

الكتاب الشامل في تعلم برمجة كلاسات PHP

فهرس الكتاب:

- ١ - كلاس php: أساسيات الـ php classes
- ٢ - كلاس php: خلاصة الأساسيات المهمة لـ php classes
- ٣ - كلاس php: الدرس الثالث: الاستغلال العملى لكلاس قواعد البيانات
- ٤ - كلاس php: الدرس الرابع: استغلال كلاس رفع الملفات
- ٥ - كلاس php: الدرس الخامس: استغلال كلاس كتابة وعرض المقالات
- ٦ - كلاس php: الدرس السادس: كيف تفكر وتخطط كمبرمج قبل برمجة كلاسك الخاصة
- ٧ - كلاس php: الدرس السابع: كيف تبرمج كلاس دخول صفحة محمية
- ٨ - كلاس php: الدرس الثامن: شرح برمجة كلاس سجل الزوار من الصفر

كلاس php : أساسيات ال php classes :

أولاً: صورته للفهرس الخاص بالدرس والذي سنقوم بشرحه خطوه خطوه بأذن الله:



نبدأ على بركة الله.....

اولا: class :

تعريف الكلاس class: هي عبارة عن مجموعة من المتغيرات \$ وال **functions** يتم استخدامها سويا، بحيث ينتج لنا ما يسمى الكلاس، والتي توفر الكثير من الوقت، فهناك كلاسات جاهزه يستخدمها الكثير من المبرمجين في برمجياتهم ، فمجرد معرفتهم بأساسيات استخدامها، وهذا ما سنشرحه، يمكنك أن تستخدم اي كلاس تريدها، وحتى ولو لم تكن مبرمج محترف وقادر على كتابة اي كلاس ، فهي ستعطى لك دفعه برمجية قوية في برمجياتك اذا كانت قدراتك محدوده في البرمجه.

مثال: مثلا يوجد كلاس اسمها كلاس قواعد البيانات ، وكودها موجود [هنا](#) , وهي عبارة عن كلاس يتم استخدامها للتعامل مع قواعد البيانات ، من اتصال وجلب وادراج وحذف وادخال بيانات.....الخ

تخيل انك بمعرفه بسيطه بأساسيات الكلاسات، ستتمكن استخدام هذا الملف الذي قد يكون معقد بالنسبة لك بكل سهولة، وهذا ما سيثبته درسنا ان شاء الله , والدروس القادمة بأذن الله.....

الصيغة العامه للاستخدام:

كود:

```
class classname {
$......
function.....
}
```

مثال عملي : نفرض اني أريد عمل كلاس تستخدم لكتابة اسمي:
سيكون الكود كالاتي:

كود:

```
<?php
class myname {

    function myname(){

        echo "my name is mohamed amin\n";
        echo "<br>";
    }
}
?>
```

الان ، سوينا هذه الكلاس البسيطة جدا، لكن نريد ان نعرف كيف سنستخدمها حتى نظهر الجملة
my name is mohamed amin

حتى نقوم بذلك لابد ان نقوم بخطوتين:

- 1- عمل متغير وتحديد هذا الكلاس لاستخدامها باستخدام الكلمة **new**
- 2- الدخول لاي **function** او متغير \$ داخل الكلاس لاستخدامه باستخدام علامة الدخول >-

سنقوم بعمل الخطوه الاولى:
نسمى اى متغير ونكتب الكلمة + **new** اسم الكلاس السابق

كود:

```
$use=new myname;
```

الحين قمنا بعمل متغير وحددنا الكلاس **myname** لاستخدامها
نقوم الان بعمل الخطوه الثانيه:

نكتب نفس اسم المتغير اللى مستخدم الكلاس + ندخل للفانكشن **myname** باستخدام >-

كود:

```
$use->myname();
```

الان لو كتبنا الكود كامل :

كود:

```
class myname {  
  
    function myname(){  
  
        echo "my name is mohamed amin\n";  
        echo "<br>";  
    }  
}  
  
$use=new myname;  
$use->myname();  
  
?>
```

لو استدعيت الملف فى المتصفح، ستجد الجملة راح تظهر لك **my name is mohamed amin**

حتى نهضم هذه الجزئيه ، نأخذ أكثر من مثال عملي.....

مثال عملي ١:

كود:

```
<?php
class personal {
    public $site="www.advphp.com";

    function fullname(){
        echo " my private site is {$this->site} ";
    }
}

$usepersonal= new personal ;
$usepersonal->fullname();

?>
```

النتائج:

كود:

```
my private site is www.advphp.com
```

استفدنا من المثال السابق، انه في حال أردت الدخول لاي متغير او فانكشن يمكنك استخدام الجملة

كود:

```
$this->var
```

حيث var هي اسم المتغير المراد الدخول له، ولاحظ اننا لن نكتب العلامة \$ قبل اسم المتغير

مثال عملي ٢:

كود:

```
<?php
class simplemath {
    public $num=18;
    function math(){
        $num2= 2;
        $sum = $this->num + $num2;
        echo " the sum of $this->num and $num2 is $sum ";
    }
}

$usepersonal= new simplemath ;
$usepersonal->math();

?>
```

النتائج:

كود:

```
the sum of 18 and 2 is 20
```

ثانياً extends :

ال extends بكل اختصار هي تكملة او اضافة ممكن تضيفها لاي كلاس، تخيل انك كاتب كلاس كبيره جدا ومعقده، ونفرض انك تود ان تضيف لها بعض السطور البرمجية بدون الدخول فيها، لذلك سنقوم بعمل كلاس جديد امتدادا للكلاس الاصلية بحيث تشمل هذه الكلاس الممتده الكلاس الاصلية بكافة اوامرها+الكلاس الجديد بكافة اوامرها.....

الصيغه العامه للاستخدام:

كود:

```
class classname2 extends classname1 {
$......
function.....
}
```

مثال عملي:

لو خذنا الكلاس السابقة والتي كودها هو:

كود:

```
class myname {

    function myname(){

        echo "my name is mohamed amin\n";
        echo "<br>";
    }
}

$use=new myname;
$use->myname();
```

نريد ان نضيف للكلاس السابق كلاس ممتد لها ليظهر اسمي المستعار الثاني لذلك سيكون الكود كالاتي:

كود:

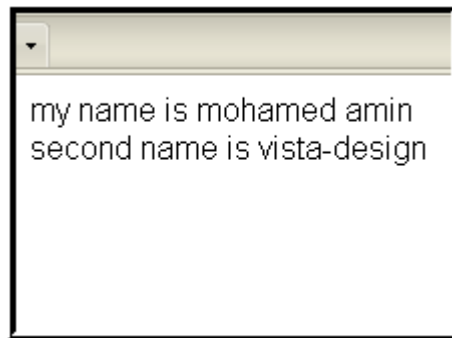
```
class secondname extends myname{  
    function displaysecond(){  
        echo "second name is vista-design";  
    }  
}  
  
$useextend= new secondname();  
$useextend->displaysecond();
```

الناتج لهذه الكلاس سوف يكون كالآتي:

كود:

```
my name is mohamed amin  
second name is vista-design
```

شوفت الناتج: يشمل الكلاس الاصلية + الكلاس الممتده.....



ثالثا: **constructors** :

ال **constructors** او المشيد ، هي اول **function** يتم كتابتها فى الكلاس ,والتي يتم استدعاءها تلقائيا بدون استدعاءها وتنفيذها بمجرد استخدام الكلاس.....

الصيغة العامة:

كود:

```
class classname {  
    function __construct() {  
    }  
}
```


الحين حتى نثبت انه يتم استدعاءها تلقائيا بدون ان ندخل >- لل , function سنقوم بعمل كلاس بسيطه وسنقوم بعمل متغير جديد لاستخدام هذه الكلاس فقط، ونشوف الناتج...

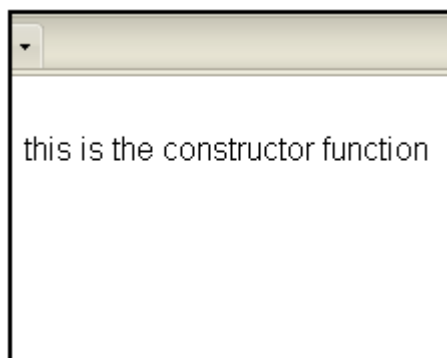
كود:

```
class useconstructor{  
    function __construct(){  
        echo "<br>";  
        echo "this is the constructor function";  
    }  
}  
  
$useconstructor= new useconstructor();
```

الناتج:

كود:

```
this is the constructor function
```



لاحظ اننا لم نستخدم عبارة الدخول لل function كالاتى:

كود:

```
$useconstructor->construct();
```

لان هذه الفانكشن يتم استدعاءها بمجرد عمل متغير لاستخدام الكلاس.....

رابعاً: **visibility** :

ال **visibility** فى الكلاسات، يقصد بها تصاريح استخدام ما بداخل الكلاس، فهناك ٣ أنواع من التصاريح: **public-** وهو التصريح العام، والذي يصرح لاستخدام الكلاس من اى مكان سواء داخل او خارج الكلاس **protected-** يقتصر على استخدام الكلاس من خلال الكلاسات المتوارثه فقط (الممتده) **private-** وهو التصريح الخاص المقتصر فقط على استخدام الكلاس نفسها فقط

الصيغة العامة:

كود:

```
class classname {
public $var="";
protected $var="";
private $var="";
}
```

مثال عملي:

كود:

```
class visibile{

    public $public='public';
    protected $protected='protected';
    private $private='private';

    function usevisible(){

        echo $this->public;
        echo $this->protected;
        echo $this->private;

    }
}

$vis= new visibile();
echo $vis->public; // works
echo "<br>";
echo $vis->usevisible();
```

النتائج:

كود:

```
public
publicprotectedprivate
```

```
public
publicprotectedprivate
```

خامسا: scope resolution operator :

ال SRO هي خاصية في الكلاسات تستخدم للإشارة لمتغير \$ او function داخل كلاس ما, ولا بد ان يكون المتغير حتى نشير له ان يكون ثابت static

مثال عملي:

كود:

```
class myname2 {
    public static $thename="mohamed amin";
    function declarescope(){
        echo self::$thename;
    }
}
$use= new myname2;
$use->declarescope();
```

النتائج:

كود:

```
mohamed amin
```

```
mohamed amin
```

سادسا: constant :

وهو المتغير الثابت في الكلاس، الذي قيمته لن تتغير ايما اردت استخدامه من الكلاس

الصيغة العامة:

كود:

```
class classname{  
  
const varname="";  
  
}
```

حيث ال varname هي اسم المتغير، ولاحظ اننا لم نلحق المتغير بالعلامة \$

مثال عملي:

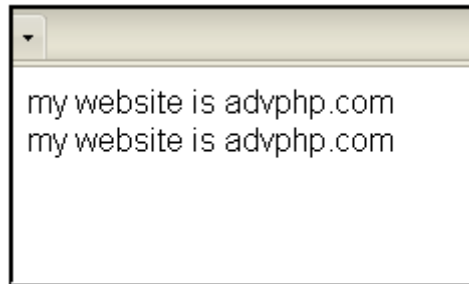
كود:

```
class myclass{  
  
    const somevar="my website is advphp.com";  
  
    function useconstant(){  
        echo self::somevar;  
    }  
}  
  
echo myclass::useconstant();  
echo "<br>";  
echo myclass::somevar;
```

النتائج:

كود:

```
my website is advphp.com  
my website is advphp.com
```



بهيك ، نكون انتهينا من ما يخص الكلاسات من أساسيات مهمه للدروس القادمة ،
والدروس القادمة ستكون تكثيف قوى فى الكلاسات، بحيث سنعرض أكثر من كلاس ونشرحها سويا، مع
استغلالها بالكامل، بحيث نكون هضمنا موضوع الكلاسات سويا بشكل جيد

ملاحظة: حتى تتذكر كافة أساسيات الكلاسات: لقد قمت بعمل ملف بسيط به كافة أكواد الاساسيات والأمثلة
العملية مجعه سويا ،

الكود:

كود:

```
<?php
////////////////////////////////////
# class file learning
# done by vista-design
# www.advphp.com
////////////////////////////////////
class myname {

    function myname(){

        echo "my name is mohamed amin\n";
        echo "<br>";
    }
}

$use=new myname;
$use->myname(); // u will find that the echo is doubled as the
function has the same name of the class,so it consider a constructor
function not a regular function

////////////////////////////////////
#extends
////////////////////////////////////

/* class secondname extends myname{

    function displaysecond(){

        echo "second name is vista-design";
    }
}
}
```

```

    $useextend= new secondname();
    $useextend->displaysecond(); // this will print (my name is.... +
second name is....)
    */

////////////////////////////////////
#autoload
////////////////////////////////////

    //auto loading of certain class from afile

    /*function __autoload($classname){

        require_once $classname. '.php';
    }

    $useclass = new myclass1();
    $useclass = new myclass2(); */

////////////////////////////////////
#construcotrs
////////////////////////////////////
/* class useconstructor{

    function __construct(){
        echo "<br>";
        echo "this is the constructor function";
    }
}

    $useconstructor= new useconstructor(); // as u see the function will
load automatically without accessing using $this
    */

////////////////////////////////////
#visibility using public protected private
////////////////////////////////////

/*
class visibile{

    public $public='public';
    protected $protected='protected';
    private $private='private';

    function usevisible(){

        echo $this->public;
        echo $this->protected;
        echo $this->private;

    }
}

    $vis= new visibile();
    echo $vis->public; // works
    echo "<br>";

```

```

echo $vis->usevisible();

*/

////////////////////////////////////
# scope resolution operator :: allows access to static, constant,
and overridden members or methods of a class.
////////////////////////////////////

/*
class myname2 {

    public static $thename="mohamed amin"; // should be static if u
want to use ::

    function declarescope(){

        echo self::$thename;

    }

}

$use= new myname2;
$use->declarescope();

*/

////////////////////////////////////
#constant: don't use the $ symbol to declare or use them ,
#The value must be a constant expression, not (for example) a
variable, a class member, result of a mathematical operation or a
function call.
////////////////////////////////////

class myclass{

    const somevar="mohamed amin"; // don't do $somevar

    function useconstant(){

        echo self::somevar;

    }
}

echo myclass::useconstant();
echo "<br>";
echo myclass::somevar;

?>

```



كلاس php: خلاصة الأساسيات المهمة لـ php classes

اختصارا للدرس السابق الخاص بالأساسيات ، والتي تعتبر أهم نقطة في دروس الكلاسات, قمت بعمل جدول مبسط لكافة الأساسيات شامل الخاصية والصيغة العامة والمثال العملي ، حتى تتذكر او ترجع لهذا الجدول في حال نسيت اى نقطة في الأساسيات دون الرجوع لكافة الدرس السابق

[انظر لآخر الصفحة لترى الخلاصة بالكامل](#)

| مثال عملي | الصيغة العامة | الخاصية |
|---|---|------------------|
| <pre>class myname { function myname(){ echo "my name is vista-design\n"; echo " "; } }</pre> | <pre>class classname{variables...functions}</pre> | class |
| <pre>\$useclass= new myname; \$useclass->myname(); //الناتج >> my name is vista- design</pre> | <pre>\$var= new classname; \$var->functionname or variable name</pre> | استخدام class |
| <pre>class myage extends myname{ function myage(){ echo "my age is 24 years old"; } }</pre> <pre>//الناتج >>myname is vistadesign myage is 24 years old</pre> | <pre>class classname2 extends classname1 {variables...functions}</pre> | extends |
| <pre>class myname{ function __construct(){ echo "this function is the constructor of the class"; } }</pre> | <pre>function __construct() {}</pre> | constrtuctor |

| | | |
|---|--|----------------------------|
| } | | |
| <pre> class visibile{ public \$public='public'; protected \$protected='protected'; private \$private='private'; function usevisibile(){ echo \$this->public; echo \$this->protected; echo \$this->private; } } </pre> | <pre> public \$ protected \$ private \$ </pre> | visibility |
| <pre> class myname2 { public static \$name="mohamed amin"; function declarescope(){ echo self::\$name; } } </pre> | <pre> :: </pre> | scoper resolution operator |
| <pre> class myclass{ const somevar="mohamed amin"; function useconstant(){ echo self::somevar; } } </pre> | <pre> const variable </pre> | constant |



كلاس php: الدرس الثالث: الاستغلال العملى لكلاس قواعد البيانات

نستكمل معكم اخوانى سلسلة دروس كلاسات php, وان شاء الله سوف نبدأ الاستغلال الحقيقى للكلاسات بدءا من هذا الدرس.

بالنسبة لاستغلال الكلاسات ، سوف تكون دروس هذه الجزئيه مقسمه لجزئين:

- 1- ثلاثة دروس تطبيقات عملية لكيفية استغلال الكلاسات الجاهزه
- 2- ثلاثة دروس تطبيقات عملية لكيفية كتابة كلاسات بسيطه من ابداعك باستخدام التفكير الذاتى oop

وبدأ من هذا الدرس سوف نبدأ باول درس لاستغلال الكلاسات الجاهزة ، حتى نتعود على أشكال كتابة كود الكلاسات كاملا، بحيث تكون دربت نفسك عليها، وأخذت فكرة بسيطة عن طريقة كتابتها قبل الخوض فى الثلاث دروس الأخرى الخاصة بكتابة الكلاسات المكتوبه بيدك أنت.....

نبدأ على بركة الله.....

استغلال كلاس قواعد البيانات



قبل أن أبدأ الشرح [حمل الملفات الآتية من هنا](#)

وتشمل ٣ ملفات:

Mysql.class.php- وهو ملف كلاس الاتصال + التأمين بقاعدة البيانات
 QueryResult.class.php- وهو خاص بجلب البيانات من القاعدة بطرق مختلفة (جدول...xml-الخ)
 usage.php- وهو خاص باستغلال هاتان الكلاسان (واللى راح نكتبه بأنفسنا ونتدرب عليه)

اولا **Mysql.class.php**:

الكود:

كود:

```

<?php
/**
 * Mysql.class.php
 * @author Charles Reace (www.charles-reace.com)
 * @version 1.0
 * @license http://opensource.org/licenses/gpl-license.php GNU Public License
 */
/**
 * class used for db query results
  
```

```

*/
require_once 'QueryResult.class.php';

/**
 * "Singleton" pattern MySQL database class
 * @package Database
 */
class Mysql
{
    // USER MODIFIABLE DATABASE SETTINGS //

    private $dbHost = 'localhost';
    private $dbUser = 'root';
    private $dbPwd = 'root';
    private $database = 'class';
    private $connx = NULL;
    private $error = '';

    // DO NOT MODIFY BELOW THIS LINE //

    private static $instance;

    /**
     * Private constructor: prevents direct creation of object
     */
    private function __construct()
    {
        $this->connect();
    }

    /**
     * Singleton method (only allow one instance of this class)
     * @return object
     */
    public static function singleton()
    {
        if (!isset(self::$instance))
        {
            $c = __CLASS__;
            self::$instance = new $c;
        }

        return self::$instance;
    } // end singleton()

    /**
     * Prevent users from cloning this instance
     * @return void
     */
    public function __clone()
    {
        trigger_error('Clone is not allowed.', E_USER_ERROR);
    }

    /**
     * Connect to MySQL and select database
     * @return boolean
     */
    private function connect()
    {
        $connx = @mysql_connect($this->dbHost, $this->dbUser, $this->

```

```

>dbPwd);
    if($connx != FALSE)
    {
        $this->connx = $connx;
        $db = mysql_select_db($this->database, $this->connx);
        if($db == FALSE)
        {
            $this->error = "Unable to select DB: " . mysql_error();
            user_error($this->error, E_USER_WARNING);
            return(FALSE);
        }
        return(TRUE);
    }
    $this->error = "Unable to connect to DBMS: " . mysql_error();
    user_error($this->error, E_USER_WARNING);
    return(FALSE);
} // end connect()

/**
 * Get database connection resource ID
 * @return resource
 */
public function getConnection()
{
    return($this->connx);
}

/**
 * Sanitize input for use in SQL
 * @param string|integer|float $input
 * @return string|integer|float
 */
public function sanitize($input)
{
    $input = trim($input);
    if(!is_numeric($input))
    {
        if(get_magic_quotes_gpc())
        {
            $input = stripslashes($input);
        }
        $input = "'" . mysql_real_escape_string($input) . "'";
    }
    return($input);
}

/**
 * Execute SELECT query (or any query that returns result rows)
 * @param string $sql
 * @return object
 */
public function select($sql)
{
    if(!$this->connx)
    {
        $this->error = "Cannot process query, no DB connection.";
        user_error($this->error, E_USER_WARNING);
        return(FALSE);
    }
    $result = mysql_query($sql, $this->connx);
    if($result)

```

```

    {
        if(mysql_num_rows($result))
        {
            return(new QueryResult($result, $this->connx));
        }
        else
        {
            return(0);
        }
    }
else
{
    $this->error = "Query failed ($sql): " . mysql_error();
    user_error($this->error, E_USER_WARNING);
    return(FALSE);
}
}

/**
 * Execute query that does not return result rows (e.g.: INSERT)
 * @param string $sql
 * @return integer
 */
public function modify($sql)
{
    if(!$this->connx)
    {
        $this->error = "Cannot process query, no DB connection.";
        user_error($this->error, E_USER_WARNING);
        return(FALSE);
    }
    $result = mysql_query($sql, $this->connx);
    if($result)
    {
        return(mysql_affected_rows($this->connx));
    }
    else
    {
        $this->error = "Query failed ($sql): " . mysql_error();
        user_error($this->error);
        return(FALSE);
    }
}
}
}

```

ما يهمنا معرفته في هذه الكلاس لاستغلالها فيما بعد هو الاتي:

١-بيانات الاتصال بالقاعدة:

كود:

```

        private $dbHost = 'localhost';
private $dbUser = 'root';
private $dbPwd = 'root';
private $database = 'class';
private $connx = NULL;
private $error = '';

```

٢- تأمين الكلاس من الاتصال بقاعدة البيانات: والتأمين هنا لا يقصد به تأمين البيانات، لكن تأمين الكلاس نفسها من حيث الخصوصية في الاتصال+private منع تكرار الاتصال واستنساخه+الاتصال الاحادي غير المتعدد-singleton

كود:

```
private function __construct()
{
    $this->connect();
}

/**
 * Singleton method (only allow one instance of this class)
 * @return object
 */
public static function singleton()
{
    if (!isset(self::$instance))
    {
        $c = __CLASS__;
        self::$instance = new $c;
    }

    return self::$instance;
} // end singleton()

/**
 * Prevent users from cloning this instance
 * @return void
 */
public function __clone()
{
    trigger_error('Clone is not allowed.', E_USER_ERROR);
}
```

3- فانكشن الاتصال بقاعدة البيانات:

كود:

```
private function connect()
{
    $connx = @mysql_connect($this->dbHost, $this->dbUser, $this->dbPwd);
    if($connx != FALSE)
    {
        $this->connx = $connx;
        $db = mysql_select_db($this->database, $this->connx);
        if($db == FALSE)
        {
            $this->error = "Unable to select DB: " . mysql_error();
            user_error($this->error, E_USER_WARNING);
            return(FALSE);
        }
        return(TRUE);
    }
    $this->error = "Unable to connect to DBMS: " . mysql_error();
    user_error($this->error, E_USER_WARNING);
    return(FALSE);
} // end connect()
```

4- فانكشن اختيار وعد عدد صفوف الجدول

كود:

```
public function select($sql)
{
    if(!$this->connx)
    {
        $this->error = "Cannot process query, no DB connection.";
        user_error($this->error, E_USER_WARNING);
        return(FALSE);
    }
    $result = mysql_query($sql, $this->connx);
    if($result)
    {
        if(mysql_num_rows($result))
        {
            return(new QueryResult($result, $this->connx));
        }
        else
        {
            return(0);
        }
    }
    else
    {
        $this->error = "Query failed ($sql): " . mysql_error();
        user_error($this->error, E_USER_WARNING);
        return(FALSE);
    }
}
```

هذا ما يهمننا في هذه الكلاس ,

ثانيا: [QueryResult.class.php](#)

1- فانكشن جلب البيانات على هيئة جدول:

كود:

```
public function getTable($headers = FALSE, $labels = NULL)
{
    if(!mysql_data_seek($this->result, 0))
    {
        return(false);
    }
    $table = "<table class='dbresult'>\n";
    if($headers)
    {
        $table .= "<tr>";
        if(is_array($labels))
        {
            foreach($labels as $label)
            {
                $table .= "<th>$label</th>";
            }
        }
        else
        {

```



```

        $num = mysql_num_fields($this->result);
        for($six = 0; $six < $num; $six++)
        {
            $table .= "<th>".mysql_field_name($this-
>result,$six)."</th>";
        }
        $table .= "</tr>\n";
    }
    while($row = mysql_fetch_row($this->result))
    {
        $table .= "<tr>";
        foreach($row as $val)
        {
            $table .= "<td>$val</td>";
        }
        $table .= "</tr>\n";
    }
    $table .= "</table>\n";
    return($table);
}

```

2- فانشكن جلب البيانات على هيئة xml

كود:

```

public function getXml()
{
    mysql_data_seek($this->result, 0);
    $xml = "<?xml version='1.0' encoding='ISO-8859-1'?>\n<data>\n";
    $count = 1;
    while($row = mysql_fetch_assoc($this->result))
    {
        $xml .= " <record row='$count'>\n";
        foreach($row as $key => $val)
        {
            $xml .= " <$key>$val</$key>\n";
        }
        $xml .= " </record>\n";
        $count++;
    }
    $xml .= "</data>";
    return($xml);
} // end getXml()

```

هذا أهم أشياء والتي منها سنستغل الكلاس جيدا.....

الآن: نأخذ مثال عملي شامل لاستغلال هذه الكلاس



اولاً: عمل قاعدة البيانات:
١- نسوى قاعدة بيانات ونسميها class
٢- نسوى جدول نسميه users, يتكون من ٤ حقول id,name,country,email وهذا هو كود عمل القاعدة جاهز

كود:

```
-- phpMyAdmin SQL Dump
-- version 2.10.3
-- http://www.phpmyadmin.net
--
: localhostالمزود--
-- أنشئ في: ٢٣ يناير ٢٠١١ الساعة ١٨:٢٣
-- إصدار المزود: ٥,٠,٥١
-- PHP إصدار: ٥,٢,٦

SET SQL_MODE="NO_AUTO_VALUE_ON_ZERO";

--
: قاعدة البيانات `class`--
--
-----
--
: بنية الجدول `users`--
--

CREATE TABLE `users` (
  `id` int(11) NOT NULL auto_increment,
  `name` varchar(150) NOT NULL,
  `country` varchar(150) NOT NULL,
  `mail` varchar(150) NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=MyISAM DEFAULT CHARSET=utf8 AUTO_INCREMENT=5 ;

--
: إرجاع أو إستيراد بيانات الجدول `users`--
--

INSERT INTO `users` VALUES (1, 'vista-design', 'egypt',
'admin@vistacompany.org');
INSERT INTO `users` VALUES (2, 'mohamed amin', 'ksa', 'm@m.com');
INSERT INTO `users` VALUES (3, 'mazn eldosary', 'syria',
'sore@tron.com');
INSERT INTO `users` VALUES (4, 'ahmed elmohandsy', 'u.a.e',
'e555@ae.com');
```

ثانياً: عمل الاسكربت واستغلال الكلاسات:

- 1- سوى مجلد جديد ونسميه dbclass
- 2- ضع ملفين الكلاس mysql.class.php + queryresulty.class.php به
- 3- سوى ملف جديد اسمه usage.php

اولاً: افتح ملف mysql.class.php وقم بظبط اعدادت الاتصال بقاعدة البيانات

```
private $dbHost = 'localhost';  
private $dbUser = 'root';  
private $dbPwd = 'root';  
private $database = 'class';  
private $connx = NULL;  
private $error = '';
```

2- مرحلة الاستغلال وعمل الاسكربت:

الان نفتح ملف , usage.php وراح نكتب به كودنا لعمل هذا الاسكربت البسيط جداً:

1-اولاً نستدعي ملف الكلاس:

كود:

```
require_once 'Mysql.class.php';
```

2- كود الاستغلال:

راح نأخذ طرق مختلفة لاستغلال هذه الكلاس.....

اولاً:دائماً تذكر هذه الجملة للاستغلال (**متغير وتأمين وفانكشن في كلاس**)

متغير: وهو المتغير الخاص اللي راح نلعب عليه
تأمين : وهي فانكشن لتأمين اتصال الكلاس بالقاعدة
فانكشن: وهي اللي راح نلعب بها لجلب البيانات
في كلاس: وهي عمل كلاس جديد لضم كل السابق

نبدأ.....

1- نسوي كلاس اسمها مثلا: test

كود:

```
class Test {  
  
}
```

2- ثم نخصص متغير لنلعب عليه:

كود:

```
private $db;
```

3- ثم نسوي خطوة التأمين ، وراح تكون عبارة عن فانكشن constructor, راح يتم استدعاءها في كل مره يتم استخدام هذه الكلاس فيها

كود:

```
public function __construct()  
{  
    $this->db = Mysql::singleton();  
}
```

تفسير الكود: قمنا بعمل فانكشن constructor , ودخلنا للمتغير، ثم اشرن لفانكشن singleton الخاصه بالتأمين الموجوده بالكلاس mysql في ملف الكلاس mysql.class.php

4- ثم نسوي فانكشن : وهي اللي راح نستغل بها الكلاس: listcustomers: نسوي الفانكشن وراح نسميها:

كود:

```
public function listCustomers() {  
  
}
```

الان علينا ان نختار فانكشن ال select والتي تقوم بالاتصال وعد حقول الجدول , (راجع فانكشن select في ملف mysql.class.php لتعرف وظيفتها)

كود:

```
public function listCustomers() {  
  
    $result = $this->db->select('SELECT * FROM users');  
  
}
```

بعد ذلك ، بعد ما اتصلنا بالقاعدة وعدينا حقوق الجدول بواسطة هذه الفانكشن (select) سنقوم ب جلب بيانات الجدول باستخدام فانكشن gettable الموجوده في ملف queryresulty.class.php

كود:

```
public function listCustomers()  
{  
    $result = $this->db->select('SELECT * FROM users');  
    return $result->gettable(true);  
}
```

لاحظ ان ال true يقصد بها انه راح يرجع باسماء الحقول , طب، نفرض اننا نريد جلب بيانات الزبائن على هيئة ملف xml بسيطه، نستبدل فانكشن gettable السابقه والتي تجلب البيانات على هيئة جدول html مثل ما شوفنا , بفانكشن getxml الموجوده في نفس ملف الكلاس queryresulty.class.php (راجعها)

كود:

```
public function listCustomers()  
{  
    $result = $this->db->select('SELECT * FROM users');  
    return $result->getXml();  
}
```

```
<?xml version='1.0' encoding='ISO-8859-1'?>  
<data>  
  <record row='1'>  
    <id>1</id>  
    <name>mohamed</name>  
    <age>amin</age>  
  </record>  
  <record row='2'>  
    <id>2</id>  
    <name>ahmed</name>  
    <age>salm</age>  
  </record>  
</data>
```

بهيك نكون خالصنا كلاس الاستغلال.....
الان علينا تنفيذ هذه الكلاس.....
١- نسوى متغير لاستخدام الكلاس

كود:

```
$test = new Test();
```

2- نطبع فانكشن الاستغلال listcustomers

كود:

```
echo $test->listCustomers();
```

بحيث يعطينا كل عضويات الزبائن

| id | name | country | mail |
|----|------------------|---------|------------------------|
| 1 | vista-design | egypt | admin@vistacompany.org |
| 2 | mohamed amin | ksa | m@m.com |
| 3 | mazn eldosary | syria | sore@tron.com |
| 4 | ahmed elmohandsy | u.a.e | e555@ae.com |



كلاس php: الدرس الرابع: استغلال كلاس رفع الملفات

نستكمل معكم أخواني زوار موقعنا الكرام، فبعد الانقطاع المطول نظرا للظروف التي تعرضت لها بلدى الحبيبه مصر، فها أنا أرجع معكم بفضل الله لنستكمل ما بدأنا به،
فى هذا الدرس سوف نستكمل الجزء الخاص باستغلال الكلاسات الجاهزه، قبل ان نخوض فى دروس كتابة الكلاسات بأنفسنا، وهذا الدرس هو مثال لاستغلال كلاس جاهزه وهى كلاس رفع الملفات،

اولا: حمل ملفات الدرس من [هنا](#)

دعونا الآن نرسم خطة وفهرس شرح الدرس:

- 1- شرح تركيبه ملف الكلاس `upload_class.php`
- 2- شرح مثال عملي لاستغلال الكلاس لعمل اسكربت لرفع الملفات

الان ، انظر لهذا المخطط المبسط لتعرف باختصار فكرة الدرس:

شرح استغلال كلاس رفع الملفات

تركيبة الكلاس file_upload

functions

فانكشن رفع الملف

function upload

فانكشن جلب معلومات الملف

function get_uploaded_file_info

اسم الملف \$the_file

الاسم المؤقت \$the_temp_file

مجلد الملف \$upload_dir

تسمية الملف \$replace

التحقق من تسمية الملف \$do_filename_check

امتداد الملف \$extensions

أقصى حجم للملف \$max_size

متغيرات \$

File upload script:

Max. filesize = 256000 bytes.

Select a file...

تصفح...

New name?

(without extension!)

Replace ?

Validate filename ?

Submit

```
$max_size = 1024*250;
```

```
$my_upload = new file_upload; عمل متغير لاستغلال الكلاس
```

```
$my_upload->upload_dir = "/upload/files/";
```

```
$my_upload->extensions = array(".png", ".zip", ".pdf", ".rar");
```

```
if(isset($_POST['Submit'])) {
```

```
    $my_upload->the_temp_file = .....
```

```
    $my_upload->the_file = .....
```

```
    $my_upload->replace = .....
```

```
    $my_upload->do_filename_check = .....
```

```
    $new_name = .....
```

```
}
```

```
}
```

استغلال الكلاس

عالم مطوري الويب
لتعليم البرمجة والتصميم
advphp.com

اولا: شرح تركيبية ملف الكلاس upload_class.php

كما نرى فى المخطط السابق ان الكلاس هى عبارة عن مجموعة من المتغيرات والفانكشن.....
فى الكلاس سنحدد متغيرات مهمه وهى:

- 1- متغير خاص باسم الملف المرفوع
- 2- متغير خاص بالاسم المؤقت للملف المرفوع
- 3- متغير لمجلد الملف
- 4- متغير لاعادة تسمية الملف
- 5- متغير للتحقق من تسمية الملف
- 6- متغير لتحديد امتداد الملف
- 7- متغير لتحديد اقصى حجم للملف المرفوع
- + متغيرات أخرى.....

رمز:Code

```
var $the_file; //اسم الملف
var $the_temp_file; //الاسم المؤقت للملف
var $upload_dir; //مجلد الملف
var $replace; //لاعادة تسمية الملف المرفوع
var $do_filename_check; //التحقق من تسمية الملف
var $max_length_filename = 100;
var $extensions; // لتحديد الامتدادات
```

ثم فانكشنز functions للتعامل مع الملف المرفوع:

1- فانكشن لرفع الملف upload function

رمز:Code

```
function upload($to_name = "") {
    $new_name = $this->set_file_name($to_name);
    if ($this->check_file_name($new_name)) {
        if ($this->validateExtension()) {
            if (is_uploaded_file($this->the_temp_file)) {
                $this->file_copy = $new_name;
                if ($this->move_upload($this->the_temp_file, $this->file_copy)) {
                    $this->message[] = $this->error_text($this->http_error);
                    if ($this->rename_file) $this->message[] = $this->error_text(16);
                    return true;
                }
            }
        } else {
            $this->message[] = $this->error_text($this->http_error);
            return false;
        }
    } else {
        $this->show_extensions();
        $this->message[] = $this->error_text(11);
        return false;
    }
} else {
    return false;
}
```

2- فانكشن لجلب معلومات الملف المرفوع function get_uploaded_file_info

رمز: Code

```
function get_uploaded_file_info($name) {
    $str = "File name: ".basename($name)."\n";
    $str .= "File size: ".filesize($name)." bytes\n";
    if (function_exists("mime_content_type")) {
        $str .= "Mime type: ".mime_content_type($name)."\n";
    }
    if ($img_dim = getimagesize($name)) {
        $str .= "Image dimensions: x = ".$img_dim[0]."px, y =
        ".$img_dim[1]."px\n";
    }
    return $str;
}
```

فانكشنز كثيره أخرى.....
ويمكنك الرجوع لكود الكلاس كاملاً من الملفات المرفقة
الآن ، عرفنا تركيب الكلاس الأساسية والأشياء المهمة في الكلاس.....

ثانياً: شرح مثال عملي لاستغلال الكلاس لعمل اسكربت لرفع الملفات

سنقوم الآن بعمل اسكربت بسيط لرفع الملفات مستغلين هذه الكلاس السابقة

نشوف صورة للاسكربت:

File upload script:

Max. filesize = 256000 bytes.

Select a file...

New name? (without extension!)

Replace ?

Validate filename ?

- اولا: دعونا نعرف فكرة الاستغلال:
- 1- تحديد أقصى حجم للملف المرفوع
 - 2- تحديد مكان المجلد الذي سيتم رفع الملفات بداخله
 - 3- تحديد الامتدادات المسموح برفعها
 - 4- تنفيذ ما سيحدث عند الضغط على زر الرفع submit
 - 5- عمل فورم لرفع الملفات

نتوكل على الله ونبدأ بالاستغلال.....

1- نستدعي ملف الكلاس:

رمز:Code

```
include ("upload_class.php");
```

a1-نسوي متغير جديد لاستخدام الكلاس file_upload الموجوده داخل ملف الكلاس

رمز:Code

```
$my_upload = new file_upload;
```

2-نحدد اقصى حجم للملف المرفوع

رمز:Code

```
$max_size = 1024*250;
```

3- نحدد مكان المجلد الذي سيتم رفع الملفات بداخله:

رمز:Code

```
$my_upload->upload_dir = $_SERVER['DOCUMENT_ROOT']."/upload/files/";
```

لاحظ :

رمز:Code

```
$_SERVER['DOCUMENT_ROOT']
```

تعني المجلد الجذر للسيرفر ، يعنيwww

4- نحدد الامتدادات المسموح برفعها:

رمز:Code

```
$my_upload->extensions = array(".png", ".zip", ".pdf",".rar");
```

5- تنفيذ ما سيحدث عند الضغط على زر الرفع submit

رمز: Code

```
if(isset($_POST['Submit'])) {
    $my_upload->the_temp_file = $_FILES['upload']['tmp_name'];
    $my_upload->the_file = $_FILES['upload']['name'];
    $my_upload->http_error = $_FILES['upload']['error'];
    $my_upload->replace = (isset($_POST['replace'])) ? $_POST['replace']
    : "n";
    $my_upload->do_filename_check = (isset($_POST['check'])) ?
    $_POST['check'] : "n";
    $new_name = (isset($_POST['name'])) ? $_POST['name'] : "";
    if ($my_upload->upload($new_name)) {
        $full_path = $my_upload->upload_dir.$my_upload->file_copy;
        $info = $my_upload->get_uploaded_file_info($full_path);
    }
}
```

تفسير الكود السابق:

1- سنستخدم الحالة الشرطية لتنفيذ في حالة تم الضغط على الزر

رمز: Code

```
if(isset($_POST['Submit'])) {
    .....
    ....
    ...
}
```

2- نحدد مجلد مؤقت لتخزين الملف المرفوع

رمز: Code

```
$my_upload->the_temp_file = $_FILES['upload']['tmp_name'];
```

ملاحظة: تذكر دالة `$_files` تستخدم لرفع الملفات والصيغه العامه هي:

رمز: Code

```
$_files['input name']['name or type or size of uploaded file']
```

3- نحدد اسم الملف المرفوع

رمز: Code

```
$my_upload->the_file = $_FILES['upload']['name'];
```

4- نحدد ما اذا تم تعليم خيار اعادة تسميه الملف ام لا

رمز: Code

```
$my_upload->replace = (isset($_POST['replace'])) ? $_POST['replace'] : "n";
```

ملاحظه : تذكر ternary operator المستخدمه في المثال السابق وهي تحل محل else....if

مثال: الجملة السابقة تساوى بالظبط الجملة الاتيه:

رمز: Code

```
if(isset($_post['replace'])){
$replace=$_POST['replace'];
}
else
{
$replace="";
}
```

5- نحدد ما اذا تم كتابة اسم جديد للملف في خانة اعادة تسمية الملف يتم تنفيذ ذلك ، او يتم تركه كما هو بالاسم الاصلى

رمز: Code

```
$new_name = (isset($_POST['name'])) ? $_POST['name'] : "";
```

6- ثم نستخدم فانكشن ال upload لرفع الملف

رمز: Code

```
if ($my_upload->upload($new_name)) {
$full_path = $my_upload->upload_dir.$my_upload->file_copy;
$info = $my_upload->get_uploaded_file_info($full_path);
}
```

7- واخيرا نسوى الفورم الخاص برفع الملفات كما ترونه بالصوره السابقه للاسكربت:

رمز: Code

```
<h3>File upload script:</h3>
<p>Max. filesize = <?php echo $max_size; ?> bytes.</p>
<form name="form1" enctype="multipart/form-data" method="post"
action="<?php echo $_SERVER['PHP_SELF']; ?>">
<input type="hidden" name="MAX_FILE_SIZE" value="<?php echo
$max_size; ?>"><br>
<label for="upload">Select a file...</label>

<input type="file" name="upload" size="30">

<br clear="all">
<label for="name">New name?</label><input type="text" name="name"
size="20">
(without extension!) <br clear="all">
<label for="replace">Replace ?</label><input type="checkbox"
name="replace" value="y"><br clear="all">
<label for="check">Validate filename ?</label><input name="check"
type="checkbox" value="y" checked><br clear="all">
<input style="margin-left:120px;" type="submit" name="Submit"
value="Submit">
</form>
```

File upload script:

Max. filesize = 256000 bytes.

Select a file...

New name? (without extension!)

Replace ?

Validate filename ?

بهيك نكون عرفنا بشكل مبسط وخفيف كيفية استغلال كلاس رفع الملفات.....



كلاس php: الدرس الخامس: استغلال كلاس كتابة وعرض المقالات

نستكمل معكم زوار موقعنا الكرام دروس دورة كلاسات ال , php وها نحن نصل لنهاية المرحلة الاولى من الدورة وهو كيفية استغلال الكلاسات الجاهزه، والتي بعدها سأبدأ بدروس كيفية كتابة الكلاسات من الصفر من تخطيط وتنظيم وترتيب للأفكار بشكل يجعلك oriented لابتكار أفكار جديدة فى برمجياتك.

اولا: [حمل ملفات الدرس من هنا](#)

ملاحظه: هذه الكلاس خاصة بعمل مقالات على هيئة ملفات فى مجلدات ، وليست كمقالات مخزنه فى قاعدة البيانات ، وهذا كنوع من التسهيل وتقليل تعقيد الكود للسهوله

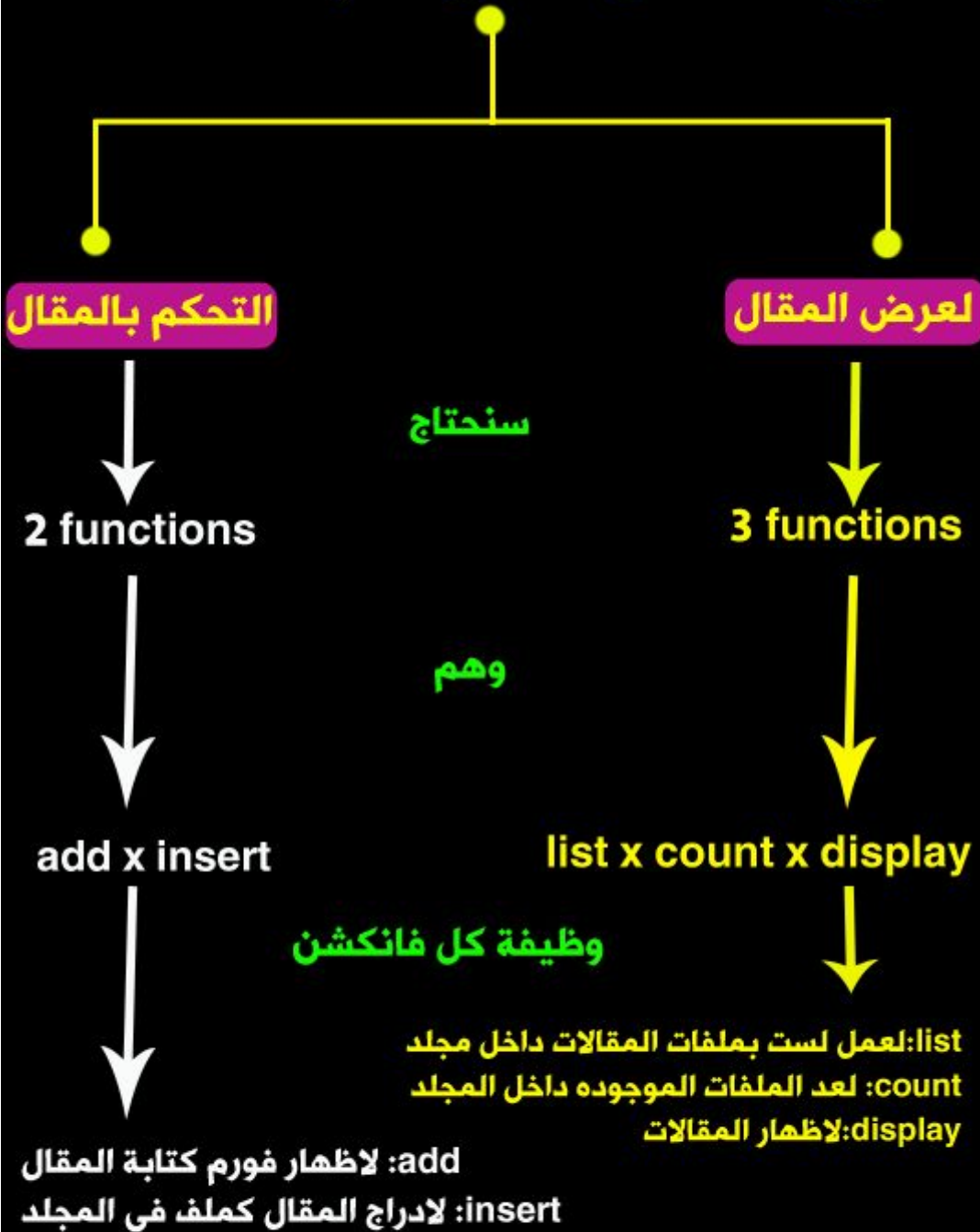
ثانيا: دعونا نعرف فهرس الدرس :

- 1- شرح كلاس كتابة وعرض المقالات
- 2- كيفية استغلال الكلاس بمثال عملي

اولا: شرح كلاس كتابة وعرض المقالات:

نشوف هذا المخطط لنفهم تركيبه الكلاس:

شرح استغلال كلاس كتابة المقالات



عالم مطورى الويب لتعليم البرمجة والتصميم

WWW.ADVPHP.COM



كما نلاحظ في المخطط، ان هذه الكلاس عبارة عن ٥ فانكشنز مقسمة ل:
اولا:- ٣ فانكشنز خاصة بعرض المقال وهم:

getNewsList: لعمل لست بملفات المقالات داخل مجلد
getNewsCount: لعد الملفات الموجوده داخل المجلد
displayNews: لاطهار المقالات
وفي المخطط وظيفة كل فانكشنز
وهذا هو كودهم:

رمز:Code

```
function getNewsList(){
    $this->newsList = array();

    // Open the actual directory
    if ($handle = @opendir($this->newsDir)) {
    // Read all file from the actual directory
    while ($file = readdir($handle)) {
        if (!is_dir($file)) {
            $this->newsList[] = $file;
        }
    }
    }

    rsort($this->newsList);
    return $this->newsList;
}

function getNewsCount(){
    if ($this->newsCount == -1) $this->getNewsList();
    $this->newsCount = sizeof($this->newsList);
    return $this->newsCount;
}

function displayNews(){
    $list = $this->getNewsList();

    echo "<table class='newsList'>";
    foreach ($list as $value) {
    $newsData = file($this->newsDir.DIRECTORY_SEPARATOR.$value);
    $newsTitle = $newsData[0];
        $submitDate = $newsData[1];
        unset ($newsData['0']);
        unset ($newsData['1']);

        $newsContent = "";
        foreach ($newsData as $value) {
            $newsContent .= $value;
        }
    echo "<tr> <th align='left'>$newsTitle</th>
        <th class='right'>$submitDate</th> </tr>";

    echo "<tr><td colspan='2'>".$newsContent."<br/></td></tr>";
    }
    echo "</table>";
    if (sizeof($list) == 0){
        echo "<center><p>No news at the
moment!</p><p>&nbsp;</p></center>";
    }
}
```

ثانياً: ٢ فانكشنز للتحكم بالمقال:
displayAddForm: لظهار فورم كتابة المقال
insertNews: لادارج المقال كملف في المجلد

وهذا هو كودهم:

رمز: Code

```
function displayAddForm(){
?>
    <script language="javascript" type="text/javascript"
src="/js/tiny_mce.js"></script>
<script language="javascript" type="text/javascript">
tinyMCE.init({
mode : "textareas",
theme : "advanced",
theme_advanced_buttons3 : "",
theme_advanced_toolbar_align : "center",
theme_advanced_toolbar_location : "top",
});
</script>
    <form action="<?php echo $_SERVER['PHP_SELF']; ?>" method="post">
    News title:<br/>
    <input type="text" name="title" size="40"/><br/><br/>
    Content:<br/>
    <textarea name="newstext" rows="15" cols="67"></textarea><br/>
    <center><input type="submit" name="submit" value="Save"
/></center>
    </form>

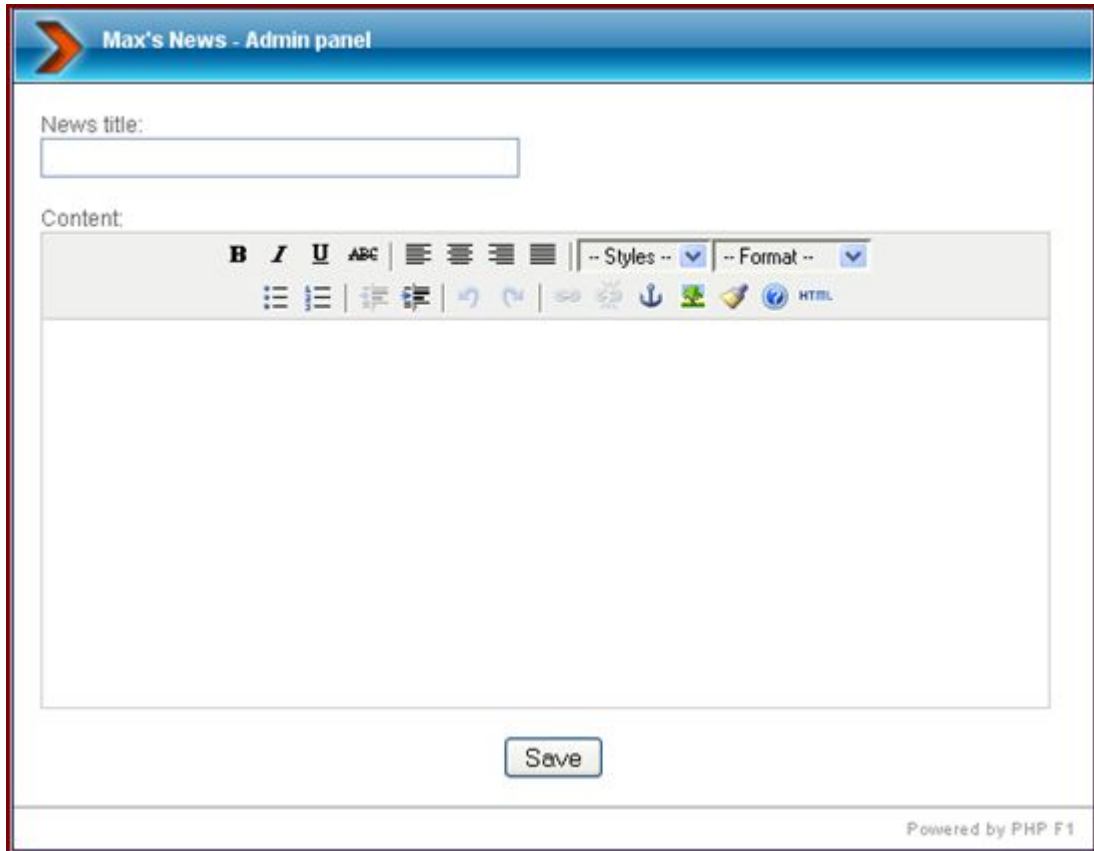
<?php
}

function insertNews(){
    $newsTitel = isset($_POST['title']) ? $_POST['title'] :
'Untitled';
    $submitDate = date('Y-m-d g:i:s A');
    $newsContent = isset($_POST['newstext']) ? $_POST['newstext'] :
'No content';

    $filename = date('YmdHis');
    if (!file_exists($this->newsDir)){
        mkdir($this->newsDir);
    }
    $f = fopen($this-
>newsDir.DIRECTORY_SEPARATOR.$filename.".txt", "w+");
    fwrite($f,$newsTitel."\n");
    fwrite($f,$submitDate."\n");
    fwrite($f,$newsContent."\n");
    fclose($f);

    header('Location:index.php');
}
}
```

وهذه صورة لوحة التحكم البسيطة لظهار فورم كتابة المقال



بحيث يكون كامل الكلاس كما هو مرفق بالملفات الخاصة بالدرس

ثانياً: كيفية استغلال الكلاس بمثال عملي
 استغلال هذه الكلاس من أسهل الأشياء، فمع لمسة جمالية ، فقط سنستدعي فانكشنز اظهر المقالات
displayNews

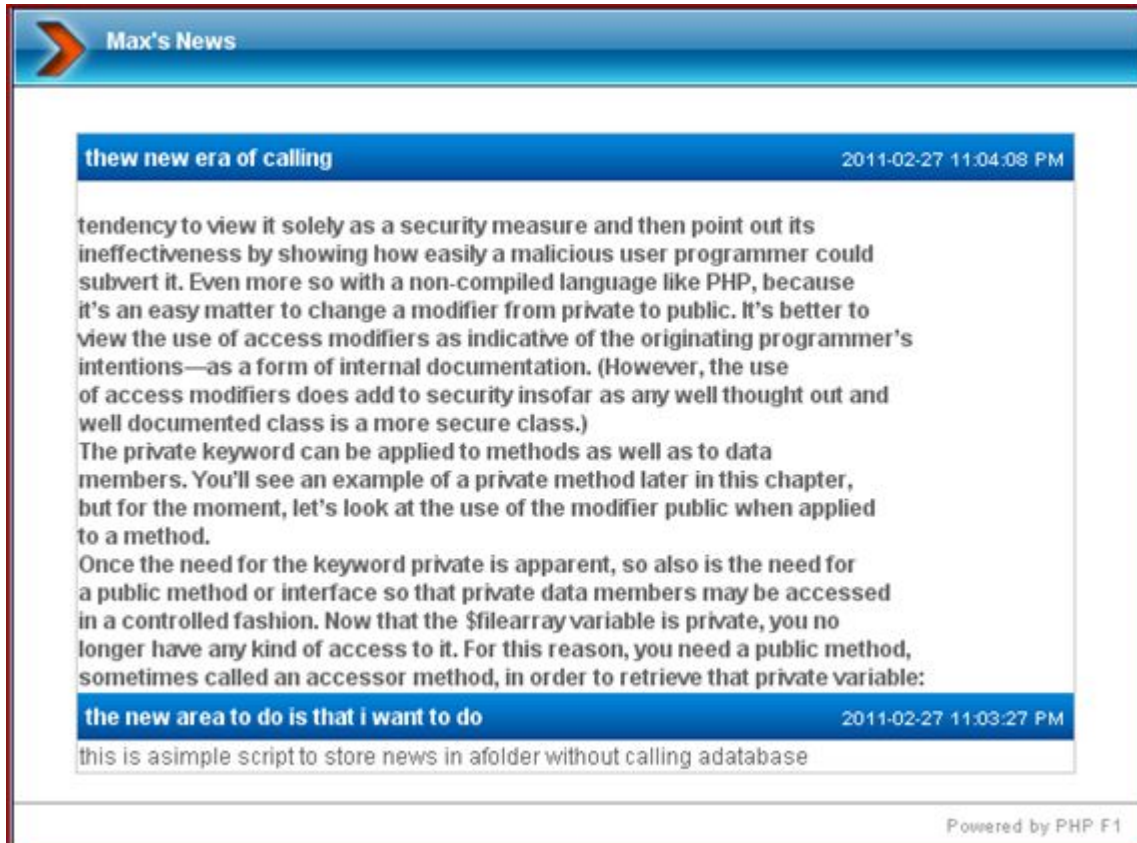
رمز: Code

```
<?php
require_once("maxNews.class.php");
$newsHandler = new maxNews();
?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Max's News</title>
<link href="/style/style.css" rel="stylesheet" type="text/css" />
</head>

<body>
<div id="container">
<div id="header"><div id="header_left"></div>
<div id="header_main">Max's News</div><div
id="header_right"></div></div>
<div id="content">
<?php $newsHandler->displayNews(); ?>
</div>
</body>
</html>
```

```
</div>
<div id="footer"><a href="http://www.phpf1.com"
target="_blank">Powered by PHP F1</a></div>
</div>
```

وهذه صورة للمقالات المخزنة في مجلد: news:



Max's News

thew new era of calling 2011-02-27 11:04:08 PM

tendency to view it solely as a security measure and then point out its ineffectiveness by showing how easily a malicious user programmer could subvert it. Even more so with a non-compiled language like PHP, because it's an easy matter to change a modifier from private to public. It's better to view the use of access modifiers as indicative of the originating programmer's intentions—as a form of internal documentation. (However, the use of access modifiers does add to security insofar as any well thought out and well documented class is a more secure class.)

The private keyword can be applied to methods as well as to data members. You'll see an example of a private method later in this chapter, but for the moment, let's look at the use of the modifier public when applied to a method.

Once the need for the keyword private is apparent, so also is the need for a public method or interface so that private data members may be accessed in a controlled fashion. Now that the \$filearray variable is private, you no longer have any kind of access to it. For this reason, you need a public method, sometimes called an accessor method, in order to retrieve that private variable:

the new area to do is that i want to do 2011-02-27 11:03:27 PM

this is a simple script to store news in a folder without calling a database

Powered by PHP F1



كلاس php: الدرس السادس: كيف تفكر وتخطط كمبرمج قبل برمجة كلاساتك الخاصة

نستكمل معكم زوار موقعنا الكرام **دورة برمجة كلاسات الPhp**, وكنا قد أخذنا ٥ دروس عرفنا فيهم الأساسيات، و٣ أمثلة عملية لكلاسات مختلفة وفهمنا كيف نستغلهم في برمجياتنا الخاصة لتوفير الوقت، ووعدتكم اننا سننتقل بعد هذه الخطوه، للخطوه الأهم وهي كيف تبرمج بنفسك وتطلق العنان لقدراتك لتبرمج كلاسات php خاصة بك ومن برمجيتك ومن الصفر، وهذا ما سنتحدث عنه في هذا الدرس والدروس القادمة، راجيا من العلى القدير ان أكون قدمت شىء جديد للمبرمج العربى، فهذه الدرس جهد شخصى منى لم اقتبس فكرتها من اى موقع، ولانى أعلم مدى أهميتها او لم يسبقنى أحد بشرحها بهذه الطريقه، فأنتى أقوم بكتابة هذه الدروس بعناية فأنقه حتى تخرج بشكل مفهوم وفى نفس الوقت مبسطه لمن يقرأها،

الدرس السادس: كيف تفكر وتخطط كمبرمج قبل برمجة كلاساتك الخاصة:

التخطيط والتفكير قبل برمجة اى شىء هو أهم شىء، ولك أن تتخيل انك لن تصبح مبرمج الا بدون ان تكون مبرمج يفكر ويخطط قبل ان يقدم على برمجة شىء، فالبرمجة ماهى معركة بين عقلك وكودك تحتاج لان تخطط لها حتى تنجح وتنتصر فيها بإخراج شىء جديد على الساحة البرمجية.

لذلك سنتعلم سويا فى هذا الدرس كيفية التخطيط والتفكير قبل البدء ببرمجة اى شىء تريده، اتبع هذه الطريقة فى اى شىء تريد برمجته فيما بعد

خطوات التخطيط والتفكير:

- 1-لا بد ان تملك الأساسيات فى جيبك قبل البرمجة
- 2-لا بد ان تحضر ورقه وقلم لتخطط ما ستقوم ببرمجته
- 3-لا بد ان تقوم برسم مخطط على جهازك لما ستقوم ببرمجته
- 4-لا بد ان تضع المخطط بجانب المحرر الذى تكتب فيه كودك لتطبق مخططك وتحوله لقطعه برمجية
- 5-لا بد ان تنظم كودك بكتابة فكرة كل كود قبل كتابة الكود
- 6-لا بد ان تكتب كودك كهيكل عام، ثم تبدأ بتخصيصه لما تريد ان تفعله به
- 7-لا بد ان تضع تعليقاتك الخاصه بجانب كل كود لتتذكر فكرته او تسهل لمن يقرأ كودك ان يفهمه

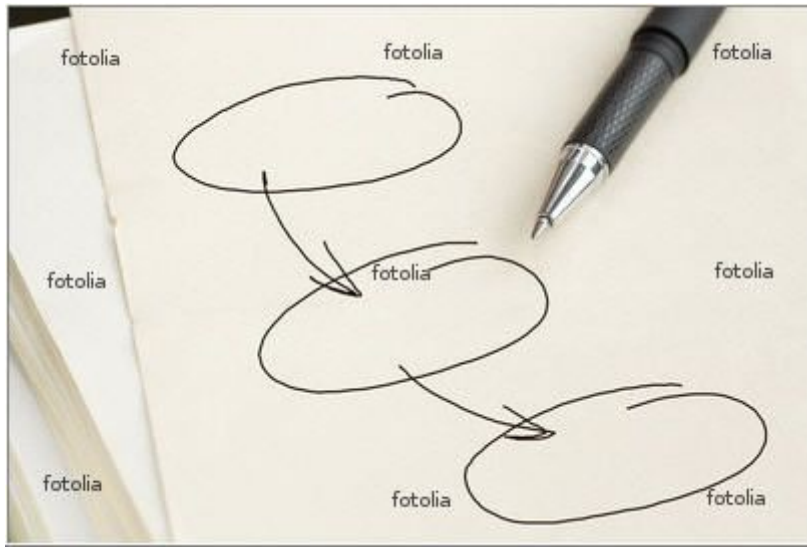
نبدأ على بركة الله.....

1-لابد ان تملك الأساسيات فى جيبك قبل البرمجة

قبل كل شيء ، لابد ان تملك أساسيات البرمجة فى جيبك، فمن غير المعقول انك ستقوم بكتابة كود وأنت لا تعرف أساسيات ما ستكتبه ، لذلك لابد أن تكون أتممت مرحلة انك على دراية بأساسيات `html/xhtml-css-php-mysql` على الأقل حتى تقوم ببرمجة شيء عملي يخدمك ويخدم مستخدمى قطعك البرمجية التى انتجتها،

طبعا الأساسيات متوفر لها دروس كثيرة جدا يمكنك البحث عنها او من خلال مدرسة الويب w3schools

2-لابد ان تحضر ورقه وقلم لتخطط ما ستقوم ببرمجته



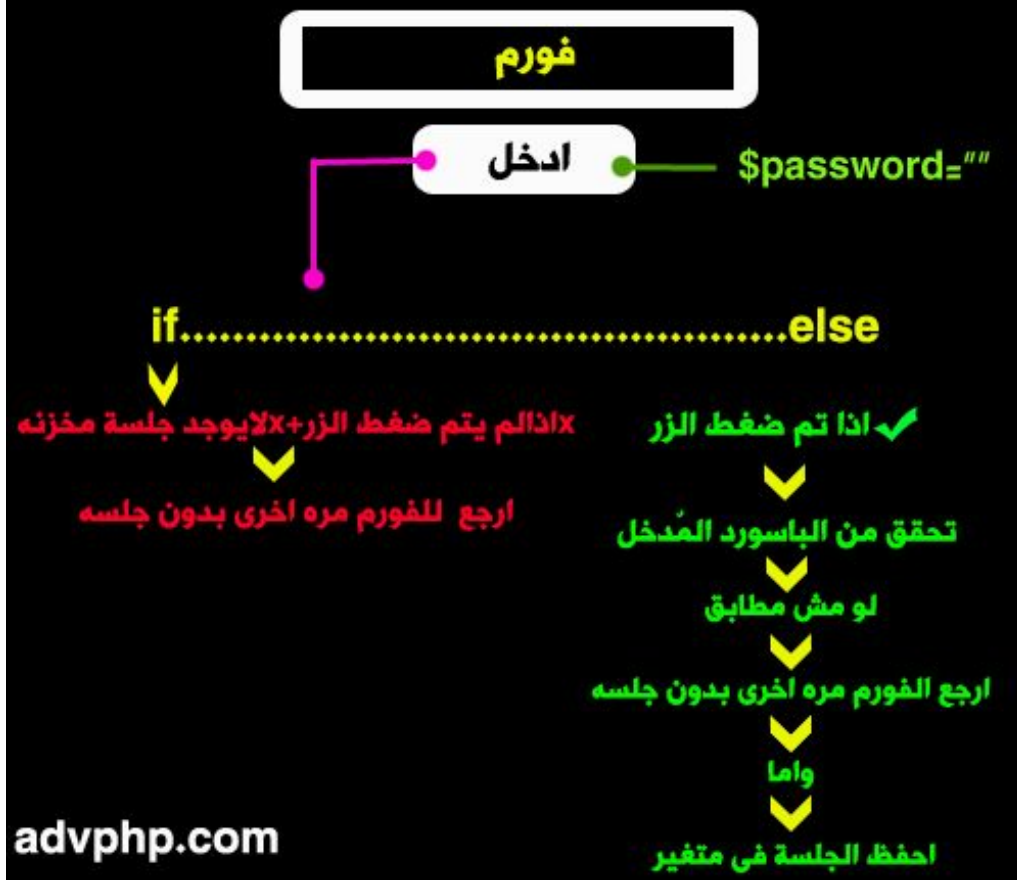
قد تجد هذه الجملة غريبه، لكن ثق تماما ان هذا هو الصواب، فأنت كالمهندس المعماري الذى يخطط بقلمه وورقته مخطط المبني الذى سيتم انشاءه ، لذلك عليك ان تحضر ورقه وقلم قبل كل شيء وتقوم بكتابة مافى بالك من أفكار فى الورقه ، ثم تقوم بتنظيمها خطوه خطوه بحيث فى النهايه تخرج بمخطط مختصر يعطى لك الطريق الصحيح لكتابة كودك بالكامل من الالف للياء

3- لابد ان تقوم برسم مخطط على جهازك لما ستقوم ببرمجته

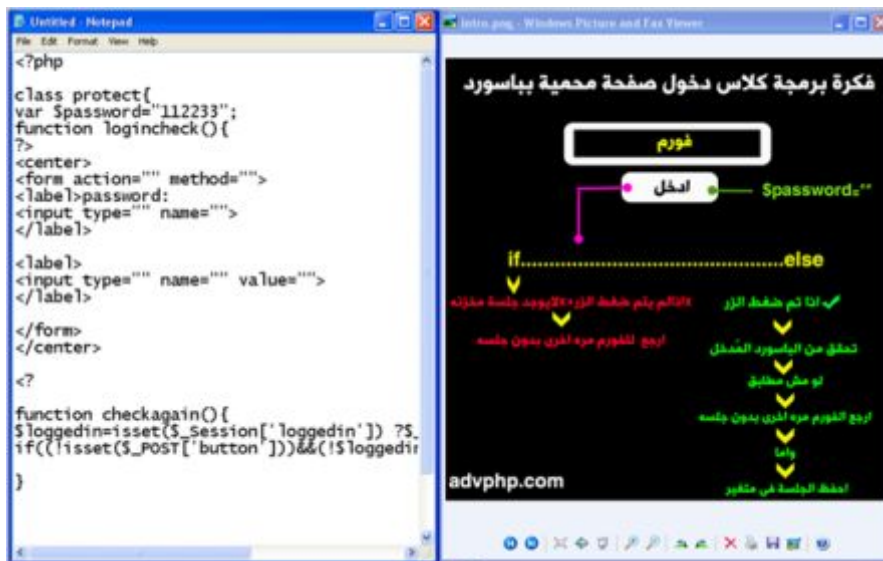
بعد كتابة افكارك وتنظيمها على هيئة مخطط فى الورقه التى كتبتها، قم برسم هذا المخطط على جهازك بالفوتوشوب، حتى يسهل لك النظر اليه وأنت تقوم ببرمجة ملفاتك

مثال:

فكرة برمجة كلاس دخول صفحة محمية بباسورد



4- لا بد ان تضع المخطط بجانب المحرر الذى تكتب فيه كودك لتطبق مخططك وتحواله لقطعه برمجية من الأشياء التى ستسهل لك كتابة كودك هو ان تضع مخططك بجانب المحرر الذى تكتب فيه كالاتى:



5-لابد ان تنظم كودك بكتابة فكرة كل كود قبل كتابة الكود

بمجرد البدء بكتابة كودك، قم بكتابة فكرة كل كود لكل الصفحة التي تبرمجها قبل ان البدء بكتابة اى كود

مثال:

رمز:Code

```
<?
//بيانات الاتصال بقاعد البيانات لوكل ويوزر وباس وداتا بيز
الاتصال بقاعدة البيانات//

//عد الحقول

لوب لارجاع داتا الحقول//

//form فورم تسجيل الدخول
?>
```

6-لابد ان تكتب كودك كهيكل عام، ثم تبدأ بتخصيصه لما تريد ان تفعله به

كتابة الهيكل العام قبل تخصيص كودك أيضا من الطرق الرائعة لتسهيل كتابة كودك ما اعنيه بهذه الجملة هو الاتي:

تخيل مثلا أريد كتابة كود فورم لتسجيل الدخول، فقبل ان اكتب الكود بقيم كل جزء فيه ، سأقوم اولاً بكتابة الهيكل العام له كالآتي:

رمز:Code

```
<form action="" method="">
<label>username:
<input type="" name="">
</label>

<label>password:
<input type="" name="">
</label>

<label>
<input type="" name="" value="">
</label>
```



```
</form>
```

ثم بعد ذلك أبدأ بكتابة قيم كل كود

رمز: Code

```
<form action="action.php" method="post">
<label>username:
<input type="text" name="usernrm">
</label>

<label>password:
<input type="password" name="passn">
</label>

<label>
">دخول<input type="submit" name="subbtn" value="
</label>

</form>
```

```
<?php
|
class protect{
var $password="112233";
function logincheck(){
?>
<center>
<form action="" method="">
<label>password:
<input type="" name="">
</label>

<label>
<input type="" name="" value="">
</label>

</form>
</center>

<?>
```

قم بتطبيق ذلك على كافة كودك البرمجي، ستجد متعه وسهوله في ذلك

7-لا بد ان تضع تعليقاتك الخاصه بجانب كل كود لتتذكر فكرته او تسهل لمن يقرأ كودك ان يفهمه

كتابة تعليقاتك بجانب كل كود من الاشياء المهمه التي ستذكرك بفكرة كل كود ، والتي ستسهل لك اذا احببت فيما بعد ان ترجع لقراءة الكود بسرعة,

مثال:

رمز:Code

```
$loggedin = isset($_SESSION['loggedin']) ? $_SESSION['loggedin'] :  
false;//if..else  
تذكر العملية الثلاثية التي تشبه الحالة الشرطية
```

وهكذا.....

أعتقد الان فهمنا كيف سنخطط جيدا لما سنقوم به قبل كل برمجيه تريد عملها ,
ترقبوا الدروس القوية القادمة والتي سنقوم سويا ببرمجة الكلاسات بانفسنا ، وستروا الاستغلال العملى لكل
كلمه ذكرتها في هذا الدرس



كلاس php: الدرس السابع: كيف تبرمج كلاس دخول صفحة محمية

نستكمل معكم اخواني الكرام دروس دورة برمجة الكلاسات ، وكنا قد أخذنا في الدرس الاخير كيف سنخطط لبرمجة كلاس ال php او اى شىء تريد برمجته تمهيدا لهذا الدرس والدروس القادمة.

فى هذا الدرس سوف نبدأ بشرح كيف تبرمج بنفسك كلاس ال php

وسوف نأخذ اول مثال عملى وخفيف ، وهو كيفية برمجة كلاس لحماية صفحة باسورد دخول

حمل ملفات الدرس من [هنا](#)

طبعاً ، أذكركم بأاساسيات التفكير من الدرس السابق قبل ان نقوم بعمل اى شىء

- 1-لابد ان تملك الأساسيات فى جيبك قبل البرمجة
- 2-لابد ان تحضر ورقه وقلم لتخطط ما ستقوم ببرمجته
- 3-لابد ان تقوم برسم مخطط على جهازك لما ستقوم ببرمجته
- 4-لابد ان تضع المخطط بجانب المحرر الذى تكتب فيه كودك لتطبق مخططك وتحوله لقطعه برمجية
- 5-لابد ان تنظم كودك بكتابة فكرة كل كود قبل كتابة الكود
- 6-لابد ان تكتب كودك كهيكل عام ، ثم تبدأ بتخصيصه لما تريد ان تفعله به
- 7-لابد ان تضع تعليقاتك الخاصة بجانب كل كود لتتذكر فكرته او تسهل لمن يقرأ كودك ان يفهمه

نتوكل على الله ونأخذ هذه الأساسيات خطوه خطوه ، وشاهدوا ما سننتجه بإتباع هذا النظام فى البرمجة

1- لابد ان تملك الأساسيات فى جيبك قبل البرمجة

من الأساسيات التى لابد ان تتقنها لعمل صفحة دخول محمية باسورد هو الأتى:

- 1-التعامل مع الفورمز forms وكيفية ربطها مع , php يمكنك القاء نظره من [w3schools](#)
- 2-التعامل مع الجلسات sessions بما أننا سنستخدمها فى تخزين الجلسات عند الدخول ، من [w3schools](#)
- 3-التعامل مع الحالة الشرطية if..else وطبعاً أساسيات الكلاسات

2-لا بد ان تحضر ورقه وقلم لتخطط ما ستقوم ببرمجته

سنحضر ورقه وقلم ، وسنفكر قليلا ونسأل نفسنا؟

-أريد ان أقوم بعمل صفحة عبارته عن فورم من خانة واحده لادخال باسورد، بحيث اطابقه بباسورد مخزن بحيث لو صحيح سوف أدخل لصفحة محمية.

-نرسم الفورم و زر ال..... submit

-الان ، كيف سأفكر بعمل هذا الفورم للتحقق من الباسورد التي كتبتة ؟

اولا: بما اننا سنحدد باسورد نطابقه باللي راح يدخله المستخدم في الفورم، سنقوم بعمل متغير يوجد به الباسورد الصحيح

ثانيا: علينا ان نقوم بعملية تحقق باستخدام الحالة الشرطية, if..else بحيث سيوجد عندنا حالتين لا ثالث لهم سوف يحدث واحد منهما

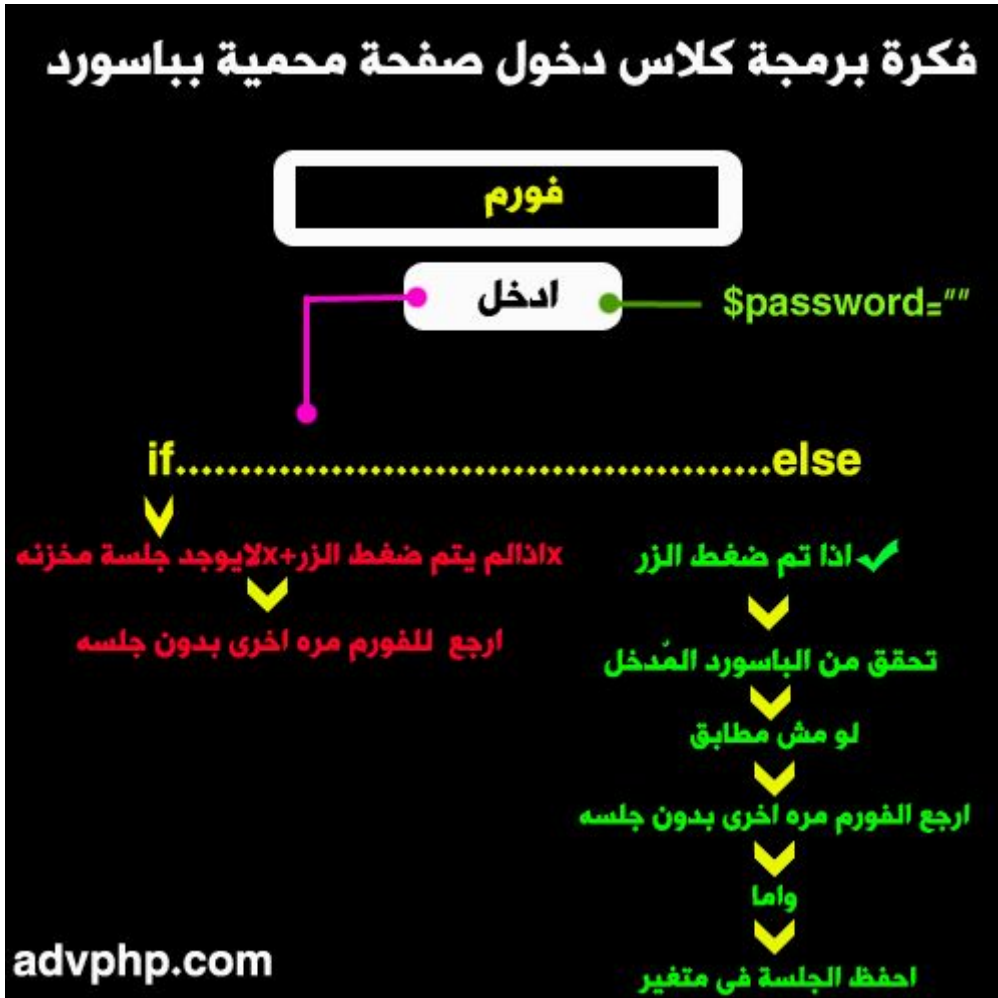
1-ان لا يتم اي كتابة اي شيء في الفورم وبالتالي لن يتم عمل مطابقة للباسورد وتخزين جلسة بذلك

2-واما ان يتم كتابة شيء في الفورم ، ومن ثم يتم مطابقته بالباسورد الصحيح، بحيث لو خطأ يرجع بالفورم مره أخرى ، ولو صح ، راح يتم عمل جلسة جديده وراح ندخل الصفحة المحمية

هذه هي الفكرة بشرح مبسط،

3-لا بد ان تقوم برسم مخطط على جهازك لما ستقوم ببرمجته

بعد ان كتبنا أفكارنا وما سنقوم به ، دعنا نحول هذا الكلام لمخطط بسيط على جهازنا حتى نبدأ نشتغل من خلاله في برمجة هذه الكلاس.....



4-لابد ان تضع المخطط بجانب المحرر الذى تكتب فيه كودك لتطبق مخططك وتحوله لقطعه برمجية

قم بفتح صورة المخطط وضعها بجانب المحرر اللى راح تكتب فيه.....

5-لابد ان تنظم كودك بكتابة فكرة كل كود قبل كتابة الكود

سنقوم بكتابة العناوين الرئيسية لكل كود قبل البدء فى كتابتها.....

نشوف.....

رمز:Code

```
*/متغير للباسورد/*
*/فانكشن لكود الفورم/*
*/فانكشن للتحقق من الباسورد المدخل قبل دخول الصفحة الخمية/*
*/اذا لم يتم ضغط الزر ولم يتم تسجيل جلسة/*
*/واما اذا تم ضغط الزر--تحقق من الباسورد-لو مش مطابق- ارجع للفورم مره
اخرى بدون جلسه واما احفظ الجلسة فى متغير/*
*/أبدأ الجلسة/*
*/متغير جديد لاستخدام الكلاس/*
*/نفذ فانكشن التحقق/*
```

6-لابد ان تكتب كودك كهيكل عام، ثم تبدأ بتخصيصه لما تريد ان تفعله به

بعد ما كتبنا فكرة كل كود، دعنا نبدأ فى التكويد وكتابة الهيكل العام للكلاس.....

رمز:Code

```
<?php
class protect{
protected $passowrd="";
function showform(){
?>

<form action="" method="">

<label>password:
<input type="" name="" >
</label>

<label>
<input type="" name="" value="">
```

```

</label>

<?php
function login(){
if(){
}
else
{
}

}

}

session_start();
$protector= new protect():
$protector->login();

?>

```

ثم سنبدأ بتخصيصه بحيث نحول فكرة مخططنا لشيء عملي بالفعل، وسنقوم بالاتي:

1-نحدد الباسورد مطابقتة باللى راح يدخله الزائر فى الفورم

رمز:Code

```
protected $password = 'test';
```

2-نحدد قيم الفورم الخاص بادخال الباسورد

رمز:Code

```

<form action="<?php echo $_SERVER['PHP_SELF'];?>"
method="post">
    <center>
        <label>Password:
            <input name="passwd" type="password"
size="20" />
        </label><br/>
        <label>
            <input type="submit" name="submitBtn"
value="Login" />
        </label>
    </center>
</form>

```

3-الان نأتى لأقوى شيء فى برمجة هذا الكلاس,

-الان علينا تطبيق الحالة الشرطية الاولى وهى لو لم يتم ضغط الزر ولم يتم تسجيل جلسة ،سنرجع بدون جلسة ونرجع للفورم مره اخرى ،لكن قبل ان نقوم بعمل ذلك ،علينا ان نقوم بعمل متغير للتحقق من وجود جلسة بالفعل ام لا،لذلك سنقوم بعمل متغير وسنتحقق كالتالى:

رمز:Code

```
$loggedin = isset($_SESSION['loggedin']) ?  
$_SESSION['loggedin'] : false;
```

سبق وشرحتكم العملية الثلاثية , ternary operator والجملة السابقة تساوى بالظبط الاتى:

رمز:Code

```
if(isset($_session['loggedin']))  
{  
    $loggedin=$_session['loggedin'];  
}  
else  
{  
    $loggedin=false;  
}
```

-ثم نطبق الحالة الشرطية.....

رمز:Code

```
f ( (!isset($_POST['submitBtn'])) && (!$loggedin)){  
    $_SESSION['loggedin'] = false;  
    $this->showform();  
    exit();  
}
```

-نكمل بقية الحالة الشرطية، واما else ان يتم ضغط الزر

رمز:Code

```
else if (isset($_POST['submitBtn'])) {
```

سنقوم بالتحقق من الباسورد المدخل ونطابقه بالصحيح

رمز:Code

```
$pass = isset($_POST['passwd']) ? $_POST['passwd'] : '';
```

بحيث لو كان الباسورد اللى كتبه الزائر لم يطابق الاصل، سنرجع للفورم مره اخرى

رمز:Code

```
if ($pass != $this->password) {  
    $_SESSION['loggedin'] = false;  
    $this->showform();  
    exit();  
}
```

واما يتم تخزين الجلسة ، ويعنى ان الباسورد اللى انكتب صحيح

رمز:Code

```
else {
    $_SESSION['loggedin'] = true;
}
```

ثم نغلق الكلاس بحيث تكون الكلاس بالكامل كالآتى:

رمز:Code

```
<?php

class Protect{
protected $password = 'test';

    function showform(){
?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8"
/>
    </title>كلاس دخول صفحة محمية <title>

</head>
<body>

        <form action="<?php echo $_SERVER['PHP_SELF'];?>"
method="post">
            <center>
                <label>Password:
                    <input name="passwd" type="password"
size="20" />
                </label><br/>
                <label>
                    <input type="submit" name="submitBtn"
value="Login" />
                </label>
            </center>
        </form>

    </body>
<?php
}

function login(){
    /*if(isset($_session['loggedin']))
    {
        $loggedin=$_session['loggedin'];
```


ثم بعد ذلك يمكنك عمل اي صفحة محمية، وكل ما عليك هو جلب صفحة الكلاس

مثال:

رمز:Code

```
<?php require_once("Protect.class.php"); ?>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8"
/>
  </title> تطبيق عملي لكلاس دخول صفحة عممية </title>
</head>

<body>
Hello this is the first page. <a href="/second.php">Go to the
next protected page</a>
<BR />
<a href="/<?php session_destroy() ?>" > return to the form
again</a>
</body>
```

بهيك يكون انتهينا من الدرس متبعين كافة الأسس التي شرحتها، والتي أعتقد لو طقناها في كل شيء
راح يكون سهل لنا برمجة اي شيء طالما جعلنا أساسنا هو التفكير والتخطيط



كلاس:php الدرس الثامن:شرح برمجة كلاس سجل الزوار من الصفر

نستكمل معكم زوار موقعنا الكرام دروس دورة برمجة كلاسات,php وهذا الدرس سيكون ختام دروس دورتنا,بحيث نكون خرجنا من هذه الدورة بقدره جيده فى برمجة الكلاسات, نبدأ بالدرس الاخير وهو شرح برمجة كلاس سجل الزوار من الصفر

حمل ملفات الدرس من [هنا](#)

قبل أن أبدأ:الدرس فيه أفكار وأشياء كثير مجمعه,حاول أن تلم بكل نقطه فى الدرس لتستفيد أكثر نذكركم قبل أن أبدأ بالخطوات التى اعتادنا ان نمشى عليها سويا فى اى شغل لنا.....

- 1-لابد ان تملك الأساسيات فى جيبك قبل البرمجة
- 2-لابد ان تحضر ورقه وقلم لتخطط ما ستقوم ببرمجته
- 3-لابد ان تقوم برسم مخطط على جهازك لما ستقوم ببرمجته
- 4-لابد ان تضع المخطط بجانب المحرر الذى تكتب فيه لتطبق مخططك وتحوله لقطعه برمجية
- 5-لابد ان تنظم كودك بكتابة فكرة كل كود قبل كتابة الكود
- 6-لابد ان تكتب كودك كهيكل عام,ثم تبدأ بتخصيصه لما تريد ان تفعله به
- 7-لابد ان تضع تعليقاتك الخاصه بجانب كل كود لتتذكر فكرته او تسهل لمن يقرأ كودك ان يفهمه

نتوكل على الله.....

1-لابد ان تملك الأساسيات فى جيبك قبل البرمجة:

سجل الزوار هو عبارة عن اسكربت يمكنك وضعه على موقعك، بحيث اذا أحب اي زائر ان يترك لك رسالة، يمكن لك او لاي شخص ان يقرأها، وهذه صورته للصورة النهائية لاسكربت سجل الزوار

3erbhnh 2011-03-26 1:44:23 PM

PHP Date / Time Constants

PHP: indicates the earliest version of PHP that supports the constant. PHP Date / Time Constants

PHP: indicates the earliest version of PHP that supports the constant. PHP Date / Time Constants

PHP: indicates the earliest version of PHP that supports the constant.

[Next »](#)

Name:

Email:

Your message:

لذلك لو تخيلنا سوينا ماذا نريد ان نكون على دراية من أساسيات فى هذا الاسكربت:
طبعا لو أنا كتبت رسالة وتركتها فأنا بحاجة للاتي:

1-التعامل الجيد مع [forms](#) الذى سأكتب من خلاله الرسالة

2-التعامل الجيد مع [دوال الوقت](#) لطباعة وقت ادراج الرسالة

3-فهم فكرة ال [pagination](#) لتعدد صفحات الرسائل

4-التعامل الجيد مع [دوال فتح وقراءة وتعديل الملفات](#)

>> [file, fopen, fwrite, mkdir, file_exists, opendir](#)، حيث سنقوم بتخزين الرسائل

على هيئة ملفات بدون ان نتعامل مع قاعدة البيانات

5-التعامل الجيد مع دوال الحماية مثل [htmlspecialchars, trim](#)

هذا كشيء أساسي من الاساسيات بجانب أشياء أخرى سنتطرق لها أثناء شرح الدرس
ك.....unset,nl2br

2-لابد ان تحضر ورقه وقلم لتخطط ما ستقوم ببرمجته

بعد مراجعة الاساسيات جيدا،سننتقل لخطوة التفكير وكيف سنبرمج هذه الكلاس،
اذا تخيلنا في مخيلتنا شكل لاسكربت سجل الزوار، سيكون عبارة عن

- 1-فورم لكتابة الرسالة يتكون من الاسم والايميل والرسالة وزر الاعتماد
- 2-جدول لعرض المقالات المخزنه فوق الفورم ،وحتى نعرض هذه الرسائل علينا ان نقوم بخطوتين لتحقيق ذلك:

A-عمل فانكشن لعد وعمل لست بالرسائل المخزنه `list`

b-عمل فانكشن لعرض هذه الرسائل

3-مكان لاطهار خاصية تعدد الصفحات السابق،التالي.....

هذه كفكره مبسطة.....

3-لابد ان تقوم برسم مخطط على جهازك لما ستقوم ببرمجته

دعنا نحول فكرتنا لمخطط حقيقي على جهازنا

فكرة برمجة كلاس سجل الزوار `guest book`

1 form فورم كتابة الرسالة
2 insert فانكشن لتخزين الرسالة
3 list فانكشن عمل لست بالرسائل المخزنه
display فانكشن لاطهار الرسائل المخزنه
pagination تعدد الصفحات

advphp.com

4-لابد ان تضع المخطط بجانب المحرر الذى تكتب فيه كودك لتطبق مخططك وتحوله لقطعه برمجية
قم بفتح صورة المخطط وضعها بجانب المحرر اللى راح تكتب فيه.....

5-لابد ان تنظم كودك بكتابة فكرة كل كود قبل كتابة الكود
سنقوم بكتابة العناوين الرئيسية لكل كود قبل البدء فى كتابتها.....
نشوف.....

رمز:Code

```
<?php
//
//المتغيرات الاساسية

//فانكشن اعتماد الرسالة بعد كتابتها

//فانكشن عمل لست بالرسائل المخزنه

//فانكشن اظهار الرسائل وتعدد الصفحات+الفرم

//فانكشن الفورم

//فانكشن ادراج الرسالة وتخزينها
//
?>
```

6-لابد ان تكتب كودك كهيكل عام، ثم تبدأ بتخصيصه لما تريد ان تفعله به
من هنا سأبدأ بكتابة الجزء الأهم فى موضوعنا , وهو تحويل فكرتنا ومخططنا لكلاس حقيقى

اولا:المتغيرات:

- 1-متغير للمجلد المٌخزن بداخله الرسائل (بدون الحاجه لتخزين الرسائل فى قاعدة البيانات)
- 2-متغير لتحديد صيغة التاريخ والوقت
- 3-متغير لعدد الرسائل بالصفحة الواحدة
- 4-متغير يخص الرسائل ككل(list)

وهذا هى المتغيرات:

رمز:Code

```
<?php
class maxGuestbook{
    public $messageDir = 'messages';
    public $dateFormat = 'Y-m-d g:i:s A';//g=24 hours - g:i:s hours
seconds minutes A AM PM
    public $itemsPerPage = 5;
    public $messageList;
```

ثانياً: سنقوم بعمل فانكشن فورم كتابة الرسالة وسيتكون من الخانات (الاسم والايمل ونص الرسالة والزر للاعتماد)

رمز:Code

```
function displayAddForm(){
?>
<form action="<?php echo $_SERVER['PHP_SELF']; ?>" method="post">
    Name:<br/>
    <input type="text" name="name" size="30"/><br/><br/>
    Email:<br/>
    <input type="text" name="email" size="30"/><br/><br/>
    Your message:<br/>
    <textarea name="message" rows="7" cols="49"></textarea><br/>
    <center><input type="submit" name="submit" value="Save"
/></center>
</form>
```

ثالثاً: سنقوم بعمل فانكشن تخزين الرسالة التي تمت كتابتها على هيئة ملف مخزن داخل المجلد , **messages** والذي حددنا مكانه من خلال المتغير **\$messageDir** ولنقوم بعمل هذه الفانكشن سنتبع الآتي:

1- نقوم بمعمل متغيرات لنخزن بها القيم المرسله من فورم الاسم والايمل ونص الرسالة بجانب وقت وتاريخ كتابة الرسالة

رمز:Code

```
function insertMessage(){
    $name = isset($_POST['name']) ? $_POST['name'] : 'Anonymous';
    $email = isset($_POST['email']) ? $_POST['email'] : '';
    $submitDate = date($this->dateFormat);
    $content = isset($_POST['message']) ? $_POST['message'] : '';
```

وأذكركم مره أخرى بالعملية الثلاثية, **ternary operator** والتي تساوى, **if...else** فلو أخذنا اول متغير وهو **\$name** فهو يساوى الجملة الآتية:

رمز:Code

```
if(isset($_POST['name'])){\n$name=$_POST['name'];\n}\nelse\n{\n$name="anonymous";\n}
```

وقس على ذلك بقية الجمل.....

2-نقوم بعمل حالة شرطية بحيث لو لم يتم كتابة اى شىء فى خانة الاسم ، سنستبدلها بكلمة **anonymous** اى مجهول مع تأمين الخانة بدالة **trim** لازالة المسافات البيضاء و**special characters** , وأيضاً لو عدد حروف نص الرسالة اقل من ٥ سنخرج.....

رمز:Code

```
if (trim($name) == '') $name = 'Anonymous';//if empty ,write\nanonymous\nif (strlen($content)<5) {\nexit();\n}
```

3-نقوم بعمل متغير خاص بحفظ اسم الرسالة، وسوف نجعل الرسائل تحفظ بأسم تاريخ ووقت ادراجها

رمز:Code

```
$filename = date('YmdHis');
```

يعنى لو مثلا الرسالة تم كتابتها بتاريخ ١-٤-٢٠١١ الساعة ١١,٣٠ سيتم تخزين الملف اللى به الرسالة باسم ٢٠١١٤١١٣٠

4-نقوم بعمل متغير احتياطي بحيث اذا لم يوجد مجلد لتخزين الرسائل، تقوم هذه الجزئية بعمل المجلد اوتوماتيكيا

رمز:Code

```
if (!file_exists($this->messageDir)){ //checks file or directory is\npresent or not\nmkdir($this->messageDir);//make a directory called messages\n}
```

5-وأخيراً، نفتح ملف جديد ليخزن الرسالة المكتوبة وندرج بداخله بيانات الفورمز ككل الاسم والايمل ونص الرسالة وتاريخ ووقت ادراجها

رمز:Code

```
$f = fopen($this->messageDir.DIRECTORY_SEPARATOR.$filename.".txt","w+");//w+
read+write
fwrite($f,$name."\n");//write in the file
fwrite($f,$email."\n");
fwrite($f,$submitDate."\n");
fwrite($f,$content."\n");
fclose($f);
```

ملاحظة fopen: خاصة بعمل ملف جديد، و **DIRECTORY_SEPARATOR** تشير للسلاش /، ويمكنك استبدالها بالسلاش مباشرة، لكن يفضل عملها بكتابة هذه الجملة، لأنه قد توجد بعض النسخ تحفظ المسار بالسلاش العكسية| وبالتالي لن يتم قراءة مسار الملف بشكل صحيح

بحيث في النهاية تكون الفانكشن ككل كالآتي:

رمز:Code

```
function insertMessage(){
    $name = isset($_POST['name']) ? $_POST['name'] : 'Anonymous';
    $email = isset($_POST['email']) ? $_POST['email'] : '';
    $submitDate = date($this->dateFormat);
    $content = isset($_POST['message']) ? $_POST['message'] : '';

    if (trim($name) == '') $name = 'Anonymous';//if empty ,write
anonymous
    if (strlen($content)<5) {
        exit();
    }

    $filename = date('YmdHis');//h 24hrs
    اسم الملف الى هيتخزن هيكون
    تاريخ ووقت كتابته
    if (!file_exists($this->messageDir)){ //checks file or directory
        موجود
        ليس موجود
        mkdir($this->messageDir);//make a directory called messages
    }
    $f = fopen($this->messageDir.DIRECTORY_SEPARATOR.$filename.".txt","w+");//w+
read+write
fwrite($f,$name."\n");//write in the file
fwrite($f,$email."\n");
fwrite($f,$submitDate."\n");
fwrite($f,$content."\n");
fclose($f);
}
```

رابعا: سنقوم بعمل فانكشن لجلب عدد الرسائل ككل (list) وحتى نقوم بذلك، نتبع الآتي:

1-جعل المتغير السابق الخاص بعرض الرسائل مخزن للرسائل على هيئة مصفوفة

رمز:Code

```
function getMessageList(){  
    $this->messageList = array();  
}
```

2-نفتح المجلد المخزن بداخله الرسائل،ونسوى لوب على الرسائل اللى بداخله،بحيث نخزنها فى النهاية فى المصفوفة

رمز:Code

```
function getMessageList(){  
    $this->messageList = array();  
  
    // Open the actual directory  
    if ($handle = @opendir($this->messageDir)) {  
        // Read all file from the actual directory  
        while ($file = readdir($handle)) {  
            if (!is_dir($file)) {  
                $this->messageList[] = $file;//save file in  
the array  
            }  
        }  
    }  
}
```

ثم اخيرا،نرتب الرسائل بترتيب عكسى ونرجع فى النهاية بالمتغير اللى خزنا فيه الرسائل على هيئة مصفوفة

رمز:Code

```
function getMessageList(){  
    $this->messageList = array();  
  
    // Open the actual directory  
    if ($handle = @opendir($this->messageDir)) {  
        // Read all file from the actual directory  
        while ($file = readdir($handle)) {  
            if (!is_dir($file)) {  
                $this->messageList[] = $file;//save file in  
the array  
            }  
        }  
    }  
  
    rsort($this->messageList);//sorts an array by the values in  
reverse order h-->d-->c  
  
    return $this->messageList;  
}
```

خامسا: سنقوم بعمل فانكشن ل اظهار الرسائل+displayتعدد الصفحاتPAGINATION
هذه أصعب جزئية في الموضوع فركز فيها جيدا.....

1-علينا ان نقوم بعمل متغير لتخزين فانكشن عد الرسائل المخزنه السبقه سنستخدمه لاحقا

رمز:Code

```
$list = $this->getMessageList();
```

2-حتى نحدد عدد الرسائل المراد ظهورها في الصفحة الواحد ،علينا ان نحدد نقطة البداية ونقطة النهاية

كفكرة عامه:

نقطة البداية: هي اول مقال سيظهر في الصفحة
نقطة النهاية: هي اخر مقال سيظهر في الصفحة

كيف نقوم بعمل نقطتي البداية والنهاية؟

نقطة البداية: بعملية حسابية بسيطة وذلك بطرح واحد من الصفحة التي أنت عليها مضروبه في عدد الرسائل
بالصفحة الواحده والتي حددناها سابقا في المتغير \$itemsPerPage

رمز:Code

```
$startItem = ($page-1)*$this->itemsPerPage;
```

نضرب مثال:نفرض انى واقف على الصفحة الثالثة من الرسائل، بذلك ستصبح العملية الحسابية كالاتى:

رمز:Code

```
(3-1)*5=2*5=10
```

و ١٠ تعنى ان اول مقال سيظهر في الصفحة الثالثة هو رقم ١١ ،لاننا كما قلنا سابقا اننا حفظنا الرسائل في مصفوفة , array وكما تعلم ان المصفوفه تبدأ قيمها من الصفر ٠،١،٢،٣،٤،...الخ

نقطة النهاية:يعمل حالة شرطية بحيث لو نقطة البداية + عدد الرسائل بالصفحة الواحدة أكبر من عدد الرسائل الكلية المخزنه،ستكون نقطة النهاية هي آخر رسالة في, list

رمز:Code

```
if (($startItem + $this->itemsPerPage) > sizeof($list)) $endItem = sizeof($list);
```

واما ان تكون نقطة النهاية هي مجموع نقطة البداية+ عدد الرسائل بالصفحة الواحدة

رمز:Code

```
else $endItem = $startItem + $this->itemsPerPage;
```

يعنى لو فرضنا ان نقطة البداية (اول رسالة فى الصفحة) هى ٥ اى اننا فى الصفحة الثانية، ستكون نقطة النهاية فى الصفحة الثانية هى

رمز:Code

```
$endItem = 5+5=10
```

اى الرسالة رقم ١٠

3-ثم نقوم بعمل لوب لعد الرسائل بالصفحة الواحده، بحيث كل ملف يمر عليه يتم قراءته كجزء من مصفوفة دالةfile

رمز:Code

```
for ($i=$startItem;$i<$endItem;$i++){
    //foreach ($list as $value) {
    $value = $list[$i];
```

ثم نقوم بعمل ثلاث متغيرات، ننحزن بها، مسار الملف، والاسم والايميل وتاريخ ووقت كتابة المقال

رمز:Code

```
$data = file($this->messageDir.DIRECTORY_SEPARATOR.$value);
//directory_separator=/ \,file:read entire file into an array,
file=messages./.nameoffile
$name = trim($data[0]);//first line of file which will be
for the name
$email = trim($data[1]);//2nd
line.....email
    $submitDate = trim($data[2]);//3rd
line.....submitdate
```

ثم نخرج قيم هذه المتغيرات من المصفوفه بدال unset

رمز:Code

```
unset ($data['0']); //
يعنى شلنا اول سطر اللى هو الاسم من المصفوفه عشان
نستخدمه لوحده
    unset ($data['1']);
    unset ($data['2']);
```

ثم نقوم بعمل لوب على المصفوفه الخاصة بدالة file

رمز:Code

```
$content = "";
    foreach ($data as $value) {
        $content .= $value;//The Concatenation Operator $a =
"Hello ";$a .= "World!"; now $a=$a."world" = hello world
    }
```

أخيرا نضع الكود السابق ككل داخل جدول، ونقوم بطباعة القيم المرسله من الرسائل المخزنه بحيث الى هذه اللحظة من الفانكشن، سيكون الكود ككل كالآتي:

رمز:Code

```
function displayGuestbook($page=1){
    $list = $this->getMessageList();
    //echo "<center><a href='add.php'>Leave a
message</a></center>";
    echo "<table class='newsList'>";

    //Get start point(article) and end point
    $startItem = ($page-1)*$this->itemsPerPage;//pagen u are on-
1*5=0    0 is the first article in the list array-->1
    if (($startItem + $this->itemsPerPage) > sizeof($list))
    $endItem = sizeof($list);//6>5 ,enditem is last item in the list 5--
>6 article
    else $endItem = $startItem + $this->itemsPerPage; //0+5=5

    for ($i=$startItem;$i<$endItem;$i++){
        //foreach ($list as $value) {
        $value = $list[$i];
    $data = file($this->messageDir.DIRECTORY_SEPARATOR.$value);
    //directory_separator=/ \,file:read entire file into an array,
    file=messages./.nameoffile
    $name = trim($data[0]);//first line of file which will be
    for the name
    $email = trim($data[1]);//2nd
    line.....email
        $submitDate = trim($data[2]);//3rd
    line.....submitdate
    يعني شلنا اول سطر الى هو الاسم من          unset ($data['0']); //
    المصوفه عشان نستخدمه لوحده
        unset ($data['1']);
        unset ($data['2']);

        $content = "";
        foreach ($data as $value) {
            $content .= $value;//The Concatenation Operator $a =
"Hello ";$a .= "World!"; now $a=$a."world" = hello world
        }

    echo "<tr>
        <th align='left'><a href=\"mailto:$email\">$name</a></th>
        <th class='right'>$submitDate</th>
        </tr>";

    echo "<tr><td
colspan='2'>".nl2br(htmlspecialchars($content))."<br/></td></tr>";//n
l2br print </br> automatically,htmlspecialchars convert special cha
&gt; &lt; as spae <>& to entity as &nbsp;
    }
    echo "</table>";
    if (sizeof($list) == 0){
        echo "<center><p>No messages at the
moment!</p><p>&nbsp;</p></center>";
    }
}
```

أخيراً، يتبقى عمل خاصية تعدد الصفحات ضمن هذه الفانكشن

رمز:Code

```
if (sizeof($list) > $this->itemsPerPage){
    echo "<div id=\"navigation\">";
    if ($startItem == 0) {
        if ($endItem < sizeof($list)){
            echo "<div id=\"nright\"><a
href=\"\".$_SERVER['PHP_SELF'].\"?page=\".($page+1).\" \" >Next
&raquo;</a></div>";
        } else {
            // Nothing to display
        }
    } else {
        if ($endItem < sizeof($list)){
            echo "<div id=\"nleft\"><a
href=\"\".$_SERVER['PHP_SELF'].\"?page=\".($page-1).\" \" >&laquo;
Prev</a></div>";
            echo "<div id=\"nright\"><a
href=\"\".$_SERVER['PHP_SELF'].\"?page=\".($page+1).\" \" >Next
&raquo;</a></div>";
        } else {
            echo "<div id=\"nleft\"><a
href=\"\".$_SERVER['PHP_SELF'].\"?page=\".($page-1).\" \" >&laquo;
Prev</a></div>";
        }
    }

    echo "<br/></div><br/>";
}
echo "<hr />";
```

خاتمة هذه الفانكشن باظهار الفورم الخاص بكتابة الرسالة بطبع الفانكشن التي سويناها سابقا
displayAddForm

بهيك تصيح الكلاس جاهزة للعمل وهذه هو كودها ككل

رمز:Code

```
<?php
class maxGuestbook{
    public $messageDir = 'messages';
    public $dateFormat = 'Y-m-d g:i:s A';//g=24 hours - g:i:s hours
seconds minutes A AM PM
    public $itemsPerPage = 5;
    public $messageList;

function processGuestbook(){
    if (isset($_POST['submit'])) {
        $this->insertMessage();
    }
    $page = isset($_GET['page']) ? $_GET['page'] : 1;
```

```

    $this->displayGuestbook($page);
}

function getMessageList(){

    $this->messageList = array();

    // Open the actual directory
    if ($handle = @opendir($this->messageDir)) {
    // Read all file from the actual directory
    while ($file = readdir($handle)) {
        if (!is_dir($file)) {
            $this->messageList[] = $file;//save file in
the array
        }
    }
    }

    rsort($this->messageList);//sorts an array by the values in
reverse order h-->d-->c

    return $this->messageList;
}

function displayGuestbook($page=1){
    $list = $this->getMessageList();
    //echo "<center><a href='add.php'>Leave a
message</a></center>";
    echo "<table class='newsList'>";

    //Get start point(article) and end point
    $startItem = ($page-1)*$this->itemsPerPage;//pagen u are on-
1*5=0    0 is the first article in the list array index-->1
    if (($startItem + $this->itemsPerPage) > sizeof($list))
    $endItem = sizeof($list);//6>5 ,enditem is last item in the list 5--
>6 article
    else $endItem = $startItem + $this->itemsPerPage; //0+5=5

    for ($i=$startItem;$i<$endItem;$i++){
        //foreach ($list as $value) {
        $value = $list[$i];
    $data = file($this->messageDir.DIRECTORY_SEPARATOR.$value);
    //directory_separator=/ \,file:read entire file into an array,
file=messages./.nameoffile
    $name = trim($data[0]);//first line of file which will be
for the name
    $email = trim($data[1]);//2nd
line.....email
        $submitDate = trim($data[2]);//3rd
line.....submitdate
    يعني شلنا اول سطر اللي هو الاسم من          unset ($data['0']); //
المصفوفه عشان نستخدمه لوحده
        unset ($data['1']);
        unset ($data['2']);

        $content = "";
        foreach ($data as $value) {
            $content .= $value;//The Concatenation Operator $a =
"Hello ";$a .= "World!"; now $a=$a."world" = hello world
        }
}

```

```

echo "<tr>
    <th align='left'><a href=\"mailto:$email\">$name</a></th>
    <th class='right'>$submitDate</th>
</tr>";

echo "<tr><td
colspan='2'>".nl2br(htmlspecialchars($content))."<br/></td></tr>";//n
l2br print </br> automatically,htmlspecialchars convert special cha
&gt; &lt;      as spae <>& to entity as &nbsp;
}
echo "</table>";
if (sizeof($list) == 0){
    echo "<center><p>No messages at the
moment!</p><p>&nbsp;</p></center>";
}
// Create pagination
if (sizeof($list) > $this->itemsPerPage){
    echo "<div id=\"navigation\">";
    if ($startItem == 0) {
        if ($endItem < sizeof($list)){
            echo "<div id=\"nright\"><a
href=\"\".$_SERVER['PHP_SELF'].\"?page=\".($page+1).\"\" >Next
&raquo;</a></div>";
        } else {
            // Nothing to display
        }
    } else {
        if ($endItem < sizeof($list)){
            echo "<div id=\"nleft\"><a
href=\"\".$_SERVER['PHP_SELF'].\"?page=\".($page-1).\"\" >&laquo;
Prev</a></div>";
            echo "<div id=\"nright\"><a
href=\"\".$_SERVER['PHP_SELF'].\"?page=\".($page+1).\"\" >Next
&raquo;</a></div>";
        } else {
            echo "<div id=\"nleft\"><a
href=\"\".$_SERVER['PHP_SELF'].\"?page=\".($page-1).\"\" >&laquo;
Prev</a></div>";
        }
    }

    echo "<br/></div><br/>";
}
echo "<hr />";
$this->displayAddForm();
}

function displayAddForm(){
?>
    <form action="<?php echo $_SERVER['PHP_SELF']; ?>" method="post">
    Name:<br/>
    <input type="text" name="name" size="30"/><br/><br/>
    Email:<br/>
    <input type="text" name="email" size="30"/><br/><br/>
    Your message:<br/>
    <textarea name="message" rows="7" cols="49"></textarea><br/>
    <center><input type="submit" name="submit" value="Save"
/></center>
    </form>

<?php
}

```



```

function insertMessage(){
    $name    = isset($_POST['name']) ? $_POST['name'] : 'Anonymous';
    $email    = isset($_POST['email']) ? $_POST['email'] : '';
    $submitDate = date($this->dateFormat);
    $content = isset($_POST['message']) ? $_POST['message'] : '';

    if (trim($name) == '') $name = 'Anonymous';//if empty ,write
anonymous
    if (strlen($content)<5) {
        exit();
    }

    اسم الملف الى هيتخزن هيكون
    تاريخ ووقت كتابته
    $filename = date('YmdHis');//h 24hrs
    if (!file_exists($this->messageDir)){ //checks file or directory
        موجود
        is present or not
        mkdir($this->messageDir);//make a directory called messages
    }
    $f = fopen($this->
messageDir.DIRECTORY_SEPARATOR.$filename.".txt","w+");//w+
read+write
    fwrite($f,$name."\n");//write in the file
    fwrite($f,$email."\n");
    fwrite($f,$submitDate."\n");
    fwrite($f,$content."\n");
    fclose($f);

```

؟>

ليكون الناتج النهائي بهذا الشكل بعد لمسة جمالية ب CSS

PHP Date / Time Constants

PHP: indicates the earliest version of PHP that supports the constant. PHP Date / Time Constants

PHP: indicates the earliest version of PHP that supports the constant. PHP Date / Time Constants

PHP: indicates the earliest version of PHP that supports the constant.

[Next »](#)

Name:

Email:

Your message:

Save