**1.** PHP nədir? PHP-də data type-lər hansılardır?

**2.** PHP nədir? PHP-də variable və constant məntiqini izah edin.

**3.** PHP-də istifadə olunan "array"-lar haqqında məlumat verin . Arrayların növləri hansılardır?

**4.** PHPMyadmin haqqında məlumat verin, verilənlər bazası ilə PDO vasitəsilə əlaqə yaradılması məntiqini izah edin.

**5.** PDO nədir? Verilənlər bazası ilə PDO vasitəsilə əlaqə yaradılması və verilənlər bazasına datanın PDO istifadəsi ilə insert edilməsi məntiqlərini izah edin.

**6.** Verilənlər bazasından datanın PDO istifadəsi ilə çəkilməsi və silinməsi məntiqlərini izah edin.

**7**. PDO nədir? PDO ilə update əməliyyatını izah edin.

**8.** PHP-də AJAX və Jquery məntiqilərini izah edin.

**9.** PHP-də İf-Else, Switch-case, Ternary Operatorları haqqında məlumat verin

**10.** PHP-də “File Handling” məntiqini izah edin. touch(), fopen(), fclose(), fwrite() funksiyalarını izah edin.

**11**. PHP-də “File Handling” məntiqini izah edin. is\_file(), file\_exist(), feof(), readfile(), funksiyalarını izah edin.

**12.** OOP-də trait nədir? \_\_construct, \_\_destruct funksiyaları haqqında məlumat verin.

**13**. PHP-də Funksiyalar, “call by value” və “call by reference” məntiqini izah edin.

**14.** Super Global dəyişənlər, Session və Cookie.

**15**. Super Global dəyişənlər, POST və GET.

**16.** String funksiyalar haqqında məlumat verin, nümunələr göstərin.

**17**. Riyazi funksiyalar haqqında məlumat verin, nümunələr göstərin.

**18.** PHP-də şərt operatorları və dövrlər hansılardır?

**19.** MySqli nədir? Verilənlər bazası ilə MySqli(Procedural) vasitəsilə əlaqə yaradılması və verilənlər bazasına datanın MySqli istifadəsi ilə insert edilməsi məntiqlərini izah edin.

**20.** Verilənlər bazasından datanın MySqli (Procedural) istifadəsi ilə çəkilməsi və silinməsi məntiqlərini izah edin.

**21.** Mysqli nədir? MySqli (Procedural) ilə update əməliyyatını izah edin.

**22.** PHP-də operatorlar hansılardır? PHP-də date funksiyaları haqqında məlumat verin.

**23**. Exception, error, try, catch, throw məntiqlərini izah edin.

**24.** PHP-OOP-də static, parent, self məntiqlərini izah edin.

**25.** PHP-OOP-də public, private, protected məntiqlərini izah edin.

**26**. PHP-də funksiyaların növləri hansılardır? Void və return funksiyalar haqqında məlumat verin.

**27.** PHP-də funksiyalar haqqında məlumat verin. func\_num\_args(), func\_get\_arg(), function\_exists() funksiyaları haqqında məlumat verin.

**28.** PHP-də arraylar haqqında məlumat verin. array\_push(), array\_pop(), array\_shift(), array\_unshift() funksiyaları haqqında yazın.

**29.** HTML Form-u yazın, bu form-dan gələn datanı filterləmək üçün “validate()” adli funksiya yazin.

**30.** “text.txt” filemizin olub olmadığını yoxlayın, əgər yoxdursa, yaradın.

**31.** Type-si checkbox olan HTML Form-u yazın, PHP-də seçilən value-ni və ya value-ləri ekrana çap edin.

**32.** Type-si radio olan HTML Form-u yazın, PHP-də seçilən value-ni ekrana çap edin.

**33.** Type-si select olan HTML Form-u yazın, PHP-də seçilən value-ni ekrana çap edin.

**34**. PHPMailer vasitəsi ilə mail göndərilən zaman istifadə olunan funksiyaları yazın və izah edin.

**35.** ‘text.txt’ filemiz var. Bu fileni acib, filenin en sonuna “text elave edildi” yazisi yazin.

**36**. ‘text.txt’ filemiz var. Bu fileni acib, filede evvelden olan melumatlar silib, “text elave edildi” yazisi yazin.

**37.** Type-si color olan HTML Form-u yazın, inputun HEX kodunu ekrana çıxarın.

**38.** Verilmiş table-də id-si 2 olan sətri MySqli(Procedural) vasitəsilə silin. Data silinərsə, "data silindi", əks halda "data silinmədi" yazısı çap edilsin (table adı: usersinfo, database adı: users).



**39.** Verilmiş table-də id-si 1 və 3 olan sətrləri MySqli(Procedural) vasitəsilə silin. Data silinərsə, "data silindi", əks halda "data silinmədi" yazısı çap edilsin (table adı: usersinfo, database adı: users).

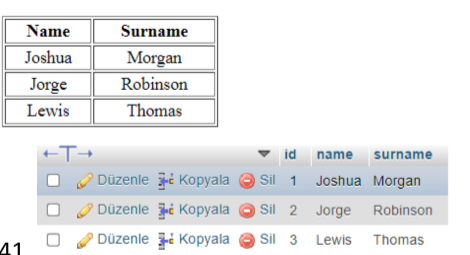


**40.** Verilmiş table-yə $name = "Lewis", $surname = "Hamilton" dəyərlərini PDO vasitəsilə insert edin. Data əlavə edilərsə, "əlavə edildi", əks halda "əlavə edilmədi"

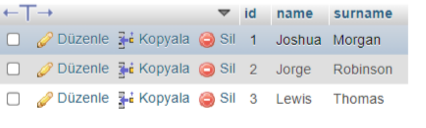
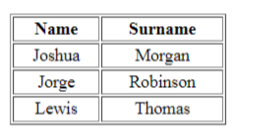
yazısı çap edilsin (table adı: usersinfo, database adı: users).



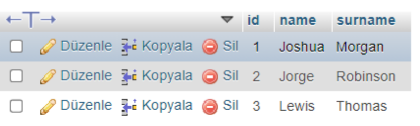
**41.** (database adı: users, table adı: usersinfo) Verilmiş table-də olan dataları PDO vasitəsilə HTML Table-yə yazdırın və aşağıdakı kimi çap edin.



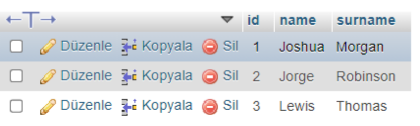
**42.** (database adı: users, table adı: usersinfo) Verilmiş table-də olan dataları MySqli(Procedural) vasitəsilə HTML Table-yə yazdırın və aşağıdakı kimi çap edin.

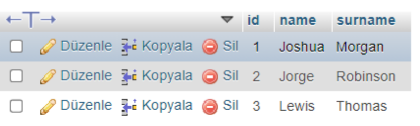
**43.** Verilmiş table-də id-si 2 olan sətri PDO vasitəsilə silin. Data silinərsə, "data silindi", əks halda "data silinmədi"

****

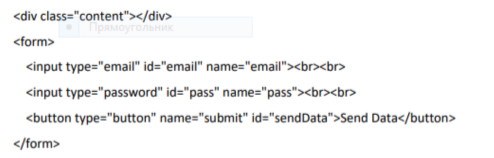
**44.** Verilmiş table-də id-si 2 olan sətri PDO vasitəsilə update edin, name=”Adrian” və surname=”Harris” olsun. Data update olarsa, "data update oldu", əks halda "data update olmadı" yazısı çap edilsin (table adı: usersinfo, database adı: users).

****

**45.** Verilmiş table-də id-si 2 olan sətri MySqli(Procedural) vasitəsilə update edin, name=”Adrian” və surname=”Harris” olsun. Data update olarsa, "data update oldu", əks halda "data update olmadı" yazısı çap edilsin (table adı: usersinfo, database adı: users).

****

**46.** Button-a click edilən zaman verilmiş form-dakı dataları "proses.php" səhifəsinə Jquery və AJAX istifadə edərək göndərin

****

**47.** $adminEmail = someone@mail.ru; $adminParol ="12345”; adli iki deyisenimiz var, formdan daxil olan email ve parolun, bu email ve parolla ust uste dusmesini

yoxlayin. Eyni olarsa ekrana “ugurlu”, ferqli olarsa, “yanlis” yazisi cixsin.

**48.** “text.txt” filesini acin ve ilk setrini və ilk simvolunu oxuyun.

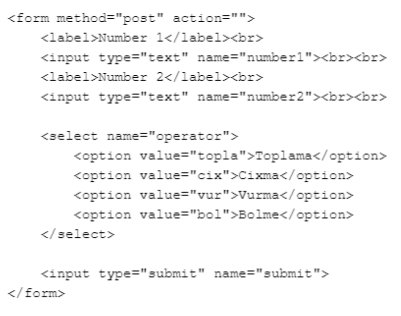
**49.** Formda inputdan soz daxil edilir ve hemin soz Mysqlde tablede “surname” sütunu üzrə axtarilir ve netice olaraq cixsin (simple search) (PDO)

****

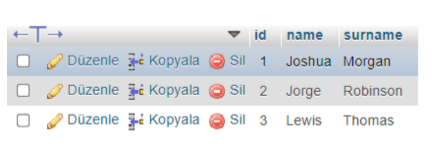
**50**. Formda inputdan soz daxil edilir ve hemin soz Mysqlde tablede “surname” sütunu üzrə axtarilir ve netice olaraq cixsin (simple search) (MySqli (Procedural)).

****

**51.** “Simple Calculator” classı yazın. Switch ve case istifadə edin.



**52.** Verilmiş table-yə $name = "Lewis", $surname = "Hamilton" dəyərlərini MySqli(Procedural) vasitəsilə insert edin. Data əlavə edilərsə, "əlavə edildi", əks halda "əlavə edilmədi" yazısı çap edilsin (table adı: usersinfo, database adı: users).



***CAVABLAR***

***1. PHP nədir? PHP-də data type-lər hansılardır?***

PHP (Hypertext Preprocessor) bir skript dilidir və əsasən server tərəfində işlədilir. HTML-ə kömək edən dinamik veb səhifələri və veb tətbiqləri yaratmaq üçün istifadə olunur. PHP, veb proqramlaşdırma üçün nəzərdə tutulmuşdur və veb səhifələrinə məlumat əlavə etmək, oxumaq və redaktə etmək, faylları yükləmək və sair kimi əməliyyatları həyata keçirmək üçün işlənilir.

PHP, həm də dəyişənlərin tipini ***avtomatik*** olaraq təyin edən dinamik bir dildir, bu da proqramlaşdırma prosesini sürətləndirir, lakin eyni zamanda ***tip xətalarına*** səbəb ola bilər.

* **Scalar (Skalyar)Tiplər:**

**Bool (Boolean):** *true və false* dəyərlərini müəyyən edir

**Int (Integer)**: Tam ədəd tipli dəyişənlər üçün istifadə olunur.

**Float və ya double:** *Onluq ədədləri özündə saxlayır*

**String:** Mətn, hərf və ya sözləri özündə saxlayır. Tək və ya qoşa dırnaqda verilir

* **Compound (Birləşik) Tiplər:**

**Array:** Birdən çox dəyəri bir sıra kimi saxlayır

**Object:** *siniflərin instanslarıdır*

* **Special (Xüsusi) Tiplər:**

**Resource**: *Xarici bir resursa* istinad edən dəyişəndir.

**NULL:** Dəyişənə hələ bir dəyər təyin edilmədiyi zaman və ya null dəyəri verildiyi zaman bu tipə malik olur.

***2. PHP nədir? PHP-də variable və constant məntiqini izah edin.***

PHP (Hypertext Preprocessor) bir skript dilidir və əsasən server tərəfində işlədilir. HTML-ə kömək edən dinamik veb səhifələri və veb tətbiqləri yaratmaq üçün istifadə olunur. PHP, veb proqramlaşdırma üçün nəzərdə tutulmuşdur və veb səhifələrinə məlumat əlavə etmək, oxumaq və redaktə etmək, faylları yükləmək və sair kimi əməliyyatları həyata keçirmək üçün işlənilir.

PHP, həm də dəyişənlərin tipini ***avtomatik*** olaraq təyin edən dinamik bir dildir, bu da proqramlaşdırma prosesini sürətləndirir, lakin eyni zamanda ***tip xətalarına*** səbəb ola bilər.

**Variable (Dəyişənlər):** Dəyişənlər məlumatları saxlamaq üçün istifadə olunan adlandırılmış yaddaş bölgələridir. PHP-də dəyişənlər dollardan ($) sonra adlandırılırlar və böyük kiçik hərfləri ayırmaq üçün case-sensitive-dir. Dəyişənlərin dəyərləri dinamik olaraq dəyişə bilir və fərqli data tiplərini (string, integer, float, boolean, və s.) qəbul edə bilərlər.

**Constant (Sabitlər):** Sabitlər dəyişənlərlə oxşar olaraq məlumatları saxlamaq üçün istifadə olunan adlandırılmış yaddaş bölgələridir. Əsas fərq, Bir dəfə təyin edildikdən sonra sabitlərin dəyəri dəyişdirilə bilməz və sabitlər **$** (dollar) işarəsi ilə başlamır. Sabitlər define() funksiyası ilə təyin olunur və yenə də case-sensitive-dir. Sabitlərin əhatə sahəsi globaldir. Yəni, sabit bir dəfə təyin edildikdən sonra *proqramın hər yerindən istifadə* edilə bilər.

**define("PI", 3.14);**

**echo Pİ;**

***3. PHP-də istifadə olunan "array"-lar haqqında məlumat verin . Arrayların növləri hansılardır?***

PHP-də “array” adlanan xüsusi dəyişənlər birdən çox dəyəri bir dəyişəndə saxlamağa imkan verir. Arraylar məlumatları qruplaşdırmağa və idarə etməyə imkan verən dəyişənlərdir.

**1.Indexed Arrays (Indeksli Arraylar):** *rəqəmsal indekslərlə* işləyir. İndeks *avtomatik* olaraq təyin edilir *və ya proqramçı* tərəfindən təyin edilə bilər.

**$colors = array("red", "green", "blue");**

burada "red" 0, "green" 1 və "blue" 2 nömrəli indeksə malikdir.

**2. Associative Arrays (Assosiativ Arraylar):** Bu arraylar hər bir dəyəri bir açar (key) ilə əlaqələndirir. Açarlar adətən **string** tipində olur.

**$ages = array("Peter" => 35, "Ben" => 37, "Joe" => 43);**

**3. Multidimensional Arrays (Çoxölçülü Arraylar):** Bu arraylar *bir və ya daha çox arrayı özündə saxlayan* arraylardır.

**$contacts = array(**

**array(**

**"name" => "Peter Parker",**

**"email" => "peterparker@mail.com",**

**),**

**array(**

**"name" => "Harry Potter",**

**"email" => "harrypotter@mail.com",**

**)**

**);**

Bu nümunədə hər bir kontakt bir **associativ array** təşkil edir və bu arraylar bir **indeksli arrayın** daxilində yerləşdirilib.

***4. PHPMyadmin haqqında məlumat verin, verilənlər bazası ilə PDO vasitəsilə əlaqə yaradılması məntiqini izah edin.***

PHPMyAdmin verilənlər bazalarını idarə etmək üçün istifadə edilən açıq mənbəli bir veb tətbiqdir. Bu alət PHP ilə yazılmışdırvə verilənlər bazasını idarə etmək, cədvəllər yaratmaq, dəyişdirmək, silmək, SQL sorğuları icra etmək, istifadəçi hüquqlarını idarə etmək və digər verilənlər bazası idarəçilik funksiyalarını həyata keçirməyə imkan verir.

**PDO (PHP Data Objects)** PHP-də verilənlər bazaları ilə işləmək üçün istifadə edilən bir obyekt-yönümlü kitabxanadır.

**PDO ilə Əlaqənin Qurulması :**

**1.PDO Obyektinin Yaradılması:** İlk addım, yeni bir PDO obyekti yaratmaqdır. Bu, verilənlər bazası serverinə əlaqə qurmaq üçün lazım olan məlumatları (host, verilənlər bazası adı, istifadəçi adı, parol) daxil edərək edilir.

**$dsn = 'mysql:host=localhost;dbname=testdb';**

**$user = 'username';**

**$password = 'password';**

**try {**

**$pdo = new PDO($dsn,$user, $password);**

**} catch (PDOException $e) {**

**echo 'Connection failed: ' . $e->getMessage();**

**}**

**2. Sorğuların İcra Edilməsi:** Əlaqə qurulduqdan sonra, **query(), prepare(), və execute()** metodları ilə SQL sorğuları icra edilə bilər. **prepare()** metodunda, SQL sorğusu hazırlanır, **execute()** ilə isə bu sorğu icra edilir. **fetch()** metodunda isə nəticələr çıxarılır.

**$stmnt = $pdo->prepare("SELECT \* FROM users WHERE id = ?");**

**$stmnt->execute([1]);**

**$user = $stmnt->fetch();**

**3. Əlaqənin Bağlanması:** PDO obyektinin işi bitdikdə, avtomatik olaraq əlaqə bağlanır. Lakin, əlaqəni əl ilə də bağlamaq mümkündür, bunun üçün sadəcə

**$pdo = null**; ifadəsini işlətmək kifayətdir.

***5. PDO nədir? Verilənlər bazası ilə PDO vasitəsilə əlaqə yaradılması və verilənlər bazasına datanın PDO istifadəsi ilə insert edilməsi məntiqlərini izah edin.***

PDO (PHP Data Objects) PHP-də verilənlər bazası əməliyyatları üçün genişlənmiş və təhlükəsiz bir interfeys təşkil edən bir PHP moduludur. PDO, müxtəlif verilən bazaları ilə işləməyə imkan verir və bir neçə təhlükəsizlik avantajına malikdir. PDO-nun əsas məqsədi, verilən bazası əməliyyatlarını müstəqil etmək və müxtəlif verilən bazaları ilə tərəqqi etməkdir.

**PDO ilə Verilənlər Bazası Əlaqəsinin Qurulması:**

**$dsn = 'mysql:host=localhost;dbname=testdb';**

**$user = 'username';**

**$password = 'password';**

**try {**

**$pdo = new PDO($dsn, $user, $password);**

**} catch (PDOException $e) {**

**echo 'Connection failed: ' . $e->getMessage();**

**}**

**Verilənlər Bazasına Data İnsert Etmək:**

**$sql = "INSERT INTO users (name, email) VALUES (:name, :email)";**

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

**$stmt->execute ([**

**'name' => 'John Doe',**

**'email' => '**[**johndoe@example.com**](mailto:johndoe@example.com)**'**

**]);**

***6. Verilənlər bazasından datanın PDO istifadəsi ilə çəkilməsi və silinməsi məntiqlərini izah edin.***

Aşağıda, bir "users" cədvəlindən bütün istifadəçi məlumatlarını çəkmək üçün bir PDO misalı verilmişdir:

**try {**

**$sql = "SELECT \* FROM users";**

**$stmt = $conn->query($sql);**

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

**echo "Ad: " . $row['name'] . ", Yaş: " . $row['age'] . ", Email: " . $row['email'] . "<br>";**

**}**

**} catch (PDOException $e) {**

**echo "Məlumat çəkmək mümkün olmadı: " . $e->getMessage();**

**}**

**Verilənlər Bazasından Datanın PDO İstifadəsi ilə Silinməsi:**

**$nameToDelete = "John";**

**try {**

**$sql = "DELETE FROM users WHERE name = :name";**

**$stmt = $conn -> prepare($sql);**

**$stmt->bindParam(':name', $nameToDelete);**

**$stmt->execute();**

**echo "Məlumat uğurla silindi.";**

**} catch (PDOException $e) {**

**echo "Silinmək mümkün olmadı: " . $e -> getMessage();**

**}**

***7. PDO nədir? PDO ilə update əməliyyatını izah edin.***

PDO (PHP Data Objects) PHP-də verilənlər bazası əməliyyatları üçün genişlənmiş və təhlükəsiz bir interfeys təşkil edən bir PHP moduludur. PDO, müxtəlif verilən bazaları ilə işləməyə imkan verir və bir neçə təhlükəsizlik avantajına malikdir. PDO-nun əsas məqsədi, verilən bazası əməliyyatlarını müstəqil etmək və müxtəlif verilən bazaları ilə tərəqqi etməkdir.

**$idToUpdate = 1; // Yenilənəcək istifadəçinin ID-ni göstərin**

**$newName = "Jane";**

**$newAge = 30;**

**try {**

**$sql = "UPDATE users SET name = :name, age = :age WHERE id = :id";**

**$stmt = $conn -> prepare($sql);**

**$stmt -> bindParam(':name', $newName);**

**$stmt -> bindParam(':age', $newAge);**

**$stmt -> bindParam(':id', $idToUpdate);**

**$stmt -> execute();**

**echo "Məlumat uğurla yeniləndi.";**

**} catch (PDOException $e) {**

**echo "Yeniləmək mümkün olmadı: " . $e -> getMessage();**

**}**

Bu kod nümunəsində **UPDATE** əməliyyatını icra etmək üçün SQL sorğusu hazırlanır. **bindParam** funksiyası ilə SQL sorğusu üçün müstəqil dəyişənlər təyin edilir və sonra **execute** funksiyası istifadə olunaraq sorğu yerinə yetirilir.

***8. PHP-də AJAX və Jquery məntiqilərini izah edin.***

Ajax(Asynchronous JavaScript and XML) bir çox proqramlaşdırma dili ilə uyğun işləyən bir texnologiyadır. Məqsəd İstifadəçinin bütün səhifəni yenidən yükləməsinə ehtiyac olmadan dinamik olaraq yalnız lazımi məlumatları ekrana çıxarmaq və ya məlumatları serverə göndərməkdir.

**Ajax kodunun əsas quruluşu və istifadəsi:**

**$.ajax({ //ajax kodunun başlanğıcı**

**type: “POST”, // göndərmə növü GET və ya POST ola bilər.**

**url : “save.php”, //məlumatların saxlanacağı səhifənin linki**

**data : ad, // göndəriləcək məlumatlar**

**success : function (nəticə) {**

**if(nəticə== “uğurlu”){**

**alert (“Ad əlavə edildi”);**

**} else{**

**alert(“Ad əlavə etmək alınmadı”);**

**} } });**

JQuery JavaScript və HTML dokumentlarının asanlıqla manipulyasiya edilməsi və həmçinin AJAX tətbiqlərinin asanlıqla yerinə yetirilməsi üçün yaradılmış bir JavaScript kitabxanasıdır. JQuery müstəqil bir proqramlaşdırma dilidir və müxtəlif brauzerlərdə eyni şəkildə işləyir.

**$.ajax({**

**url: "server.php",**

**type: "GET",**

**success: function (data) {**

**// Serverdən gələn məlumatlar burada işlənir**

**console.log(data);**

**},**

**error: function (xhr, status, error) {**

**console.error(status + ": " + error);**

**}**

**});**

***9. PHP-də İf-Else, Switch-case, Ternary Operatorları haqqında məlumat verin***

IF-ELSE şərtləri, şərtin doğru və ya yalnış olmasına əsasən kodun müxtəlif hissələrinin icra edilməsini təmin edir. Əgər təyin edilən şərt doğru olarsa if hissəsi icra olunur, əks halda else hissəsi icra olunur.

**$age = 20;**

**if ($age >= 18) {**

**echo "Siz yetkin yaşdasınız.";**

**} else {**

**echo "Siz hələ yetkin deyilsiniz.";**

**}**

Switch-case bir çox `if-else` ifadələrinin əvəzinə istifadə edilə bilər, xüsusilə bir dəyişənin müxtəlif mümkün dəyərlərini yoxlamaq lazım olduğunda istifadə olunur.

**$day = "Cümə";**

**($day) {**

**case "Bazar":**

**echo "Bu gün Bazardır.";**

**break;**

**case "Cümə":**

**echo "Bu gün Cümədir.";**

**break;**

**default:**

**echo "Bu gün həftənin digər günlərindən biridir.";**

**}**

Ternary operatoru ‘if-else` ifadəsinin qısa formasıdır və adətən sadə şərtlər üçün istifadə olunur.

**$age = 20;**

**$message = ($age >= 18) ? "Siz yetkin yaşdasınız." : "Siz hələ yetkin deyilsiniz.";**

**echo $message;**

***10. PHP-də “File Handling” məntiqini izah edin. touch(), fopen(), fclose(), fwrite() funksiyalarını izah edin.***

PHP'də "File Handling" (Fayl İdarəetmə) məntiqi faylların oxunması, yazılması, yaradılması və silinməsi kimi əməliyyatları həyata keçirmək üçün istifadə olunur.

**fopen()** - fayla oxuma, yazma və ya hər ikisi üçün giriş imkanı verir.

**$file = fopen("example.txt", "r");**

**fwrite()** - Fayla Yazmaq

**fwrite($file, "Yeni məlumat");**

**fclose()(**Fayl Bağlamaq**) -** Bu, sistem resurslarını sərbəst buraxır və məlumat itkisinin qarşısını alır.

***11. PHP-də “File Handling” məntiqini izah edin. is\_file(), file\_exist(), feof(), readfile(), funksiyalarını izah edin.***

PHP'də "File Handling" (Fayl İdarəetmə) məntiqi faylların oxunması, yazılması, yaradılması və silinməsi kimi əməliyyatları həyata keçirmək üçün istifadə olunur.

**file\_exist() -** faylların mövcudluğunu yoxlayır

**if (file\_exists("example.txt"));**

**feof() – FİLE END OF FİLE –** php də faylın sonuna çatılıb-çatılmadığını yoxlayan funksiyadır.

**readfile() –** php də faylı oxuyan və çıxış buferinə yazan funksiyadır.

**$filename = “example.txt”;**

**Readfile($filename);**

***12. OOP-də trait nədir? \_\_construct, \_\_destruct funksiyaları haqqında məlumat verin.***

Obyekt-yönümlü proqramlaşdırmada PHP, interfeyslər və klassların bəzi məhdudiyyətləri üzərində işləmək üçün `trait`ləri təqdim edir. Bir trait, metod və xüsusiyyətlər toplusudur ki, bu metod və xüsusiyyətlər bir və ya daha çox klassda istifadə edilə bilər. Traitlər kodun yenidən istifadəsini asanlaşdırır və kodun təkrarını azaldır. Bir klass birdən çox traiti istifadə edə bilər. Traitlər klass iyerarxiyasının hissəsi olmadan kodun paylaşılmasına imkan verir.

**`\_\_construct` Metodu:** bir klassın obyekti yaradıldıqda avtomatik olaraq çağırılan metoddur. Başlanğıc dəyərlərin təyin edilməsi və başlanğıc başlatma əməliyyatları üçün istifadə olunur. Bir klassın hər obyekti üçün yalnız bir dəfə işə salınır.

**class Product {**

**public function \_\_construct() {**

**echo "Product created";**

**}**

**}**

**$product = new Product();**

**`\_\_destruct` Metodu:** bir obyektin məhv edilməzdən əvvəl avtomatik olaraq çağırılan metoddur. Obyektin məhv edilməsi zamanı təmizləmə əməliyyatları üçün istifadə edilir. PHP skripti işini bitirdikdə və ya obyekt üçün məhdudiyyətlər aradan qalxdıqda destruktor çağırılır.

**class Product {**

**public function \_\_destruct() {**

**echo "Product destroyed";**

**}**

**}**

**$product = new Product();**

**unset($product);**

***13. PHP-də Funksiyalar, “call by value” və “call by reference” məntiqini izah edin***

PHP-də funksiya, bir və ya daha çox işi yerinə yetirən kod blokudur. Funksiyalar kodun təkrar istifadəsini asanlaşdırır və proqramın oxunaqlılığını artırır.

**function greet($name) {**

**return "Salam, $name!";**

**}**

**echo greet("Saida");**

PHP-də "call by value" və "call by reference" məntiqləri, dəyişənlərlə işlərkən necə davranmalı olduğunuzu təyin etmək üçün vacibdir. **"Call by value"** daha təhlükəsiz bir yanaşma təqdim edir, çünki orijinal dəyişənlər dəyişməz qalır. **"Call by reference"** isə, xüsusilə böyük məlumat strukturları ilə işləyərkən, yaddaş istifadəsini və performansı yaxşılaşdıra bilər, lakin dəyişənləri dəyişdirmə potensialını artırır.

**Call by Value** dəyişənin dəyərinin funksiyanın parametrinə kopyalandığı məntiqdir. Bu məntiqdə, funksiya daxilində edilən dəyişikliklər çağırılan dəyişən üzərində təsir etmir.

**function addFive($number) {**

**$number += 5;**

**}**

**$myNumber = 10;**

**addFive($myNumber);**

**echo $myNumber;**

**Call by Reference** dəyişənin özünün (yəni onun yaddaşdakı ünvanının) funksiyanın parametrinə ötürüldüyü məntiqdir. Bu halda, funksiya daxilində edilən dəyişikliklər orijinal dəyişəni də dəyişdirir. Bu məntiqi tətbiq etmək üçün parametrin önünə **`&`** işarəsini qoyuruq.

**function addFive(&$number) {**

**$number += 5;**

**}**

**$myNumber = 10;**

**addFive($myNumber);**

**echo $myNumber;**

***14. Super Global dəyişənlər, Session və Cookie.***

Super global dəyişənlər, PHP-də hər hansı bir funksiya və ya skript bloku ərzində istifadə edilə bilən özəl dəyişənlərdir. Bu dəyişənlər özəl funksionallıq təmin edirlər və proqramın müxtəlif hissələrindən çox rahat şəkildə əldə edilə bilərlər.

İlkin dəyərləri müəyyənləşdirilmiş super global dəyişənlərlə tanış olmaq üçün ən yaygın olanları: GET, POST, REQUEST, SESSİON, COOKİE, SERVER, FİLES.

**Session** istifadəçi məlumatlarını server tərəfində saxlamaq və onlarla əməliyyat aparmaq üçün istifadə olunan bir mexanizmdir. Session məlumatları, istifadəçinin bir veb saytı ilə əlaqə yaratdığı və bu saytda keçirildiyi müddət ərzində saxlanılır.

**session\_start(); //Session başladılması**

**$\_SESSION['user\_id'] = 1; //Session dəyişəni təyin etmək**

**$user\_id = $\_SESSION['user\_id']; //Session dəyişənini əldə etmək**

**Cookie** brauzerdə saxlanılan məlumatlardır və istifadəçinin bir veb saytı ilə əlaqə yaratdığı müddət ərzində saxlanılır.

**setcookie("username", "john\_doe", time() + 3600, "/");**

**$username = $\_COOKIE['username'];**

***15. Super Global dəyişənlər, POST və GET.***

Super global dəyişənlər, PHP-də hər hansı bir funksiya və ya skript bloku ərzində istifadə edilə bilən özəl dəyişənlərdir. Bu dəyişənlər özəl funksionallıq təmin edirlər və proqramın müxtəlif hissələrindən çox rahat şəkildə əldə edilə bilərlər.

İlkin dəyərləri müəyyənləşdirilmiş super global dəyişənlərlə tanış olmaq üçün ən yaygın olanları: GET, POST, REQUEST, SESSİON, COOKİE, SERVER, FİLES.

**Post super qlobal dəyişəni** HTML formlarından POST metodu ilə göndərilən məlumatları əldə etmək üçün istifadə olunur. Formda method = “ post” təyin edildiyi təqdirdə, form elementlərinin dəyərləri bu dəyişəndə saxlanılır.

**<form method="post" action="process.php">**

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

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

**<input type="submit" value="Göndər">**

**</form>**

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

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

**GET Super qlobal dəyişəni** URLüzərindən göndərilən məlumatları əldə etmək üçün istifadə olunur. URL-də sorğu sətri ilə göndərilmiş məlumatlar bu dəyişəndə saxlanılır.

**<a href="page.php?id=123&category=books">Kitablar</a>**

**$id = $\_GET['id'];**

**$category = $\_GET['category'];**

***16. String funksiyalar haqqında məlumat verin, nümunələr göstərin.***

**Strlen() –** mətnin uzunluğunu təyin edir

**$text = "Hello, World!";**

**$length = strlen($text);**

**echo "$length";**

**str\_replace() –** bir mətn daxilində müəyyən bir alt mətni tapmaq və başqa bir alt mətnlə əvəz etmək üçün istifadə edilir.

**$text = "Hello, World!";**

**$new\_text = str\_replace("World", "PHP", $text);**

**echo $new\_text;**

**strops() –** bir alt mətnin mətndəki indeksini tapır.

**$text = "Hello, World!";**

**$position = strpos($text, "World");**

**echo "$position";**

**strtolower() və strtoupper** – mətndəki bütün simvolları kiçik və ya böyük hərflərə çevirir.

**$text = "Hello, World!";**

**$lowercase = strtolower($text);**

**$uppercase = strtoupper($text);**

**echo "$lowercase <br>";**

**echo "$uppercase";**

***17. Riyazi funksiyalar haqqında məlumat verin, nümunələr göstərin***

**Abs() –** ədədin müsbət qiymətini qaytarır.

**$number = -5;**

**$absolute\_value = abs($number);**

**echo "$absolute\_value";**

**sqrt() -** ədədin kvadrat kökünü qaytarır.

**$number = 25;**

**$square\_root = sqrt($number);**

**echo "$square\_root";**

**pow() -** ədədin qüvvətini tapır

**$a= 2;**

**$b = 3;**

**$result = pow($a, $b);**

**echo " $result";**

**round() –** verilmiş ədədi yaxın tam ədədə yuvarlaqlaşdırır.

**$number = 3.8;**

**$rounded = round($number);**

**echo "$rounded";**

***18. PHP-də şərt operatorları və dövrlər hansılardır?***

**ŞƏRT OPERATORLARI:**

**1.IF-ELSE –** verilmiş şərtin dorğu və ya yanlış olduğunu müəyyən edir. Əgər if ilə verilən şərt yanlış olarsa else şərtindən istifadə olunur.

**if ($a > $b) {**

**echo "$a is greater than $b";**

**} else {**

**echo "$a is not greater than $b";**

**}**

**2.SWITCH-CASE -** dəyişənin müxtəlif mümkün dəyərlərini yoxlamaq üçün istifadə olunur.

**switch ($a) {**

**case 1:**

**echo "a is 1";**

**break;**

**case 2:**

**echo "a is 2";**

**break;**

**default:**

**echo "a is neither 1 nor 2";**

**}**

**3.Ternary -** ‘if-else` ifadəsinin qısa formasıdır və adətən sadə şərtlər üçün istifadə olunur.

**$age = 20;**

**$message = ($age >= 18) ? "Siz yetkin yaşdasınız." : "Siz hələ yetkin deyilsiniz.";**

**echo $message;**

**DÖVRLƏR:**

**1. For Dövrü -** Təyin edilmiş şərtlər daxilində kod blokunu təkrarlayır.

**for ($i = 0; $i < 10; $i++) {**

**echo $i;**

**}**

**2. While Dövrü -** Bir şərt doğru olduğu müddətcə kod blokunu təkrarlayır.

**$i = 0;**

**while ($i < 10) {**

**echo $i;**

**$i++; }**

**3. Do-While Dövrü -** Kod blokunu ən azı bir dəfəicra edir və sonra şərti yoxlayaraq dövrü davam etdirir.

**$i = 0;**

**do {**

**echo $i;**

**$i++;**

**} while ($i < 10);**

**4. Foreach Dövrü -** Array və ya obyekt kimi kolleksiyaların üzərində təkrarlanma edir.

**$arr = [1, 2, 3, 4, 5];**

**foreach ($arr as $value) {**

**echo $value;**

**}**

***19. MySqli nədir? Verilənlər bazası ilə MySqli(Procedural) vasitəsilə əlaqə yaradılması və verilənlər bazasına datanın MySqli istifadəsi ilə insert edilməsi məntiqlərini izah edin.***

MySqli PHP üçün hazırlanmış bir MySQL verilənlər bazası üçün qoşulma təmin edən genişləndirilmiş moduldur. MySqli, hem prosedur (obyektsiz) hem de obyekt-yönümlü interfeyslər təqdim edir.

**MySqli əlaqəsini yaratmaq:**

**$server = "localhost";**

**$username = "istifadəçi\_adı";**

**$password = "şifrə";**

**$database = "verilən\_bazası\_adı";**

**$conn = new mysqli($server, $username, $password, $database);**

**if ($conn->connect\_error) {**

**die("Əlaqə uğursuz oldu: " . $conn->connect\_error);**

**}**

**Verilənlər Bazasına MySqli İstifadəsi ilə Datanın Əlavə Edilməsi:**

**$sql = "INSERT INTO users (username, email, password) VALUES ('john\_doe', 'john@example.com', 'password123')";**

**if ($conn->query($sql) === TRUE) {**

**echo "Yeni istifadəçi uğurla əlavə edildi.";**

**} else {**

**echo "Səhv baş verdi: " . $conn->error;**

**}**

**$conn->close();**

***20. Verilənlər bazasından datanın MySqli (Procedural) istifadəsi ilə çəkilməsi və silinməsi məntiqlərini izah edin.***

Verilənlər bazasından data çəkmək üçün **`mysqli\_query`** və **`mysqli\_fetch\_assoc`** kimi funksiyalar istifadə edilir.

**$conn = mysqli\_connect("localhost", "username", "password", "database");**

**if (!$conn) {**

**die("Qoşulma xətası: " . mysqli\_connect\_error());**

**}**

**$sql = "SELECT id, column1, column2 FROM table\_name";**

**$result = mysqli\_query($conn, $sql);**

**if (mysqli\_num\_rows($result) > 0) {**

**while($row = mysqli\_fetch\_assoc($result)) {**

**echo "id: " . $row["id"]. " - Column1: " . $row["column1"]. " - Column2: " . $row["column2"]. "<br>";**

**}**

**} else {**

**echo "0 nəticə";**

**}**

**mysqli\_close($conn);**

Data silmək üçün, yənə **`mysqli\_query`** funksiyasından istifadə edilir.

**$conn = mysqli\_connect("localhost", "username", "password", "database");**

**if (!$conn) {**

**die("Qoşulma xətası: " . mysqli\_connect\_error());**

**}**

**$sql = "DELETE FROM table\_name WHERE id=3";**

**if (mysqli\_query($conn, $sql)) {**

**echo "Qeyd uğurla silindi";**

**} else {**

**echo "Xəta: " . mysqli\_error($conn);**

**}**

**mysqli\_close($conn);**

***21. Mysqli nədir? MySqli (Procedural) ilə update əməliyyatını izah edin.***

MySqli PHP üçün hazırlanmış bir MySQL verilənlər bazası üçün qoşulma təmin edən genişləndirilmiş moduldur. MySqli, hem prosedur (obyektsiz) hem de obyekt-yönümlü interfeyslər təqdim edir.

**$update\_id = 2;**

**$new\_username = "new\_username";**

**$new\_email = "new\_email@example.com";**

**$sql\_update = "UPDATE users SET username = '$new\_username', email = '$new\_email' WHERE id = $update\_id";**

**if ($conn->query($sql\_update) === TRUE) {**

**echo "Məlumat uğurla güncəlləndi.";**

**} else {**

**echo "Səhv baş verdi: " . $conn->error;**

**}**

**$conn->close();**

***22. PHP-də operatorlar hansılardır? PHP-də date funksiyaları haqqında məlumat verin.***

**1. Arifmetik Operatorlar:**

Toplama: `+`

    Çıxma: `-`

    Vurma: `\*`

    Bölmə: `/`

    Modul (qalıq): `%`

    İnkrement: `++`

    Dekrement: `--`

**2. Təyinat Operatorları**: `=`, `+=`, `-=`, `\*=`, `/=`, `%=`

**3. Müqayisə Operatorları:**

Bərabərdir: `==`

   Dəqiq bərabərdir: `===`

    Bərabər deyildir: `!=`

    Dəqiq bərabər deyildir: `!==`

    Kiçikdir: `<`

Böyükdür: `>`

    Kiçik və ya bərabərdir: `<=`

    Böyükdür və ya bərabərdir: `>=`

**4. Məntiqi Operatorlar:**

Və: `&&` və ya `and`

    Və ya: `||` və ya `or`

    Not: `!`

**5. String Operatorları:**

    Birləşdirmə: `.`

    Birləşdirmə və təyin: `.=`

**Tarix Funksiyaları**

**date() -** Cari tarixi və vaxtı göstərir

**echo date("d/m/Y H:i:s");**

**time() –** Cari tarix və vaxt məlumamatlarını UNİX vaxt damğası kimi qaytarır

**echo time();**

**mktime() –** Bir tarixi UNİX vaxt damğasına çevirir

**$mohur = mktime(0, 0, 0, 7, 1, 2000);** // 2000-ci ilin 1 iyuluna aid zaman möhürünü qaytarır

**strtotime() -** tarix və vaxt mətnini Unix zaman möhürünə çevirir.

**$mohur = strtotime("10 September 2000");**

**date\_add() və date\_sub() -**  DateTime obyektinə müəyyən bir müddət əlavə etmək və ya çıxmaq üçün istifadə edilir.

**$date = date\_create('2024-01-01');**

**date\_add($date, date\_interval\_create\_from\_date\_string('10 days'));**

**echo date\_format($date, 'Y-m-d'); // 10 gün əlavə edir**

**getdate()** - Tarix və vaxtla bağlı bir çox məlumatı əldə edir

**$date = getdate();**

**print\_r($date);**

**checkdate()**

  **if (checkdate(12, 31, 2024)) {**

**echo "Keçərli tarixdir";**

**} else {**

**echo "Keçərli tarix deyil";**

**}**

***23. Exception, error, try, catch, throw məntiqlərini izah edin.***

**Exception -** proqramın nizamlı işləməsinə mane olmayan istisnai vəziyyətləri nümayiş etdirir. Məsələn, bir faylın mövcud olmaması, verilənlər bazasına əlaqə qurmaq üçün səhv məlumatlar, nəticədə əldə edilmiş tip və s**.**

**Error –** normal işləyiş qaydasını pozan, amma bu dəfə proqramın təkrar normal çalışmasını mümkün etməyən ciddi bir problem göstərmək üçün istifadə olunan obyekt və ya xüsusiyyətdir.

**Try** - xəta baş verə biləcək kodları əhatə edir.

**Catch** - catch` bloku, `try` blokunda baş verən istisnanı tutar və idarə edir. İstisna `catch` blokuna ötürülən obyektin növü ilə uyğun gəlməlidir.

**Throw** - İstisna yaratmaq və onu `catch` blokuna ötürməküçün `throw` ifadəsi istifadə olunur.

**try {**

**if(somethingGoesWrong()) {**

**throw new Exception("Xəta baş verdi!");**

**}**

**} catch (Exception $e) {**

**echo 'İstisna tutuldu: ' . $e->getMessage();**

**} finally {**

**echo "Finally bloku icra edildi";**

**}**

***24. PHP-OOP-də static, parent, self məntiqlərini izah edin.***

**Static** bir üzvün klassa məxus olduğunu və obyekt yaratmadan istifadə oluna biləcəyini göstərir.

**class MathOperation {**

**public static function add($a, $b) {**

**return $a + $b;**

**} }**

**$result = MathOperation::add(5, 3);**

**echo $result;**

**Parent** bir sinifin üst sinifinə (parent sinifinə) işarə edir. Obyekt yaratmaq və ya statik üzvələri işlətmək üçün istifadə olunur.

**class Animal {**

**public function makeSound() {**

**echo "Some generic sound";**

**} }**

**class Dog extends Animal {**

**public function makeSound() {**

**parent::makeSound();**

**echo "Woof! Woof!";**

**} }**

**$dog = new Dog();**

**$dog->makeSound();**

**Self** miras(inheritance) ilə işləməz və üzvü olduğu klassda təyin olunmuşsa, həmin klassın özünə aid olduğunu bildirir.

**class MyClass {**

**const SABIT = "Mən sabitəm";**

**public function showConstant() {**

**echo self::SABIT . "\n";**

**} }**

**echo MyClass::SABIT; // Sabiti çağırmaq**

**$obj = new MyClass();**

**$obj->showConstant(); // self ilə sabiti çağırmaq**

***25. PHP-OOP-də public, private, protected məntiqlərini izah edin.***

**Public** xassə və ya metodun hər yerdən əlçatan olacağını bildirir. Bu, həm sinifin daxilində, həm də sinifin xaricində olan kodlar tərəfindən çağırıla biləcəyi mənasını daşıyır.

**class Example {**

**public $publicVar = 'Public variable';**

**public function publicMethod() {**

**return 'Public method';**

**} }**

**$obj = new Example();**

**echo $obj->publicVar;**

**echo $obj->publicMethod();**

**Private** xassə və ya metodun yalnız öz sinifi daxilində əlçatan olacağını təyin edir.

c**lass PrivateExample {**

**private $privateProperty = "Mən yalnız öz sinifimdə əlçatanam";**

**private function privateMethod() {**

**return "Bu metod yalnız öz sinifimdə çağırıla bilər";**

**}**

**}**

**$obj = new PrivateExample();**

**$obj->privateProperty;**

**$obj->privateMethod();**

**Protected** xassə və ya metodun sinifin özü və ondan miras alan alt siniflərdəəlçatan olacağını bildirir.

**class Example {**

**protected $protectedVar = 'Protected variable';**

**protected function protectedMethod() {**

**return 'Protected method';**

**} }**

**class AnotherExample extends Example {**

**public function accessProtected() {**

**return $this->protectedVar;**

**} }**

**$obj = new AnotherExample();**

**echo $obj->accessProtected();**

***26. PHP-də funksiyaların növləri hansılardır? Void və return funksiyalar haqqında məlumat verin.***

Funksiyalar müəyyən bir əməliyyatı yerinə yetirən kod bloklarıdır. PHP-də funksiyaların bir neçə növü mövcuddur

**Built-in Functions -** Bunlar PHP-nin özündə quraşdırılmışdır və heç bir xüsusi quraşdırma tələb etmirlər. Məsələn, **`echo()`, `print\_r()`, `var\_dump()`, `date()`, `strlen()`**

**User-Defined Functions –** Burada istifadəçilər öz funksiyalarını təyin edə bilərlər.

**Anonymous Functions -** Adı olmayan və dəyişənlərə mənimsədilən funksiyalardır. Callback funksiyaları və ya həndəsi funksiyalar kimi istifadə oluna bilər.

**Callback Functions -** Bir funksiyanın parametri olaraq ötürülən və həmin funksiya daxilində çağırılan funksiyalar. **`array\_map()`, ‘array\_filter’ `usort()`** kimi PHP funksiyaları callback funksiyalarını qəbul edir.

**Recursive Functions** - Özünü özü çağıran funksiyalardır.

**Nested Functions –** İç-içə funksiyalar.

**Return funksiyalar -** bir və ya daha çox dəyər qaytaran funksiyalardır.

**function add($a, $b) {**

**return $a + $b;**

**}**

**$result = add(5, 3);**

**echo $result;**

**Void funksiyalar -** hər hansı bir dəyər qaytarmayan funksiyalardır. PHP 7.1-dən etibarən void tipi dəstəklənir.

**function showMessage(): void {**

**echo "This is a void function.";**

**}**

**showMessage();**

***27. PHP-də funksiyalar haqqında məlumat verin. func\_num\_args(), func\_get\_arg(), function\_exists() funksiyaları haqqında məlumat verin.***

Funksiyalar müəyyən bir əməliyyatı yerinə yetirən kod bloklarıdır. Funksiyaların parametrləri ola bilər, və bu parametrlər funksiyanın davranışını təyin edə bilər. PHP, funksiyalarla işləməyi asanlaşdıran bir sıra gömülü funksiyalartəqdim edir.

**`func\_num\_args()` -** o anda çağırılan funksiyanın neçə argument aldığını qaytarır. Yalnız funksiya daxilində istifadə edilə bilər.

f**unction test() {**

**$num\_args = func\_num\_args();**

**echo  $num\_args;**

**}**

**test(1, 2, 3);**

**`func\_get\_arg()` -** çağırılan funksiyanın müəyyən indeksdəki argumentini qaytarır. İndeks sıfırdan başlayır. Yalnız funksiya daxilində istifadə edilə bilər.

**function test() {**

**if (func\_num\_args() > 0) {**

**$arg = func\_get\_arg(0);**

**echo "İlk argument: $arg\n";**

**}**

**}**

**test('PHP', 'Python');**

**`function\_exists()`-**  Əgər göstərilən adda bir funksiya mövcuddursa `true`, əks halda `false` qaytarır.

i**f (function\_exists('test')) {**

**echo "Funksiya 'test' mövcuddur.";**

**} else {**

**echo "Funksiya 'test' mövcud deyil.";**

**}**

***28. PHP-də arraylar haqqında məlumat verin. array\_push(), array\_pop(), array\_shift(), array\_unshift() funksiyaları haqqında yazın.***

PHP-də arraylar, bir çox dəyəri bir dəyişəndə saxlamağa imkan verən məlumat strukturlarıdır. PHP, arraylərlə işləməyi asanlaşdıran bir çox gömülü funksiyalar təqdim edir.

**`array\_push()` -** arrayın sonuna bir və ya daha çox element əlavə edir. İlk parametr olaraq arrayı, sonrakı parametrlər olaraq əlavə ediləcək elementləri qəbul edir. Əlavə edilən elementlərin sayını qaytarır.

**$array = array("a", "b", "c");**

**array\_push($array, "d", "e");**

**print\_r($array);**

**`array\_pop()`-** arrayın son elementini çıxardır və onu qaytarır. Array boşdursa, `NULL` qaytarır.

**$array = array("a", "b", "c");**

**$element = array\_pop($array);**

**echo $element;**

**print\_r($array);**

**`array\_shift()`-** arrayın ilk elementini çıxartır və onu qaytarır. Array boşdursa, `NULL` qaytarır. `array\_shift()` işlədikdə arrayın qalan elementlərinin indeksləri yenidən təyin edilir.

**$array = array("a", "b", "c");**

**$firstElement = array\_shift($array);**

**echo $firstElement;**

**print\_r($array);**

**`array\_unshift()`-** arrayın başına bir və ya daha çox element əlavə edir. İlk parametr olaraq arrayı, sonrakı parametrlər olaraq əlavə ediləcək elementləri qəbul edir. Yeni arrayın uzunluğunu qaytarır.

**$array = array("b", "c");**

**array\_unshift($array, "a");**

**print\_r($array);**

***29. HTML Form-u yazın, bu form-dan gələn datanı filterləmək üçün “validate()” adli funksiya yazin.***

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

         Email: <input type="text" name="email"><br>

        <input type="submit" value="Göndər">

    </form>

<?php

function validate($email) {

    $email = trim($email);

    $email = strip\_tags($email);

    $email = htmlspecialchars($email);

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

        return false ;

           }

        return $email;}

 $gelenEmail = validate($\_POST["email"]);

?>

***30. “text.txt” filemizin olub olmadığını yoxlayın, əgər yoxdursa, yaradın.***

<?php

$dosyaAdi = 'text.txt';

if (!file\_exists($dosyaAdi)) {

    $dosya = fopen($dosyaAdi, 'w');

    fclose($dosya);

    echo "Fayl yaradıldı: $dosyaAdi";

} else {

    echo "Fayl mövcuddur: $dosyaAdi";

}

?>

***31. Type-si checkbox olan HTML Form-u yazın, PHP-də seçilən value-ni və ya value-ləri ekrana çap edin.***

<form method="post" action="veri\_isle.php">

    <label ><input type="checkbox" name="secim[]" id="secim1" value="option1"> Seçim 1</label><br>

    <label ><input type="checkbox" name="secim[]" id="secim2" value="option2"> Seçimk 2</label><br>

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

</form>

<?php

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

    if (isset($\_POST["secim"])) {

        $secilen\_veriler = $\_POST["secim"];

        if (!empty($secilen\_veriler)) {

            foreach ($secilen\_veriler as $secilen) {

                echo $secilen . "<br>";

            }

        } else {

            echo "Məlumat seçilmədi!";

        }

}

?>

***32. Type-si radio olan HTML Form-u yazın, PHP-də seçilən value-ni ekrana çap edin.***

form method="post" action="veri\_isle.php">

    <label ><input type="radio" name="secim" id="secim1" value="secim1"> Seçim 1</label><br>

    <label ><input type="radio" name="secim" id="secim2" value="secim2"> Seçim 2</label><br>

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

</form>

<?php

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

    if (isset($\_POST["secim"])) {

        $secilen\_melumat = $\_POST["secim"];

        echo "Seçilen məlumat: " . $secilen\_melumat;

    } else {

        echo "Məlumat seçilmədi!";

    }

}

?>

***33. Type-si select olan HTML Form-u yazın, PHP-də seçilən value-ni ekrana çap edin.***

<form method="post" action="veri\_isle.php">

    <select name="secim" id="secim">

        <option value="secim1">Seçim1</option>

        <option value="secim2">Seçim 2</option>

        </select>

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

</form>

<?php

if ($\_SERVER["REQUEST\_METHOD"] == "POST" && isset($\_POST['submit'])) {

    if(isset($\_POST['secim'])) {

        $secilen = $\_POST['secim'];

         echo "Seçilen məlumat " . $secilen;

    } else {

        echo "Bir məlumat seçin.";

    }

}

?>

***34. PHPMailer vasitəsi ilə mail göndərilən zaman istifadə olunan funksiyaları yazın və izah edin.***

PHPMailer, PHP ilə e-poçt göndərmək üçün geniş istifadə edilən bir kitabxanadır. Bu kitabxana, SMTP serverlər üzərindən e-poçt göndərməyi asanlaşdırır və HTML məzmunu, şifrələmə, fayl əlavələri və digər funksiyaları dəstəkləyir.

**PHPMailer-i Quraşdırmaq -** PHPMailer, **Composer** vasitəsilə asanlıqla quraşdırıla bilər.

**PHPMailer ilə E-poçt Göndərmək**

**1. PHPMailer Kitabxanasını Daxil Edin:**

    use PHPMailer\PHPMailer\PHPMailer;

    use PHPMailer\PHPMailer\Exception;

    require 'vendor/autoload.php';

**2. PHPMailer İnstansını Yaradın:**

    $mail = new PHPMailer(true);

**3. SMTP Parametrlərini Quraşdırın:**

     $mail->SMTPDebug = 2;

    $mail->isSMTP();

    $mail->Host = 'smtp.example.com';  // SMTP server ünvanı

    $mail->SMTPAuth = true;

    $mail->Username = 'email@example.com';  // SMTP istifadəçi adı

    $mail->Password = 'password';  // SMTP parol

    $mail->SMTPSecure = 'tls';  // Şifrələmə üsulu (‘tls’ və ya ‘ssl’)

    $mail->Port = 587;  // SMTP port (TLS üçün 587, SSL üçün 465)

**4. E-poçt Parametrlərini Quraşdırın:** Kimə göndərildiyi, kimdən göndərildiyi, mövzu və məzmun kimi e-poçt məlumatlarını təyin edin.

    $mail->setFrom('from@example.com', 'Göndərən Adı');

    $mail->addAddress('to@example.com', 'Alan Adı');

    $mail->Subject = 'E-poçt Mövzusu';

    $mail->Body    = 'Bu e-poçtun mətni';

    $mail->AltBody = 'e-poçt proqramları üçün sadə mətn.';

**5. E-poçtu Göndərin:**

    try {

        $mail->send();

    } catch (Exception $e) {

        echo 'Xəta: ', $mail->ErrorInfo; }

***35. ‘text.txt’ filemiz var. Bu fileni acib, filenin en sonuna “text elave edildi” yazisi yazin.***

<?php

$dosyaAdi = 'text.txt';

$dosya = fopen($dosyaAdi, 'a');

if ($dosya) {

$yazi = "text elave edildi\n";

fwrite($dosya, $yazi);

fclose($dosya);

echo "File uğurlu şəkildə yazıldı";

} else {

echo "File açılamadı.";

}

?>

***36. ‘text.txt’ filemiz var. Bu fileni acib, filede evvelden olan melumatlar silib, “text elave edildi” yazisi yazin.***

<?php

$file = fopen("text.txt", "w");

if ($file) {

    $text = "text əlavə edildi";

    fwrite($file, $text);

    fclose($file);

} else {

    echo "Fayl açıla bilmədi.";

}

?>

***37. Type-si color olan HTML Form-u yazın, inputun HEX kodunu ekrana çıxarın.***

<form method="post" action="process\_color.php">

    <input type="color" id="colorInput" name="color">

    <input type="submit" value="Submit">

</form>

<?php

if (isset($\_POST['submit']) {

    $selectedColor = $\_POST["color"];

    $selectedColor = ltrim($selectedColor, '#');

    $selectedColor = strtoupper($selectedColor);

    echo "#$selectedColor";

}

?>

***38. Verilmiş table-də id-si 2 olan sətri MySqli(Procedural) vasitəsilə silin. Data silinərsə, "data silindi", əks halda "data silinmədi" yazısı çap edilsin (table adı: usersinfo, database adı: users).***

<?php

$servername = "localhost";

$username = "username";

$password = "password";

$dbname = "users";

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

if (!$conn) {

    die("Connection failed: " . mysqli\_connect\_error());

}

$sql = "DELETE FROM usersinfo WHERE id=2";

if (mysqli\_query($conn, $sql)) {

    echo "Data silindi";

} else {

    echo "Data silinmədi: " . mysqli\_error($conn);

}

mysqli\_close($conn);

?>

***39. Verilmiş table-də id-si 1 və 3 olan sətrləri MySqli(Procedural) vasitəsilə silin. Data silinərsə, "data silindi", əks halda "data silinmədi" yazısı çap edilsin (table adı: usersinfo, database adı: users).***

<?php

$servername = "localhost";

$username = "username";

$password = "password";

$dbname = "users";

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

if (!$conn) {

    die("Connection failed: " . mysqli\_connect\_error());

}

$sql = "DELETE FROM usersinfo WHERE id IN (1, 3)";

if (mysqli\_query($conn, $sql)) {

    echo "Data silindi";

} else {

    echo "Data silinmədi: " . mysqli\_error($conn);

}

mysqli\_close($conn);

?>

***40. Verilmiş table-yə $name = "Lewis", $surname = "Hamilton" dəyərlərini PDO vasitəsilə insert edin. Data əlavə edilərsə, "əlavə edildi", əks halda "əlavə edilmədi" yazısı çap edilsin (table adı: usersinfo, database adı: users).***

<?php

try{

    $db = new PDO("mysql:host=localhost; dbname=pdo; charset=utf8", 'root', '12345678');

}catch(PDOException $e){

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

}

?>

<?php

require "connect.php";

if(isset($\_POST['submit'])){

    $save = $db->prepare("INSERT INTO users (name, surname) VALUES (:name, :surname)");

    $isInserted = $save->execute([

        'name' => $\_POST['name'],

        'surname' => $\_POST['surname']

    ]);

    if($isInserted){

        echo "Əlavə edildi";

    }else{

        echo "Əlavə edilmədi";

    }

}

?>

***41. (database adı: users, table adı: usersinfo) Verilmiş table-də olan dataları PDO vasitəsilə HTML Table-yə yazdırın və aşağıdakı kimi çap edin.***

<?php

require "connect.php";

$sql = $db->prepare("SELECT name, surname FROM usersinfo");

$sql->execute();

?>

<table border="1" width="200">

    <tr>

        <th>Ad</th>

        <th>Soyad</th>

    </tr>

    <?php

    while($dataGet = $sql->fetch(PDO::FETCH\_ASSOC)){

        ?>

        <tr>

            <td><?php echo htmlspecialchars($dataGet["name"]); ?></td>

            <td><?php echo htmlspecialchars($dataGet["surname"]); ?></td>

        </tr>

        <?php

    }

    ?>

</table>

<?php

?>

***42. (database adı: users, table adı: usersinfo) Verilmiş table-də olan dataları MySqli(Procedural) vasitəsilə HTML Table-yə yazdırın və aşağıdakı kimi çap edin.***

<?php

$host = 'localhost';

$user = 'username';

$password = 'password';

$dbname = 'users';

$conn = mysqli\_connect($host, $user, $password, $dbname);

if (!$conn) {

    die("Connection failed: " . mysqli\_connect\_error());

}

$query = "SELECT name, surname FROM usersinfo";

$result = mysqli\_query($conn, $query);

?>

<table border="1" width="200">

    <tr>

        <th>Ad</th>

        <th>Soyad</th>

    </tr>

    <?php

        while($dataGet = mysqli\_fetch\_assoc($result)){

        ?>

        <tr>

            <td><?php echo htmlspecialchars($dataGet["name"]); ?></td>

            <td><?php echo htmlspecialchars($dataGet["surname"]); ?></td>

        </tr>

        <?php

    }

    ?>

</table>

<?php

mysqli\_close($conn);

?>

***43. Verilmiş table-də id-si 2 olan sətri PDO vasitəsilə silin. Data silinərsə, "data silindi", əks halda "data silinmədi"***

<?php

$host = 'localhost';

$dbname = 'users';

$username = 'your\_username';

$password = 'your\_password';

try {

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

    $sql = $pdo->prepare("DELETE FROM usersinfo WHERE id = :id");

    $sql->bindValue(':id', 2);

    $sql->execute();

    if ($sql->rowCount() > 0) {

        echo "Data silindi";

    } else {

        echo "Data silinmədi";

    }

} catch (PDOException $e) {

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

}

?>

***44. Verilmiş table-də id-si 2 olan sətri PDO vasitəsilə update edin, name=”Adrian” və surname=”Harris” olsun. Data update olarsa, "data update oldu", əks halda "data update olmadı" yazısı çap edilsin (table adı: usersinfo, database adı: users).***

<?php

require "connect.php";

if(isset($\_POST['updateRow'])){

    $updateId = $\_POST['id'];

    $updateAd = $\_POST['ad'];

    $updateSoyad = $\_POST['soyad'];

$update = $db->prepare("UPDATE userinfo set ad=:ad, soyad=:soyad WHERE id={$updateId}");

    $update->execute(

        [

            'ad' => $updateAd,

            'soyad' => $updateSoyad,

        ]

        );

        if($update){

            echo "update olundu";

        }else{

            echo "update olunmadi";

        }

    }

?>

***45. Verilmiş table-də id-si 2 olan sətri MySqli(Procedural) vasitəsilə update edin, name=”Adrian” və surname=”Harris” olsun. Data update olarsa, "data update oldu", əks halda "data update olmadı" yazısı çap edilsin (table adı: usersinfo, database adı: users).***

<?php

$servername = "server\_adi";

$username = "istifadeci\_adi";

$password = "parol";

$database = "databasei\_adi";

$conn = mysqli\_connect($servername, $username, $password, $database);

if (!$conn) {

    die("Database bağlantı xətası: " . mysqli\_connect\_error());

}

if (isset($\_POST['submit']) {

    $id = $\_POST["id"];

    $ad = $\_POST["ad"];

    $soyad = $\_POST["soyad"];

    $sql = "UPDATE tablo\_adi SET ad = '$ad', soyad = '$soyad' WHERE id = $id";

    if (mysqli\_query($conn, $sql)) {

        echo "Data update olundu";

    } else {

        echo "Data update olunmadı: " . $sql . "<br>" . mysqli\_error($conn);

    }

}

mysqli\_close($conn);

?>

***46. Button-a click edilən zaman verilmiş form-dakı dataları "proses.php" səhifəsinə Jquery və AJAX istifadə edərək göndərin***

$(document).ready(function(){

    $("#sendData").click(function(){

        var email = $("#email").val();

        var password = $("#pass").val();

        $.ajax({

            type: "POST",

            url: "proses.php",

            data: { email: email, pass: password },

            success: function(response){

                               console.log(response);

            },

            error: function(xhr, status, error){

                        console.error(error);

            }

        });

    });

});

***47. $adminEmail = someone@mail.ru; $adminParol ="12345”; adli iki deyisenimiz var, formdan daxil olan email ve parolun, bu email ve parolla ust uste dusmesini yoxlayin. Eyni olarsa ekrana “ugurlu”, ferqli olarsa, “yanlis” yazisi cixsin.***

<?php

$adminEmail = "someone@mail.ru";

$adminParol = "12345";

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

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

if ($adminEmail == $formEmail && $adminParol == $formParol) {

    echo "ugurlu";

} else {

    echo "yanlis";

}

?>

***48. “text.txt” filesini acin ve ilk setrini və ilk simvolunu oxuyun.***

<?php

$dosyaAdi = 'text.txt';

if (file\_exists($dosyaAdi)) {

    $dosya = fopen($dosyaAdi, 'r');

    $ilkSetr = fgets($dosya);

    $ilkSimvol = fgetc($dosya);

    fclose($dosya);

    if ($ilkSetr!== false $ilkSimvol !== false) {

        echo  $ilkSetr;

        echo  $ilkSimvol;

    } else {

        echo "Faylda məlumat yoxdur.";

    }

} else {

    echo "Fayl mövcud deyil: $dosyaAdi";

}

?>

***49. Formda inputdan soz daxil edilir ve hemin soz Mysqlde tablede “surname” sütunu üzrə axtarilir ve netice olaraq cixsin (simple search) (PDO)***

<?php

     require "connect.php";

     if(isset($\_POST['submit'])){

     $search\_keyword = $\_POST['search\_keyword'];

    $statement = $conn->prepare("SELECT \* FROM userinfo WHERE surname LIKE :search\_keyword");

    $keyword = '%' . $search\_keyword . '%';

    $statement->execute(

   [

'search\_keyword'=> $keyword

  ]

);

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

        echo "ID: " . $row['id'] . ", Ad: " . $row['name'] . ", Soyad: " . $row['surname'] . "<br>";

    }

?>

***50. Formda inputdan soz daxil edilir ve hemin soz Mysqlde tablede “surname” sütunu üzrə axtarilir ve netice olaraq cixsin (simple search) (MySqli (Procedural)).***

<?php

$servername = "server\_adi";

$username = "istifadeci\_adi";

$password = "parol";

$database = "database\_adi";

$conn = mysqli\_connect($servername, $username, $password, $database);

if (!$conn) {

    die("Database bağlantı xətası: " . mysqli\_connect\_error());

}

if (isset($\_POST['submit']) {

    $search\_keyword = $\_POST["search\_keyword"];

    $sql = "SELECT \* FROM userinfo WHERE surname LIKE '%$search\_keyword%'";

    $result = mysqli\_query($conn, $sql);

    if (mysqli\_num\_rows($result) > 0) {

        while ($row = mysqli\_fetch\_assoc($result)) {

            echo "ID: " . $row["id"] . " - Kelime: " . $row["surname"] . "<br>";

        }

    } else {

        echo "Nəticə yoxdur";

    }

}

mysqli\_close($conn);

?>

***51. “Simple Calculator” classı yazın. Switch ve case istifadə edin.***

<?php

class SimpleCalculator {

public function calculate($num1, $num2, $operator) {

switch ($operator) {

case '+':

return $num1 + $num2;

case '-':

return $num1 - $num2;

case '\*':

return $num1 \* $num2;

case '/':

if ($num2 != 0) {

return $num1 / $num2;

} else {

return "Error: Division by zero!";

}

default:

return "Error: Invalid operator!";

}

}

}

?>