PHP

**#1.1(15) Обрезание и харакири**

- (**5**) Ситуация: мы обрезали первый “символ” в строке “Маша”: substr(“Маша”, 1). Как на экране будет выглядеть остаток?

За умови, якщо вказано кодування(UTF-8) substr, то буде обрізано 1 байт слова, тобто буква «М» буде поділена навпіл. Тобто виведеться «??аша».

- (**1**) Всегда ли кол-во символов, показываемое функцией strlen(), соответствует реальности? Когда соответствует, а когда нет?

Ні, так як ця функція рахує кількість байт, а не символів.

- (**1**) Какой ф-ей точно узнать кол-во символов в строке?

        mb\_strlen($str) або iconv\_strlen($str)

- (**8**) Как можно всегда точно получить кол-во символов именной ф-ей strlen() без написания доп. кода (например, ф-я strlen() используется в CMS в очень многих местах, и заменить ее нельзя либо очень трудоемко). Как можно без доп. кода нормально обрезать строки ф-ей substr()?

Необхідно замінити кодування на (windows-1251).

**#1.2(4) Версии PHP**

Как известно, есть разные версии PHP. Нужно перечислить основные отличия между версиями PHP 5.3, 5.4, 5.5. Озвучить собственное мнение о PHP 7. Тезисно.

Основні відмінності:

·        Версія 5.4 в порівнянні з 5.3

1.   Були введені трейти

2.   Можливість розіменовувати масиви, що повертаються функціями.

3.   Короткий запис масивів

4.   Добавлений бінарний формат задання чисел.

5.   Можливість отримати доступ до члена класа при створені екземпляра.

6.   Встроєнний веб-сервіс.

·        Версія 5.5 в порівнянні з 5.4

1.   Були додані генератори

2.   Хешування паролів

3.   Foreach підтримує list();

4.   Можливість звертатися до елеменів рядків(string) як до масивів(по індексу)

·        Ставлення до PHP 7

        Ставлення є досить позитивним. Враховуючи відгуки на різних джерелах інформації, він є досить перспективним, проте я його поки так і не використовував. Є досить велика кількість змін, причому по ідеї вони по мінімуму конфліктують з попередньою версією.

**#2(4). Длинная Нина**

Нужно записать большую строковую переменную. Большую - имеется ввиду несколько строк или даже несколько десятков строк: большой SQL запрос, длинный кусок статического текста, кусок html. Внутри строки есть переменные, которые нужно подставить при использовании переменной.

Пример

**Шла Нина по $way. Навстречу ей шел $nina->boyfriend**

Предложите варианты записи такой переменной.

Можна так :

echo ‘<div id=”road”><a href=”/road”>Дорога</a></div>’;

$query = mysql\_query("SELECT road\_name FROM road WHERE road\_name=’$way’);

$row = mysql\_fetch\_array($query);

echo “Вона йшла по вулиці, яка мала назву ”.$row[‘road\_name’].” і нарешті прийшла”;

echo "Шла Нина по $way .Навстречу ей шел $boyfriend";

echo "Шла Нина по ".$way.".Навстречу ей шел ".$boyfriend;

**#3(10). Регулярки и циркулярки**

Напишите шаблоны регулярных выражений и сами ф-и для того, чтобы:

-**(3)** оставить в строке только цифры, то есть из выражений вида ‘%%1dd23 jj$$45’ получить 12345

$str = *preg\_replace*("/\D/", "", "%%1dd23 jj$$45");

$str = eregi\_replace("([^0-9])", "", "%%1dd23 jj$$45");

В $str будуть тільки цифри(12345)

-**(3)** получить содержимое, заключенное в скобки: **(***вот это нужно получить***)**

*preg\_match('/\((.+)\)/', "Текст за дужками(Текст в дужках)", $str);*echo $str[1];

В $str буде Текст в дужках

-**(4)** предложите ф-ю валидации автомобильного номера Украины (пример: АА1234СС, то есть две буквы, четыре цифры, опять две буквы). Обратите внимание, что в номерах используются только те буквы украинского алфавита, которые по написанию совпадают с английскими.

$string = "АA1234СС";

function auto\_validation($string)

{

$alpha = '(А|В|С|Е|Н|І|К|М|О|Р|Т|Х|A|B|C|E|H|I|K|M|O|P|T|X)';

$validaion = *preg\_match*("#$alpha{2}\d{4}$alpha{2}#", $string);

return ($validaion);

}

**#5(5)** **Уникальная уникальность**

Вы вытащили из БД ассоциативный массив вот такого вида

array(

        array(

‘id’ => 111,

‘field2’ => 222,

‘field3’ => 333,

),

        array(

‘id’ => 33,

‘field2’ => 3333,

‘field3’ => 321

),

...

);

то есть каждый вложенный массив - строка таблицы.

id - параметр, который полностью определяет все значения вложенного массива, он должен быть уникальным, но он может повторяться в таблице БД (например, дважды внесли одну и ту же строку).

Обеспечьте переформирование массива таким образом, чтобы в нем не было повторяющихся вложенных массивов. Структуру массива можно менять (добавлять/убирать ключи и пр.)

function unique\_array($argument\_array)

{

 $unique\_id = array();

 $unique\_array = array();

 foreach($argument\_array as $current\_array)

 {

     if(!*in\_array*($current\_array['id'],$unique\_id))

     {

         $unique\_id[] = $current\_array['id'];

         $unique\_array[] = $current\_array;

     }

 }

 return $unique\_array;

}

$result = unique\_array($array);

*print\_r*($result);

В результаті отримаємо масив без повтореннь по полю “id”

**#6(5) Звонок на email**

В таблицу БД вставляются значения: name, last\_name, email, phone.

Комбинация email и phone полностью определяет пользователя, остальные поля произвольны и особой роли не играют.

Напишите SQL-запрос и структуру таблицы, которая обеспечит уникальность комбинации email + phone в таблице, даже если пытаться вставлять одни и те же данные много раз.

Разумеется, выполнение такого запроса не должно вызывать ошибок и исключений от движка БД.

*CREATE TABLE IF NOT EXISTS* Clients(

*id* INT(8) *PRIMARY KEY UNSIGNED NOT* NULL *AUTO\_INCREMENT*,

*name* VARCHAR(10) *NOT* NULL,

*last\_name* VARCHAR(10) *NOT* NULL,

*email* VARCHAR(10) *NOT* NULL,

*phone* VARCHAR(20) NOT NULL,

*UNIQUE* KEY (*email*, *phone*)

) ENGINE=InnoDB AUTO\_INCREMENT=0 DEFAULT CHARSET=utf8

Забезпечить унікальність полів email тв phone . Система просто не надасть користувачу можливість додати клієнта з однаковими поштою, та телефоном.

**#7(5)** **Школьная задача**

Дан код

$b = \_\_\_;

if ($b == 1 && $b==2 && $b==3){

echo(«ok»);

}

Что нужно подставить в $x чтобы было выведено в консоль «ok».

$b = true;

if ($b == 1 && $b == 2 && $b == 3) {

 echo("ok");

}

Кожна значення, що є більшим 0, має значення TRUE;

**#8(5) Какое все зеленое!**

Есть вот такой кусок разметки и стилей страницы

<style>

table .c { color: red }

.a .c { color: green }

</style>

<table id="t" >

<tr>

<td class="c">Текст</td>

</tr>

</table>

Предложите несколько вариантов решений на JS, которые сделают слово “Текст” зеленым. Предложите варианты на чистом JS и с использованием jQuery. (Каждый вариант после 2-х прибавляет 1 балл, то есть за 8 вариантов Вы получите 5 + (8-2) = 11 баллов).

* На JS

*document*.getElementById("t").*className* = 'a'; - якщо #t має якийсь клас то видалить його і поставить замість нього .а

*document*.getElementsByClassName("c").style.color = "green";

*t*.className += ‘a’;

$(“#t”).addClass(‘a’);

*document*.getElementsByTagName("td").style.color = "green"

* На Jquery

$("#t tr").addClass('a'); Просто додасть класс а тегу <tr>

$("#t").toggleClass('a'); також додасть клас .а але якщо у <tr> він уже є то його видалить

$(“#t td”).addClass(‘a’);

$("td").css("color:green")ж

**#9(10) Тайные категории**

Структура отображения вложенных категорий упрощенно выглядит так:

<ul class="catalog-list">

        <li class="cat-item cat-item-9">Категория 1</li>

        <li class="cat-item cat-item-28"> Категория 2

<ul class="children">

                               <li class="cat-item cat-item-22">Вложенная 1</li>

                               <li class="cat-item cat-item-21">Вложенная 2</li>

</ul>

</li>

</ul>

**(5)** Напишите скрипт с использованием jQuery, который обеспечит следующее:

- в обычном состоянии список вложенных категорий свернут (скрыт)

- при наведении мыши список раскрывается

- при убирании мыши с родительской категории список вложенных опять скрывается

$(document).ready(function(){

 $('.catalog-list li ul').hide();

 $('.catalog-list li').hover(function () {

     clearTimeout($.data(this,'timer'));

     $('ul',this).stop(true,true).slideDown(200);

 }, function () {

     $.data(this,'timer', setTimeout($.proxy(function() {

         $('ul',this).stop(true,true).slideUp(200);

     }, this), 100));

 });

});

Спочатку ховаємо всі елементи з допомогою .hide(), При наведенні (.hover) показуємо знову(.sideDown()).

**#10** **(20) Товары за долАры**

**(5)** В таблице БД goods находятся товары (поля: id (уникальное), name). В таблице selected\_goods (одно поле id) находятся id отобранных товаров.

Вывести **одним** SQL запросом названия (name) отобранных товаров.

Предложите два варианта такого запроса (**+5** баллов за второй вариант).

Сравните быстродействие двух вариантов запросов (**+10** баллов).

1)Без підзапиту

[**SELECT**](http://localhost/pma/url.php?url=http%3A%2F%2Fdev.mysql.com%2Fdoc%2Frefman%2F5.5%2Fen%2Fselect.html&token=d20417f1962be565b4a4927d7dabf1db) goods.name **FROM** goods, selectedGoods

**WHERE** goods.id [=](http://localhost/pma/url.php?url=http%3A%2F%2Fdev.mysql.com%2Fdoc%2Frefman%2F5.5%2Fen%2Fcomparison-operators.html%23operator_equal&token=d20417f1962be565b4a4927d7dabf1db) selectedGoods.id

2) З правим обєднанням

[**SELECT**](http://localhost/pma/url.php?url=http%3A%2F%2Fdev.mysql.com%2Fdoc%2Frefman%2F5.5%2Fen%2Fselect.html&token=d20417f1962be565b4a4927d7dabf1db) goods.name **FROM** goods

[**RIGHT**](http://localhost/pma/url.php?url=http%3A%2F%2Fdev.mysql.com%2Fdoc%2Frefman%2F5.5%2Fen%2Fstring-functions.html%23function_right&token=d20417f1962be565b4a4927d7dabf1db) **JOIN** selectedGoods **ON** goods.id [=](http://localhost/pma/url.php?url=http%3A%2F%2Fdev.mysql.com%2Fdoc%2Frefman%2F5.5%2Fen%2Fcomparison-operators.html%23operator_equal&token=d20417f1962be565b4a4927d7dabf1db) selectedGoods.id

        3) З підзапитом

**SELECT** name **FROM** goods

**WHERE** id [**IN**](http://localhost/pma/url.php?url=http%3A%2F%2Fdev.mysql.com%2Fdoc%2Frefman%2F5.5%2Fen%2Fcomparison-operators.html%23function_in&token=d20417f1962be565b4a4927d7dabf1db) ([**SELECT**](http://localhost/pma/url.php?url=http%3A%2F%2Fdev.mysql.com%2Fdoc%2Frefman%2F5.5%2Fen%2Fselect.html&token=d20417f1962be565b4a4927d7dabf1db) id **FROM** selectedGoods)

Найкращим за швидкодією буде перший запит так, як об’єднання та підзапити дуже уповільнюють час виконнаня запиту.

**#11(7) На волю, в пампасы**

**(7)** Опишите несколько способов переадресации на другой URL - из PHP и из JS.. Перечислите достоинства и недостатки каждого способа.

* PHP

Тут ми прописуемо:

<?php  header('Location: http://www.google.com'); exit(); ?>

Також можна відправити функції з різними кодами статусу сервера наприклад :

<?php  header('Location: http://www.google.com', true, 301);  exit(); ?>

Інші варіанти переадресації :

<?php header("Content-Location: <http://www.google.com>'"); ?>

<?php header("Request-URI: <http://www.google.com>'"); ?>

* JS

<script language="JavaScript">

window.location.href = "http://www.google.com"

</script>

Також можна перенаправляти метатегом

<meta http-equiv="refresh" content="0;url=http://www.google.com" />

**#12(20) Обмен по текущему курсу**

**(5)**На странице работает яваскрипт, выполняя определенные действия через промежуток времени *delta*

Значение delta задается в PHP-скрипте. Как это значение передать яваскрипту?

  <?php

 $delta = 5;

 echo '<script language="javascript">

     var *delta* = "'.$delta.'";

 </script>';

  ?>

По суті викликати JS посеред блоку PHP.

        Або ж можна в самому скрипті зробити щось типу :

  <?php

$delta = 5;

  ?>

 <script>

        var delta = “<?php echo $delta;?>”;

 </script>

**(15)** Другая ситуация. На странице есть два независимых, не вложенных друг в друга аяксовых обработчика событий, выполненных с использование jQuery ajax, причем второй должен принять данные от первого.

Обработчики между собой не связаны и обрабатывают разные события.

Как передать результат выполнения первого аякс-обработчика второму?

function *makeAjaxs*() {

$.ajax({

    type: "POST",

    contentType: "application/json; charset=utf-8",

    url: "handler.php", /\*Target function that will be return result\*/

    data: '{"data":"' + param + '"}',

    dataType: "json",

    success: function(data) {

        $.ajax({

            type: "POST",

            contentType: "application/json; charset=utf-8",

            url: "handler2.php",

            data: '{"data":"' + data + '"}',

            dataType: "json",

            success: function(data) {

                alert("Success!");

            }

error: function(result) {

    alert(“Error! Second AJAX ERROR!”);

}

        });

    }

    error: function(result) {

           alert(“Error! First AJAX ERROR!”);

    }

});

}

Один AJAX треба викликати в методі SUCCESS другого та завдаяки цьому передати інформацію вже з першого AJAX запита в інший.

**#13(25)Hi-Load News**

Представьте себе, что у Вас есть ресурс - агрегатор новостей. Боты сыплют новости со всего Интернета в табличку news. Таблица содержит поля time и text - соотв. время новости и текст новости.

На главную страницу сайта нужно выдать пять последних, то есть самых свежих новостей. Свежесть определяется значением поля time.

(5) Предложите SQL-запрос, который это сделает, используя ORDER BY

[**SELECT**](http://localhost/pma/url.php?url=http%3A%2F%2Fdev.mysql.com%2Fdoc%2Frefman%2F5.5%2Fen%2Fselect.html&token=d20417f1962be565b4a4927d7dabf1db) \* **FROM** news

**ORDER** **BY** TIME **DESC**

**LIMIT** 5

(10) Есть предположение, что сортировка с помощью ORDER BY в запросе может его резко затормозить на больших - 1 миллион записей и более - таблицах.

Предложите SQL-запрос, обходящий это явление.

        Можна використовувати індекси, так як вони підвищують ефективність виконання запитів. Тобто ми можемо створиити індекс на поле time.

CREATE INDEX TimeIndex ON news(time);

(10) Исследуйте и сравните быстродействие двух описанных выше запросов для разного кол-ва записей в таблицах.

Якщо ми не будемо створювати додаткових індексів, і це буде основне поле, по якому ми будемо робити запити, то для будь-якої кількості записів за допомогою індексу запити будуть виконуватись швидше.

**#14(15) Git...**

(5) Какую операцию с файлами очень не рекомендуется делать при использовании в проекте системы контроля версий Git?

Загалом не рекомендується використовувати rebase до вітки яка була запушена в публічний репозиторій, бо можуть з’явитися проблеми зв’язані з merge’м старої вітки (яка зберігається в когось, хто не встиг її викачати з репозиторію), це може привести до конфліктів і появи коммітів з одинаковими датами.

(10) Предложите варианты структуры системы работы над проектом с использованием Git. Нужно:

- обеспечить работу нескольких разработчиков над проектом и внесение ими всеми изменений в код

- обеспечить четкое понимание, кто, какие изменения вносил и когда.

- выдачу финального кода в веб-директорию с возможностью быстро откатить изменения обратно

По суті все це Git вже забезпечує. Краще щоб всі працювали над різними файлами. Завжди є вітка, в якій міститься робоча версія продукту. Також є вітка, для забезпечення можливості додавання якогось нового функціоналу. Також Git забезпечує можливість в будь-який момент можна відкотитися до будь-якої версії.

**#15(15)Linux forever!**

Вы купили хостинг - выделенный сервер. Он чист как слеза - на нем есть только Linux (Debian) и SSH для доступа через сеть.

Ваш сервер доступен только через сеть - через SSH.

Опишите Ваши действия по инсталляции на сервер всего ПО, необходимого для превращения его в рабочий сервер для веб-разработки и работы создаваемых веб-приложений.

Просто встановлюємо LAMP на сервек

Нам необхідно оновити репозиторій :

apt-get update

Встановлюємо Apache :

apt-get install apache2

Команда необхідна для того щоб взнати VPS IP адресу :

ifconfig eth0 | grep inet | awk '{ print $2 }'

Встановлення БД MySQL :

apt-get install mysql-server

mysql\_secure\_installation  
Встановлення PHP :

apt-get install php5 php-pear php5-mysql

Перезавантаження Apache :

service apache2 restart