri nədir? 9](#_Toc169788621)

[12. SSH və HTTPS vasitəsilə Git repository-nin klon edilməsi 10](#_Toc169788622)

[13. İnformasiyanın qlobal şəbəkədə yerləşdirilməsi. 11](#_Toc169788623)

[14. Domain və Hosting anlayışı. FTP protokolu 13](#_Toc169788624)

[15. DNS və nameserver arasındakı fərqlər 15](#_Toc169788625)

[16. Hosting alarkən diqqət edilməli məqamlar 16](#_Toc169788626)

[17. Veb serverlər 17](#_Toc169788627)

[18. Paylanıla bilən və virtual veb serverlər 18](#_Toc169788628)

[19. DNS necə işləyir? 20](#_Toc169788629)

[20. Html strukturu, “Tag” anlayışı, proqram kodunun düzgün tərtib edilməsi mərhələləri 21](#_Toc169788630)

[21. HTML-də teq və atribut anlayışı 22](#_Toc169788631)

[22. HTML-də “boş element” anlayışı 23](#_Toc169788632)

[23. HTML elementlərindən “id” atributu ilə “class” atributu arasında fərq nədir? 24](#_Toc169788633)

[24. HTML teqləri vasitəsilə cədvəllərin tərtib edilməsi. 24](#_Toc169788634)

[25. HTML cədvəlində iki və ya daha çox sətir və ya sütunu bir sətir və ya sütuna necə birləşdirə bilərik? 25](#_Toc169788635)

[26. HTML-də link necə təyin edilir və “target” atributu nə üçündür? 26](#_Toc169788636)

[27. Nizamlı və nizamsız HTML siyahıları 27](#_Toc169788637)

[28. HTML formaları və forma elementləri 28](#_Toc169788638)

[29. HTML formasının vacib atributları hansılardır? 29](#_Toc169788639)

[30. HTML formasında input elementinin vacib atributları hansılardır? 30](#_Toc169788640)

[31. CSS nədir? Üstünlükləri və sintaksisi 30](#_Toc169788641)

[32. CSS-in HTML səhifəsinə inteqrasiya edilməsi üsulları 31](#_Toc169788642)

[33. CSS framework-ləri. Bootstrap. Bootstrapın HTML səhifəsinə inteqrasiya edilməsi 33](#_Toc169788643)

[34. CSS-də “class” və “id” arasındakı fərqi izah edin 34](#_Toc169788644)

[35. CSS-də universal selector dedikdə nə başa düşürsünüz? 36](#_Toc169788645)

[36. POST – GET anlayışı və iş prinsipi 37](#_Toc169788646)

[37. Veb Proqramlaşdırma dilləri 38](#_Toc169788647)

[38. PHP və sintaksisi 40](#_Toc169788648)

[39. PHP scriptlərin təhlükəsizliyi 42](#_Toc169788649)

[40. PHP forma validasiyası 43](#_Toc169788650)

[41. PHP-də açar sözlərin və dəyişənlərin böyük/kiçik hərflərə həssaslığı 44](#_Toc169788651)

[42. PHP-də dəyişənlərin elan edilməsi və çapa verilməsi 45](#_Toc169788652)

[43. PHP data tipləri hansılardır? 45](#_Toc169788653)

[44. PHP-də lokal, qlobal və statik dəyişənlər 46](#_Toc169788654)

[45. PHP-də funksiyalar və sintaksisi 47](#_Toc169788655)

[46. PHP şərhlərində hansı simvollar istifadə edilir? 48](#_Toc169788656)

[47. JavaScript data tipləri hansılardır? 48](#_Toc169788657)

[48. Javascript şərhlərində hansı simvollar istifadə edilir? 49](#_Toc169788658)

[49. Javascript kodları ilə yazılmış 3+2+”7″ nəticəsi nədir? Səbəbi izah edin. 49](#_Toc169788659)

[50. JavaScript. Üstünlükləri və çatışmazlıqları. HTML sənədində tətbiq üsulları 50](#_Toc169788660)

[51. Java və JavaScript arasındakı fərqlər nələrdir? 51](#_Toc169788661)

[52. Verilənlər bazası. MySQL 53](#_Toc169788662)

[53. MYSQL və SQL arasındakı fərqi izah edin 54](#_Toc169788663)

[54. PhpMyAdmində username və password-untəyin edilməsi. MySQL VB-na daxil olma qaydası 56](#_Toc169788664)

[55. PhpMyAdmində VB-nın və cədvəllərin yaradılması mərhələləri. Cədvəllərdə əsas və xarici açar anlayışı 57](#_Toc169788665)

[56. MYSQL və PHP faylı arasında əlaqə üsulları 58](#_Toc169788666)

[57. MYSQL-də CRUD əməliyyatları 59](#_Toc169788667)

[58. MYSQL – də bazanın export və import edilməsi qaydası 59](#_Toc169788668)

[59. Kompüter şəbəkələri. Lokal və qlobal şəbəkə. 59](#_Toc169788669)

[60. Kompüter şəbəkələrinin aparat təminatı 60](#_Toc169788670)

[61. Şəbəkə topologiyaları 61](#_Toc169788671)

[62. Şəbəkə protokolları. İP ünvan 63](#_Toc169788672)

[63. Şəbəkə avadanlıqları 63](#_Toc169788673)

[64. Hub və switch fərqi nədir? 64](#_Toc169788674)

[65. OSİ modeli və səviyyələrin izahı 65](#_Toc169788675)

[66. İnkapsulyasiya və deinkapsulyasiya nədir? 66](#_Toc169788676)

[67. Kabel növləri 67](#_Toc169788677)

[68. Bulud texnologiyaları 68](#_Toc169788678)

[69. Bulud texnologiyalarının strukturu və iş prinsipi 70](#_Toc169788679)

[70. SHA-1, SHA-2 və SHA-256 Hash Alqoritmləri Arasındakı Fərqləri izah edin 71](#_Toc169788680)

[71. GitHub-da repository yaradın və lokal kompüterə klon edin. 72](#_Toc169788681)

## 1. Github versiya nəzarət sistemi

[**Git**](https://git-scm.com/) bir versiya kontrol sistemidir (Version Control System — VCS). Biz kod yazmağa başladıqdan etibarən ilk faylımızı yaratmaqla onun ilk versiyasını yaratmış oluruq. Və hər dəfəsində onun üzərində dəyişiklik edib yaddaşda saxladıqca onun yeni versiyalarını yaradırıq. Bir fayl üzərində edilən dəyişikliklər yaddaşa qeyd edilmədiyi müddətdə həmin faylın yeni versiyası yaranmış sayılmır. Version control biz fayl üzərində hər dəyişiklik etdikdən sonra onun dəyişiklikdən əvvəlki halı və ya halları ilə sonrakı halı arasındakı fərqləri ortaya çıxarır. Git arxitekturasından istifadə edən bir neçə serverlər mövcuddur. [GitHub](https://github.com/), [GitLab](https://about.gitlab.com/), [Bitbucket](https://bitbucket.org/)-dir bunlardan biridir.

GitHub, əsasən Git üzərində qurulmuş və proqramçılar və yazılım inkişaf komandaları üçün məhsuldarlığı artıran bir platformadır. İşlənən layihələr üçün müxtəlif alətlər və xidmətlər təklif edir. Githubun əsas funksiyaları aşağıdakılardır:

1. Repositoriyalar - GitHub-da hər bir layihə repositoriya olaraq tanımlanır. Hər bir repositoriyada kod faylları, sənədlər, görüşlər və digər məlumatlar saxlanılır.

2. Versiya Nəzarəti - Git vasitəsilə GitHub, hər hansı bir faylın və ya proqram kodunun müxtəlif versiyalarını idarə edir. İstifadəçilər dəyişiklikləri izləyə bilər və lazımi hallarda əvvəlki versiyalara geri qayıda bilərlər.

3. Branchlar - Layihələrdə fərqli işləmə xətləri yaradıla bilər. Bu, yeni funksionallıqların inkişaf etdirilməsi, səhvlərin düzəldilməsi və ya fərqli məqsədlər üçün istifadə olunur. Branch-lar daha sonra pull request (PR) vasitəsilə əsas branch-a birləşdirilə bilər.

4. Pull Request və Sorğular (Issues) - Pull request, bir istifadəçinin dəyişikliklərini əsas branch-a təklif etmək üçün istifadə olunur. Sorğular (issues) isə səhvləri, təklifləri və ya layihə ilə bağlı digər mövzuları izah etmək və idarə etmək üçün istifadə olunur.

5. İcra və Tərcümələmə (Continuous Integration /Continuous Deployment) - GitHub Actions xidməti ilə icra proseslərini (CI/CD) avtomatlaşdırmaq mümkündür. Bu, kodun avtomatik sınaqdan keçirilməsi, yüklənməsi və ya digər avtomatlaşdırılmış tədbirlər üçün istifadə olunur.

6. İnkişafı Təminat - GitHub, açıq mənbəli inkişaf modelini dəstəkləyir və digər istifadəçilərlə əlaqə yaradır. İstifadəçilər layihələri izləyə, star qoymağa və ya bəyəndiklərini paylaşmağa imkan verir.

GitHub, proqramçılar üçün əsasən əməkdaşlığı və koordinasiyanı asanlaşdırmaq üçün dizayn edilmişdir. Bu platforma sahib olmaqla, istifadəçilər inkişaf layihələrini daha səmərəli idarə edə və işləmələrini tənzimləyə bilərlər.

## 2. Git-də repository nədir?

Git-də repository (repo), məlumatları, faylları, sürücü verilənlərini və tənzimləmə tarixlərini saxlamaq üçün istifadə olunan bir mənbə məlumatlar deposudur. Əsasən bir layihə, proqram, və ya digər məlumatları saxlamaq üçün istifadə olunur.

Git-də repository-lər, kodlamanın müəyyən bir vaxtdakı hər bir halını (commit) və müəyyən bir fayl vəziyyətini (snapshot) saxlayan, həmçinin müxtəlif versiyaları və branch-ları idarə edən üsuldur. Hər bir repository, özünəməxsus bir ad (adres) və bütün faylların və tarixçənin olduğu bir yaddaş aləti ilə təchiz olunmuşdur.

Repository-lər, bir proqramın mənbə məlumatlarının, konfiqurasiya fayllarının, əlavədə olaraq şəxsi məlumatlar və s. saxladığı məkanlar kimi düşünə bilərsiniz. Hər bir commit (dəyişikliklər paketi) repository-də yeni bir versiyaya əlavə olunur və bu dəyişiklikləri əvvəlki halından asılı olaraq baxmaq və müqayisə etmək mümkündür.

Git-də repository-lər əsasən iki növ olur: yerli (local) və uzaq (remote). Yerli repository, local kompüterinizdə saxlanılan və local fayllarınıza müraciət edən repository-dir. Uzaq repository isə əsasən bir mənbə təminat platforması (GitHub, GitLab, Bitbucket kimi) üzərində saxlanılan və daha geniş bir əlaqə və əməkdaşlıq imkanı təmin edən repository-dir.

Müstəqil bir repository, proqramçılar və inkişaf komandaları üçün əhəmiyyətli bir alətdir, çünki onlar kodun istənilən versiyasına asanlıqla keçid edə bilər, dəyişiklikləri nəzərdən keçirə bilər və fərqli inkişaf xətləri yarada bilərlər.

## 3. Git və Github fərqi

Git və GitHub iki fərqli mənada olan, lakin bir-birilə əlaqəli olan iki mənbə kod idarəetmə vasitəsidir.

**Git** açıq mənbəli bir versiya nəzarət sistemi (VCS)-dir. Bu, proqramçıların və yazılım inkişaf komandalarının kodlarını idarə etmək, dəyişiklikləri izləmək, fərqli versiyalar arasında keçid etmək və hər hansı bir zamanda əvvəlki bir halına qayıtmaq üçün istifadə olunan bir alətdir. Git, proqramçıların yerli kompüterlərində (local repository) işləyə biləcəyi bir mənbə kod idarəetmə sistemidir. Həmçinin, fərqli branch-ları (inkişaf xətləri) yaratmaq, fərqli inkişaf nöqtələri üzərində işləmək və mənbə kodunu digər proqramçılarla paylaşmaq imkanı verir.

Git-in əsas funksiyaları aşağıdakılardır:

* Kodun müxtəlif versiyalarını idarə etmək və tarixçəsini saxlamaq.
* Kompüterinizdə faylların və tarixçənin saxlanıldığı, yaradıldığı bir alət
* Mövcud olan repository-lərə kompüterinizdə işləmək üçün komanda sətrindən müraciət etməyə imkan verir.

**GitHub** isə Git-dən istifadə edən və git versiya nəzarəti olan bütün layihələrə müraciət edən bir platformadır. GitHub, git repository-lərini internetdə saxlayan bir hosting xidmətidir və müxtəlif əməkdaşlıq, inkişaf, və iş birliyi alətləri təklif edir. GitHub, proqramçılar və inkişaf komandaları üçün kodlarını saxlamaq, idarə etmək, digər istifadəçilərlə birləşdirmək, müzakirə etmək və s. üçün bir platforma tərəfdaşlıq edir.

GitHub-in əsas funksiyaları aşağıdakılardır:

* Git repository-lərinin internetdə saxlanıldığı və əlaqəli digər inkişafçılarla paylaşıldığı platformadır.
* Pull request (PR), issue tracker, code review və s. kimi əməkdaşlıq alətləri ilə inkişaf komandaları arasında əlaqəni və iş birliyini təmin edir.
* İndi istifadəyə verilən repository-lərə elektron poçt alətləri, web hosting, wiki və s. kimi əlavə xidmətlər təklif edir.

## 4. Git-də branch anlayışı

Branch layihələrdə fərqli inkişaf xətləri yaratmağa imkan verən və nəticədə fərqli fəaliyyət sahələrində və ya yeni funksional nəticələr əldə etmək üçün kodların müxtəlif versiyalarını idarə etməyə yarayan əsas alətdir.

Əsas konseptlər:

1. Branch (İnkişaf Xətti): Bir Git repository-də, branch adlanan fərqli işləmə xətləri yaratmaq mümkündür. Əsas branch əsasən main və ya master olur və əsas layihənin ən son versiyasıdır. İstifadəçilər bu əsas branch-dən ayrılaraq fərqli işləmə xətləri yarada, yeni funksionallıqlar əlavə edə və ya səhvləri düzəldə bilərlər.

2. Yeni Branch Yaratmaq: Yeni bir branch yaratmaq üçün git branch komandası istifadə olunur. Məsələn, git branch new-feature yazmaqla new-feature adlı yeni bir branch yaradıla bilər.

3. Aktiv Branch: Aktiv (çalışan) branch, proqramçının hazırda işlədiyi branch-dır. Aktiv branch-da əsasən dəyişikliklər edilir və commitlər əlavə olunur.

4. Branch-lər arasında keçid: git checkout komandası vasitəsilə fərqli branch-lər arasında keçid edilə bilər. Məsələn, git checkout new-feature yazaraq new-feature adlı branch-a keçid edilir.

5. Branch-ların birləşdirilməsi: İşin bitdikdən sonra, fərqli branch-lər əsas branch-ə birləşdirilə bilər. Bu əməliyyat üçün git merge komandası istifadə olunur.

Əlavə Məlumatlar

- Branch-lar və əməkdaşlıq: Bir layihədə, fərqli inkişafçılar fərqli işləmə xətləri yaradaraq eyni anda müxtəlif fəaliyyət sahələrində işləyə bilərlər. Bu, işin tənzimlənməsini və effektivliyini artırır.

- Temporary (müvəqqəti) Branch-lər: Qısa müddətli fəaliyyət üçün müvəqqəti branch-lər yaradıla bilər, məsələn, səhvlərin düzəldilməsi üçün, təcrübə işləmələri üçün və s.

Git-də branch anlayışı, iş birliyi və inkişafda böyük əhəmiyyətə malikdir və effektiv və tənzimlənmiş kod idarəçiliyini təmin edir.

## 5. Git-də commitin mahiyyəti

Git-də commit əməliyyatı, mənbə kod idarəetmə prosesində ən əsas və əhəmiyyətli addımlardan biridir. Commit, proqramçının yaratdığı dəyişiklikləri (kod dəyişiklikləri, fayl əlavələri, silmələr və s.) yaddaşda sabitləmək və tarixçələmək üçün istifadə olunur. Git-də commitlər, layihədə müəyyən bir vaxtda (müxtəlif fəaliyyət nöqtələri, məsələn, bir fikir əlavə olunduqda, bir səhv təhlil edildikdə və ya funksional təcrübələri aparmaq üçün) kodun hansısa bir versiyasının qeydini almaq üçün istifadə olunur.

Commit prosesi ümumiyyətlə aşağıdakı adımları əhatə edir:

1. Dəyişikliklərin seçilməsi: İstifadəçi, repository-dəki faylları dəyişdirir, əlavə edir və ya silir. Bu dəyişiklikləri commit etmək istəyirsə, əvvəlcə bu dəyişiklikləri seçməli və birlikdə commit edilməsi üçün hazırlamaq lazımdır.

2. Commit məlumatlarının daxil edilməsi: Hər commit, bir mesajla təsvir olunur. Bu mesaj, commit-də edilən dəyişikliklərin qısa bir izahını və ya nəticələrinin nə olduğunu açıqlayır. Məsələn, Bug fix for login page validation , Add new feature for user profile , kimi.

3. Commit etmək: Commit mesajı daxil edildikdən sonra, istifadəçi bu dəyişiklikləri yaddaşda sabitləmək üçün “git commit -m commit\_message” komandasını işə salır. Bu əməliyyat commit tarixini, faylların və dəyişikliklərin şəxsiyyətlərini (hash) yaradır və repository-dəki tarixçədə yerləşdirir.

Commit əhəmiyyəti:

- Versiya nəzarəti: Commitlər, repository-də bir kodun hansısa bir versiyasını təyin etmək və müəyyənləşdirmək üçün əsas alındıqda yararlıdır. Bu, proqramçının və ya inkişaf komandasının zamanla kodun dəyişmələrini nəzarət altında saxlamaq və proqramın hər hansı bir vaxtda əvvəlki bir vəziyyətinə qayıtmağını təmin edir.

- Tarixçələmə: Commitlər, repository-də bir tarixçə yaradır və hər bir commit, həmin tarixdə edilən dəyişiklikləri göstərir. Bu, proqramın inkişaf tarixini və proqramçıların hər bir dəyişikliyə dair izahatlarını müəyyən etməyə kömək edir.

- İş birliyi: Commitlər, inkişaf komandasının üzvləri arasında iş birliyini təmin etmək üçün əsasən əhəmiyyətlidir. Hər bir commit mesajı, başqalarının nəzərindən keçirilə bilər, müzakirə oluna bilər və təsdiqlənə bilər.

Git-də commitlər, effektiv və tənzimlənmiş bir kod idarəetmə prosesini təmin edir və proqramçıların layihədə hər hansı bir vaxtda müxtəlif mənbə kod versiyaları arasında asanlıqla keçid etməsinə imkan verir.

## 6. Git repository-ni local kompüterinizə necə klon edə bilərsiz?

Git repository-ni local kompüterinizə klonlamaq, repository-dəki bütün faylları və tarixçəni kompüterinizdə yerli bir kopyaya köçürmək üçün əsas metodlardan biridir. Onlar arasında əsas fərq var ki, SSH 1 kompüterdən bir neçə github hesabıyla işləməyə imkan verir. Amma HTTPS isə 1 github hesabıyla işləməyə imkan yaradır.İki əsas yol var:

1. HTTPS ilə Klonlama

1. Repository URL-nin əldə edilməsi: İnternetdəki Git hosting servisində (məsələn, GitHub, GitLab, Bitbucket) repository-nin HTTPS URL-ni əldə edin. URL quick setup səhifəsindən kopyalanır.

2. Git Bash və ya Terminalda git clone komandasının istifadəsi:

**git clone <repository\_URL>**

Burada <repository\_URL> yerinə əldə etdiyiniz HTTPS URL-ni əlavə edin. Məsələn: **git clone https://github.com/username/repository.git**

3. Klonlama prosesinin başlaması: Komandanı işə saldıqdan sonra Git, repository-dəki bütün faylları və tarixçəni kompüterinizə klonlamağa başlayacaq. Yükləmə başa çatdıqdan sonra, repository kompüterinizdəki bir qovluqda yerləşdiriləcək.

2. SSH ilə Klonlama:

1. SSH açarlarının qurulması: Əvvəlcə SSH açarını (SSH key pair) generasiya etməlisiniz.

**ssh -keygen -t rsa -b 4096 -C your\_email@example.com**

Githubda olan gmailinizlə eyni olmalıdır. Daha sonra sizə şifrə yerləşən faylın adresini təqdim edir.

2. Repository URL-nin əldə edilməsi: Repository-nin SSH URL-ni əldə edin. Bu, quick setub səhifəsindən əldə edilə bilər və SSH protokolü ilə başlayır (git@hosting\_service:user/repository.git).

3. Git Bash və ya Terminalda git clone komandasının istifadəsi:

**git clone <repository\_SSH\_URL>**

Məsələn:

**git clone git@github.com:username/repository.git**

4. SSH açarlarının təsdiq edilməsi: İlk dəfə klonlama zamanı SSH açarınızı təsdiq etməyə icazə veriləcək. Bu, əlavə daxili əməliyyatlar olmadan davam etmək üçün əsasən bir dəfəlik prosesdir.

## 7. git add . əmri nədir?

git add . komandası, Git repository-dəki bütün dəyişiklikləri stage (səhnə) mərhələsinə əlavə etmək üçün istifadə olunur. Bu komanda, işlənmiş və ya yeni yaradılmış bütün faylları, repository-də yerləşən bütün dəyişiklikləri və ya silmələri qeydiyyata almaq üçün istifadə olunur. İstifadəçi tərəfindən dəyişilmiş bütün fayllar, növbəti git commit komandasında təsdiqlənmək üçün hazır olacaqdır.

Ətraflı izah:

- . işarəsi: Bu, cari kataloq (current directory) və bütün alt qovluqlarda dəyişikliklərin əlavə olunması deməkdir. Git, cari iş qovluğunda və alt qovluqlarında olan bütün faylları və onların dəyişikliklərini əlavə edəcək.

- git add: Bu, Git-in dəyişiklikləri stage mərhələsinə (hazırlama mərhələsi) əlavə etmək üçün olan komandasıdır. Stage mərhələsi, növbəti commit (təsdiq) əməliyyatında yer alacaq dəyişiklikləri və faylları bildirir.

## 8. git commit -m əmri nədir?

git commit -m komandası, Git-də dəyişikliklərinizi təsdiqləmək üçün istifadə olunan və commit mesajı təyin etmək üçün istifadə olunan bir komandadır.

Ətraflı İzah:

- git commit : Bu komanda, stage mərhələsində olan bütün dəyişiklikləri yaddaşa təsdiq etmək üçün istifadə olunur. Stage mərhələsi, git add komandası ilə dəyişikliklərin hazırlanmasından sonra gəlir.

- -m seçimi: Bu, kommit mətni (commit message) əlavə etmək üçün təyin olunmuş bir seçimdir. -m seçimi ilə birlikdə gələn əlavə mətn, commit haqqında qısa bir açıqlama və ya mesajdır.

- Commit Mesajı: -m seçimi ilə birlikdə yazılan mətn, commit əməliyyatının məzmununu təyin edir. Bu mətn, commitin içindəki dəyişiklikləri və ya əlavə edilmiş funksionallıqları qısa şəkildə təsvir edir. Məsələn:

git commit -m “Fix issue with login validation”

Bu əmr, Fix issue with login validation commit mesajını təyin edərək, staging mərhələsində olan bütün dəyişiklikləri yaddaşa təsdiqləyəcəkdir.

## 9. git push əmri nədir?

git push əmri, yerli repository-də (local repository) edilən dəyişiklikləri uzaq repository-yə (remote repository) göndərmək üçün istifadə olunur. Bu əmrin əsas funksiyası, yerli repository-dəki commit-ləri uzaq repository-yə sinxronlaşdırmaqdır.

Ətraflı İzah:

- git push : Bu əmr, yerli branch-dakı commit-ləri uzaq repositoryyə göndərir. Əksər hallarda, yerli branch uzaq repositorydəki eyni adlı branch-a sinxronlaşdırılır.

- Uzaq repository: Git-də uzaq repository, layihənin mərkəzi və ya paylaşılan versiyasını saxlayan bir yerdir. Ən çox istifadə olunan uzaq repository xidmətləri GitHub, GitLab, Bitbucket və digərləridir.

git push Əmrinin İstifadəsi:

1.Əvvəlcə, layihədə dəyişikliklər edirik və onları commit edirik.

git add .

git commit -m Your commit message

2. Dəyişiklikləri uzaq repository-yə göndərmək üçün git push əmrini istifadə edin: git push

## 10. git pull əmri nədir?

git pull əmri, uzaq repository-dəki dəyişiklikləri yerli repository-yə (local repository) gətirmək və birləşdirmək (merge) üçün istifadə olunur. Bu əmr, git fetch və git merge əmrlərinin kombinasiyasıdır və uzaq repository-dən yenilikləri alıb yerli branch-ə tətbiq edir.

Ətraflı İzah:

- git pull : Bu əmr, uzaq repository-dəki dəyişiklikləri gətirir və yerli branch-da birləşdirir (merge edir). Əgər yerli branch-da dəyişikliklər varsa və uzaq branch-da da dəyişikliklər varsa, bu zaman birləşdirmə əməliyyatı yerinə yetirilir.

git pull Əmrinin İstifadəsi:

1. Uzaq repository-dəki dəyişiklikləri gətirmək və birləşdirmək:

git pull

Bu əmr, default (defolt) uzaq repository-dən və default branch-dən yenilikləri gətirir və cari yerli branch-də birləşdirir.

2. Müəyyən bir branch-ı uzaq repository-dən gətirmək və birləşdirmək:

Əgər müəyyən bir branch-dan yenilikləri gətirmək istəyirsinizsə, uzaq repository-nin adını və branch-ın adını göstərməlisiniz:

git pull origin branch-name

Burada, origin uzaq repository-nin adı, branch-name isə yenilikləri gətirmək istədiyiniz branch-in adıdır.

Əlavə Məlumatlar:

- Fetch və Merge əmrlərinin kombinasiyası: git pull əmrinin əslində git fetch və git merge əmrlərinin birləşməsi olduğunu bilmək əhəmiyyətlidir. git fetch uzaq repository-dən yenilikləri gətirir, amma avtomatik olaraq birləşdirmir, git merge isə bu yenilikləri cari yerli branch-də birləşdirir.

## 11. git status və git status -s fərqləri nədir?

git status və git status -s əmrləri, Git-də işləyən iş qovluğundakı və ya stage (hazırlıq) mərhələsindəki faylların vəziyyətini göstərmək üçün istifadə olunur. Onların fərqli təqdimat formatları var və fərqli detallı məlumatları təmin edirlər.

**git status** - Bu komanda, iş qovluğunuzun cari vəziyyətini ətraflı şəkildə göstərir. Bu, daha uzun və oxunaqlı bir formatda təqdim edilir və aşağıdakı məlumatları təmin edir:

- Hər hansı bir dəyişiklik edilmiş, əlavə olunmuş və ya silinmiş fayllar haqqında məlumat verir.

- Hər hansı bir commit ediləcək dəyişikliklər, stage mərhələsində olan və ya olmayan dəyişikliklər.

- Hər hansı bir izlənilməyən (untracked) fayllar.

- Hər hansı bir branch-də olduğunuz və uzaq repository ilə fərqlilikləri (divergence).

Nümunə:

$ git status

On branch main

Your branch is up to date with origin/main .

Changes to be committed:

(use git reset HEAD <file>... to unstage)

modified: file1.txt

new file: file2.txt

Changes not staged for commit:

(use git add <file>... to update what will be committed)

(use git restore <file>... to discard changes in working directory)

modified: file3.txt

Untracked files:

(use git add <file>... to include in what will be committed)

file4.txt

**git status -s** (və ya **git status –short** ) - Bu komanda, iş qovluğunuzun cari vəziyyətini daha qısa və sıxılmış formatda göstərir. Bu, daha qısa, sadə və daha sürətli bir nəzərdən keçirmə təmin edir. Aşağıdakı simvollar vasitəsilə fayl vəziyyətini bildirir:

- ?? : İzlənilməyən (untracked) fayllar.

- A : Yeni əlavə olunmuş (added) fayllar.

- M : Dəyişdirilmiş (modified) fayllar.

- D : Silinmiş (deleted) fayllar.

- R : Yenidən adlandırılmış (renamed) fayllar.

- C : Kopyalanmış (copied) fayllar.

Nümunə:

$ git status -s

M file3.txt

A file2.txt

?? file4.txt

Bu formatda, sol tərəfdəki hərf, stage mərhələsində olan dəyişiklikləri, sağ tərəfdəki hərf isə iş qovluğundakı dəyişiklikləri göstərir.

## 12. SSH və HTTPS vasitəsilə Git repository-nin klon edilməsi

Git repository-ni local kompüterinizə klonlamaq, repository-dəki bütün faylları və tarixçəni kompüterinizdə yerli bir kopyaya köçürmək üçün əsas metodlardan biridir. Onlar arasında əsas fərq var ki, SSH 1 kompüterdən bir neçə github hesabıyla işləməyə imkan verir. Amma HTTPS isə 1 github hesabıyla işləməyə imkan yaradır.İki əsas yol var:

1. HTTPS ilə Klonlama

1. Repository URL-nin əldə edilməsi: İnternetdəki Git hosting servisində (məsələn, GitHub, GitLab, Bitbucket) repository-nin HTTPS URL-ni əldə edin. URL quick setup səhifəsindən kopyalanır.

2. Git Bash və ya Terminalda git clone komandasının istifadəsi:

**git clone <repository\_URL>**

Burada <repository\_URL> yerinə əldə etdiyiniz HTTPS URL-ni əlavə edin. Məsələn: **git clone https://github.com/username/repository.git**

3. Klonlama prosesinin başlaması: Komandanı işə saldıqdan sonra Git, repository-dəki bütün faylları və tarixçəni kompüterinizə klonlamağa başlayacaq. Yükləmə başa çatdıqdan sonra, repository kompüterinizdəki bir qovluqda yerləşdiriləcək.

2. SSH ilə Klonlama:

1. SSH açarlarının qurulması: Əvvəlcə SSH açarını (SSH key pair) generasiya etməlisiniz.

**ssh-keygen -t rsa -b 4096 -C your\_email@example.com**

Githubda olan gmailinizlə eyni olmalıdır. Daha sonra sizə şifrə yerləşən faylın adresini təqdim edir.

2. Repository URL-nin əldə edilməsi: Repository-nin SSH URL-ni əldə edin. Bu, quick setub səhifəsindən əldə edilə bilər və SSH protokolü ilə başlayır (git@hosting\_service:user/repository.git).

3. Git Bash və ya Terminalda git clone komandasının istifadəsi:

**git clone <repository\_SSH\_URL>**

Məsələn:

**git clone git@github.com:username/repository.git**

4. SSH açarlarının təsdiq edilməsi: İlk dəfə klonlama zamanı SSH açarınızı təsdiq etməyə icazə veriləcək. Bu, əlavə daxili əməliyyatlar olmadan davam etmək üçün əsasən bir dəfəlik prosesdir.

## 13. İnformasiyanın qlobal şəbəkədə yerləşdirilməsi.

İnformasiyanın qlobal şəbəkədə (internetdə) yerləşdirilməsi, məlumatların internet vasitəsilə hər kəs üçün əlçatan olmasını təmin etmək prosesidir. Bu, müxtəlif yollarla həyata keçirilə bilər. Veb sayt yaradılması və yerləşdirilməsi - Veb sayt yaratmaq və onu internetdə yerləşdirmək, məlumatları qlobal şəbəkədə paylaşmaq üçün ən geniş yayılmış üsullardan biridir. Bunun üçün bir neçə addımlar var:

1. Veb saytın hazırlanması: HTML, CSS və JavaScript kimi texnologiyalar vasitəsilə veb saytınızı yaradın. Bu proses üçün müxtəlif veb dizayn və inkişaf platformalarından istifadə edə bilərsiniz (məsələn, WordPress, Wix, Squarespace və s.).

2. Domen adı alınması: Veb saytın internetdəki adıdır. İp adres yerinə domen istifadə edirik. Veb saytınız üçün unikal bir domen adı qeydiyyatdan keçirin. Məsələn, example.com . Domen adları müxtəlif registratorlar vasitəsilə alınır (GoDaddy, Namecheap və s.).

Domen adına nümunə: <https://naa.edu.az>

Bu domendə https - bizim protokolumuzdur. naa - aşağı səviyyəli domenimiz. edu - alt domendir, təhsil müəssisəsi olduğunu bildirir. az - yüksək səviyyəli domendir.

* org – qeyri kommersiya təşkilatı
* gov – dövlət təşkilatları
* net – şəbəkə agentlikləri
* int – beynəlxalq təşkilatlar

**Alt domenlər**

Coğrafi zonanı bildirir

* az –Azərbaycan
* tr – Türkiyə
* de – Almaniya
* uk – Böyük Britaniya

**Yüksək səviyyəli domen**

3. Veb hostinq xidmətinin seçilməsi: Veb saytınızı internetdə yerləşdirmək üçün hostinq xidməti seçirik. Bu xidmətlər veb saytınızı serverlərdə yerləşdirir və onu qlobal şəbəkədə əlçatan edir. Məsələn, Bluehost, HostGator, SiteGround və s.

Hosting alarkən müəyyən şeylərə diqqət yetirməliyik:

* Düzgün hosting şirkəti seçilməlidir
* Ehtiyaclar müəyyənləşdirilməlidir
* Database
* Disk Sahəsi
* Traffic
* Email sayı
* Büdcənizi müəyyənləşdirin, endirim olan zaman təklifləri diqqətlə incələyin
* Yedəkləmə

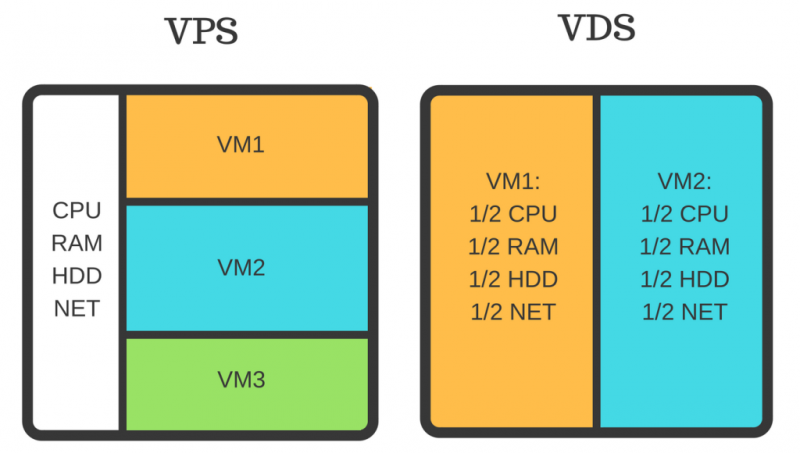
4. Serverlər: Veb Server (en. Web-server) – istifadəçilərin İnternetdəki Veb səhifələrə və digər verilənlərə daxil olmasına şərait yaradan xüsusi proqramlarla təmin olunmuş kompüterdir. İnformasiyaların saxlanılmasını, təşkilini və göndərilməsini təmin edir. Veb-brauzerdən qəbul etdiyi sorğu əsasında soruşulan sənədin elektron surətini istifadəçiyə göndərir. Belə sorğuların emal edilməsi və yerinə yetirilmə ardıcıllığı HTTP protokolu vasitəsilə yerinə yetirilir. İnternet üçün nəzərdə tutulmuş serverlərin tipləri:

* veb-serverlər
* elektron poçt serverləri
* FTP serverlər, fayllarla mübadilə üçün nəzərdə tutulmuşdur
* İnternet-telefoniyanın (IP telefoniya) işini təmin edən serverlər
* İnternet vasitəsilə radio və video translyasiyalar sistemi.

Veb serverlər müxtəlif olurlar. Bu muxtəliflik serverlərin işlədikləri əməliyyat sistemlərindəki fərqlərdən ibarətdir. Hal hazırda İnternet şəbəkələrində ən çox iki növ server proqramlarından istifadə olunur:

* Microsoft Internet Information Server (**IIS**) - Window NT/2000 sistemi əsasında işləyən.
* **Apache** – OC UNIX standartına uyğun platforma üçün nəzərdə tutulmuşdur.

Əgər veb sayt böyük bir proyetkdirsə bu zaman hosting xidməti deyil, virtual veb serverlər icarəyə götürülür.

 Virtual serverin iş prinsipi və xüsusiyyətləri: İş prinsipi və funksional xüsusiyyətlərinə görə virtual server fiziki serverdən çox az fərqlənir. O sayt, məlumat bazası, domen, ssh, ftp, e-poçt istifadəçilərinin yerləşdirməsilə bağlı məhdudiyyətə malik deyil. İnzibatçı bu parametrlər üzərində tam hüquqlara malikdir. İstənilən proqram təminatı real fiziki serverdə olduğu kimi quraşdırıla, dəyişdirilə və ya əvəz edilə bilər. Bizim serverlərdə yerləşən hər bir virtual server özünün IP-ünvanına malikdir və bu ünvan yalnız ayrılmış serverə məxsusdur.

Virtual serverin veb-hostinqlə müqayisədə üstünlükləri:

Tam idarəetmə: virtual serverin sahibləri root (UNIX üçün) və ya Administrator (Windows üçün) səviyyəsində hüquqlara malik olur və konkret layihə tələblərinə uyğun istənilən proqram təminatı və əməliyyat sistemini quraşdıra bilir. Serverin ayarlanması tam olaraq inzibatçıdan asılıdır.

Minimal məhdudiyyət: sayt, məlumat bazası, domen, ssh, ftp və e-poçt istifadəçiləri yalnız serverin öz resursları səviyyəsində məhdudiyyətə malikdir.

Müstəqillik: host-maşında yerləşən digər virtual serverlər - qonşular bir-birindən təcrid olunub və iş fəaliyyəti zamanı bir-birindən resurs çəkmir. Həmçinin veb-hostinqdən fərqli olaraq özünə məxsus IP-ünvanına malikdirlər.

Virtual serverin veb-hostinqlə müqayisədə mənfi cəhətləri:

Qiymət: virtual serverin qiyməti adətən veb-hostinqdən daha yüksək olur. Ancaq biz çalışırıq ki, bu fərqi getdikcə daha da minimuma endirək.

İnzibatçılığa ehtiyac: virtual serverin idarəetməsi fiziki serverin inzibatçılığı ilə eynidir. Buna görə istifadəçiyə server inzibatçılığı sahəsində müəyyən bilik və bacarıqlara malik olmaq lazımdır

## 14. Domain və Hosting anlayışı. FTP protokolu

Domain - internetdə bir veb saytın ünvanıdır. İstifadəçilər, domain adı vasitəsilə veb sayta daxil ola bilirlər. Məsələn, www.example.com bir domain adıdır.

- Domain adı strukturu: Domain adı, əsasən üç hissədən ibarətdir:

- Top-Level Domain (TLD): Ən yuxarı səviyyəli domain, domain adının son hissəsidir (məsələn, .com , .net ).

- Second-Level Domain (SLD): Bu, domain adının əsas hissəsidir və istifadəçi tərəfindən seçilir (məsələn, example ).

- Subdomain: Domain adının ən sol tərəfində yerləşən əlavə hissədir və əsas domain adının alt hissəsini təşkil edir (məsələn, www subdomain olaraq www.example.com ).

Hosting (veb hosting) - veb sayt fayllarını saxlayan və onları internet istifadəçilərinə əlçatan edən xidmətlərdir. Hosting provayderləri serverlər vasitəsilə bu xidməti təqdim edir.

- Hostinq növləri:

- Paylaşılan hostinq (Shared Hosting): Bir serverdə bir neçə veb saytın yerləşdirilməsi. İqtisadi və başlanğıc səviyyəli layihələr üçün uyğundur.

- Virtual Şəxsi Server (VPS Hosting): Paylaşılan hostinqə nisbətən daha çox mənbə təmin edən, lakin hələ də bir serveri paylaşan virtual serverlər.

- Fiziki Server (Dedicated Hosting): Tək bir istifadəçi üçün ayrılmış tam server. Yüksək performans və mənbələr tələb edən layihələr üçün nəzərdə tutulmuşdur.

- Bulud Hostinq (Cloud Hosting): Bir neçə serverin mənbələrini birləşdirərək təqdim edilən hostinq növü. Yüksək mövcudluq və elastiklik təmin edir.

FTP (File Transfer Protocol) - faylları bir kompüterdən digərinə ötürmək üçün istifadə olunan standart bir şəbəkə protokoludur. İnternet və ya yerli şəbəkələr vasitəsilə faylların ötürülməsi üçün geniş istifadə edilir.

FTP-nin Əsas Anlayışları:

- FTP Server: Faylların saxlandığı və istifadəçilərin bu fayllara daxil olub ötürə bildiyi server.

- FTP Klient: İstifadəçilərin FTP serverə qoşulmaq üçün istifadə etdiyi proqram. Məsələn, FileZilla, WinSCP və s.

FTP ilə Bağlantı Qurmaq:

FTP vasitəsilə bir serverə qoşulmaq üçün istifadəçi adı və parol tələb edilir. Aşağıda FTP bağlantısının nümunəsi verilmişdir:

1. FTP Klient proqramının açılması: Məsələn, FileZilla kimi bir FTP klienti istifadə edərək bağlantı qurmaq.

2. FTP Serverinə qoşulmaq:

- Host: FTP serverin ünvanı (məsələn, ftp.example.com ).

- İstifadəçi adı: FTP serverə daxil olmaq üçün istifadəçi adı.

- Parol: FTP serverə daxil olmaq üçün parol.

- Port: FTP üçün standart olaraq 21 nömrəli port istifadə edilir.

3. Faylların ötürülməsi: FTP bağlantısı qurulduqdan sonra faylları serverə yükləyə (upload) və ya serverdən endirə (download) bilərsiniz.

FTP-nin təhlükəsizlik çatışmazlıqlarına görə, Secure FTP (SFTP) və FTP over SSL/TLS (FTPS) kimi daha təhlükəsiz alternativlər də mövcuddur:

- SFTP (Secure File Transfer Protocol): SSH (Secure Shell) vasitəsilə faylları təhlükəsiz şəkildə ötürür.

- FTPS (FTP Secure): SSL/TLS vasitəsilə faylları təhlükəsiz şəkildə ötürür.

## 15. DNS və nameserver arasındakı fərqlər

DNS (Domain Name System) və nameserver terimləri internetdə domain adlarının və IP ünvanlarının idarə olunmasında əhəmiyyətli rollar oynayan mənbələrdir, lakin fərqlidirlər:

DNS (Domain Name System) - domain adlarının IP ünvanları ilə əlaqələndirilməsi və bu məlumatların paylaşılmasında istifadə olunan sistemdir.

- Əməliyyat: İnternetdə bir domain adı axtarışı başladıqda, DNS, həmin domain adının mənsub olduğu IP ünvanını tapır və geri qaytarır. Bu, domain adları ilə gecikməsiz işləməyə və mənbələrin sayt, e-poçt serverləri kimi xidmətlərinə çatdırılmasına imkan verir.

- DNS serverləri: DNS serverləri, bu məlumatları saxlayan və təmin edən serverlərdir. İnternetdə bir çox DNS server mövcuddur, həmçinin iş yerlərinin və internet servis təminatçılarının öz DNS serverləri də olur.

Nameserver - DNS sisteminin bir hissəsidir və məlumatları saxlayan və təmin edən DNS serverlərini göstərir. Bir domain adının hansı DNS serverlərinə (nameserverlərinə) nəzər salmalı olduğunu göstərir.

- Nameserverlərin konfiqurasiyası: Hər domain adının bir nameserver konfiqurasiyası var. Bu, domain adının məlumatlarının hansı DNS serverlərində saxlandığını göstərir.

- DNS serverlərinin mənsub olduğu nameserverlər: Hər bir DNS serveri (məsələn, “ns1.example.com”, “ns2.example.com”) nameserverlər olur. Bu serverlər, həmin domain adına mənsub olmuş IP ünvanlarını və digər DNS məlumatlarını saxlar.

Fərqlər

- Məqsəd: DNS, domain adları ilə IP ünvanları arasında əlaqələrin olmasına xidmət edir. Nameserver isə həmin DNS məlumatlarının idarə edilməsini təmin edir və domain adının hansı DNS serverlərinə yönəlməli olduğunu göstərir.

- İdarəetmə: DNS, geniş vəzifələrdə məlumatların idarə olunması üçün geniş istifadə olunur, ən çox domain adlarının IP ünvanlarına çevrilə bilməsinin sürətli və effektiv şəkildə həyata keçirilməsi məqsədilə. Nameserverlər isə konkret domain adlarına aid DNS məlumatlarının saxlanıldığı serverlərdir.

- Qoşulma: DNS, istifadəçilər və mənbələr arasında qoşulmaq üçün ümumi bir terminologiya kimi istifadə olunur. Nameserver isə konkret bir serverin funksional terminologiyasıdır, həmçinin bir domain adının konfiqurasiyasında göstərilən adıdır.

## 16. Hosting alarkən diqqət edilməli məqamlar

Hosting - internetdə informasiyanın serverdə fiziki yaddaşda yerləşdirilməsi üçün xidmətdir. Hosting xidməti seçərkən diqqət edilməli məqamlar var:

1. Düzgün Hosting Şirkəti Seçilməlidir: Tanınmış və etibarlı hosting şirkətləri seçilməlidir. Müştəri rəyləri və müstəqil təhlilləri araşdırmaq faydalı olar. 7/24 dəstək verən hosting şirkətləri üstünlük təşkil etməlidir. Məsələlər tez həll olunmalıdır. Şirkətin təklif etdiyi texniki imkanlar və xidmətlərin keyfiyyəti nəzərə alınmalıdır.

2. Ehtiyaclar Müəyyənləşdirilməlidir: Veb saytınızın növü (blog, e-ticarət, korporativ veb səhifə və s.) ehtiyaclarınızı təyin edəcək. Gözlənilən ziyarətçi sayı və trafik həcmi nəzərə alınmalıdır. Yüksək trafik olan saytlar üçün güclü hosting xidmətləri lazımdır.

3. Database (Verilənlər Bazası): MySQL, PostgreSQL, MariaDB və s. kimi hansı verilənlər bazası sistemini dəstəklədiyi nəzərə alınmalıdır. Verilənlər bazası məlumatlarının müntəzəm olaraq yedəklənməsi imkanı olmalıdır. Verilənlər bazasının performansı və sürəti yoxlanılmalıdır.

4. Disk Sahəsi: Veb saytınızın ölçüsünə və fayl ehtiyaclarına uyğun kifayət qədər disk sahəsi təmin edilməlidir. SSD disklər daha sürətli performans təmin edir. SSD hostinglərə üstünlük vermək lazımdır.

5. Traffic (Trafik): Məlumatların sürətli və fasiləsiz axınını təmin edən bant genişliyi tələb olunmalıdır. Aylıq trafik limiti və əlavə trafikin qiyməti haqqında məlumatlı olmaq lazımdır.

6. Email Sayı: Hosting paketində nə qədər e-poçt hesabının daxil olduğu yoxlanmalıdır. E-poçt xidmətlərinin keyfiyyəti və təhlükəsizlik tədbirləri nəzərə alınmalıdır.

7. Büdcənizi Müəyyənləşdirin: Hosting xidmətinin qiyməti büdcənizə uyğun olmalıdır. Uzunmüddətli müqavilələrdə endirimlər təklif olunur. Endirimlər və promo kodları nəzərə alaraq seçim etmək daha sərfəli ola bilər. Şərtlər və qaydalar diqqətlə oxunmalıdır.

8. Yedəkləmə: Məlumatların avtomatik yedəklənməsi imkanı olmalıdır. Yedəkləmələrin nə qədər tez-tez aparıldığı vacibdir (günlük, həftəlik və s.). Yedəkləmələrin nə qədər müddət saxlandığı və bərpa prosesi haqqında məlumatlı olmaq lazımdır.

## 17. Veb serverlər

Veb serverlər veb saytların istifadəçilərə təqdim olunmasında əsas rol oynayan texnologiyalardır. Onlar istifadəçilərin brauzerlərindən gələn tələbləri qəbul edir, bu tələbləri emal edir və müvafiq cavabları (məsələn, HTML səhifələr, şəkillər, videolar və s.) göndərir. Aşağıda veb serverlər və onların idarə olunması haqqında ətraflı məlumat verilmişdir:

**1. Populyar Veb Serverlər**

Apache HTTP Server: Dünyada ən çox istifadə olunan veb serverlərdən biridir. Açıq mənbəli (open-source) və geniş yayılmışdır. Modul əsaslı arxitektura sayəsində genişləndirilə biləndir. Çox sayda konfiqurasiya variantları və dəstəklədiyi müxtəlif platformalar (Linux, Windows, macOS) ilə geniş imkanlar təqdim edir.

Nginx: Yüksək performans və az resurs tələbi ilə tanınır. Yüklənmiş saytlar üçün statik faylları tez bir zamanda çatdırmaqda üstünlük təşkil edir. Load balancing (yük balansı) və reverse proxy imkanları təqdim edir. Veb server və eyni zamanda proxy server kimi istifadə edilə bilər.

Microsoft Internet Information Services (IIS): Windows əməliyyat sistemində işləyir. ASP.NET və digər Microsoft texnologiyaları ilə yaxşı inteqrasiya olunur. Güclü idarəetmə və təhlükəsizlik xüsusiyyətlərinə malikdir.

**2. Veb Serverlərin İdarə Edilməsi**

Konfiqurasiya:

* Apache: Konfiqurasiya faylı “httpd.conf” və ya “.htaccess” vasitəsilə tənzimlənir. Modullar vasitəsilə əlavə funksionallıq əlavə edilə bilər.
* Nginx: Konfiqurasiya faylı “nginx.conf” vasitəsilə tənzimlənir. Bloklar və direktivlər istifadə edilərək yüksək dərəcədə fərdiləşdirilə bilər.
* IIS: İdarəetmə Paneli (IIS Manager) vasitəsilə vizual olaraq konfiqurasiya edilir. XML əsaslı konfiqurasiya faylları istifadə olunur.

Təhlükəsizlik:

* SSL/TLS: Məlumatların təhlükəsiz ötürülməsi üçün SSL/TLS sertifikatları quraşdırılmalıdır.
* Firewall: Veb serverə girişləri nəzarət altına almaq üçün firewall qaydaları tənzimlənməlidir.
* Güncəlləmələr: Veb server proqramlarının müntəzəm olaraq güncəllənməsi təhlükəsizlik boşluqlarının qarşısını almaq üçün vacibdir.

Performans Optimizasiyası:

* Keşləmə (Caching): Müxtəlif növ keşləmə texnikaları (məsələn, opcache, content caching) istifadə edilərək performans artırıla bilər.
* Compression: Gzip və ya Brotli kimi kompresiya metodları ilə məlumatlar sıxışdıraraq istifadəçilərə daha sürətli təqdim edilə bilər.
* Load Balancing: Bir neçə server arasında yükün bölüşdürülməsi ilə performans və etibarlılıq artırıla bilər.

Yedəkləmə və Bərpa:

* Verilənlərin Yedəklənməsi: Məlumatların müntəzəm olaraq yedəklənməsi hər hansı bir məlumat itkisinə qarşı qorunmanı təmin edir.
* Bərpa Planı: Fövqəladə hallar üçün bərpa planı hazırlanmalı və test edilməlidir.

**3. Monitoring və Loglar**

Log Faylları:

* Access Logs: İstifadəçilərin müraciətləri və səhifə yükləmələri haqqında məlumatlar saxlanılır.
* Error Logs: Xəta və səhvlərin təfərrüatları qeyd edilir.

Monitoring Alətləri:

* Nagios, Zabbix: Sistem və şəbəkə resurslarının izlənməsi üçün istifadə edilə bilər.
* New Relic, Datadog: Performans monitoringi və analiz alətləri olaraq geniş imkanlar təqdim edir.

4. Skalabillik(scalability) və Redundans

Horizontal Skalabillik: Yüklənmiş saytlar üçün əlavə serverlər əlavə edərək trafik həcmi paylaşıla bilər.

Vertikal Skalabillik: Mövcud server resurslarının (CPU, RAM) artırılması ilə performans yaxşılaşdırıla bilər.

Redundans: Faylların və verilənlər bazasının ehtiyat nüsxələri müxtəlif serverlərdə saxlanılaraq yüksək əlçatanlıq təmin edilir.

## 18. Paylanıla bilən və virtual veb serverlər

Paylanıla bilən və virtual veb serverlər veb saytların və tətbiqlərin idarə olunmasında əhəmiyyətli rol oynayır. Onlar genişlənə bilən, etibarlı və səmərəli hosting mühiti təmin edirlər. Bu serverlər, xüsusilə yüksək trafik, kompleks tətbiqlər və resurs ehtiyacı olan veb saytlar üçün əlverişlidir.

**Paylanıla bilən veb serverlər** (Distributed Web Servers) sistem resurslarını müxtəlif serverlər arasında bölüşdürərək yükü balanslaşdırır və yüksək performans təmin edir.

Xüsusiyyətləri:

* + Yük Balanslama (Load Balancing): Trafik həcmi bir neçə server arasında paylanır. Bu, performansı artırır və tıxacların qarşısını alır.
  + Yüksək Əlçatanlıq (High Availability): Bir serverdə problem yaranarsa, digər serverlər avtomatik olaraq işə düşür, beləliklə, xidmət fasiləsiz davam edir.
  + Miqyaslanabilmə (Scalability): Trafik və resurs tələblərinə uyğun olaraq serverlər asanlıqla əlavə edilə və ya çıxarıla bilər.

İstifadə Halları:

* + Böyük Veb Saytlar və Tətbiqlər: Yüksək trafik və mürəkkəb əməliyyatlara malik veb saytlar üçün idealdır.
  + E-ticarət Saytları: Tez-tez dəyişən məhsul məlumatları və yüksək ziyarətçi sayı olan saytlar üçün uyğun gəlir.
  + Media və Yayım Xidmətləri: Böyük miqdarda media fayllarının paylanmasını təmin edir.

**Virtual veb serverlər** (Virtual Web Servers) - fiziki serverin virtualizasiyası ilə yaradılan mühitlərdir. Hər bir virtual server müstəqil əməliyyat sisteminə malikdir və resursları (CPU, RAM, disk sahəsi və s.) bölüşür.

Xüsusiyyətləri:

* + Virtualizasiya Texnologiyası: VMware, Hyper-V, KVM kimi texnologiyalarla fiziki server üzərində bir neçə virtual server yaradılır.
  + İzolyasiya: Hər virtual server digərlərindən müstəqil işləyir və bir serverdə problem yaranarsa, digərlərinə təsir etmir.
  + Miqyaslanabilmə (Scalability): Resurslar dinamik olaraq artırıla və ya azaldıla bilər.
  + Çeviklik: İstifadəçilər öz virtual serverlərini ehtiyaclarına uyğun olaraq idarə edə bilərlər.

İstifadə Halları:

* + Kiçik və Orta Ölçülü Veb Saytlar: Resursları daha səmərəli istifadə etməyə imkan verir.
  + Test və İnkişaf Mühitləri: Yeni tətbiqlərin və ya yeniləmələrin sınaqdan keçirilməsi üçün mühitlər yaradır.
  + Bulud Xidmətləri: AWS, Azure, Google Cloud kimi bulud platformaları virtual serverləri istifadə edir.

Paylanıla bilən və virtual veb serverlər veb saytların və tətbiqlərin idarə edilməsində müasir və effektiv həllər təqdim edir. Onların seçimi və konfiqurasiyası, ehtiyaclarınıza uyğun olaraq optimallaşdırılmalıdır.

## 19. DNS necə işləyir?

DNS (Domain Name System) internetin işləməsi üçün vacib olan bir sistemdir. O, domen adlarını (məsələn, example.com) IP ünvanlarına (məsələn, 192.0.2.1) çevirir. Bu çevirmə sayəsində istifadəçilər asanlıqla yadda saxlanıla bilən domen adlarından istifadə edərək internetdəki resurslara (veb saytlar, e-poçt serverləri və s.) daxil ola bilirlər. DNS necə işlədiyini daha ətraflı izah edək:

1. Domen Adı Sorğusu - İstifadəçi veb brauzerdə bir domen adı yazır (məsələn, www.example.com). Brauzer həmin domenin IP ünvanını tapmaq üçün DNS sorğusu göndərir.

2. Yerli DNS Keşi - Brauzer və ya əməliyyat sistemi əvvəlcə yerli keşi yoxlayır. Əgər axtarılan domen üçün müvafiq IP ünvan keşdə varsa, DNS sorğusu burada yekunlaşır və cavab brauzerə qaytarılır.

3. Rekursiv DNS Serveri - Əgər yerli keşdə müvafiq cavab yoxdursa, DNS sorğusu internet provayderinin (ISP) rekursiv DNS serverinə yönləndirilir. Bu server istifadəçinin əvəzinə digər DNS serverlərinə sorğular göndərərək düzgün IP ünvanını tapmağa çalışır.

4. Root DNS Serverləri - Rekursiv DNS serveri əvvəlcə root DNS serverlərinə sorğu göndərir. Root serverlər domen adının ilk hissəsinə (məsələn, .com, .net) əsasən sorğunu müvafiq top-level domain (TLD) serverlərinə yönləndirir.

5. TLD DNS Serverləri - TLD serverləri, müəyyən TLD (məsələn, .com) üçün məsuliyyət daşıyan serverlərdir. Bu serverlər domen adının sonrakı hissəsinə (məsələn, example.com) əsasən sorğunu həmin domenin avtoritativ DNS serverlərinə yönləndirir.

6. Avtoritativ(səlahiyyətli) DNS Serverlər - Avtoritativ DNS serverlər müəyyən domen üçün məlumatları saxlayır. Bu serverlər sorğunu alır və müvafiq domenin IP ünvanını rekursiv DNS serverinə qaytarır.

7. IP Ünvanın Cavabı - Rekursiv DNS serveri avtoritativ DNS serverindən aldığı IP ünvanını istifadəçinin brauzerinə qaytarır. Bu məlumat yerli keşdə saxlanılır ki, növbəti dəfə eyni sorğu daha tez cavablandırılsın.

8. Brauzerin Tələb Göndərməsi - Brauzer aldığı IP ünvanını istifadə edərək müvafiq veb serverə HTTP və ya HTTPS tələbi göndərir. Veb server həmin tələbi qəbul edir və istifadəçinin brauzerinə uyğun cavab qaytarır (məsələn, veb səhifə).

DNS Serverlərin Tipləri

* + Root DNS Serverlər: DNS sisteminin ən yüksək səviyyəsində yerləşirlər və TLD serverlərinə yönləndirmə təmin edirlər.
  + Top-Level Domain (TLD) Serverlər: Müəyyən TLD-lər üçün məlumat saxlayırlar (məsələn, .com, .net, .org).
  + Avtoritativ DNS Serverlər: Müəyyən bir domen üçün orginal DNS məlumatlarını saxlayırlar və sorğulara cavab verirlər.
  + Rekursiv DNS Serverlər: İstifadəçilərin sorğularını alır və digər DNS serverlərinə müraciət edərək düzgün IP ünvanını tapır.

DNS Sorğularının Növləri

* A (Address) Rekordu: Domen adını IPv4 ünvanına çevirir.
* AAAA (Quad A) Rekordu: Domen adını IPv6 ünvanına çevirir.
* CNAME (Canonical Name) Rekordu: Domen adını digər bir domen adı ilə dəyişdirir.
* MX (Mail Exchange) Rekordu: E-poçt serverlərinin IP ünvanlarını saxlayır.
* NS (Name Server) Rekordu: Domen üçün avtoritativ name serverlərin adlarını göstərir.
* TXT (Text) Rekordu: Domen haqqında əlavə mətn məlumatları saxlayır.

## 20. Html strukturu, “Tag” anlayışı, proqram kodunun düzgün tərtib edilməsi mərhələləri

HTML (HyperText Markup Language) veb səhifələrin qurulmasında istifadə olunan əsas dilidir. HTML sənədi veb səhifənin strukturunu təyin edir və brauzerlərə həmin səhifəni necə göstərməli olduqlarını bildirir.

HTML sənədinin əsas strukturu aşağıdakı elementlərdən ibarətdir:

<!DOCTYPE html>

<html>

<head>

<title>Veb Səhifə Başlığı</title>

<meta charset=“UTF-8”>

<!-- Stil, meta məlumatlar və digər başlıq məlumatları buraya daxil edilir -->

</head>

<body>

<!-- Səhifənin məzmunu buraya daxil edilir -->

</body>

</html>

Açıklama:

* + “<!DOCTYPE html>“: Bu deklarasiya sənədin HTML5 olduğunu bildirir.
  + “<html>“: HTML sənədinin kök elementi. Bütün digər elementlər bu elementin içində yerləşir.
  + “<head>“: Meta məlumatlar, stil cədvəlləri və ssenarilər (scripts) bu bölmədə yerləşir.
  + “<title>“: Səhifənin başlığı. Brauzerin başlıq çubuğunda göstərilir.
  + “<meta charset=“UTF-8”>“: Sənədin kodlaşdırmasını təyin edir. UTF-8 ən geniş istifadə olunan kodlaşdırma formatıdır.
  + “<body>“: Veb səhifənin məzmunu (mətn, şəkillər, linklər və s.) bu bölmədə yerləşir.

HTML “tag” (etiket) elementlərin başlayıb bitdiyini göstərir. Hər bir HTML etiketi “<etiket\_adı>“ və “</etiket\_adı>“ şəklində yazılır.

Tagların Tipləri

1. Blok Etiketlər (Block Tags): Məzmunu yeni sətirdən başlayır və tam genişliyi əhatə edir. Məsələn: “<div>“, “<p>“, “<h1>“, “<ul>“.

2. Inline Etiketlər (Inline Tags): Məzmunu sətir daxilində yerləşir. Məsələn: “<span>“, “<a>“, “<img>“, “<strong>“.

3. Öz-özünə bağlanan Etiketlər (Self-closing Tags): Qapanma etiketi yoxdur, yalnız açılış etiketi var. Məsələn: “<img>“, “<br>“, “<hr>“, “<input>“.

Proqram Kodunun Düzgün Tərtib Edilməsi Mərhələləri

1. Planlama: Veb səhifənin məqsədi, hədəf auditoriyası və funksionallığı haqqında məlumatların toplanması. Səhifənin strukturunun və naviqasiyasının planlaşdırılması.

2. HTML Strukturunun Yaradılması: HTML sənədinin əsas strukturu və bölmələri yaradılır. Məzmun və məlumatlar uyğun HTML elementləri ilə işarələnir.

3. Stil və Dizayn: Veb səhifənin görünüşü və dizaynı üçün CSS (Cascading Style Sheets) istifadə edilir. Müxtəlif cihazlara uyğun olaraq dizayn edilir (məsələn, mobil və masaüstü).

4. İnteraktivlik: Veb səhifəyə dinamik və interaktiv funksiyalar əlavə etmək üçün JavaScript istifadə edilir. jQuery, React, Vue.js kimi kitabxana və çərçivələrdən istifadə edilə bilər.

5. Test və Debug: HTML, CSS və JavaScript kodlarının səhvlərini yoxlamaq və düzəltmək. Veb səhifənin müxtəlif brauzerlərdə düzgün işlədiyini yoxlamaq.

6. Nəzarət və Baxım: Veb səhifənin məzmunu və funksionallığı müntəzəm olaraq yenilənir.

## 21. HTML-də teq və atribut anlayışı

HTML (HyperText Markup Language) veb səhifələrin qurulması üçün istifadə olunan dildir. HTML-də elementlər (taglar) və atributlar səhifənin strukturu və məzmununu təyin etmək üçün istifadə olunur.

HTML-də elementlər “<tag>“ və “</tag>“ açılış və bağlanma etiketləri ilə təyin olunur. Hər bir tag vəzifəsinə uyğun olaraq məzmunu işarələyir. Məsələn:

* + “<p>“: Paraqrafı təyin edir.
  + “<h1>“, “<h2>“, “<h3>“: Başlıqları təyin edir.
  + “<a>“: Linkləri təyin edir.
  + “<img>“: Şəkilləri təyin edir.

Bəzi taglar öz-özünə bağlanan taglardır və yalnız açılış tagi istifadə olunur. Məsələn:

- “<img>“: Şəkil əlavə etmək üçün.

- “<br>“: Yeni sətir əlavə etmək üçün.

Tag-lərin içində, ya da açılış tag-ində, bir və ya daha çox atribut əlavə edilə bilər. Atributlar elementin davranışını, görünüşünü və ya başqa xüsusiyyətlərini təyin edir. Məsələn, “<a>“ (link) elementi üçün “href” atributu linkin mənbə ünvanını təyin edir:

<a href="https://www.example.com">Example Website</a>

Burada “href” atributu “https://www.example.com” ünvanına sahib olan linki göstərir.

“<img>“ elementində “src” atributu şəkilin mənbəsini təyin edir:

<img src="image.jpg" alt="Şəkilin Təsviri">

“<a>“ elementində “target” atributu linkin açılacağı pəncərənin tipini təyin edir

<a href="https://www.example.com" target="\_blank">Example Website</a>

## 22. HTML-də “boş element” anlayışı

HTML-də "boş element" anlayışı, yalnız açılış etiketi olan elementlərdir. Bu elementlər, öz-özünə bağlanan elementlər kimi adlandırılabilir, çünki onların qapanış etiketi yoxdur. Ən çox müşahidə olunan boş elementlərdən biri “<img>“ elementidir. Bu element şəkilləri və ya digər multimedia obyektləri səhifəyə əlavə etmək üçün istifadə olunur. “<img>“ elementi tək başına açılış etiketi ilə istifadə olunur və heç bir iç məzmunu yoxdur. Məsələn:

<img src="image.jpg" alt="Şəkilin Təsviri">

Burada “src” atributu şəkilin mənbəsini (“image.jpg”) təyin edir və “alt” atributu şəklin təsvirini təyin edir.

Başqa bir nümunə “<br>“ elementidir ki, bu element yeni sətir əlavə etmək üçün istifadə olunur. “<br>“ elementi də yalnız açılış etiketi ilə istifadə olunur:

Text1<br>Text2

Bu nümunədə, “Text1” və “Text2” mətni fərqli sətirlərdə göstərilir.

Digər boş elementlərə misal olaraq “<input>“, “<hr>“, “<meta>“, “<link>“ kimi elementləri də göstərmək mümkündür. Bu elementlər də yalnız açılış etiketi ilə istifadə olunur və içində heç bir məzmun yoxdur, yalnız atributları ilə funksionallığı təyin edirlər.

<input> elementi istifadəçidən məlumat daxil etməsini tələb edir və mətn, rəqəm, şifrə və ya fayl yükləmək üçün istifadə olunur.

<hr> elementi yatay xətt təyin edir. HTML-də səhifədən məzmun hissələrini ayıran görsənir.

<meta> elementi HTML səhifəsinin məlumatlarını təyin edir. Əsasən <head> bölməsində istifadə olunur və səhifə haqqında məlumatları (məsələn, səhifə dilini, mətn kodlaşdırmasını və s.) təyin etmək üçün istifadə olunur.

<link> elementi səhifəyə başqa sənədlərin (məsələn, CSS cədvəlləri) və ya resursların (məsələn, şriftlər) bağlantılarını əlavə etmək üçün istifadə olunur. Əsasən <head> bölməsində istifadə olunur.

## 23. HTML elementlərindən “id” atributu ilə “class” atributu arasında fərq nədir?

HTML elementlərində "id" atributu ilə "class" atributu arasında əsas fərq məhz unikal olmaları və təkrarlanmaq imkanıdır:

**“id” Atributu:**

* Unikal olmalıdır: Bir HTML sənədi içində hər bir “id” atributu yalnız tək bir elementə təyin edilə bilər. Başqa sözlə, bir “id” atributu təyin edilmiş elementin sayı qlobal olaraq bir səhifədə yalnız bir dəfə istifadə olunmalıdır.
* Stil və işləmələrdə istifadə edilir: CSS (Cascading Style Sheets) və JavaScript koda elementə müraciət etmək üçün təyin edilir. Məsələn, “ header { ... }” CSS quraları ilə “id="header"“ atributu olan elementə təyin edilə bilər və JavaScriptdə “document.getElementById(“header”)” funksiyası ilə elementə müraciət edilə bilər.

**“class” Atributu:**

* Birdən çox elementdə istifadə edilə bilər: Bir “class” atributu eyni və ya müxtəlif elementlərdə birdən çox elementə təyin edilə bilər. Bu, eyni stil tələblərinə uyğun olan bəzi və ya bütün elementlər üçün tətbiq edilən məqamlarda istifadə olunur.
* Stil və işləmələrdə istifadə edilir: CSS (Cascading Style Sheets) ilə elementlərin görünüşünü təyin etmək üçün istifadə olunur. Məsələn, “.menu { ... }” CSS quraları ilə “.menu” klassı olan bütün elementlərə tətbiq edilə bilər.

## 24. HTML teqləri vasitəsilə cədvəllərin tərtib edilməsi.

HTML-də “<table>“, “<tr>“, “<th>“, “<td>“ kimi teqlər vasitəsilə cədvəllər (tables) tərtib edilir. Bu teqlər, məlumatların qruplaşdırılması üçün əlverişli və mümkün olduğu halda strukturallaşdırılmasına imkan verir. Əsas Cədvəl Teqləri:

“<table>” teqi bir cədvəl elementini təyin edir. İçindəki məzmun cədvəl formatında yerləşir.

<table>

<!-- table rows (tr) and cells (td or th) will go here -->

</table>

“<tr>“ teqi bir cədvəl sətrini (row) təyin edir. İçindəki məzmun sətrin daxilindəki hüceyrələrə yerləşir.

<tr>

<!-- table header cells (th) or table data cells (td) will go here -->

</tr>

“<th>“ teqi bir cədvəl başlığını (table header) təyin edir. İstifadəçiyə cədvəl sətrinin başlığını göstərmək üçün istifadə olunur.

<th>Header Mətni</th>

“<td>“ teqi bir cədvəl datalarını (table data) təyin edir. İstifadəçiyə cədvəl sətrindəki məlumatları göstərmək üçün istifadə olunur.

<td>Data Mətni</td>

Aşağıda bir HTML cədvəlinin sadə nümunəsini görə bilərsiniz:

<table border="1">

<tr>

<th>Ad</th>

<th>Soyad</th>

<th>Yaş</th>

</tr>

<tr>

<td>Əli</td>

<td>Məmmədov</td>

<td>25</td>

</tr>

<tr>

<td>Məryəm</td>

<td>İbrahimova</td>

<td>30</td>

</tr>

</table>

Bu nümunədə, bir cədvəl yaradılıb, “border="1"“ atributu ilə kənarların göstərilməsi təyin edilir. İlk “<tr>“ teqi başlıqları (“<th>“) təyin edir, altındakı “<tr>“ teqləri (“<td>“) isə məlumatları təyin edir.

## 25. HTML cədvəlində iki və ya daha çox sətir və ya sütunu bir sətir və ya sütuna necə birləşdirə bilərik?

HTML cədvəllərində iki və ya daha çox sətir və ya sütunu birləşdirmək üçün “rowspan” və “colspan” atributları istifadə olunur. Bu atributlar “<td>“ və “<th>“ elementlərinə təyin olunur və onların üstündə hansı sıra və ya sütunların birləşdirilməsini təyin edir.

“rowspan” Atributu - bir “<td>“ və ya “<th>“ elementini birdən çox sətirə birləşdirmək üçün istifadə olunur. Bu atribut, hansı sətrə birləşdirmək istədiyimizi göstərir.

<table border="1">

<tr>

<th>Ad</th>

<th>Soyad</th>

<th>Yaş</th>

</tr>

<tr>

<td>Əli</td>

<td rowspan="2">Məmmədov</td>

<td>25</td>

</tr>

<tr>

<td>Məryəm</td>

<td>30</td>

</tr>

</table>

Bu nümunədə, "Məmmədov" soyadı ikinci sətrdə iki sətri birləşdirir. Yani, bu soyad iki sətrdə görünür.

“colspan” Atributu - isə bir “<td>“ və ya “<th>“ elementini birdən çox sütuna birləşdirmək üçün istifadə olunur.

<table border="1">

<tr>

<th colspan="2">İstifadəçi Məlumatları</th>

</tr>

<tr>

<td>Ad</td>

<td>Soyad</td>

</tr>

<tr>

<td>Əli</td>

<td>Məmmədov</td>

</tr>

<tr>

<td colspan="2">Yaş: 25</td>

</tr>

</table>

Bu nümunədə, "İstifadəçi Məlumatları" başlığı iki sütunu birləşdirir. Həmçinin "Yaş: 25" məlumatı da iki sütunu birləşdirir.

## 26. HTML-də link necə təyin edilir və “target” atributu nə üçündür?

HTML-də link (hərəkət) təyin etmək üçün “<a>“ (anchor) elementi istifadə olunur. Bu element “href” (hypertext reference) atributu ilə linkin ünvanını təyin edir. Həmçinin “target” atributu, linkin necə açılacağını təyin etmək üçündür.

“**<a>**“ elementi, hərəkət (link) təyin etmək üçün istifadə olunur. “href” atributu ilə hərəkətin ünvanı təyin edilir. Məsələn:

<a href="https://www.example.com">Example Website</a>

Bu nümunədə, “https://www.example.com” ünvanına sahib olan "Example Website" adlı link təyin olunur.

“**target**” atributu, linkin necə açılacağını göstərir (defolt olaraq self-dir). İstifadə olunan dəyərlər aşağıdakılardır:

* “**\_self**”: Hərəkət, həmin səhifədə yüklənir. Bu, əsas olaraq, linki açdığınız səhifədə yeni məzmunu göstərmək üçün istifadə olunur.
* “**\_blank**”: Hərəkət, yeni bir brauzer pəncərəsində (və ya yeni bir tabda) yüklənir. Bu, istifadəçini əsas səhifədən uzaqda başqa məzmunu baxmağa imkan verir.
* “**\_parent**”: Hərəkət, əsas səhifəni aşıb, üst kateqoriyadakı səhifədə yüklənir.
* “**\_top**”: Hərəkət əsas pəncərədə yüklənir.

## 27. Nizamlı və nizamsız HTML siyahıları

HTML-də nizamlı (ordered) və nizamsız (unordered) siyahılar yaratmaq üçün “<ol>“ və “<ul>“ elementləri istifadə olunur. Hər iki elementin içində “<li>“ (list item) elementləri yerləşdirilir. Məqsədə uyğun olaraq, nizamlı siyahılar rəqəmlərlə və ya hərflərlə, nizamsız siyahılar isə simvol, məznə və ya başqa ikonlarla təqdim edilə bilər.

Nizamlı (Ordered) Siyahılar - əsasən sıralı məlumatların göstərilməsi üçün istifadə olunur. İki növü mövcuddur: rəqəmlərlə və hərflərlə göstərilən siyahılar.

Rəqəmlərlə Nizamlı Siyahılar (“<ol type="1">“)

<ol>

<li>İlk maddə</li>

<li>İkinci maddə</li>

<li>Üçüncü maddə</li>

</ol>

Bu nümunədə, siyahıda standart rəqəmlərlə maddələr göstərilir.

Hərflərlə Nizamlı Siyahılar (“<ol type="A">“)

<ol type="A">

<li>İlk maddə</li>

<li>İkinci maddə</li>

<li>Üçüncü maddə</li>

</ol>

Bu nümunədə isə siyahıda hərflərlə maddələr göstərilir.

Nizamsız (Unordered) Siyahılar - siyahılanmış məlumatların sırasız şəkildə göstərilməsi üçün istifadə olunur. İkonlar və ya başqa simvollar istifadə olunur.

Nizamsız Siyahılar (“<ul>“)

<ul>

<li>İlk maddə</li>

<li>İkinci maddə</li>

<li>Üçüncü maddə</li>

</ul>

İkon ilə Nizamsız Siyahılar (“<ul>“)

<ul>

<li><i class="fa fa-check"></i> İlk maddə</li>

<li><i class="fa fa-check"></i> İkinci maddə</li>

<li><i class="fa fa-check"></i> Üçüncü maddə</li>

</ul>

Bu nümunədə, Font Awesome kimi ikon fontları istifadə edilərək maddələr ikonlar ilə göstərilir.

style="list-style-type:circle" və style="list-style-type:disc" ilə içi boş və dolu disk olaraq işarə qoymaq olar.

## 28. HTML formaları və forma elementləri

HTML formaları, istifadəçilərin məlumat göndərmək üçün internetdə məlumatların toplanması və istifadə edilməsi üçün əlverişli bir mekanizmadır. HTML formaları, bir veb səhifəsində müxtəlif form elementləri (input sahələri, seçimlər, düymələr və s.) ilə təşkil olunur.

HTML Forması “<form>“ - Bu element, formanın bütün məzmununu əhatə edir və formanın yollanması zamanı necə işləndiyini təyin edir.

<form action="/submit\_form.php" method="post">

<!-- Form elementləri burada yerləşdirilir -->

</form>

* + “action” atributu, formanın məlumatlarının yollanacağı ünvanı göstərir.
  + “method” atributu, HTTP protokolunun istifadə olunacağı formanın yollanma metodu təyin edir. Ən çox istifadə olunanlar “post” və “get” metodlarıdır.

Form Elementləri - formanın içində yerləşdirilən əsas elementlər form məlumatlarının toplanması üçün istifadə olunur. Aşağıda ən çox istifadə olunan form elementlərinin nümunələri verilmişdir:

**Input Sahələri:**

* Mətn sahəsi (“<input type="text">“): İstifadəçidən mətn daxil etməsi üçün.

<input type="text" name="username" placeholder="İstifadəçi adı">

* Parol sahəsi (“<input type="password">“): Gizli mətnlər üçün.

<input type="password" name="password" placeholder="Parol">

* + Email sahəsi (“<input type="email">“): E-poçt ünvanı daxil etmək üçün.

<input type="email" name="email" placeholder="E-poçt ünvanı">

* + Nömrə sahəsi (“<input type="number">“): Yalnız rəqəmlər daxil etmək üçün.

<input type="number" name="quantity" min="1" max="10">

**Seçim Elementləri:**

* + Seçim çərçivəsi (“<select>“): İstifadəçinin bir və ya bir neçə seçim etməsinə imkan verir.

<select name="country">

<option value="az">Azərbaycan</option>

<option value="tr">Türkiyə</option>

<option value="ru">Rusiya</option>

</select>

* + - * Radio düymələri (“<input type="radio">“): Yalnız bir seçim etmək üçün.

<input type="radio" name="gender" value="male"> Kişi

<input type="radio" name="gender" value="female"> Qadın

* Checkbox (“<input type="checkbox">“): Birdən çox seçim etmək üçün.

<input type="checkbox" name="interests" value="football"> Futbol

<input type="checkbox" name="interests" value="basketball"> Basketbol

**Düymələr:**

* + - * Göndərmə düyməsi (“<input type="submit">“): Formanın məlumatlarını yolla.

<input type="submit" value="Göndər">

* + - * Sıfırla düyməsi (“<input type="reset">“): Formdakı məlumatları sıfırla.

<input type="reset" value="Sıfırla">

## 29. HTML formasının vacib atributları hansılardır?

HTML formasının vacib atributları aşağıdakılardır:

**1. action**: Formanın məlumatlarının yollanacağı ünvanı göstərir. Bu atribut forma təyin olunmalıdır.

<form action="/submit\_form.php" method="post">

<!-- Form elementləri burada yerləşdirilir -->

</form>

**2. method**: Formanın məlumatlarının yollanma metodunu təyin edir. Ən çox istifadə olunan dəyərlər “get” və “post” metodlarıdır.

* “get”: bu metod sorğuların parametri brauzerin axtarış hissəsində görünərək göndərir.
* “post”: bu metodsa, HTTP sorğuların gövdəsində gizlənir və daha çox məlumat əlavə edilə bilər.

<form action="/submit\_form.php" method="post">

<!-- Form elementləri burada yerləşdirilir -->

</form>

**3. name**: Form elementlərinin adını təyin edir.

<input type="text" name="username" placeholder="İstifadəçi adı">

**4. id**: Form elementinə unikal təyin edilmiş bir id verir. Bu, CSS və JavaScript kimi digər texnologiyalar ilə form elementlərinə istənilən zaman məlumat əlavə etmək və ya tənzimləmək üçün vacibdir.

<input type="text" id="username" name="username" placeholder="İstifadəçi adı">

**5. type**: “<input>“ elementləri üçün inputun növünü təyin edir. Məsələn, “text”, “password”, “checkbox”, “radio”, “submit” kimi dəyərlər mövcuddur.

<input type="text" name="username" placeholder="İstifadəçi adı">

## 30. HTML formasında input elementinin vacib atributları hansılardır?

HTML “<input>“ elementinin əsas və vacib atributları aşağıdakılardır:

**1. type**: inputun növünü təyin edir. Məsələn, “text”, “password”, “checkbox”, “radio”, “email”, “number” kimi dəyərlər mövcuddur. Type atributu əsasən inputun məzmun və istifadəçi tərəfindən daxil olunacaq məlumatın növünü təyin edir.

<input type="text" name="username">

**2. name**: inputun adını təyin edir. Bu atribut, input elementinə ad verir və server tərəfindən form məlumatlarının işlənməsi üçün vacibdir.

<input type="text" name="username">

**3. id**: inputun unikal təyin edilmiş id-sini təyin edir. Bu atribut, CSS və JavaScript kimi digər texnologiyalarla inputa istənilən zaman məlumat əlavə etmək və ya tənzimləmək üçün vacibdir.

<input type="text" id="username" name="username">

## 31. CSS nədir? Üstünlükləri və sintaksisi

CSS (Cascading Style Sheets) veb səhifələrin tərtibatı və dizaynını müəyyənləşdirmək üçün istifadə olunan bir texnologiyadır. HTML ilə birlikdə işləyir və HTML sənədlərinin görünüşünü dəyişmək, tərtib etmək və vizual olaraq daha cəlbedici etmək üçün nəzərdə tutulmuşdur. CSS, bir çox üstünlükləri və sadə sintaksisi ilə geniş istifadə olunur.

CSS-in Üstünlükləri

1. Separation of Concerns (Məsuliyyətlərin Ayrılması) - HTML məzmunun strukturunu təmin edir, CSS isə bu məzmunun görünüşünü və tərtibatını müəyyənləşdirir. Bu yanaşma məzmun və dizaynı ayrı saxlamaqla daha təmiz və oxunaqlı kod yazmağa imkan yaradır.

2. Yenidən İstifadə - CSS tərtibatını bir neçə veb səhifədə təkrar istifadə etmək mümkündür. Bu, kodun təkrarlanmasının qarşısını alır və dizayn dəyişikliklərini daha asan və sürətli edir.

3. Dizaynın Konsistensiyası - Bir CSS faylı bütün veb saytlara tətbiq edilərək, saytın bütün səhifələrində eyni dizayn və tərtibatı təmin edə bilər.

4. Daha Sürətli Yükləmə - CSS faylları keşlənə (cache) bilər, yəni bir dəfə yükləndikdən sonra brauzer tərəfindən yadda saxlanılaraq gələcəkdə daha sürətli yüklənə bilər.

5. Qabiliyyət və Təkmilləşmə - CSS inkişaf etdiricilərə daha mürəkkəb və interaktiv dizaynlar yaratmağa imkan verən bir çox xüsusiyyət və metod təklif edir (məsələn, animasiyalar, media sorğuları).

CSS Sintaksisi çox sadədir və əsasən üç əsas hissədən ibarətdir: selektor (selector), xassə (property), və dəyər (value).

Əsas Strukturu:

selektor {

xassə: dəyər;

}

Nümunə:

body {

background-color: lightblue;

}

h1 {

color: navy;

text-align: center;

}

p {

font-family: Arial, sans-serif;

font-size: 14px;

}

CSS həmçinin daha mürəkkəb selektorları dəstəkləyir, məsələn:

- Sinif Selektorları: “.myClass”

- ID Selektorları: “myId”

- Universal Selektor: “ \* ”

- Atribut Selektorları: “a[href]”

## 32. CSS-in HTML səhifəsinə inteqrasiya edilməsi üsulları

CSS-in HTML səhifəsinə inteqrasiya edilməsi üçün üç əsas üsul var: Sətiriçi CSS (İnline CSS), Daxili CSS (İnternal CSS) və Xarici CSS (External CSS). Bu üsulların hər biri fərqli vəziyyətlərdə istifadə oluna bilər.

1. Sətiriçi CSS (İnline CSS) - tək bir HTML elementinə xüsusi stil tətbiq etmək üçün istifadə olunur. Bu üsulda “style” atributu elementin özündə göstərilir.

<!DOCTYPE html>

<html>

<head>

<title>Daxili Stil Misalı</title>

</head>

<body>

<h1 style="color: blue; text-align: center;">Mənim Veb Səhifəm</h1>

<p style="font-size: 20px; color: green;">Bu, bir abzasdır.</p>

</body>

</html>

2. Daxili CSS (İnternal CSS) - bir HTML sənədinin başlıq hissəsində (“<head>”) “<style>” etiketləri arasında yerləşdirilir. Bu üsul, yalnız həmin HTML səhifəsinə aid olan stil qaydalarını təyin etmək üçün istifadə olunur.

<!DOCTYPE html>

<html>

<head>

<title>Daxili Stil Cədvəli Misalı</title>

<style>

body {

background-color: f0f0f0;

}

h1 {

color: blue;

text-align: center;

}

p {

font-size: 20px;

color: green;

}

</style>

</head>

<body>

<h1>Mənim Veb Səhifəm</h1>

<p>Bu, bir abzasdır.</p>

</body>

</html>

3. Xarici CSS (External CSS) - ayrıca bir CSS faylında saxlanılır və bir və ya bir neçə HTML sənədinə link vasitəsilə daxil edilir. Bu üsul, stil qaydalarını təkrar istifadə etmək və veb saytın müxtəlif səhifələrində eyni dizaynı tətbiq etmək üçün ən yaxşı üsuldur.

HTML (index.html):

<!DOCTYPE html>

<html>

<head>

<title>Xarici Stil Cədvəli Misalı</title>

<link rel="stylesheet" type="text/css" href="styles.css">

</head>

<body>

<h1>Mənim Veb Səhifəm</h1>

<p>Bu, bir abzasdır.</p>

</body>

</html>

CSS (styles.css):

body {

background-color: f0f0f0;

}

h1 {

color: blue;

text-align: center;

}

p {

font-size: 20px;

color: green;

}

## 33. CSS framework-ləri. Bootstrap. Bootstrapın HTML səhifəsinə inteqrasiya edilməsi

CSS framework-ləri, veb inkişaf etdiricilər üçün təkrarlanan tapşırıqları sadələşdirən və layihə inkişafını sürətləndirən əvvəlcədən hazırlanmış CSS və JavaScript kod kitabxanalarıdır. Bu framework-lər, dizaynın və yerləşimlərin asanlıqla həyata keçirilməsi üçün təməl komponentləri təmin edir.

Məşhur CSS Framework-ləri:

**1. Bootstrap**: Twitter tərəfindən yaradılmış və ən populyar CSS framework-lərdən biridir. Responsiv dizayn və geniş komponentlər dəstinə malikdir.

**2. Foundation**: ZURB tərəfindən hazırlanmışdır. Mobil cihazlar üçün optimallaşdırılmış və yüksək fərdiləşdirilə bilən komponentlər təklif edir.

**3. Bulma**: Müasir və sadə dizayn prinsiplərinə əsaslanan bir framework-dür. Responsiv və mobil dostu dizaynlar üçün uyğundur.

**4. Tailwind CSS**: Utility-first yanaşması ilə stil qaydalarını birləşdirərək daha çevik və sürətli inkişaf təmin edir.

**Bootstrap** - veb saytların sürətli və asan inkişaf etdirilməsi üçün nəzərdə tutulmuş açıq mənbəli CSS və JavaScript framework-dür. Bootstrap-in əsas xüsusiyyətlərinə aşağıdakılar daxildir:

* Responsiv Dizayn: Mobil və müxtəlif ekran ölçülərinə uyğun gəlir.
* Əvvəlcədən Hazırlanmış Komponentlər: Düymələr, naviqasiya çubuqları və daha çoxu.
* Əsas Şəbəkə Sistemi (Grid System): Layout yaratmaq üçün sadə və güclü bir sistem.
* Uyğun Genişləndirmə və Fərdiləşdirmə: İstəyə uyğun CSS və JavaScript kodu əlavə edə bilərsiniz.

Bootstrap-ın HTML Səhifəsinə İnteqrasiya etməyin iki əsas yolu var: Bootstrap CDN (Content Delivery Network) istifadə etmək və Bootstrap fayllarını yerli olaraq yükləyib istifadə etmək.

1. Bootstrap CDN İlə İnteqrasiya - Bootstrap-ın sürətli və asan şəkildə inteqrasiyası üçün CDN istifadə edə bilərsiniz. Bu üsulda, Bootstrap faylları uzaq bir serverdən yüklənir.

<!DOCTYPE html>

<html>

<head>

<title>Bootstrap Misalı</title>

<link href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css" rel="stylesheet">

</head>

<body>

<div class="container">

<h1 class="text-center">Mənim Veb Səhifəm</h1>

<p class="lead">Bu, Bootstrap ilə hazırlanmış sadə bir nümunədir.</p>

<button class="btn btn-primary">Bootstrap Düyməsi</button>

</div>

<!-- Bootstrap JS və jQuery -->

<script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></script>

<script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.9.3/dist/umd/popper.min.js"></script>

<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>

</body>

</html>

2. Bootstrap Fayllarını Yerli Olar Yükləyib İnteqrasiya

Bootstrap-ın yerli olaraq yüklənməsi və istifadəsi üçün əvvəlcə Bootstrap fayllarını [Bootstrap saytından](https://getbootstrap.com/) yükləyin və layihənizə əlavə edin.

HTML Səhifəsinə Əlavə:

<!DOCTYPE html>

<html>

<head>

<title>Bootstrap Misalı</title>

<!-- Yerli Bootstrap CSS -->

<link href="css/bootstrap.min.css" rel="stylesheet">

</head>

<body>

<div class="container">

<h1 class="text-center">Mənim Veb Səhifəm</h1>

<p class="lead">Bu, Bootstrap ilə hazırlanmış sadə bir nümunədir.</p>

<button class="btn btn-primary">Bootstrap Düyməsi</button>

</div>

<!-- Yerli Bootstrap JS və jQuery -->

<script src="js/jquery.min.js"></script>

<script src="js/popper.min.js"></script>

<script src="js/bootstrap.min.js"></script>

</body>

</html>

## 34. CSS-də “class” və “id” arasındakı fərqi izah edin

CSS-də “class” və “id” seçiciləri HTML elementlərini üslublandırmaq üçün istifadə edilən əsas seçicilərdən ikisidir. Hər ikisinin də müəyyən xüsusiyyətləri və istifadə sahələri var.

**Class** - “class” atributu, bir qrup elementi üslublandırmaq üçün istifadə olunur.

- Sintaksis: Nöqtə (“.”) ilə başlanır.

- Xüsusiyyətlər:

- Eyni sinif adı birdən çox elementə təyin edilə bilər.

- Həm üslublandırma, həm də JavaScript üçün istifadə edilə bilər.

- Bir elementə bir neçə sinif adı təyin oluna bilər.

<!DOCTYPE html>

<html>

<head>

<link rel="stylesheet" type="text/css" href="styles.css">

</head>

<body>

<p class="highlight">Bu abzas birinci sinifə malikdir.</p>

<p class="secondary">Bu abzas ikinci sinifə malikdir.</p>

</body>

</html>

CSS (styles.css):

.highlight {

color: blue;

font-weight: bold;

}

.secondary {

color: green;

font-style: italic;

}

**id** - “id” atributu, unikal bir elementi üslublandırmaq üçün istifadə olunur.

- Xüsusiyyətlər:

- Eyni “id” yalnız bir dəfə istifadə edilə bilər.

- Həm üslublandırma, həm də JavaScript üçün istifadə edilə bilər.

- Elementin unikal identifikatoru kimi xidmət edir.

<!DOCTYPE html>

<html>

<head>

<link rel="stylesheet" type="text/css" href="styles.css">

</head>

<body>

<h1 id="main-title">Bu, əsas başlıqdır.</h1>

<p id="intro">Bu abzas giriş abzasıdır.</p>

</body>

</html>

CSS (styles.css):

main-title {

color: red;

font-size: 24px;

}

intro {

color: gray;

font-size: 18px;

}

Fərqlərin Yekunu:

1. Unikallıq:

- “class”: Birdən çox elementə tətbiq edilə bilər.

- “id”: Yalnız bir element üçün istifadə olunmalıdır.

2. Sintaksis:

- “class”: “.className”

- “id”: “idName”

3. Prioritet:

- CSS-də “id” seçiciləri “class” seçicilərinə nisbətən daha yüksək prioritetə malikdir. Bu o deməkdir ki, eyni element üçün həm “class”, həm də “id” üslubu təyin edilərsə, “id” üslubu tətbiq ediləcək.

## 35. CSS-də universal selector dedikdə nə başa düşürsünüz?

CSS-də universal selector (“\*”) bütün HTML elementlərini seçmək üçün istifadə olunur. Bu seçici, veb səhifəsindəki hər bir elementə üslub tətbiq etmək üçün geniş şəkildə istifadə edilə bilər.

Universal Selector - Universal selector (“\*”) bütün elementləri hədəf alır, yəni bir veb səhifəsindəki hər bir HTML elementinə tətbiq olunur.

Sintaksis

\* {

property: value;

}

1. Bütün Elementlərin Margin və Padding Parametrlərini Sıfırlamaq:

Çox vaxt, brauzerlərin fərqli default stil parametrlərini ləğv etmək üçün universal selector istifadə olunur.

\* {

margin: 0;

padding: 0;

}

Bu kod, bütün HTML elementlərinin margin və padding parametrlərini sıfırlayır, beləliklə veb səhifənizdə daha tutarlı bir dizayn əldə edə bilərsiniz.

2. Bütün Elementlərə Ümumi Stil Tətbiq Etmək:

\* {

box-sizing: border-box;

}

Bu kod, bütün elementlərə “box-sizing: border-box;” təyin edir. Bu, padding və border parametrlərinin elementin ümumi ölçüsünə daxil olmasını təmin edir.

Üstünlüklər:

* Sadəlik və Rahatlıq: Bütün elementlərə ümumi üslub tətbiq etmək üçün asan bir üsuldur.
* Default Stil Parametrlərini Ləğv Etmək: Brauzer default stil parametrlərini sıfırlamaq üçün istifadə edilə bilər.
* Debug: Veb səhifənizin layoutunu anlamaq və debugging etmək üçün faydalıdır.

Dezavantajlar:

* Performans: Bütün elementlərə stil tətbiq edildiyi üçün böyük və mürəkkəb veb səhifələrdə performans problemlərinə səbəb ola bilər.
* Çox Geniş Təsir: Bütün elementlərə təsir etdiyi üçün gözlənilməz nəticələrə səbəb ola bilər və layihənizdə istəmədiyiniz dəyişikliklərə gətirib çıxara bilər.

## 36. POST – GET anlayışı və iş prinsipi

HTTP (Hypertext Transfer Protocol) protokolunda iki əsas metod, POST və GET, məlumatların serverlə mübadiləsi üçün istifadə edilir. Bu metodlar veb inkişafında əsas rol oynayır və müxtəlif məqsədlər üçün istifadə olunur.

GET Metodu - Məlumatların serverdən alınması üçün istifadə olunur. Bu metod URL-də verilən məlumatları göstərərək serverə sorğu göndərir.

Xüsusiyyətləri

1. GET sorğuları məlumatları URL-də (sorğu sətirində) kodlayır.

http://example.com/page?param1=value1&param2=value2

2. Eyni GET sorğusu bir neçə dəfə təkrarlandıqda nəticə dəyişməz qalır. Yəni, GET sorğusu məlumatları dəyişdirmir.

3. GET sorğuları brauzer və server tərəfindən keşlənə bilər.

4. URL uzunluğu məhdud olduğu üçün GET sorğuları vasitəsilə göndərilən məlumatlar məhdud miqdarda olur.

<form action="/search" method="get">

<input type="text" name="query">

<button type="submit">Axtar</button>

</form>

Bu formada istifadəçi məlumatı URL vasitəsilə göndərilir, məsələn:

http://example.com/search?query=sorğu

POST Metodu - Məlumatların serverə göndərilməsi və serverdə yeni resurs yaradılması və ya mövcud resursun yenilənməsi üçün istifadə olunur.

Xüsusiyyətləri

1. POST sorğuları məlumatları HTTP sorğusunun bədənində göndərir.

2. Eyni POST sorğusunun təkrarlanması fərqli nəticələrə səbəb ola bilər.

3. POST sorğuları adətən keşlənmir.

4. POST sorğuları vasitəsilə göndərilən məlumat miqdarı nəzəri olaraq məhdud deyil.

<form action="/submit" method="post">

<input type="text" name="username">

<input type="password" name="password">

<button type="submit">Göndər</button>

</form>

Bu formada istifadəçi məlumatı HTTP sorğusunun bədənində göndərilir və URL-də görünmür.

GET və POST Metodlarının İş Prinsipi

1. GET Metodu:

- İstifadəçi və ya tətbiq URL-də məlumatları daxil edir.

- Brauzer və ya HTTP müştərisi GET sorğusu yaradır və serverə göndərir.

- Server sorğunu qəbul edir, tələb olunan məlumatı emal edir və müvafiq cavabı geri göndərir.

- Brauzer və ya HTTP müştərisi serverin cavabını alır və istifadəçiyə göstərir.

2. POST Metodu:

- İstifadəçi və ya tətbiq forma və ya başqa üsulla məlumatları daxil edir.

- Brauzer və ya HTTP müştərisi POST sorğusu yaradır və məlumatları HTTP sorğusunun gövdəsində serverə göndərir.

- Server sorğunu qəbul edir, göndərilən məlumatı emal edir və müvafiq cavabı geri göndərir.

- Brauzer və ya HTTP müştərisi serverin cavabını alır və istifadəçiyə göstərir.

## 37. Veb Proqramlaşdırma dilləri

Veb proqramlaşdırma dilləri, veb saytların və veb tətbiqlərin inkişafı üçün istifadə olunan proqramlaşdırma dilləridir. Bu dillər frontend və backend inkişafı üçün müxtəlif məqsədlərə xidmət edir.

**Frontend** **Dilləri** veb saytın və ya tətbiqin istifadəçilərə təqdim olunan hissəsini yaratmaq üçün istifadə olunur. Bu dillər brauzer tərəfindən işlənir və veb səhifələrin interfeysini və qarşılıqlı əlaqəsini təmin edir.

1. HTML (HyperText Markup Language) - Veb səhifənin strukturunu və məzmununu təyin edir.

- Xüsusiyyətlər:

* Mətn, şəkil, video və digər məzmun elementlərini təşkil edir.
* Elementlərin yerləşdirilməsi və əlaqələndirilməsi üçün istifadə olunur.

2. CSS (Cascading Style Sheets) - Veb səhifələrin görünüşünü və tərzini təyin edir.

- Xüsusiyyətlər:

* Şriftlər, rənglər, ölçülər, arxa planlar və layoutlar kimi vizual xüsusiyyətləri idarə edir.
* Responsiv dizayn və media sorğuları vasitəsilə müxtəlif cihazlara uyğunluq təmin edir.

3. JavaScript - Veb səhifələrə dinamiklik və qarşılıqlı əlaqə əlavə edir.

- Xüsusiyyətlər:

* Hadisə idarəetməsi, formaların doğrulanması, animasiyalar və s. funksiyaları təmin edir.
* Bir çox kitabxana və framework-lər (məsələn, jQuery, React, Angular, Vue.js) ilə genişləndirilə bilər.

**Backend Dilləri** veb serverdə işləyən və veb tətbiqlərin məntiqi, məlumat emalı və məlumat bazası idarəçiliyini təmin edən dillərdir. Bu dillər serverin işini və veb tətbiqinin funksionallığını idarə edir.

1. PHP (Hypertext Preprocessor) - Dinamik veb məzmun yaratmaq və server tərəfli skriptlər yazmaq.

- Xüsusiyyətlər:

* Məlumat bazası ilə əlaqə qurma və məlumat emalı üçün geniş istifadə olunur.
* WordPress, Joomla və Drupal kimi populyar CMS-lərdə istifadə olunur.

2. Python - Veb tətbiqlərin və API-lərin yaradılması.

- Xüsusiyyətlər:

* Django və Flask kimi güclü framework-lərlə istifadə olunur.
* Məntiqi və oxunaqlı sintaksis ilə tanınır.

3. Ruby - Veb tətbiqlərin sürətli inkişafı.

- Xüsusiyyətlər:

* Ruby on Rails kimi populyar framework ilə geniş istifadə olunur.
* Veb tətbiqlərin strukturlaşdırılması və MVC (Model-View-Controller) arxitekturası təmin edir.

4. Java - Yüksək performanslı və miqyaslana bilən veb tətbiqlərin inkişafı.

- Xüsusiyyətlər:

* Spring və Hibernate kimi güclü framework-lərlə istifadə olunur.
* Təhlükəsizlik və güclü tip yoxlaması ilə tanınır.

5. Node.js - JavaScript-in server tərəfli işlənməsi.

- Xüsusiyyətlər:

* Asinxron və hadisə yönümlü bir modelə əsaslanır.
* Express.js kimi framework-lərlə geniş istifadə olunur.

6. C - Microsoft texnologiyaları ilə veb tətbiqlərin yaradılması.

- Xüsusiyyətlər:

* ASP.NET Core kimi framework-lərlə istifadə olunur.
* Güclü tip yoxlaması və inteqrasiya alətləri ilə tanınır.

**Digər Əsas Texnologiyalar və Dillər**

1. SQL (Structured Query Language) - Məlumat bazası idarəetməsi və sorğuları yazmaq.

- Xüsusiyyətlər:

* Məlumatların saxlanması, idarə edilməsi və sorğulanması üçün istifadə olunur.
* MySQL, PostgreSQL, SQLite kimi məlumat bazası sistemlərində istifadə edilir.

2. TypeScript - JavaScript-in genişləndirilmiş versiyası.

- Xüsusiyyətlər:

* Statik tip yoxlaması və daha yaxşı kod idarəçiliyi təmin edir.
* Angular kimi framework-lərlə geniş istifadə olunur.

3. GraphQL - API-lərin yaradılması və məlumatların sorğulanması üçün daha çevik bir dil.

- Xüsusiyyətlər:

* Məlumatların dəqiq şəkildə sorğulanması və yalnız lazım olan məlumatların alınması üçün istifadə olunur.

## 38. PHP və sintaksisi

PHP (Hypertext Preprocessor) server tərəfli skript dilidir və dinamik veb səhifələrin yaradılmasında geniş şəkildə istifadə olunur. PHP-in sintaksisi sadə və öyrənilməsi asan olduğu üçün veb inkişafında populyar seçimdir.

PHP Faylının Quruluşu

PHP kodu “<?php ... ?>“ etiketi arasında yazılır və bu etiketi HTML sənədində də istifadə edə bilərsiniz.

<!DOCTYPE html>

<html>

<head>

<title>PHP Nümunəsi</title>

</head>

<body>

<h1>PHP ilə Salam Dünya!</h1>

<?php

echo "Salam Dünya!";

?>

</body>

</html>

Əsas Sintaksis Elementləri

1. Dəyişənlər - PHP-də dəyişənlər “$” simvolu ilə başlanır və adları rəqəm, hərf və alt xətdən (“\_”) ibarət ola bilər. Lakin ad rəqəmlə başlamalıdır. Dəyişən adında böyük kiçik hərf həssaslığı var. Yəni $Ad və $ad fərqli dəyişəndir.

<?php

$ad = "Ali";

$yas = 25;

echo "Ad: " . $ad . ", Yaş: " . $yas;

?>

2. Məlumat Tipləri - PHP-də bəzi əsas məlumat tipləri var: tam ədəd (integer), üzən nöqtəli ədəd (float), mətn (string), məntiqi (boolean), massiv (array), obyekt (object), NULL.

<?php

$integer = 10;

$float = 10.5;

$string = "Salam";

$boolean = true;

$array = array(1, 2, 3);

$object = (object) ['property' => 'value'];

$null = NULL;

?>

3. Operatorlar - PHP-də riyazi, məntiqi, müqayisə və digər operatorlar mövcuddur.

<?php

$a = 5;

$b = 10;

$c = $a + $b; // 15

$d = $a \* $b; // 50

$e = $b / $a; // 2

$equal = ($a == $b); // false

$not\_equal = ($a != $b); // true

$greater = ($b > $a); // true

$and = ($a < $b && $b > $c); // false

$or = ($a > $b || $b > $c); // false

?>

4. İfadələr və Dəyişənlərin Çıxarılması - PHP-də “echo” və “print” funksiyaları mətn və dəyişənləri çıxarmaq üçün istifadə olunur.

<?php

echo "Salam Dünya!";

print "Bu da bir başqa mesaj.";

$ad = "Ali";

echo "Ad: " . $ad;

?>

5. Şərt Bəyanatları (if-else) - PHP-də şərt bəyanatları məntiqi ifadələri qiymətləndirmək və müvafiq kodları icra etmək üçün istifadə olunur.

<?php

$yas = 20;

if ($yas >= 18) {

echo "Böyükdür.";

} else {

echo "Kiçikdir.";

}

?>

6. Dövr Strukturları (for, while, foreach) - PHP-də müxtəlif dövr strukturları mövcuddur.

<?php

// for dövrü

for ($i = 0; $i < 5; $i++) {

echo "Nömrə: " . $i . "<br>";

}

// while dövrü

$j = 0;

while ($j < 5) {

echo "Nömrə: " . $j . "<br>";

$j++;

}

// foreach dövrü

$numbers = array(1, 2, 3, 4, 5);

foreach ($numbers as $number) {

echo "Nömrə: " . $number . "<br>";

}

?>

7. Funksiyalar - Funksiyalar kodun yenidən istifadə edilə bilən hissələrini yaratmaq üçün istifadə olunur.

<?php

function salam($ad) {

return "Salam, " . $ad;

}

echo salam("Ali");

?>

Məlumat Bazası Əlaqəsi - PHP-də məlumat bazası ilə əlaqə qurmaq üçün geniş istifadə olunan üsul PDO (PHP Data Objects) və ya MySQLi-dir. Aşağıda PDO-nun istifadəsinə dair bir nümunə verilmişdir:

<?php

try {

$pdo = new PDO("mysql:host=localhost;dbname=example\_db", "username", "password");

$pdo->setAttribute(PDO::ATTR\_ERRMODE, PDO::ERRMODE\_EXCEPTION);

$stmt = $pdo->query("SELECT FROM users");

while ($row = $stmt->fetch(PDO::FETCH\_ASSOC)) {

echo "Ad: " . $row['ad'] . ", Email: " . $row['email'] . "<br>";

}

} catch (PDOException $e) {

echo "Bağlantı xətası: " . $e->getMessage();

}

?>

## 39. PHP scriptlərin təhlükəsizliyi

PHP, internetin ən populyar proqramlaşdırma dillərindən biridir və çox geniş istifadə olunur. Lakin, PHP kodunun təhlükəsizliyi ilə bağlı bəzi məsələlər olur ki, bu məsələlərə diqqət etmək əhəmiyyətlidir:

1. SQL İnjection: İstifadəçi məlumatlarını bazaya daxil edərkən daxili işarələri yoxlamalı və parametrizasiya üsullarından istifadə etməlisiniz ki, SQL injeksiya hücumlarından qorunasanız.

$username = $\_POST['username'];

$password = $\_POST['password'];

// SQL sorgusunu hazırlama

$sql = "SELECT FROM users WHERE username = ? AND password = ?";

$stmt = $pdo->prepare($sql);

$stmt->execute([$username, $password]);

2. XSS (Cross-Site Scripting): İstifadəçilərin daxil etdiyi məlumatları təhlükəsiz bir şəkildə ekrana çıxarmaq üçün “htmlspecialchars” və ya “strip\_tags” kimi funksiyalardan istifadə edin.

echo htmlspecialchars($user\_input);

3. Form validation: İstifadəçi məlumatlarını qəbul edən form elementlərinin düzgün validasiya edilməsi və təhlükəsizliyin yoxlanılması (məsələn, e-poçt ünvanları və şifrələrin minimal tələbləri kimi).

4. Dosya işləmələri: İstifadəçilərin yüklədiyi faylların güvənliyini təmin etmək üçün fayl adlarını yoxlamaq, yalnız müəyyən fayl tiplərinə icazə vermək və yükləmələrə məhdudiyyətlər təyin etmək lazımdır.

5. Session security: Session id-lərinin təhlükəsiz və qorunan bir şəkildə idarə olunması üçün HTTPS istifadə etmək və session id-lərinin hər dəfə dəyişdirilməsi məsləhət görülür.

session\_regenerate\_id(true);

6. Dependency management: PHP və PHP paketləri üçün təhlükəsizlik güncəlləmələri və patch-lər üçün sisteminizi yeniləyin.

## 40. PHP forma validasiyası

PHP ilə forma validasiyası həssas və istifadəçi dostu bir funksionallığı əlavə etmək üçün əhəmiyyətlidir. Bu, istifadəçilərin forma məlumatlarını düzgün şəkildə daxil etdiklərini yoxlamaq və tələblərə uyğunluğunu yoxlamaq üçün əsaslı bir mühiti təmin edir. Aşağıda PHP ilə sadə bir forma validasiyası nümunəsi göstərilmişdir.

Formanın HTML hissəsi

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Forma Validasiyası</title>

<style>

.error { color: red; }

</style>

</head>

<body>

<h2>Qeydiyyat Forması</h2>

<form method="post" action="<?php echo htmlspecialchars($\_SERVER["PHP\_SELF"]); ?>">

<label for="ad">Adınız:</label><br>

<input type="text" id="ad" name="ad"><br>

<label for="email">Email ünvanınız:</label><br>

<input type="text" id="email" name="email"><br>

<label for="parol">Şifrə:</label><br>

<input type="password" id="parol" name="parol"><br>

<label for="confirm\_parol">Şifrənin təkrarı:</label><br>

<input type="password" id="confirm\_parol" name="confirm\_parol"><br><br>

<input type="submit" name="submit" value="Təsdiqlə">

</form>

<?php

// Forma göndərilməsindən sonra validasiya

if ($\_SERVER["REQUEST\_METHOD"] == "POST") {

$ad = $\_POST['ad'];

$email = $\_POST['email'];

$parol = $\_POST['parol'];

$confirm\_parol = $\_POST['confirm\_parol'];

$errors = [];

// Adın boş olmaması yoxlanılır

if (empty($ad)) {

$errors['ad'] = "Adınızı daxil edin."; }

// Emailin düzgün formatda olub-olmadığı yoxlanılır

if (!filter\_var($email, FILTER\_VALIDATE\_EMAIL)) {

$errors['email'] = "Düzgün email ünvanı daxil edin."; }

// Şifrənin minimum uzunluğu yoxlanılır

if (strlen($parol) < 6) {

$errors['parol'] = "Şifrə minimum 6 simvoldan ibarət olmalıdır."; }

// Şifrənin təkrarı ilə uyğunluğu yoxlanılır

if ($parol !== $confirm\_parol) {

$errors['confirm\_parol'] = "Şifrələr uyğun gəlmir."; }

// Əgər səhv yoxdursa məlumatların göstərilməsi

if (empty($errors)) {

echo "<h3>Qeydiyyat məlumatlarınız:</h3>";

echo "Ad: " . htmlspecialchars($ad) . "<br>";

echo "Email: " . htmlspecialchars($email) . "<br>";

// Şifrə təhlükəsizliyi üçün yoxlanmayacaq, sadəcə göstəriləcək

echo "Şifrə: " . str\_repeat('', strlen($parol)) . "<br>"; }

else {

// Səhvlərin göstərilməsi

echo "<h3>Səhvlər:</h3><ul>";

foreach ($errors as $error) {

echo "<li class='error'>" . $error . "</li>"; }

echo "</ul>"; }

}

?>

</body>

</html>

İzahat:

1. HTML Forması: HTML forması “method="post"“ ilə göndərilir və “action” atributu “$\_SERVER["PHP\_SELF"]” ilə özünə yönəldilir. Formada ad, email, şifrə və şifrənin təkrarı sahələri var.

2. PHP Validasiyası: PHP hissəsində “$\_POST” istifadə olunaraq formadan gələn məlumatlar alınır. “$\_SERVER["REQUEST\_METHOD"] == "POST"“ şərti ilə forma göndərilməsini yoxlayırıq. Əgər forma göndərilibsə, daxil edilmiş məlumatlar validasiya edilir:

- “empty()” funksiyası ilə adın boş olub-olmadığı yoxlanılır.

- “filter\_var()” funksiyası ilə emailin düzgün formatda olub-olmadığı yoxlanılır (“FILTER\_VALIDATE\_EMAIL” filteri ilə).

- Şifrənin minimum uzunluğu “strlen()” funksiyası ilə yoxlanılır.

- Şifrənin təkrarı ilə uyğunluğu daxilində “===“ operatoru ilə müqayisə edilir.

3. Səhvlərin İşlənməsi və Göstərilməsi: Əgər səhvlər varsa, onlar “foreach” dövrü ilə “ul” elementində göstərilir. Əks halda, qeydiyyat məlumatları təsdiqlənərək göstərilir.

## 41. PHP-də açar sözlərin və dəyişənlərin böyük/kiçik hərflərə həssaslığı

Açar Sözlər (Keywords) - PHP-də bütün açar sözlər (keywords) kiçik hərflərlə yazılır. Bu açar sözlər dilin sintaksisini təyin edir və hər hansı bir variant (böyük hərflərlə, yalnız baş hərf böyük olaraq) qəbul edilmir. Məsələn:

- “if”, “else”, “while”, “for”, “foreach”, “function”, “class” kimi açar sözlər kiçik hərflərlə yazılır.

Dəyişənlər (Variables) - dəyişənlərin adları istifadəçilər tərəfindən təyin olunur və PHP-də həm böyük, həm də kiçik hərflərlə yazılan dəyişənlər fərqli kimi qəbul olunur. Məsələn:

$ad = "Ali";

$Ad = "Məhəmməd";

$AD = "Əli";

Yuxarıdakı nümunədə “$ad”, “$Ad” və “$AD” dəyişənləri fərqli dəyişənlər kimi qəbul olunur və hər biri özünə məxsus dəyərə malikdir.

## 42. PHP-də dəyişənlərin elan edilməsi və çapa verilməsi

PHP-də dəyişənlər adları ilə müəyyənləşdirilən məlumatların saxlanılması üçün istifadə olunur. Aşağıda PHP-də dəyişənlərin necə elan edilməsi və istifadə olunması barədə ətraflı məlumat verilmişdir:

**Dəyişənlərin Elan Edilməsi** - PHP-də dəyişənlər “$” işarəsi ilə başlayır və adları öz istifadə edilməkləri üçün təyin edilmiş adla başlayır. Dəyişən adları aşağıdakı qaydalar üzrə olmalıdır:

* Dəyişən adları “$” işarəsi ilə başlamalıdır.
* Dəyişən adları yalnız hərflərdən və rəqəmlərdən ibarət olmalıdır.
* Dəyişən adları alt xətlər (\_) ilə başlaya bilər.
* Dəyişən adları heç bir zaman rəqəm ilə başlaya bilməz.

Məsələn, aşağıdakı kodda “$name” adlı bir dəyişən elan olunur və sonra istifadə olunur:

<?php

$name = "John Doe";

echo $name; // Çıxış: John Doe

?>

**Dəyişənləri Çap Etmək** - Dəyişənləri çap etmək üçün “echo” və ya “print” istifadə oluna bilər. Dəyişən adı “$” işarəsi ilə birlikdə yazılır. Məsələn:

<?php

$name = "John Doe";

echo $name; // Çıxış: John Doe

?>

## 43. PHP data tipləri hansılardır?

PHP-də müxtəlif data tipləri mövcuddur. Hər biri müxtəlif növ və dəyərlərdə saxlanılan məlumatları işləmək üçün istifadə olunur.

1. Integer (Tam Ədəd): Tam ədədlər, məsələn, “-100”, “0”, “42” kimi. PHP-də integer tipləri 32 bit və 64 bit kimi olur.

2. Float (Onluq Növ): Onluq və ya rəqəmli tiplər, məsələn, “3.14”, “2.71828” kimi. Bu tiplər float və ya double olaraq da adlandırılır.

3. String (Sözlər): Mətnləri və simvolları saxlamaq üçün istifadə olunur. Mətnlər “ **“ ”** ” və ya “ **‘ ’** “ təklikləri arasında yazılır, məsələn, “"Hello, World!"“, “'PHP Programming'“.

4. Boolean (Məntiqi): Yalnız iki dəyəri ola bilən data tipləri: “true” və “false”. Məntiqi operatorlar və şərt ifadələri ilə istifadə olunur.

5. Array (Qrup): Bir sıra ilə indekslənmiş məlumat tipləri qrupu. PHP-də əsasən iki növ array mövcuddur: indexed arrays (indekslənmiş qruplar) və associative arrays (əlaqəli qruplar).

6. Object (Obyekt): İstifadəçi tərəfindən yaradılan və ya ilkinləşdirilmiş bir sınıfın məlumatlarını saxlamaq üçün istifadə olunan məlumat tipləri.

7. NULL: Hər hansı bir dəyəri olmayan və ya boş dəyəri göstərmək üçün istifadə olunur.

## 44. PHP-də lokal, qlobal və statik dəyişənlər

PHP-də dəyişənlər fərqli sahələrdə və məqsədlər üçün təyin oluna bilər. İşlənən mühitə və dəyişənin istifadə məqsədini nəzərə alaraq, lokal, qlobal və statik dəyişənlər fərqlənir:

Lokal Dəyişənlər - əsasən funksiyalar və metodlar daxilində təyin olunan dəyişənlərdir. Bu dəyişənlər yalnız daxili funksiyalarda istifadə olunur və funksiya daxilində təyin edilən məzmunun yalnız funksiya işlənərkən fəaliyyət göstərir. Funksiyon tamamlandıqdan sonra lokal dəyişənlər qarışılıqdan çıxarılır və yaddaşdan silinir. Məsələn:

function hello() {

$message = "Hello, World!";

echo $message;

}

hello(); // Çıxış: Hello, World!

// $message dəyişəni bu növbədə lokal olaraq qalır və başqa funksiyalar və ya skript tərəfinə görə məlumat əldə edə bilməz.

Qlobal Dəyişənlər - skriptin əsas hissəsində və ya funksiyaların xaricində təyin olunan dəyişənlərdir. Bu dəyişənlər skriptin daxilində hər hansı bir funksiya və ya metodun tərəfindən istifadə oluna bilər və hər zaman yaddaşda saxlanılır. Qlobal dəyişənləri “$GLOBALS” təsvirindən istifadə edərək istifadə edə bilərsiniz. Məsələn:

$name = "John Doe";

function greet() {

global $name;

echo "Hello, " . $name!;

}

greet(); // Çıxış: Hello, John Doe!

Burada “$name” qlobal dəyişən olaraq təyin olunub və “greet()” funksiyası içində istifadə edilir.

Statik Dəyişənlər - daxili funksiyonlarda və ya metodlarda istifadə olunan xüsusi dəyişənlərdir. Bu dəyişənlər funksiya daxilində istifadə edilən dəyişənlərə oxşar olaq, lakin funksiya yenidən çağırılanda dəyişənin məzmunu yenidən təyin olunmur və qalır. Məsələn:

function counter() {

static $count = 0;

$count++;

echo $count . "<br>";

}

counter(); // Çıxış: 1

counter(); // Çıxış: 2

counter(); // Çıxış: 3

Burada “$count” statik dəyişən kimi təyin olunub və funksiya hər dəfə çağırılanda dəyişənin dəyəri artırılaraq saxlanılır.

## 45. PHP-də funksiyalar və sintaksisi

PHP-də funksiyalar, proqram tərəfində yazılmış təkrar edilən əməlləri icra etmək üçün istifadə olunan bloklardır. Funksiyalar, kodunuzun daha düzgün, təkrarlanabilir və idarə olunması daha asandır.

Funksiya Təyinatı - PHP-də funksiya təyin etmək üçün “function” açar sözü istifadə olunur. Funksiya təyinatı aşağıdakı sintaksisə uyğun olmalıdır:

function function\_name($parameter1, $parameter2, ...) {

// funksiya daxilində icra ediləcək əməllər

return $result; // opsional olaraq bir dəyər qaytarılabilir

}

Ən əsas hissələr:

- **function\_name**: Adlar funksiyanı təyin edərkən dəyişənlərdən fərqli olaraq böyük və kiçik hərflərə həssasdır.

- **$parameter1, $parameter2, ...**: Funksiyaya gələn parametrlər. Parametrlər funksiya təyinatında opsiyonaldır və daxili olaraq istifadə olunur.

- **return**: Funksiyanın işlədikdən sonra qaytaracağı dəyəri təyin edir. “return” ifadəsi istəyə bağlıdır və funksiyanın işinə və məqsədinə görə dəyişir.

Funksiya çağırmaq üçün sadəcə funksiyanın adını və parametrlərini qeyd edin:

// Funksiya təyinatı

function greet($name) {

echo "Hello, $name!";

}

// Funksiya çağırışı

greet("John"); // Çıxış: Hello, John!

PHP-də funksiya parametrləri varsayılan dəyərlər təyin edə bilərsiniz:

function greet($name = "Guest") {

echo "Hello, $name!";

}

greet(); // Çıxış: Hello, Guest!

greet("Jane"); // Çıxış: Hello, Jane!

Statik Və Global Dəyişənlər

Funksiya daxilində qlobal və statik dəyişənlərə müraciət edə bilərsiniz:

$global\_var = "Global variable";

function test() {

global $global\_var;

echo $global\_var;

}

test(); // Çıxış: Global variable

Statik dəyişənlər funksiyanın hər dəfə çağırılmasında yenidən təyin olunmur:

function counter() {

static $count = 0;

$count++;

echo $count . "<br>";

}

counter(); // Çıxış: 1

counter(); // Çıxış: 2

## 46. PHP şərhlərində hansı simvollar istifadə edilir?

PHP şərhlərində istifadə olunan simvollar aşağıdakılardır:

1. Bir Sətirlik Şərhlər:

- İki kəsir xətti (“//”): Bu simvolların arxasından gələn hər şey şərh olaraq qəbul edilir və həmin sətir kompilyasiya edilmir.

// Bu bir sətirlik şərhdir

- Diyez işarəsi (“#”): Bu da eyni funksiyanı yerinə yetirir və yalnız həmin sətiri şərh olaraq qəbul edir.

# Bu da bir sətirlik şərhdir

2. Çox Sətirlik Şərhlər:

- Şərhlər blokunu açmaq və bağlamaq üçün “/” və “/” istifadə olunur. Bu metod bir neçə sətirlik şərhlər üçün istifadə olunur.

/\*

Bu çox sətirlik bir şərhdir

və bir neçə sətiri əhatə edir.

\*/

## 47. JavaScript data tipləri hansılardır?

JavaScript-də məlumat tipləri əsasən iki kateqoriyaya bölünür: primitiv tiplər və mürəkkəb tiplər.

Primitiv Tiplər - bu tiplər sadə və dəyişməz (immutable) dəyərlərdir:

**1. Number**: Rəqəmləri təmsil edir. Həm tam ədəd (integer), həm də onluq (floating-point) rəqəmlər bu tipə daxildir.

let age = 25; // Tam ədəd

let temperature = 98.6; // Onluq ədəd

**2. String**: Mətn məlumatlarını təmsil edir. İkiqat dırnaq (" "), tək dırnaq (' ') və ya backtick (` `) istifadə edilə bilər.

let name = "John Doe";

let greeting = 'Hello, World!';

let template = `This is a template string`;

**3. Boolean**: Yalnız iki dəyər qəbul edir: `true` və `false`.

let isActive = true;

let isFinished = false;

**4. Undefined**: Dəyişənə dəyər təyin edilmədikdə bu tipə malik olur.

let notAssigned;

console.log(notAssigned); // undefined

**5. Null**: Məqsədli olaraq boş və ya mövcud olmayan bir dəyəri təmsil edir.

let emptyValue = null;

**6. Symbol**: Unikal və dəyişməz bir dəyəri təmsil edir. Simvollar, əsasən, obyekt açarları kimi istifadə olunur.

let uniqueId = Symbol('id');

**7. BigInt**: Çox böyük rəqəmləri təmsil etmək üçün istifadə olunur. Bu, standart “Number” tipinin təhlükəsiz saxladığı rəqəm diapazonunu aşan dəyərləri saxlamaq üçün yaradılmışdır.

let bigNumber = BigInt(9007199254740991);

let anotherBigNumber = 9007199254740991n;

8. Object: Açar-dəyər cütlərindən ibarət bir kolleksiya və ya kompleks məlumat strukturudur. Obyektlərin xüsusiyyətləri və metodları ola bilər.

let person = {

name: "Alice",

age: 30,

isStudent: true

};

## 48. Javascript şərhlərində hansı simvollar istifadə edilir?

Bir Sətirlik Şərhlər - Tək bir sətri şərh etmək üçün istifadə olunur və “//” simvolları ilə işarələnir. Bu simvolların ardınca gələn hər şey həmin sətirdə şərh olaraq qəbul edilir.

// Bu bir sətirlik şərhdir

let x = 5; // Bu da bir sətirlik şərhdir

Çox Sətirlik Şərhlər - Bir neçə sətri şərh etmək üçün istifadə olunur və “/\*” ilə başlayır, “\*/” ilə bitir. Bu, şərhin başlanğıcını və sonunu işarələyir.

/\*

Bu çox sətirlik bir şərhdir.

Bir neçə sətri əhatə edir.

\*/

let y = 10;

## 49. Javascript kodları ilə yazılmış 3+2+”7″ nəticəsi nədir? Səbəbi izah edin.

JavaScript-də `3 + 2 + "7"` ifadəsinin nəticəsi `57` olacaq. Bunun səbəbini başa düşmək üçün JavaScript-in operator overloading qaydalarını və növ çevirilmələrini izah etmək lazımdır.

1. Əvvəlcə “3 + 2” hesablanır: Burada həm “3”, həm də “2” rəqəm (number) tipindədir. JavaScript-də iki rəqəm arasında “+” operatoru istifadə edildikdə, bu rəqəmlər cəmlənir. “3 + 2” = “5”.

2. Sonra nəticə “5” ilə “"7"“ (string) toplanır: Bu nöqtədə “5” rəqəm tipindədir, lakin “"7"“ string tipindədir. JavaScript-də bir rəqəmlə bir string arasında “+” operatoru istifadə edildikdə, rəqəm avtomatik olaraq stringə çevrilir və concatenation (birləşdirmə) əməliyyatı həyata keçirilir. “5” rəqəmi stringə çevrilir və “"5"“ olur. “"5" + "7"“ = “"57"“.

## 50. JavaScript. Üstünlükləri və çatışmazlıqları. HTML sənədində tətbiq üsulları

JavaScript-in Üstünlükləri

**1. Platformdan Asılı Olmaması**: JavaScript müxtəlif əməliyyat sistemlərində və veb brauzerlərdə işləyir. Bu onu çox platformalı tətbiqlər üçün ideal edir.

**2. Zəngin Kitabxana və Framework-lər**: jQuery, React, Angular, Vue.js kimi çoxlu kitabxana və framework-lər var. Bu, inkişaf müddətini qısaldır və funksionallığı artırır.

**3. Dəstək və İcma**: JavaScript çox böyük və aktiv bir icma tərəfindən dəstəklənir. Bu, çoxlu resurslar və dəstək əldə etməyə imkan verir.

**4. DOM Manipulyasiyası**: JavaScript DOM-u dəyişmək, istifadəçi hadisələrini işləmək və dinamik məzmun yaratmaq üçün güclüdür.

JavaScript-in Çatışmazlıqları

**1. Təhlükəsizlik**: JavaScript-in açıq mənbə kodu istifadəçi tərəfindən görünə bilər və bu, təhlükəsizlik problemlərinə yol aça bilər. Kodun manipulyasiyası və zərərli skriptlərin daxil edilməsi mümkündür.

**2. Uyğunluq Problemləri**: Müxtəlif brauzerlər JavaScript-i fərqli şəkildə dəstəkləyə bilər. Bu, xüsusilə köhnə brauzerlərdə uyğunluq problemlərinə səbəb ola bilər.

**3. Performans Məsələləri**: Çox böyük və mürəkkəb JavaScript kodları brauzerin işləmə sürətini yavaşlada bilər. Bu, istifadəçi təcrübəsini mənfi təsir edə bilər.

**4. Dinamik Yazı Sistemi**: JavaScript-də dinamik yazı sistemi bəzən səhvləri tapmağı və düzəltməyi çətinləşdirə bilər. Bu, kodun daha az etibarlı olmasına səbəb ola bilər.

JavaScript-in HTML sənədinə bir neçə fərqli üsulla tətbiq edilə bilər:

**1. Daxili Skript (Inline Script)**: JavaScript birbaşa HTML elementlərinin “onclick”, “onmouseover” və digər hadisə atributlarında yazıla bilər.

<button onclick="alert('Hello, World!')">Click Me</button>

**2. Skripti Başlıq və ya Gövdə İçində Yerləşdirmək**: JavaScript “<script>“ etiketi istifadə edilərək HTML sənədinin “<head>“ və ya “<body>“ bölmələrində yazıla bilər.

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Document</title>

<script>

// Başlıq içində JavaScript

function showMessage() {

alert('Hello from the head!');

}

</script>

</head>

<body>

<button onclick="showMessage()">Click Me</button>

<script>

// Gövdə içində JavaScript

console.log('Hello from the body!');

</script>

</body>

</html>

3. Xarici Skript (External Script): JavaScript kodu ayrı bir “.js” faylına yazıla bilər və həmin fayl HTML sənədinə “<script src="path/to/script.js"></script>“ etiketi ilə əlavə edilə bilər.

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Document</title>

<script src="script.js"></script>

</head>

<body>

<button onclick="showMessage()">Click Me</button>

</body>

</html>

// script.js faylı

function showMessage() {

alert('Hello from the external script!');

}

## 51. Java və JavaScript arasındakı fərqlər nələrdir?

Java və JavaScript arasındakı fərqlər çoxdur və onlar arasında əsas fərqləri anlamak, hər iki dili daha yaxşı istifadə etmək üçün vacibdir.

1. Məqsəd və İstifadə Sahəsi:

- **Java**:

* Ümumi məqsədli proqramlaşdırma dili olaraq, masaüstü, veb, mobil tətbiqlər və hətta server tərəfli tətbiqlər üçün istifadə olunur.
* Java Virtual Machine (JVM) üzərində işləyir və bir dəfə yazılan kod hər yerdə işləyə bilir ("write once, run anywhere").

- **JavaScript**:

* Əsasən veb brauzerlərdə işləyən və veb səhifələrin qarşılıqlı fəaliyyətini təmin etmək üçün istifadə olunan skript dilidir.
* Əsasən müştəri tərəfli (client-side) proqramlaşdırma üçün nəzərdə tutulmuşdur, lakin Node.js ilə server tərəfli (server-side) də istifadə edilə bilir.

2. Sintaksis:

- **Java**:

* Sintaksisi C/C++ dillərinə bənzəyir və ciddi tip yoxlanışı var.
* Obyekt yönümlü proqramlaşdırma (OOP) prinsipinə əsaslanır.
* Tiplər compile vaxtı yoxlanılır (statically typed).

- **JavaScript**:

* Daha sadə və çevik bir sintaksisə malikdir.
* Dinamik tiplidir, yəni tiplər icra zamanı müəyyən edilir (dynamically typed).
* Obyekt yönümlü xüsusiyyətləri dəstəkləyir, lakin prototip əsaslı bir obyekt modelinə malikdir.

3. Run və Performans:

- **Java**:

* Kod əvvəlcə bytecode-a compile edilir və sonra JVM üzərində işlədilir.
* Performansı yüksəkdir və böyük miqyaslı tətbiqlərdə geniş istifadə olunur.

- **JavaScript**:

* Kod birbaşa brauzer tərəfindən icra edilir.
* Müştəri tərəfli tətbiqlər üçün optimallaşdırılmışdır, lakin Node.js ilə server tərəfli proqramlaşdırmada da istifadə edilə bilir.
* Dinamik təbiətinə görə performansı bəzi hallarda Java-dan aşağı ola bilər.

4. Mühit:

- **Java**:

* Java Development Kit (JDK) və Integrated Development Environment (IDE) (məsələn, Eclipse, IntelliJ IDEA) ilə işləyir.
* Yazılı kod JVM-də çalışır.

- **JavaScript**:

* Hər hansı bir mətn redaktoru ilə yazıla bilər və veb brauzerlərdə işlədir.
* Node.js ilə server tərəfli istifadə üçün dəstəklənir.
* Çoxlu sayda kitabxana və framework (məsələn, React, Angular, Vue.js) mövcuddur.

5. Obyekt Yönümlülük:

- **Java**:

* Tamamilə obyekt yönümlü bir dildir və hər şey siniflər və obyektlər əsasında qurulur.
* İrsi (inheritance), interfeyslər (interfaces), polimorfizm (polymorphism) kimi OOP konsepsiyalarını dəstəkləyir.

- **JavaScript**:

* Prototip əsaslı obyekt yönümlü proqramlaşdırmanı dəstəkləyir.
* Sinif əsaslı OOP konsepsiyaları ES6 ilə əlavə edilmişdir, lakin əsasında prototiplər var.

6. İstifadəçilər və Tətbiq Sahələri

- **Java**:

* Bankçılıq, korporativ tətbiqlər, mobil tətbiqlər (Android) və böyük miqyaslı sistemlərdə geniş istifadə olunur.

- **JavaScript**:

* Veb inkişafında dominant rola malikdir. Veb səhifələrin qarşılıqlı fəaliyyətini təmin etmək, istifadəçi interfeysləri yaratmaq və müasir veb tətbiqlər üçün istifadə edilir.

## 52. Verilənlər bazası. MySQL

MySQL açıq mənbə kodlu, çox istifadə olunan əlaqəli verilənlər bazası idarəetmə sistemidir (RDBMS). MySQL, məlumatların səmərəli və etibarlı şəkildə saxlanması və idarə olunması üçün SQL (Structured Query Language) istifadə edir. Oracle Corporation tərəfindən dəstəklənir və müxtəlif platformalarda işləyir.

MySQL-in Əsas Xüsusiyyətləri

1. **Performans və Sürət**: MySQL yüksək performanslı və sürətli bir RDBMS-dir. Bu, həm kiçik layihələrdə, həm də geniş miqyaslı tətbiqlərdə effektivdir.
2. **Etibarlılıq və Təhlükəsizlik**: Güclü təhlükəsizlik mexanizmləri və məlumatların bütövlüyünü qorumaq üçün müxtəlif vasitələr təklif edir.
3. **Platformdan Asılı Olmama:** MySQL Windows, Linux, macOS və digər platformalarda işləyir.
4. **Çox İstifadəçi Dəstəyi:** MySQL çoxlu istifadəçinin eyni vaxtda verilənlər bazasına daxil olmasına imkan verir.
5. **Replication və High Availability**: Verilənlər bazasının nüsxələrini yaratmaq və yüksək mövcudluq təmin etmək üçün replikasiya dəstəyi var.
6. **İnteqrasiya**: MySQL, PHP, Java, Python və digər proqramlaşdırma dilləri ilə asanlıqla inteqrasiya edilə bilər.

MySQL-də Əsas SQL Əməliyyatları

1. Verilənlər Bazasının Yaradılması:

CREATE DATABASE mydatabase;

2. Cədvəlin Yaradılması:

CREATE TABLE users (

id INT AUTO\_INCREMENT PRIMARY KEY,

username VARCHAR(50) NOT NULL,

email VARCHAR(100),

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP

);

3. Məlumat Əlavə Edilməsi:

INSERT INTO users (username, email) VALUES ('john\_doe', 'john@example.com');

4. Məlumatın Seçilməsi:

SELECT \* FROM users;

5. Məlumatın Yenilənməsi:

UPDATE users SET email = 'john\_new@example.com' WHERE username = 'john\_doe';

6. Məlumatın Silinməsi:

DELETE FROM users WHERE username = 'john\_doe';

## 53. MYSQL və SQL arasındakı fərqi izah edin

MySQL və SQL tez-tez qarışdırılan, lakin fərqli olan iki termindir.

SQL (Structured Query Language) - SQL, əlaqəli verilənlər bazası idarəetmə sistemlərində (RDBMS) məlumatları idarə etmək və manipulyasiya etmək üçün istifadə olunan standart proqramlaşdırma dilidir. SQL xüsusiyyətləri:

* SQL dili məlumatların daxil edilməsi, seçilməsi, yenilənməsi və silinməsi üçün istifadə olunur.
* SQL müxtəlif əmrlərdən ibarətdir, məsələn, “SELECT”, “INSERT”, “UPDATE”, “DELETE”, “CREATE TABLE”, “DROP TABLE”, və s.
* SQL dili ANSI (American National Standards Institute) və ISO (International Organization for Standardization) tərəfindən standartlaşdırılmışdır.
* SQL-in müxtəlif tətbiqləri Oracle, Microsoft SQL Server, PostgreSQL, SQLite və MySQL kimi müxtəlif verilənlər bazası idarəetmə sistemlərində istifadə edilir.

SQL Əmrlərinə Nümunələr:

-- Cədvəl yaradılması

CREATE TABLE users (

id INT PRIMARY KEY AUTO\_INCREMENT,

username VARCHAR(50) NOT NULL,

email VARCHAR(100)

);

-- Məlumat əlavə edilməsi

INSERT INTO users (username, email) VALUES ('john\_doe', 'john@example.com');

-- Məlumatın seçilməsi

SELECT \* FROM users;

-- Məlumatın yenilənməsi

UPDATE users SET email = 'john\_new@example.com' WHERE username = 'john\_doe';

-- Məlumatın silinməsi

DELETE FROM users WHERE username = 'john\_doe';

MySQL - Veb dizayn dünyasında istifadə edilən ən məşhur açıq mənbəli əlaqəli verilənlər bazası idarəetmə sistemlərindən biridir. MySQL xüsusiyyətləri:

* MySQL açıq mənbə kodlu, əlaqəli verilənlər bazası idarəetmə sistemidir (RDBMS).
* MySQL, məlumatları idarə etmək və manipulyasiya etmək üçün SQL dilindən istifadə edir.
* MySQL Oracle Corporation tərəfindən dəstəklənir.
* MySQL yüksək performanslı, çox istifadəçili və təhlükəsiz bir verilənlər bazası sistemidir.
* MySQL müxtəlif əməliyyat sistemlərində işləyə bilir, məsələn, Windows, Linux, macOS və s.
* MySQL müxtəlif proqramlaşdırma dilləri ilə (məsələn, PHP, Python, Java) asanlıqla inteqrasiya oluna bilir.

MySQL-in İstifadə Nümunəsi: MySQL-i istifadə etmək üçün əvvəlcə verilənlər bazasına qoşulmaq lazımdır. MySQL-də SQL sorğularını işlətmək üçün bir müştəri (client) istifadə edilir, məsələn, “mysql” komanda xətti vasitəsi, phpMyAdmin, MySQL Workbench və s.

<?php

$servername = "localhost";

$username = "root";

$password = "";

$dbname = "mydatabase";

// Bağlantı yaradılır

$conn = new mysqli($servername, $username, $password, $dbname);

// Bağlantı yoxlanılır

if ($conn->connect\_error) {

die("Bağlantı alınmadı: " . $conn->connect\_error);

}

// SQL sorğusu hazırlanır

$sql = "SELECT id, username, email FROM users";

$result = $conn->query($sql);

if ($result->num\_rows > 0) {

// Hər bir sətri görüntüləmək

while($row = $result->fetch\_assoc()) {

echo "id: " . $row["id"]. " - Username: " . $row["username"]. " - Email: " . $row["email"]. "<br>";

}

} else {

echo "0 nəticə";

}

// Bağlantı bağlanır

$conn->close();

?>

Əsas Fərqlər

1. Tərif:

- SQL: Verilənlər bazası ilə qarşılıqlı əlaqə yaratmaq üçün istifadə olunan standart proqramlaşdırma dilidir.

- MySQL: Verilənlər bazası idarəetmə sistemidir (RDBMS) və SQL dilindən istifadə edərək məlumatları idarə edir.

2. Funksiya:

- SQL: Dil olaraq məlumatları sorğulamaq və manipulyasiya etmək üçün istifadə olunur.

- MySQL: Verilənlər bazası sistemidir və SQL dilindən istifadə edərək məlumatları saxlayır, idarə edir və emal edir.

3. İstifadə Sahəsi:

- SQL: Müxtəlif verilənlər bazası idarəetmə sistemlərində (məsələn, Oracle, SQL Server, PostgreSQL) istifadə olunur.

- MySQL: Xüsusi olaraq MySQL verilənlər bazası idarəetmə sistemi üçün istifadə olunur, lakin SQL dilini istifadə edir.

## 54. PhpMyAdmində username və password-untəyin edilməsi. MySQL VB-na daxil olma qaydası

İlk olaraq biz PHPMyAdmin-ə daxil olmalıyıq. Orada əsas səhifədə User Accounts bölməsinə daxil oluruq. Orada qarşımıza çıxan istifadəçilərdən adı “root” və Əsas maşın adı “localhost” olan istifadəçinin qarşısında edit permissions-a klikləyirik. Change password hissəsindən password təyin edirik.

MySQL VB-ni biz PhpMyAdminlə idarə edirik. Ona daxil olmaq üçün biz ilk öncə XAMPP proqramından APACHE server və MySQL modulunu aktivləşdiririk. Daha sonra brauzerimizdə localhost/phpmyadmin/-ə daxil oluruq ilk daxil olduqda orada bizdə login hissəsi olacaq. Bu zaman ilk səfər olduğu üçün orada username root qeyd edirik, parol olmur.

## 55. PhpMyAdmində VB-nın və cədvəllərin yaradılması mərhələləri. Cədvəllərdə əsas və xarici açar anlayışı

Cədvəllərin yaradılması: Yeni baza yaratmaq üçün sol tərəfdə yuxarıda dayanan “New”-a klik edirik. Burada bizim vb-lərimiz görünür. Create database hissəsinə baza adı yazıb Create edirik. Bazanı yaratdıqdan dərhal sonra bizə ilk cədvəli yaratmaq üçün səhifə açır. Burada cədvəlin adını və istədiyimiz sayda sütun əlavə etməyə imkan verir. Müvafiq xananı doldurub “create”-ə toxunuruq. Açılan pəncərədə biz cədvəlin sütunlarının adlarını onların tipini, uzunluğunu və sütundakı verilənlərin “primary key” olması və s. təyin edirik.

Cədvəllərin əlaqələndirilməsi üçün 2 cür açar var: **Əsas açar** (**P**rimary **K**ey, **PK**) və **xarici açar** (**F**oreign **K**ey, **FK**). Əsas açar - cədvəlin sütununu unikallaşdırır. Yəni, həmin açarın qiyməti baxılan cədvəldə təkrarlana bilməz. Xarici açar – cədvəllər arasında əlaqə yaradan sahədir.

Cədvəli yaratdıqdan sonra bizim qarşımıza onun strukturu çıxır. Biz burada cədvəlin sütunlarını görürük. İstəsək əgər, buradan “change”-ə klikləyərək istənilən sütunun xüsusiyyətlərini dəyişə bilərik. Biz daha sonra cədvələ məlumat əlavə etmək üçün SQL sorğulardan istifadə etməliyik. Bunun üçün yuxarıda yerləşən SQL bölməsinə daxil oluruq. Burada bizə “İnsert” yəni daxiletmə əməliyyatı lazım olduğu üçün onu yazırıq. Əgər biz cədvəlin müəyyən bölməsini “Auto İncrement” seçsək, həmin bölmədəki qiymətlər avtomatik artır. Biz sadəcə digər bölməsini doldururuq. “Go”-ya klik etdiyimizdə sorğu göndərilir və baza müvafiq məlumatlar cədvələ doldurulur. Bu məlumatları doldurduqdan sonra cədvəlimizi görmək üçün “Browse” bölməsinə keçirik.

Daha sonra 2-ci cədvəli yaradırıq. 2-ci cədvəl üçün bir bölmə “Primary Key”-dir. Digər cədvəldən fərqli olaraq burada eyni adlı bölmə “Foreign Key”-dir. Bunu daha sonra əlaqələndirmə zamanı istifadə edirik. Əlaqə yaratmaq üçün FK olan cədvəldə “Sturucture” bölməsində “Relation view” hissəsinə keçid edirik. Burada quracağımız əlaqəyə bir ad təyin edirik. 1-ci “Column” hissəsində FK adını, daha sonra uyğun olaraq vb adı, PK yerləşən cədvəlin adı və oradakı PK olan hissənin adını qeyd edirik. Save”-i klik edərək əlaqəni yaradırıq. Daha sonra VB-mizə klikləyib oradakı “Designer” bölməsinə keçid edərək bu əlaqəni görə bilərik.

## 56. MYSQL və PHP faylı arasında əlaqə üsulları

PHP 5 və sonrakı versiyalar MySQL verilənlər bazası ilə işləyə bilər:

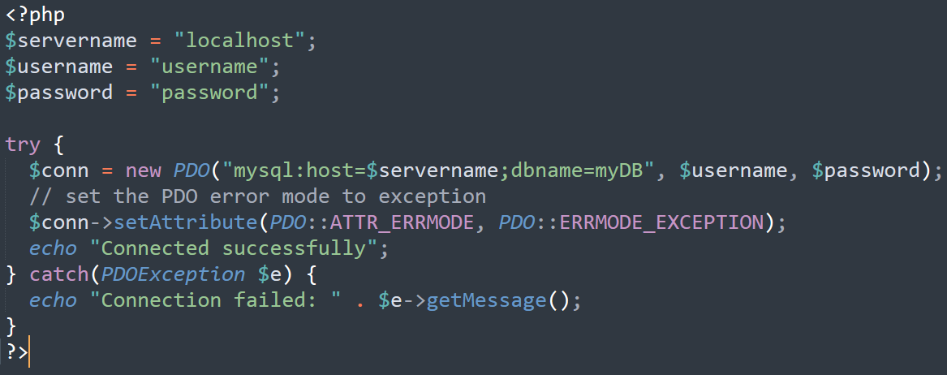
* MySQLi genişləndirilməsi ("i" təkmilləşdirilmiş (improved) deməkdir)
* PDO (PHP Data Object)

PHP-nin əvvəlki versiyaları MySQL genişlənməsindən istifadə edirdi. Ancaq bu uzadılma 2012-ci ildə ləğv edildi.

**MySQLi və PDO fərqi**

* İstər PDO, istərsə də MySQLi funksiyalar üçün Obyekt yönümlü interfeys təklif edir
* PDO-nun MySQLi-dən əsas üstünlüyü verilənlər bazası dəstəyindədir. PDO 12 müxtəlif verilənlər bazası növünü dəstəklədiyi halda, MySQLi yalnız MySQL-i dəstəkləyir.
* Layihənizi başqa verilənlər bazasından istifadə etmək üçün dəyişməli olduğunuz zaman, PDO prosesi asanlaşdırır. Beləliklə, sizə lazım olan yeganə prosedur, yeni verilənlər bazanız tərəfindən dəstəklənməyən hər hansı sintaksisdən istifadə edərsə, əlaqə sətirini və ən çox bir neçə sorğunu dəyişdirmək olacaq.

**MySQLi OBJECT-ORIENTED**  **MySQLi PROCEDURAL**

**PDO**

## 57. MYSQL-də CRUD əməliyyatları

MySQL-də CRUD əməliyyatları (Create, Read, Update, Delete), verilənlər bazası əməliyyatlarını icra etmək üçün əsas funksionallığı təşkil edir.

**1. Create (Yaratma**) - Create əməliyyatı yeni məlumatlar əlavə etmək üçün istifadə olunur. MySQL-də bu, `INSERT INTO` əmri ilə edilir. Nümunə olaraq:

INSERT INTO users (username, email, password) VALUES ('john\_doe', 'john.doe@example.com', 'password123');

**2. Read (Oxuma)** - Read əməliyyatı məlumatları oxumaq üçün istifadə olunur. Bu, `SELECT` əmri ilə icra olunur. Nümunə olaraq:

SELECT \* FROM users;

**3. Update (Yeniləmə)** - Update əməliyyatı mövcud məlumatları yeniləmək üçün istifadə olunur. Bu, `UPDATE` əmri ilə edilir. Nümunə olaraq:

UPDATE users SET email = 'john.doe.updated@example.com' WHERE username = 'john\_doe';

**4. Delete (Silinmə)** - Delete əməliyyatı məlumatları silmək üçün istifadə olunur. Bu, `DELETE FROM` əmri ilə edilir. Nümunə olaraq:

DELETE FROM users WHERE username = 'john\_doe';

## 58. MYSQL – də bazanın export və import edilməsi qaydası

Bazada yuxarıda yerləşən “Export-İmport” bölməsi vasitəsilə bazanı “export” və “import” edə bilərik.

**Export:** Orada export üçün template seçməyimizi deyir. Əgər yoxdursa, ilk öncə bir “export template” yaradırıq. Daha sonra template seçib export edirik. Bunun üçün “export” klikləyirik.

**İmport:** “İmport” bölməsində export olunmuş faylı əlavə edirik. Daha sonra “import” klikləyirik. Beləcə fayl import olunur. Lakin əgər “import” olunan fayl əvvəlcədən mövcuddursa, qarşımıza xəta çıxacaq.

## 59. Kompüter şəbəkələri. Lokal və qlobal şəbəkə.

Kompüter şəbəkələri, lokal (yerli) və qlobal (global) olmaqla iki əsas kateqoriyaya bölünür.

1. Lokal Şəbəkə (Local Area Network - LAN) - Lokal şəbəkə (LAN), bir neçə kompüter və digər cihazın bir-birilə əlaqələndiyi, genəlliklə məhdud məsafədə (bir ofis, bir binanın içi, bir universitet kampusu kimi) yerləşən şəbəkədir. LAN-ın əsas xüsusiyyətləri aşağıdakılardır:

* Məsafə: LAN, əsasən bir neçə yüz metrdən az məsafədə yerləşir.
* Təşkilat: LAN, bir yerdə olan bir təşkilat, məktəb, ofis və s. tərəfindən idarə olunur.
* Texnologiyalar: Ethernet, Wi-Fi kimi texnologiyalar LAN-da istifadə olunur.
* Məqsəd: Məlumatların yerli cəhətdən paylaşılması, printerlər, fayl serverləri, internetə giriş və s. təmin edir.

LAN nümunəsi, bir ofisdəki kompüterlərin, printerlərin və digər cihazların bir Ethernet kabelləri və ya Wi-Fi ilə birləşdiyi şəbəkədir.

2. Qlobal Şəbəkə (Wide Area Network - WAN) - Qlobal şəbəkə (WAN), daha böyük məsafələr və bir neçə fərqli yer arasında uzanır. WAN, geniş əraziyə yayılmış və xaricdəki serverlərlə, resurslarla, internetlə əlaqə qurmağı təmin edir. Qlobal şəbəkənin əsas xüsusiyyətləri aşağıdakılardır:

* Məsafə: WAN, böyük məsafələri (millətlərarası, milli səth, ölkələr arası) örtən geniş şəbəkədir.
* Təşkilat: İnternet, milli şəbəkələr, ölkə sərhədləri və telekomunikasiya şirkətləri tərəfindən idarə olunur.
* Texnologiyalar: WAN, internet protokolləri (TCP/IP, UDP, DNS kimi) və telekomunikasiya texnologiyaları (DSL, Fiber Optik, Satelit, Mobil şəbəkələr və s.) istifadə edir.
* Məqsəd: İnternet, VPN (Virtual Private Network), Cloud servisləri, milli və beynəlxalq əlaqələr, telekomünikasiya və multimedia məqsədləri üçün nəzarət təmin edir.

Qlobal şəbəkə, dünya ərazisindən keçən, fərqli providerlərin (təchizatçılar) tərəfindən idarə olunan və birləşdirilmiş internetin bütün hissələrini əhatə edir.

## 60. Kompüter şəbəkələrinin aparat təminatı

Şəbəkə avadanlıqları kompüter şəbəkələrinin qurulması və idarə olunması üçün istifadə olunan müxtəlif cihazları əhatə edir. Aşağıda əsas şəbəkə avadanlıqları verilmişdir:

1. Router (Marşrutlaşdırıcı) - Şəbəkələr arasında məlumat paketlərini yönləndirir və bir şəbəkəni digərinə qoşur. Ev, ofis və geniş miqyaslı şəbəkələrdə istifadə olunur.

2. Switch (Komutator) - Eyni şəbəkə daxilində cihazları birləşdirir və məlumat paketlərini doğru cihazlara yönləndirir. Lokal Şəbəkələrdə (LAN) geniş istifadə olunur.

3. Hub - Cihazları mərkəzi nöqtədə birləşdirir və məlumatları bütün portlara eyni zamanda göndərir. Sadə və kiçik şəbəkələrdə istifadə olunur, lakin sürət və təhlükəsizlik baxımından məhduddur.

4. Access Point (Giriş Nöqtəsi) - Simsiz şəbəkələrin (Wi-Fi) yaradılmasını təmin edir. Ev və iş yerlərində simsiz əlaqə təmin etmək üçün istifadə olunur.

5. Modem - İnternet xidmət təminatçısının (ISP) siqnallarını qəbul edir və onları istifadə üçün çevrilir. İnternetə qoşulma üçün istifadə olunur (DSL, kabel, fiber optik modemlər).

6. Network Interface Card (NIC - Şəbəkə İnterfeysi Kartı) - Kompüteri və ya digər cihazı şəbəkəyə qoşmaq üçün vacibdir. Bir çox cihazda və kompüterdə var(Ethernet, Wi-Fi və digər əlaqə növləri üçün).

7. Load Balancer (Yüklənmə Dağıtıcısı) - Məlumat axınını bir neçə server arasında bərabər şəkildə paylayır. Böyük miqyaslı veb saytlar və tətbiqlərdə.

9. Repeater (Təkrarçı) - Şəbəkə siqnalını gücləndirir və uzaq məsafələrə ötürür. Şəbəkə əhatə dairəsini genişləndirmək üçün istifadə olunur.

10. Bridge (Körpü) - İki və ya daha çox LAN-ı birləşdirir və onları eyni şəbəkə kimi işləməyə imkan verir. Müxtəlif şəbəkələrin inteqrasiyasında.

11. Cable (Kabellər) - Fiziki əlaqə üçün istifadə olunur. Ethernet kabelləri (CAT5, CAT6 və s.), fiber optik kabellər.

## 61. Şəbəkə topologiyaları

Şəbəkə topologiyası şəbəkədəki qovşaqların və birləşmələrin fiziki və məntiqi düzülüşüdür. Düyünlərə adətən açarlar, marşrutlaşdırıcılar və keçid və marşrutlaşdırıcı xüsusiyyətləri olan proqram təminatı kimi qurğular daxildir. Şəbəkə topologiyaları çox vaxt qrafik şəklində təqdim olunur.

• star • ring • mesh • bus • tree

Bus şəbəkəsi: Yerli şəbəkədir. Topologiyada hər bir qovşağın - iş stansiyası və ya digər cihazın - əsas kabelə və ya şin adlanan keçidə qoşulur. Bus-dakı bütün qoşulmuş stansiyalar tək şəbəkədəki bütün digərləri ilə əlaqə saxlaya bilir. Şəbəkə seqmenti sadə və etibarlıdır. Əgər bir stansiya işləmirsə, qalan stansiyalar hələ də bir-biri ilə əlaqə saxlaya bilirlər. Bus şəbəkələrini genişləndirmək də asandır.

Star şəbəkəsi: Yerli şəbəkədir. Topologiyada bütün qovşaqlar - fərdi kompüterlər, iş stansiyaları və ya digər qurğular - birbaşa mərkəz kimi adlandırılan ümumi mərkəzi kompüterə qoşulur. Buna görə də star şəbəkəsi tez-tez mərkəz adlanır və danışıq şəbəkəsi topologiyası mərkəzə qoşulmuş hər bir iş stansiyasına dolayı yolla qoşulur. Star şəbəkələri adətən müəssisə şəbəkələrinin giriş səviyyəsində yerləşdirilir.

Ring şəbəkəsi: Qovşaqlar qapalı dövrə konfiqurasiyasında birləşdirilir. Bəzi ring-lər məlumatları yalnız bir istiqamətdə ötürür, digərləri isə hər iki istiqamətdə ötürmə qabiliyyətinə malikdir. Bu iki istiqamətli ring şəbəkələri bus şəbəkələrindən daha davamlıdır, çünki trafik hər iki istiqamətdə hərəkət edərək bir node-a çata bilər. Sinxron Optik Şəbəkə texnologiyasına əsaslanan metro şəbəkələri ring şəbəkələrinin əsas nümunəsidir.

Mesh şəbəkəsi: Qovşaqları elə əlaqələndirir ki, şəbəkənin ən azı bəzi nöqtələri arasında çoxlu yollar mövcud olsun. Bütün qovşaqlar bütün digər qovşaqlara birbaşa bağlıdırsa və yalnız bəzi qovşaqların digərləri ilə çoxlu əlaqəsi varsa, şəbəkə tam şəbəkəli (fully meshed) hesab olunur. Çoxlu yolların birləşdirilməsi dayanıqlığı artırır, həm də xərcləri artırır. Tam mesh (full mesh) şəbəkə topologiyasında hər bir qovşaq birbaşa bütün digər qovşaqlara qoşulur. Qismən mesh topologiyasında yalnız bəzi qovşaqlar bir-birinə birbaşa bağlanır.

Tree şəbəkəsi: bir kök qovşağından ibarətdir və bütün digər qovşaqlar bir iyerarxiyada birləşdirilir. Topologiyanın özü star konfiqurasiyasında birləşdirilir. Məlumat mərkəzi şəbəkələri də daxil olmaqla bir çox böyük Ethernet keçid şəbəkələri ağac kimi konfiqurasiya edilmişdir. Tree topologiyası böyük şəbəkələr, xüsusən də qovşaqların bir yerdə qruplaşdırıldığı şəbəkələr üçün idealdır.

Hybrid şəbəkə: Hibrid topologiyanın üstünlüyü onun çevikliyidir ki, siz şəbəkəni istədiyiniz kimi uyğunlaşdıra bilərsiniz. Ehtiyaclarınıza ən uyğun olan şəkildə star və bus, star və ring və ya daha çoxunu birləşdirə bilərsiniz.



## 62. Şəbəkə protokolları. İP ünvan

Şəbəkə protokolları və İP ünvanlar şəbəkələrdə məlumatların ötürülməsi və ünvanlaşdırılması üçün vacib anlayışlardır.

Şəbəkə Protokolları - şəbəkədəki cihazların bir-biri ilə necə ünsiyyət qurmasını və məlumatları necə mübadilə etməsini müəyyənləşdirən qaydalar toplusudur. Bəzi əsas şəbəkə protokollarına aşağıdakılar daxildir:

1. TCP/IP (Transmission Control Protocol/Internet Protocol): İnternetin əsas protokolu olan TCP/IP məlumatların etibarlı şəkildə ötürülməsini təmin edir. TCP, məlumat paketlərinin səhvsiz və düzgün ardıcıllıqla çatmasını təmin edir, IP isə paketlərin ünvanlanmasını və marşrutlandırılmasını həyata keçirir.

2. HTTP(HTTPS) (Hypertext Transfer Protocol/Secure): Veb səhifələrin internet üzərindən ötürülməsi üçün istifadə olunan protokoldur. HTTPS təhlükəsiz versiyasıdır. SSL/TLS ilə şifrələnir.

3. FTP (File Transfer Protocol): Faylların bir kompüterdən digərinə ötürülməsi üçün istifadə olunur.

4. SMTP (Simple Mail Transfer Protocol): Elektron poçtların göndərilməsi üçün istifadə olunan protokoldur.

5. DNS (Domain Name System): Domen adlarını İP ünvanlara çevirmək üçün istifadə olunan sistemdir.

İP Ünvan - internetə qoşulan hər bir cihazın unikal identifikatorudur. İP ünvanlar iki əsas versiyada mövcuddur: IPv4 və IPv6.

IPv4 - 32-bit uzunluğunda və dörd oktetdən ibarətdir (məsələn, 192.168.1.1). Hər oktet 0 ilə 255 arasında bir dəyər ala bilər. Təxminən 4.3 milyard unikal İP ünvan təmin edə bilir.

IPv6 - 128-bit uzunluğunda və səkkiz heksadesimal blokdan ibarətdir (məsələn, 2001:0db8:85a3:0000:0000:8a2e:0370:7334). Praktik olaraq sonsuz sayda İP ünvan təmin edə bilir. IPv4 ünvanlarının tükənməsi problemini həll etmək üçün hazırlanmışdır.

## 63. Şəbəkə avadanlıqları

Şəbəkə avadanlıqları kompüter şəbəkələrinin qurulması və idarə olunması üçün istifadə olunan müxtəlif cihazları əhatə edir. Aşağıda əsas şəbəkə avadanlıqları verilmişdir:

1. Router (Marşrutlaşdırıcı) - Şəbəkələr arasında məlumat paketlərini yönləndirir və bir şəbəkəni digərinə qoşur. Ev, ofis və geniş miqyaslı şəbəkələrdə istifadə olunur.

2. Switch (Komutator) - Eyni şəbəkə daxilində cihazları birləşdirir və məlumat paketlərini doğru cihazlara yönləndirir. Lokal Şəbəkələrdə (LAN) geniş istifadə olunur.

3. Hub - Cihazları mərkəzi nöqtədə birləşdirir və məlumatları bütün portlara eyni zamanda göndərir. Sadə və kiçik şəbəkələrdə istifadə olunur, lakin sürət və təhlükəsizlik baxımından məhduddur.

4. Access Point (Giriş Nöqtəsi) - Simsiz şəbəkələrin (Wi-Fi) yaradılmasını təmin edir. Ev və iş yerlərində simsiz əlaqə təmin etmək üçün istifadə olunur.

5. Modem - İnternet xidmət təminatçısının (ISP) siqnallarını qəbul edir və onları istifadə üçün çevrilir. İnternetə qoşulma üçün istifadə olunur (DSL, kabel, fiber optik modemlər).

6. Network Interface Card (NIC - Şəbəkə İnterfeysi Kartı) - Kompüteri və ya digər cihazı şəbəkəyə qoşmaq üçün vacibdir. Bir çox cihazda və kompüterdə var(Ethernet, Wi-Fi və digər əlaqə növləri üçün).

7. Load Balancer (Yüklənmə Dağıtıcısı) - Məlumat axınını bir neçə server arasında bərabər şəkildə paylayır. Böyük miqyaslı veb saytlar və tətbiqlərdə.

9. Repeater (Təkrarçı) - Şəbəkə siqnalını gücləndirir və uzaq məsafələrə ötürür. Şəbəkə əhatə dairəsini genişləndirmək üçün istifadə olunur.

10. Bridge (Körpü) - İki və ya daha çox LAN-ı birləşdirir və onları eyni şəbəkə kimi işləməyə imkan verir. Müxtəlif şəbəkələrin inteqrasiyasında.

11. Cable (Kabellər) - Fiziki əlaqə üçün istifadə olunur. Ethernet kabelləri (CAT5, CAT6 və s.), fiber optik kabellər.

## 64. Hub və switch fərqi nədir?

Hub və switch şəbəkə cihazlarıdır və şəbəkə əlaqələrində istifadə olunur. Ancaq onların iş prinsipləri və funksionallıqları fərqlidir.

**Hub** - şəbəkə cihazıdır və əsasən aşağıdakı xüsusiyyətlərə malikdir:

1. Şəbəkə Bağlantısı: Hub kompüterləri və digər şəbəkə cihazlarını bir araya gətirərək, məlumatların ötürülməsini təmin edir.

2. Məlumatın Yayılması: Hub-a daxil olan məlumat paketləri bütün portlara göndərilir. Bu o deməkdir ki, məlumatlar şəbəkədəki bütün cihazlara yayımlanır, amma yalnız məqsəd ünvanı olan cihaz həmin məlumatı qəbul edir.

3. OSI Modeli: Hub OSI modelinin fiziki qatında (Layer 1) işləyir.

4. Bant Genişliyi: Hub-un bütün portları eyni bant genişliyini paylaşır, yəni eyni anda çoxlu məlumat göndərildikdə şəbəkə yüklənməsi və toqquşmalar (collisions) artar.

5. Ağıllı İşləmə: Hub sadə bir cihazdır və məlumat paketlərini yönləndirmək üçün heç bir ağıllı qərar vermir.

**Switch** - Switch daha ağıllı və effektiv bir şəbəkə cihazıdır və aşağıdakı xüsusiyyətlərə malikdir:

1. Şəbəkə Bağlantısı: Switch də kompüterləri və digər şəbəkə cihazlarını bir araya gətirir, amma daha ağıllı şəkildə.

2. Məlumatın Yönləndirilməsi: Switch məlumat paketlərini yalnız məqsəd ünvanı olan portlara yönləndirir. Bu, şəbəkə trafikinin effektiv idarə edilməsinə və toqquşmaların azalmasına kömək edir.

3. MAC Ünvanları: Switch hər bir cihazın MAC ünvanını öyrənir və bu ünvanları əsas götürərək məlumat paketlərini yönləndirir.

4. OSI Modeli: Switch OSI modelinin məlumat bağlantısı qatında (Layer 2) işləyir. L3 switchlər isə şəbəkə qatında (Layer 3) işləyə bilər.

5. Bant Genişliyi: Hər bir port öz ayrı bant genişliyinə malikdir, bu da şəbəkə performansını artırır.

6. Ağıllı İşləmə: Switch məlumat paketlərinin effektiv və düzgün şəkildə yönləndirilməsini təmin edən kompleks qərarlar qəbul edir.

## 65. OSİ modeli və səviyyələrin izahı

OSI (Açıq Sistemlər İnteqrasiya) modeli, kompüter şəbəkələrinin və kommunikasiya sistemlərinin qurulması və protokollərinin standartlaşdırılması üçün istifadə olunan bir referans modelidir. OSI modeli, əsasən, məlumatın bir cihazdan digərinə necə ötürüldüyünü və məlumatlar arasında necə əməliyyatlar yerinə yetirildiyini göstərir. OSI modeli, 7 mərhələdən ibarətdir və hər biri öz spesifik funksiyaları ilə məşğuldur:

1. Fiziki mərhələ (Physical Layer) - Fiziki mərhələ, məlumatın elektrik sinqalları, optik sinqallar və radio dalğaları kimi fiziki vasitələrlə necə ötürüldüyünü təyin edir. Bu mərhələdə Ethernet kabelləri, optik liflər, koaksial kablolar və ya radio dalğaları kimi fiziki infrastruktur yer alır.

2. Link mərhələsi (Data Link Layer) - Link mərhələsi, məlumatın fiziki mərhələdən necə ötürüldüyünü təhlil edir və məlumatı çoxlu cihazlar arasında necə hərəkət etdirmək üçün protokollar təyin edir. MAC (Media Access Control) adresləri, Ethernet və Wi-Fi kimi protokollar bu mərhələdə yer alır.

3. Şəbəkə mərhələsi (Network Layer) - Şəbəkə mərhələsi, məlumat paketlərinin ötürülməsi üçün yolların təyin edilməsini və məlumatın gediş-nəqliyyatını idarə edir. IP (Internet Protocol) ünvanlarını, router-ları və IP routing protokollarını daxil edir.

4. Nəqliyyat mərhələsi (Transport Layer) - Nəqliyyat mərhələsi, məlumatın gecikmə və səhvə qarşı qorunması üçün nəzərdə tutulur. TCP (Transmission Control Protocol) və UDP (User Datagram Protocol) kimi protokollar bu mərhələdə yer alır.

5. Məlumat təhlil mərhələsi (Session Layer) - Məlumat təhlil mərhələsi, məlumatın istifadəçilər arasında nəzərdə tutulmuş seanslardan (sessions) və əməliyyatlarının idarə olunmasından məsuldur.

6. Təqdimat mərhələsi (Presentation Layer) - Təqdimat mərhələsi, məlumatın qəbul edilə bilən forma hazırlanması, şifrlənməsi, şifrələrin çözülməsi və məlumatın tərcüməsi ilə məşğuldur.

7. Tətbiqat mərhələsi (Application Layer) - Tətbiqat mərhələsi, istifadəçilərin proqramlarının və tətbiqatlarının məlumatlaşmasını təmin edir. Bu mərhələdə, fayl paylaşımı, elektron poçtlar, veb brauzerləri, və buna oxşar tətbiqatlar yer alır.

## 66. İnkapsulyasiya və deinkapsulyasiya nədir?

İnkapsulyasiya (Encapsulation) - Məlumatların yuxarı səviyyə protokollarından aşağı səviyyə protokollarına ötürülərkən müəyyən protokol başlıqlarının və ya qoşulmalarının (headers) əlavə edilməsi prosesidir.

1. Tətbiq Səviyyəsi (Application Layer): Məlumat istifadəçinin tətbiqi tərəfindən yaradılır.

2. Təqdimat Səviyyəsi (Presentation Layer): Məlumatın formatlanması və şifrələnməsi aparılır.

3. Sessiya Səviyyəsi (Session Layer): Məlumatın əlaqəsi yaradılır və idarə olunur.

4. Nəqliyyat Səviyyəsi (Transport Layer): Məlumat paketlərə (segmente) bölünür və hər bir paketə nəqliyyat başlığı (məsələn, TCP başlığı) əlavə edilir.

5. Şəbəkə Səviyyəsi (Network Layer): Nəqliyyat səviyyəsindəki paketlərə şəbəkə başlığı (məsələn, IP başlığı) əlavə edilir və paketlər datagramlara çevrilir.

6. Məlumat Bağlantısı Səviyyəsi (Data Link Layer): Datagramlara məlumat bağlantısı başlığı (məsələn, MAC başlığı) və sonluğu (trailer) əlavə edilir.

7. Fiziki Səviyyə (Physical Layer): Məlumat bitlərə çevrilir və fiziki mühit vasitəsilə ötürülür.

Deinkapsulyasiya (Decapsulation) - Deinkapsulyasiya, məlumatların aşağı səviyyə protokollarından (məsələn, fiziki səviyyədən) yuxarı səviyyə protokollarına (məsələn, tətbiq səviyyəsinə) ötürülərkən həmin protokol başlıqlarının və ya qoşulmalarının çıxarılması prosesidir. Bu proses məlumatın mənbədən hədəfə doğru hərəkət edərkən hər bir səviyyədə baş verir:

1. Fiziki Səviyyə (Physical Layer): Fiziki mühitdən alınan bitlər məlumat bağlantısı səviyyəsinə ötürülür.

2. Məlumat Bağlantısı Səviyyəsi (Data Link Layer): Məlumat bağlantısı başlığı və sonluğu çıxarılır, qalan məlumat şəbəkə səviyyəsinə ötürülür.

3. Şəbəkə Səviyyəsi (Network Layer): Şəbəkə başlığı çıxarılır və qalan məlumat nəqliyyat səviyyəsinə ötürülür.

4. Nəqliyyat Səviyyəsi (Transport Layer): Nəqliyyat başlığı çıxarılır və qalan məlumat sessiya səviyyəsinə ötürülür.

5. Sessiya Səviyyəsi (Session Layer): Sessiya səviyyəsi əlaqəni idarə edir və məlumatı təqdimat səviyyəsinə ötürür.

6. Təqdimat Səviyyəsi (Presentation Layer): Məlumat formatlanır və deşifrələnir, sonra tətbiq səviyyəsinə ötürülür.

7. Tətbiq Səviyyəsi (Application Layer): Məlumat istifadəçinin tətbiqi tərəfindən qəbul edilir və istifadə olunur.

## 67. Kabel növləri

Şəbəkələrdə məlumatların ötürülməsi üçün müxtəlif kabel növləri istifadə olunur. Hər bir kabel növü öz xüsusiyyətlərinə və istifadə sahələrinə malikdir. Əsas kabel növləri aşağıdakılardır:

1. Koaksial Kabel - Koaksial kabellər şəbəkələrdə və digər rabitə sistemlərində geniş istifadə olunan kabel növüdür. Onların əsas xüsusiyyətləri:

* Struktur: Mərkəzdə bir mis tel, onu əhatə edən izolyasiya təbəqəsi, onun üzərində ekranlama (shielding) və ən son xarici plastik örtükdən ibarətdir.
* Nümunələr: RG-6, RG-59
* İstifadə sahələri: Kabel televiziyası, internet bağlantıları, analog və rəqəmsal video siqnalları.
* Üstünlükləri: Yüksək bant genişliyi və müdaxiləyə qarşı yaxşı qorunma təmin edir.

2. Bükülü Cüt Kabel (Twisted Pair Cable) - Bükülü cüt kabellər Ethernet şəbəkələrində ən çox istifadə olunan kabel növüdür. İki əsas növü var:

a) UTP (Unshielded Twisted Pair)

* Struktur: Bir-birinə bükülmüş cüt mis tellərdən ibarətdir. Üzərində əlavə ekranlama yoxdur.
* Kateqoriyalar: Cat 5, Cat 5e, Cat 6, Cat 6a, Cat 7
* İstifadə sahələri: Yerli şəbəkələr (LAN), telefon xətləri.
* Üstünlükləri: Əlverişli qiymət, quraşdırma asanlığı.

b) STP (Shielded Twisted Pair)

* Struktur: Bükülü cüt tellərin ətrafında əlavə ekranlama mövcuddur.
* Kateqoriyalar: Cat 5, Cat 6, Cat 7
* İstifadə sahələri: Elektromaqnit müdaxilənin yüksək olduğu mühitlər.
* Üstünlükləri: Daha yaxşı müdaxilə qoruması, daha yüksək bant genişliyi.

3. Optik Fiber Kabel (Fiber Optic Cable) - Optik fiber kabellər məlumatların işıq siqnalları vasitəsilə ötürülməsi üçün istifadə olunur. İki əsas növü var:

a) Single-mode Fiber

* Struktur: Daha dar nüvə (core) malikdir və işıq siqnalları bir yol ilə ötürülür.
* İstifadə sahələri: Uzun məsafəli və yüksək sürətli məlumat ötürülməsi.
* Üstünlükləri: Uzun məsafəli ötürmə, daha az siqnal zəifləməsi.

b) Multi-mode Fiber

* Struktur: İşıq siqnalları bir neçə yol ilə ötürülür.
* İstifadə sahələri: Qısa məsafəli məlumat ötürülməsi, kampus şəbəkələri.
* Üstünlükləri: Daha əlverişli qiymət, qısa məsafələrdə yüksək sürət.

4. Serial Kabel - Serial kabellər əsasən köhnə və xüsusi cihazların bir-biri ilə əlaqələndirilməsi üçün istifadə olunur.

* Struktur: RS-232, RS-485 kimi standartlara malik ola bilər.
* İstifadə sahələri: Sənaye avtomatlaşdırma sistemləri, modemlər, köhnə periferiya cihazları.
* Üstünlükləri: Sadə dizayn, xüsusi tətbiqlər üçün uyğunluq.

## 68. Bulud texnologiyaları

Bulud texnologiyaları (Cloud Computing) məlumatların saxlanılması, işlənməsi və idarə edilməsi üçün internet üzərindən xidmətlərin təmin edilməsini nəzərdə tutur. Bu texnologiyalar fərdi istifadəçilərdən tutmuş, iri müəssisələrə qədər geniş bir istifadəçi spektrinə xidmət edir və ənənəvi yerli serverlərə və fərdi cihazlara ehtiyac olmadan məlumatların saxlanılmasına və işlənməsinə imkan verir.

Bulud Texnologiyalarının Əsas Növləri:

1. İctimai Bulud (Public Cloud):

* Xüsusiyyətləri: Xidmətlər və resurslar ictimaiyyətə açıq şəkildə təmin olunur. İstifadəçilər xidmətləri internet üzərindən istifadə edirlər.
* Təchizatçılar: Amazon Web Services (AWS), Microsoft Azure, Google Cloud Platform (GCP).
* Üstünlükləri: Elastiklik, genişləndirilə bilən resurslar, az ilkin investisiya.

2. Özəl Bulud (Private Cloud):

* Xüsusiyyətləri: Resurslar və xidmətlər yalnız bir təşkilat tərəfindən istifadə olunur. Fiziki olaraq təşkilatın özündə və ya üçüncü tərəf tərəfindən idarə oluna bilər.
* Təchizatçılar: VMware, OpenStack.
* Üstünlükləri: Yüksək təhlükəsizlik və nəzarət, fərdiləşdirilmiş konfiqurasiya.

3. Hibrid Bulud (Hybrid Cloud):

* Xüsusiyyətləri: İctimai və özəl buludların birləşməsidir. Verilənlər və tətbiqlər bu iki mühit arasında paylaşdırıla bilər.
* Üstünlükləri: Daha çox çeviklik, həssas məlumatların özəl buludda saxlanılması, ictimai buludun genişləndirilməsi imkanları.

4. Cəmiyyət Buludu (Community Cloud):

* Xüsusiyyətləri: Müəyyən bir icma tərəfindən istifadə olunan bulud mühitidir. Bu icma ümumi məqsədlərə və tələblərə malik ola bilər.
* Təchizatçılar: Elmi tədqiqat müəssisələri, dövlət qurumları.
* Üstünlükləri: Xüsusi icma tələblərinə uyğunlaşdırılmışdır, resurslar ortaq istifadə oluna bilər.

Bulud Xidmət Modelləri:

1. İnfrastruktur xidmətləri (IaaS - Infrastructure as a Service):

* Xüsusiyyətləri: İstifadəçilər virtuallaşdırılmış kompüter resurslarına (serverlər, saxlama, şəbəkə) çıxış əldə edirlər.
* Təchizatçılar: AWS EC2, Google Compute Engine, Microsoft Azure.
* Üstünlükləri: İstifadəçilər öz tətbiqlərini qura və idarə edə bilərlər, yüksək səviyyədə nəzarət və çeviklik.

2. Platforma xidmətləri (PaaS - Platform as a Service):

* Xüsusiyyətləri: İstifadəçilərə tətbiqləri inkişaf etdirmək, test etmək və yerləşdirmək üçün platformalar təmin edilir.
* Təchizatçılar: Google App Engine, Microsoft Azure App Services, Heroku.
* Üstünlükləri: İnfrastrukturun idarə edilməsi haqqında narahat olmağa ehtiyac yoxdur, inkişaf prosesini sürətləndirir.

3. Proqram təminatı xidmətləri (SaaS - Software as a Service):

* Xüsusiyyətləri: İstifadəçilər internet vasitəsilə tam işləyən tətbiqlərə çıxış əldə edirlər.
* Təchizatçılar: Google Workspace, Microsoft Office 365, Salesforce.
* Üstünlükləri: Asan istifadə və idarəetmə, quraşdırma və yenilənmə ilə bağlı heç bir narahatlıq yoxdur.

Bulud Texnologiyalarının Üstünlükləri

1. Maliyyə Effektivliyi: İstifadəçilər yalnız istifadə etdikləri resurslar üçün ödəniş edirlər. Əlverişli qiymətlər, infrastruktur investisiyalarını minimuma endirir.

2. Elastiklik və Ölçəklənmə: İstifadəçilər tələb artdıqca və ya azaldıqca resursları asanlıqla genişləndirə və ya daralda bilərlər.

3. Müxtəlif Yerlərdən Giriş: İnternet bağlantısı olan hər yerdən xidmətlərə çıxış əldə etmək imkanı.

4. Avtomatik Yeniləmələr və Texniki Xidmətlər: Təchizatçılar sistemləri avtomatik olaraq yeniləyir və texniki xidmət göstərirlər.

5. Yüksək Performans: Bulud xidmətləri yüksək performanslı və etibarlı infrastruktur üzərində çalışır.

Bulud Texnologiyalarının Çətinlikləri

1. Təhlükəsizlik və Məxfilik: Həssas məlumatların qorunması və təhlükəsizlik təhdidlərinə qarşı müdafiə.

2. Asılılıq (Vendor Lock-in): Bir təchizatçıya bağlılıq, başqa təchizatçıya keçid edərkən yaranan çətinliklər.

3. Şəbəkə Performansı: İnternet bağlantısının keyfiyyətinə və sürətinə asılılıq.

## 69. Bulud texnologiyalarının strukturu və iş prinsipi

Bulud texnologiyaları (Cloud Computing) məlumatların saxlanılması, işlənməsi və idarə edilməsi üçün internet üzərindən xidmətlərin təmin edilməsini nəzərdə tutur. Bu texnologiyalar fərdi istifadəçilərdən tutmuş, iri müəssisələrə qədər geniş bir istifadəçi spektrinə xidmət edir və ənənəvi yerli serverlərə və fərdi cihazlara ehtiyac olmadan məlumatların saxlanılmasına və işlənməsinə imkan verir.

Bulud Texnologiyalarının Əsas Növləri:

1. İctimai Bulud (Public Cloud):

* Xüsusiyyətləri: Xidmətlər və resurslar ictimaiyyətə açıq şəkildə təmin olunur. İstifadəçilər xidmətləri internet üzərindən istifadə edirlər.
* Təchizatçılar: Amazon Web Services (AWS), Microsoft Azure, Google Cloud Platform (GCP).
* Üstünlükləri: Elastiklik, genişləndirilə bilən resurslar, az ilkin investisiya.

2. Özəl Bulud (Private Cloud):

* Xüsusiyyətləri: Resurslar və xidmətlər yalnız bir təşkilat tərəfindən istifadə olunur. Fiziki olaraq təşkilatın özündə və ya üçüncü tərəf tərəfindən idarə oluna bilər.
* Təchizatçılar: VMware, OpenStack.
* Üstünlükləri: Yüksək təhlükəsizlik və nəzarət, fərdiləşdirilmiş konfiqurasiya.

3. Hibrid Bulud (Hybrid Cloud):

* Xüsusiyyətləri: İctimai və özəl buludların birləşməsidir. Verilənlər və tətbiqlər bu iki mühit arasında paylaşdırıla bilər.
* Üstünlükləri: Daha çox çeviklik, həssas məlumatların özəl buludda saxlanılması, ictimai buludun genişləndirilməsi imkanları.

4. Cəmiyyət Buludu (Community Cloud):

* Xüsusiyyətləri: Müəyyən bir icma tərəfindən istifadə olunan bulud mühitidir. Bu icma ümumi məqsədlərə və tələblərə malik ola bilər.
* Təchizatçılar: Elmi tədqiqat müəssisələri, dövlət qurumları.
* Üstünlükləri: Xüsusi icma tələblərinə uyğunlaşdırılmışdır, resurslar ortaq istifadə oluna bilər.

Bulud Xidmət Modelləri:

1. İnfrastruktur xidmətləri (IaaS - Infrastructure as a Service):

* Xüsusiyyətləri: İstifadəçilər virtuallaşdırılmış kompüter resurslarına (serverlər, saxlama, şəbəkə) çıxış əldə edirlər.
* Təchizatçılar: AWS EC2, Google Compute Engine, Microsoft Azure.
* Üstünlükləri: İstifadəçilər öz tətbiqlərini qura və idarə edə bilərlər, yüksək səviyyədə nəzarət və çeviklik.

2. Platforma xidmətləri (PaaS - Platform as a Service):

* Xüsusiyyətləri: İstifadəçilərə tətbiqləri inkişaf etdirmək, test etmək və yerləşdirmək üçün platformalar təmin edilir.
* Təchizatçılar: Google App Engine, Microsoft Azure App Services, Heroku.
* Üstünlükləri: İnfrastrukturun idarə edilməsi haqqında narahat olmağa ehtiyac yoxdur, inkişaf prosesini sürətləndirir.

3. Proqram təminatı xidmətləri (SaaS - Software as a Service):

* Xüsusiyyətləri: İstifadəçilər internet vasitəsilə tam işləyən tətbiqlərə çıxış əldə edirlər.
* Təchizatçılar: Google Workspace, Microsoft Office 365, Salesforce.
* Üstünlükləri: Asan istifadə və idarəetmə, quraşdırma və yenilənmə ilə bağlı heç bir narahatlıq yoxdur.

Bulud Texnologiyalarının Üstünlükləri

1. Maliyyə Effektivliyi: İstifadəçilər yalnız istifadə etdikləri resurslar üçün ödəniş edirlər. Əlverişli qiymətlər, infrastruktur investisiyalarını minimuma endirir.

2. Elastiklik və Ölçəklənmə: İstifadəçilər tələb artdıqca və ya azaldıqca resursları asanlıqla genişləndirə və ya daralda bilərlər.

3. Müxtəlif Yerlərdən Giriş: İnternet bağlantısı olan hər yerdən xidmətlərə çıxış əldə etmək imkanı.

4. Avtomatik Yeniləmələr və Texniki Xidmətlər: Təchizatçılar sistemləri avtomatik olaraq yeniləyir və texniki xidmət göstərirlər.

5. Yüksək Performans: Bulud xidmətləri yüksək performanslı və etibarlı infrastruktur üzərində çalışır.

Bulud Texnologiyalarının Çətinlikləri

1. Təhlükəsizlik və Məxfilik: Həssas məlumatların qorunması və təhlükəsizlik təhdidlərinə qarşı müdafiə.

2. Asılılıq (Vendor Lock-in): Bir təchizatçıya bağlılıq, başqa təchizatçıya keçid edərkən yaranan çətinliklər.

3. Şəbəkə Performansı: İnternet bağlantısının keyfiyyətinə və sürətinə asılılıq.

## 70. SHA-1, SHA-2 və SHA-256 Hash Alqoritmləri Arasındakı Fərqləri izah edin

SHA (Secure Hash Algorithm) ailəsinə aid olan SHA-1, SHA-2 və SHA-256 alqoritmləri kriptoqrafik hash funksiyalarıdır. Onlar məlumatları müəyyən bir uzunluğa qədər olan hash dəyərlərinə çevirir və məlumatların təhlükəsizliyini təmin etmək üçün geniş istifadə olunur.

**SHA-1 (Secure Hash Algorithm 1)** - SHA alqoritmlərinin ilk versiyalarından biridir və 1995-ci ildə NIST (National Institute of Standards and Technology) tərəfindən standart olaraq qəbul edilmişdir.

* Hash Dəyərinin Uzunluğu: 160 bit (20 bayt).
* Təhlükəsizlik: SHA-1 hal-hazırda kriptoqrafik cəhətdən etibarlı sayılmır. 2005-ci ildə zəiflikləri aşkar edilmiş və 2017-ci ildə Google və CWI Amsterdam tərəfindən konkret olaraq toqquşma (collision) nümayiş etdirilmişdir.
* İstifadə Sahələri: Bir zamanlar SSL sertifikatları, rəqəmsal imzalar və şifrələmə sistemlərində geniş istifadə olunurdu. Hazırda yeni sistemlərdə istifadə olunmaması tövsiyə edilir.

**SHA-2** - SHA-1-də aşkar edilən zəiflikləri aradan qaldırmaq üçün 2001-ci ildə NIST tərəfindən təqdim edilmişdir. SHA-2 ailəsi bir neçə fərqli hash funksiyasını əhatə edir: SHA-224, SHA-256, SHA-384, və SHA-512.

* Hash Dəyərlərinin Uzunluğu: 224, 256, 384 və 512 bit.
* Təhlükəsizlik: SHA-2 ailəsi SHA-1-dən daha təhlükəsiz hesab olunur və indiyə qədər ciddi zəiflikləri aşkar edilməyib.
* İstifadə Sahələri: SSL/TLS sertifikatları, rəqəmsal imzalar, blokçeyn texnologiyaları və digər təhlükəsizlik tətbiqlərində geniş istifadə olunur.

**SHA-256** - SHA-2 ailəsinin bir hissəsi olaraq, xüsusi olaraq vurğulanmalı bir alqoritmdir.

* Hash Dəyərinin Uzunluğu: 256 bit (32 bayt).
* Təhlükəsizlik: SHA-256 hazırda çox etibarlı hesab olunur və geniş şəkildə istifadə olunur. 256 bitlik hash dəyəri onu toqquşmalara qarşı daha davamlı edir.
* İstifadə Sahələri: Kriptovalyutalar (məsələn, Bitcoin), SSL/TLS sertifikatları, rəqəmsal imzalar və daha çox. SHA-256-nın populyarlığı və etibarlılığı onu çox geniş tətbiq olunan hash alqoritmlərindən biri edir.

## 71. GitHub-da repository yaradın və lokal kompüterə klon edin.

Https: GitHub hesabımıza giririk. Soldakı paneldən “new” klikləyib yeni repository yaradırıq. Onun public və ya private olmasını seçirik. Daha sonra “readme” faylını olub-olmamasını, .gitignore faylı və lisenziya faylını əlavə edə bilərik. Repomuzda code bölməsindən onu lokal kompüterə kopyalamaq üçün https url-sini götürürük. Uyğun faylda sağ klikləyib GitBashHere deyirik. Daha sonra “git clone” və repo url-sini yazırıq. Repo clone edilir.

SSH: Əvvəlcə SSH açarını (SSH key pair) generasiya etməlisiniz.

**ssh-keygen -t rsa -b 4096 -C your\_email@example.com**

Githubda olan gmailinizlə eyni olmalıdır. Daha sonra sizə şifrə yerləşən faylın adresini təqdim edir. Repository-nin SSH URL-ni əldə edirik. Git Bash və ya Terminalda git clone komandasının istifadəsi: **git clone <repository\_SSH\_URL>** SSH açarlarının təsdiq edilməsi lazım gəlir. İlk dəfə klonlama zamanı SSH açarınızı təsdiq etməyə icazə veriləcək. Bu əsasən bir dəfəlik prosesdir.