

باسلام این فایل توسط مصطفی مهرجوز منبع آموزشی دیگری جمع آوری و تهیه شده که برای راهنمایی دوستان آماده شده. اگر کم و کاستی دیدید حتماً بگید درستش کنم.

کار با فایل ها و پوشه ها : Working With Files

فایل ها ابزاری هستند که ما به واسطه آنها می توانیم اطلاعات را ذخیره کنیم و بعد باز خوانی کنیم و یا برای انتقال بین دو تا سیستم ازشون استفاده کنیم .

در برنامه app1.php

```
Echo __file__ ;
```

خروجی :

C:\xampp\htdocs\7Learn.php\php-toolbox\08-file-in-php\app1.php

اما در نوار آدرس سایت URL خواهیم داشت :

<http://localhost/7learn.php/php-toolbox/08-file-in-php/app1.php>

مسیر URL و FILE دو تا چیز متفاوت هستند اما به یک چیز اشاره میکنند . نکته اینجاست که فایل فقط در همین کامپیوتر اصلی سرور معتبر است و URL از طریق بستر وب به آن دسترسی پیدا میکنیم .

در برنامه app1.php

```
Echo __file__ ;
```

```
Echo __dir__ ;
```

```
Echo dirname( __dir__ ) ;
```

```
Echo basename(__file__ ) ;
```

```
Echo basename(__dir __) ;
```

خروجی به ترتیب :

C:\xampp\htdocs\7Learn.php\php-toolbox\08-file-in-php\app1.php

C:\xampp\htdocs\7Learn.php\php-toolbox\08-file-in-php

C:\xampp\htdocs\7Learn.php\php-toolbox

app1.php

08-file-in-php

در واقع به این ترتیب نگاه کنید متوجه خواهید شد :

Dir1/dir2/dir3/... /dirn/basename

میخواهم به تابع معرفی کنم که آرایه ای از dirname و basename و extension و filename را به ما میدهد :

در برنامه app1.php

```
$info = pathinfo(__File__);
```

```
Echo "<pre>";
```

```
Print_r($info);
```

```
Echo "</pre>";
```

خروجی :

```
Array
```

```
(
    [dirname] => C:\xampp\htdocs\7Learn.php\php-toolbox\08-file-in-php
    [basename] => appl.php
    [extension] => php
    [filename] => appl
)
```

حالا اگر همین را برای __DIR__ صدا بزنید فقط extension نخواهد داشت و خروجی بصورت زیر خواهد بود .

```
Array
```

```
(
    [dirname] => C:\xampp\htdocs\7Learn.php\php-toolbox
    [basename] => 08-file-in-php
    [filename] => 08-file-in-php
)
```

میخواهیم ببینیم که یک فایل وجود دارد یا خیر :

یک فایل test.txt در همین پروژه بسازید .

در برنامه app1.php

```
<?php
```

```
function check_file_existance ($file){
```

```
    if (file_exists($file)){
```

```
        echo "FILE : $file : <span style='color:green;'> exists </span>.";
```

```
    } else {
```

```
        echo "FILE : $file : <span style='color:red;'>does not exists </span> .";
```

```
    }
```

```
    echo "<br/>";
```

به رنگ سبز

به رنگ قرمز

```

}
$dir = "C:/xampp/htdocs/7Learn.php/php-toolbox/08-file-in-php/";
$file1 = $dir . "test.txt";
$file2 = $dir . "test-1.txt";
check_file_existance($file1);
check_file_existance($file2);

```

خروجی :

```

FILE : C:/xampp/htdocs/7Learn.php/php-toolbox/08-file-in-php/test.txt : exists .
FILE : C:/xampp/htdocs/7Learn.php/php-toolbox/08-file-in-php/test-1.txt : does not exists .

```

نکته دیگر اینجاست که شما اگر خود مسیر \$dir رو هم به تابع بالا پاس بدهید به شما میگه که وجود داره همچین مسیری .

میخوام کد بالا رو کمی تغییر بدهم تا ببینیم که آیا مسیر بالا به یک فایل اشاره میکند یا به یک مسیر .

در برنامه app1.php

```
<?php
```

```

function check_file_existance ($file){
    $type = "File";
    if (is_dir($file)){
        $type = "Directory";
    }
    if (file_exists($file)){
        echo "$type : $file : <span style='color:green;'>exists </span>.";
    } else {
        echo "$type : $file : <span style='color:red;'>does not exists </span>.";
    }
    echo "<br/>";
}

$dir = "C:/xampp/htdocs/7Learn.php/php-toolbox/08-file-in-php/";
$file1 = $dir . "test.txt";

```

```
$file2 = $dir . "test-1.txt";
```

```
check_file_existance($file1);
```

```
check_file_existance($file2);
```

```
check_file_existance($dir);
```

```
check_file_existance(dirname($file2));
```

خروجی :

File : C:/xampp/htdocs/7Learn.php/php-toolbox/08-file-in-php/test.txt : **exists** .

File : C:/xampp/htdocs/7Learn.php/php-toolbox/08-file-in-php/test-1.txt : **does not exists** .

Directory : C:/xampp/htdocs/7Learn.php/php-toolbox/08-file-in-php/ : **exists** .

Directory : C:/xampp/htdocs/7Learn.php/php-toolbox/08-file-in-php : **exists** .

حالا میخوام اضافه کنم اگر فایل بود چک کنه که فایل خواندنی است یا خیر :

کافیه بعد از if اول که توی تابع آوردیم این کد را بنویسیم :

من کل فایل رو براتون میزارم و تغییرات رو با قرمز نشون میدم :

```
<?php
```

```
function check_file_existance ($file){
```

```
    $type = "File";
```

```
    if (is_dir($file)){
```

```
        $type = "Directory";
```

```
    }
```

```
    if (is_file($file)){
```

```
        if (is_readable($file)){
```

```
            echo "$file : The File is Readable .<br/>";
```

```
        } elseif (is_writable($file)){
```

```
            echo "$file : The File is Writable . <br/>";
```

```
        }
```

```
    }
```

```

if (file_exists($file)){
    echo "$type : $file : <span style='color:green;'>exists </span>.";
} else {
    echo "$type : $file : <span style='color:red;'>does not exists </span>.";
}
echo "<br/>";
}

$dir = "C:/xampp/htdocs/7Learn.php/php-toolbox/08-file-in-php/";
$file1 = $dir . "test.txt";
check_file_existance($file1);

```

خروجی :

C:/xampp/htdocs/7Learn.php/php-toolbox/08-file-in-php/test.txt : The File is Readable .
 File : C:/xampp/htdocs/7Learn.php/php-toolbox/08-file-in-php/test.txt : **exists** .

برای ساختن یک مسیر جدید توسط پی اچ پی که استاد هم توضیح دادند میشه بصورت زیر استفاده کرد :

در برنامه [app2.php](#)

```

<?php
function mkdir_if_not_exists ($dir){
    if(!file_exists($dir)){
        mkdir($dir);
    }
}

$path = __DIR__ . "/";
// $path = __DIR__ . DIRECTORY_SEPARATOR;
$newdir = $path . "new-dir";
mkdir_if_not_exists($newdir);

```

این خط با خط پایین یک کار انجام میدهند با
 این تفاوت که دومی (\) در اخر اضافه میکند .

خروجی یک صفحه وب خالی است اما در پروژه اصلی شما در آدرسی که داده اید یک پوشه به اسمی که شما مشخص کرده اید ساخته شده است .

حالا اگر بجای `mkdir` از `rmdir` استفاده کنید و شرط را از حالت `not` خارج کنید مسیری که وجود داشته باشد حذف خواهد شد . (اسم تابع را هم به دلخواه تغییر دهید .)

نکته مهم اینکه شما نمی توانید با تابع `mkdir` یک مسیر تو در تو درست کنید مثلاً

```
$newdir = $path . "new-dir/01/abc";
```

برای اینکار شما باید تنظیمات خود تابع را دستکاری کنید :

```
Mkdir($newdir , 0777 , true);
```

پسورت پیش فرض `false` است . که اگر `true` شود
می تواند برای مسیر های تو در تو استفاده شود

برای سه گروه صاحب فایل و یوزرها و کاربران اینترنتی
در باره این موضوع پنده همین قدر اطلاع دارم .

خب الان میخوایم محتوای داخل یک دایرکتوری رو بررسی کنیم :

در برنامه `app3.php`

```
<?php  
$path = dirname(__DIR__);  
echo $path . "<br/>";  
  
$files = scandir($path);  
echo "<pre>";  
print_r($files);  
echo "</pre>";
```

خروجی یک آرایه که تشکیل شده از تمام فایل ها یا دایرکتوری هایی که در مسیر خواسته شده وجود دارد :

`C:\xampp\htdocs\7Learn.php\php-toolbox`

```
Array  
(  
    [0] => .  
    [1] => ..  
    [2] => 01-string  
    [3] => 02-array  
    [4] => 03-data-time  
    [5] => 04-json  
    [6] => 05-data-type  
    [7] => 06-rand  
    [8] => 07-math-function  
    [9] => 08-file-in-php  
)
```

تک نقطه نشان دهنده فایل `app3` و دو نقطه نشان دهنده یک
پرگشت به پوشه قبلی (از پوشه `08-file-in-php`).

درواقع `scandir` یک آرایه از محتوای آدرسی که میخوایم به ما میدهد . چه شامل فایل باشد و چه شامل پوشه . فرقی هم نمی

کند که فایل از چه نوعی باشد .

نکته دیگر راجب به این تابع این است که sort این تابع بر اساس صعودی است که میشود آن را تغییر داد .

کافی است :

صعودی ، بصورت پیش فرض

scandir(\$path , SCANDIR_SORT_ASCENDING);

scandir(\$path , SCANDIR_SORT_DESCENDING);

نزولی

برای اینکه اون دو نقطه اول تابع scandir رو نمایش نده میشه از تابع زیر استفاده کنید :

```
$files = array_diff( scandir($path) , array (".", ".."));
```

تابع fopen :

```
$handle = fopen ("test.txt" , "r") ;
```

حاوی یک کد شناسایی برای منبعی است که باز کردیم . تا زمانی که fclose نیامده اثر این منبع استفاده کن و بعد منبع رو باید ببندیم .

عملیاتی که میخواهیم روی فایل انجام دهیم .

برای انجام عملیات حتما از \$handle باید استفاده شود

```
Fclose ($handle);
```

برای حالت r و r+ اگر فایلی وجود نداشته باشد برنامه خطا میدهد در حالی که برای بقیه حالات خودش اگر فایل وجود نداشته باشد فایل را میسازد .

در فایل test.txt محتوای زیر را وارد کنید :

ABCDEF

GHIJKL

MNOPQR

STUVWX

YZ

تابع fread محتوای یک فایل را با طول مشخص شده می دهد چون تابع ما خواندنی است من از fread استفاده کردم شما می توانید تابع نوشتنی بسازید مثل استاد و داخلش بنویسید .

فقط توجه داشته باشید که هر بار fread را صدا بزنید ادامه محتوا را نشان میدهد . و نکته دوم این است که شما در ویندوز برای اینکه از لاین بعد استفاده کنید خود ویندوز دو تا کاراکتر به انتهای لاین اضافه میکنه . میتونید در برنامه notepad++ گزینه نمایش تمام کاراکتر ها را بزنید و این دو تا کاراکتر را ببینید . در لینوکس یک کاراکتر داره .

ABCDEF CR LF
GHIJKL CR LF
MNOPQR CR LF
STUVWX CR LF
YZ

در واقع کاراکتر هاش را در ویندوز به صورت عکس رو به رو نشون میده :
میتونید از طریق همان notepad++ نمایش لینوکسی رو هم ببینید .

پس وقتی طول بدهید به fread این دوتا کاراکتر را هم در نظر میگیرد .

در برنامه app4.php

```
<?php
$handle = fopen("test.txt" , "r");
    $data = fread($handle, 10);
    $data1 = fread($handle, 10);
    $data2 = fread($handle, 10);
    $data3 = fread($handle, 10);
echo "<pre>";
print_r($data);
echo "<hr>";
print_r($data1);
echo "<hr>";
print_r($data2);
echo "<hr>";
print_r($data3);
fclose($handle);
```

خروجی برنامه رو خودتون تست کنید و مشاهده کنید با توجه به توضیحاتی که بهتون دادم .

امیدوارم فایل تا اینجا بدردتون خورده باشه . ان شاءالله اگ خوب بود و دوست داشتین بگین ادامه شو هم براتون کامل کنم . اگر هم نخواستید اشکالی نداره چون من فقط قصدم کمک و راهنمایی بیشتر بود .