

مقدمه :

به وسیله قابلیت های کار با فایل ها در PHP ، شما می توانید فایل های متنی و یا سایر فایل ها را باز کرده ، خوانده ، ویرایش نموده و یا جستجو نمایید .
در این بخش قصد داریم تا متدها و توابعی که برای کار با فایل ها در PHP در نظر گرفته شده اند ، را تشریح نماییم .

باز کردن یک فایل – متد (fopen) :

به وسیله این تابع در PHP ، می توانید یک فایل را باز کنید .
این تابع دو پارامتر اصلی دارد .

پارامتر اول تعیین کننده نام و آدرس فایل و

پارامتر دوم تعیین کننده نحوه باز کردن فایل است .

برای کار با این تابع ، باید خروجی آن را در یک متغیر ذخیره نمود و سپس آن متغیر را در سطح برنامه استفاده نمود .

شکل کلی استفاده از این متد به صورت زیر است :

```
< ? Php  
$ = fopen ( " نام یا آدرس فایل مورد نظر " , " نام یک متغیر دلخواه " ) ;  
> ? >
```

مثال:

```
< ? Php  
$My_File = fopen ( " Example.txt " , " r" ) ;  
> ?
```

نحوه باز شدن فایل:

پارامتر دوم تعیین کننده نحوه باز شدن فایل است . این پارامتر می تواند یکی از مقادیر جدول زیر را داشته باشد . ضمناً عملکرد هر مقدار نیز توضیح داده شده است :

| مقدار | نحوه باز کردن فایل |
|-------|---|
| r | در این حالت فایل به صورت فقط خواندنی و از ابتدای آن باز می شود . در این حالت امکان تغییر فایل برای کاربر وجود ندارد . |
| +r | در این حالت فایل به صورت خواندنی ، قابل ویرایش و از ابتدا باز می شود . در این حالت امکان تغییر فایل برای کاربر وجود دارد . |
| w | در این حالت فایل به صورت فقط نوشتنی و قابل تغییر باز می شود . در این حالت چنان فایل از قبل وجود داشته باشد ، محتویات آن پاک می شود . و اگر وجود نداشته باشد ، یک فایل جدید ایجاد می شود . |
| +w | در این حالت فایل به صورت خواندنی و نوشتنی و قابل تغییر باز می شود . در این حالت چنان فایل از قبل وجود داشته باشد ، محتویات آن پاک می شود . و اگر وجود نداشته باشد ، یک فایل جدید ایجاد می شود . |
| a | در این حالت فایل باز شده و متن تعیین شده به انتهای آن اضافه می شود . |
| x | در این حالت یک فایل جدید با نام تعیین شده ایجاد و به صورت فقط نوشتنی باز می شود . چنانچه فایل از قبل وجود داشته باشد ، برنامه پیام error صادر کرده و مقدار false را بر می گرداند . |
| +x | در این حالت یک فایل جدید با نام تعیین شده ایجاد و به صورت خواندنی و نوشتنی باز می شود . چنانچه فایل از قبل وجود داشته باشد ، برنامه پیام error صادر کرده و مقدار false را بر می گرداند . |

اگر سرور فایل را پیدا نکرده و یا وجود نداشت ، چه می شود :

اگر فایلی که در تابع fopen() تعیین کرده ایم ، وجود نداشت و یا سرور موفق نشد ، آن را پیدا کند ، برنامه یک پیام error صادر کرده و مقدار false یا ۰ را بر می گرداند .
اسکرپت زیر ، در صورتی که فایل وجود نداشته باشد ، یک پیام هشدار را به کاربر نمایش می دهد :

```
< ? php
$file = fopen ( " welcome.txt " , " r " ) or exit ( " Unable to open file ! " ) ;
? >
```

بستن یک فایل باز شده - مند : fclose()

پس از اینکه یک فایل را با دستور fopen() باز کردیم ، می توانیم آن را با دستور fclose() ببندیم .
برای این منظور باید نام متغیری که در هنگام باز کردن فایل به کار بردیم و محتویات فایل را درون آن ذخیره نمودیم را در پرانتز مقابل تابع fclose () قرار دهیم .
شکل کلی استفاده از این متد به صورت زیر است :

```
< ? php
syntax: fclose ( " نام متغیری که فایل باز شده را در آن قرار داده بودیم " ) ;
? >
```

مثال :

```
< ? php
$My_File = fopen ( " Example.txt " , " r " ) ;      // یک فایل را با این دستور باز کرده ایم

..
..
fclose ( $My_File ) ;      // فایل باز شده را بسته ایم
? >
```

چک کردن اینکه به آخر فایل رسیده ایم یا خیر ؟ - متد : feof()

تابع feof() ، چک می کند که آیا برنامه به آخر یک فایل رسیده است یا خیر.
این تابع برای جستجو و یا خواندن فایل های داده ای با حجم نامشخص استفاده می شود . در این موارد ، برنامه هنگامی که به آخر فایل برسد ، باید جستجو یا عملیات خواندن را متوقف کند .
نکته : از این تابع در زمانی که فایل را با حالت های a , w و x باز کرده اید ، نمی توانید استفاده نمایید .
شکل کلی استفاده از این متد به صورت زیر است :

```
< ? php
Syntax   feof ( نام فایل مورد نظر ) ;
? >
```

مثال : در مثال زیر یک اسکرپت طراحی شده است که فایل \$My_File را بررسی کرده و چنانچه برنامه به آخر آن رسیده باشد ، یک پیام صادر می کند :

```
< ? php
if ( feof ( $ My_File ) ) echo " End of file " ;
? >
```

خواندن یک فایل به صورت خط به خط :

به وسیله تابع `fgetc ()` در PHP ، می توانید یک فایل متنی را باز کرده و به صورت خط به خط ، بخوانید . سپس در صورت نیاز در خروجی چاپ نمایید .
این تابع فایل را به صورت خط به خط خوانده و پس از خواندن یک خط ، اشاره گر به ابتدای خط بعدی منتقل می شود .

مثال : اسکریپت مثال زیر یک فایل متنی را به وسیله یک دستور حلقه ، به صورت خط به خط خوانده و سپس در خروجی چاپ می نماید :

```
< ? php
$file = fopen( "welcome.txt", "r" ) or exit ( "Unable to open file!" ) ;
// حلقه زیر فایل را به صورت خط به خط تا رسیدن به آخر آن می خواند و در خروجی نمایش می دهد
while ( !feof ( $file ) )
{
    echo fgets ( $file ) . "< br />" ;
}
fclose( $file ) ;
? >
```

مرجع توابع کار با فایل ها

توابع کار با فایل ها در PHP ، امکان دسترسی و کار با فایل ها را در این زبان به شما می دهند . به وسیله این توابع ، می توانید فایل های روی سرور را باز کرده ، بخوانید ، تغییردهید.

فهرست توابع کار با فایلها:

| نام تابع | شرح عملکرد |
|----------------------------------|--|
| <code>basename ()</code> | این تابع ، نام یک فایل که آدرس آن تعیین شده را بر می گرداند . |
| <code>chgrp ()</code> | این تابع ، گروه کاربرانی که امکان دسترسی به یک فایل خاص را دارند ، تعیین می کند . |
| <code>chmod ()</code> | این تابع ، سطح دسترسی به یک فایل خاص را مشخص می کند . |
| <code>chown ()</code> | این تابع ، مالک (owner) فایل مورد نظر را تغییر می دهد . |
| <code>clearstatcache ()</code> | این تابع ، میزان حافظه اشغال شده توسط یک تابع یا فایل در حافظه موقت را آزاد می کند . |
| <code>copy ()</code> | این تابع ، محتویات یک فایل مورد نظر را در مکانی دیگر بر روی سرور ، کپی می کند . |
| <code>dirname ()</code> | این تابع ، نام پوشه جاری در یک آدرس داده شده به آن را بر می گرداند . |
| <code>disk_free_space ()</code> | این تابع ، میزان فضای خالی در یک پوشه که آدرس آن داده شده را بر حسب bytes بر می گرداند . |
| <code>disk_total_space()</code> | این تابع ، میزان فضای کلی یک پوشه که آدرس آن داده شده را بر حسب bytes بر می گرداند . |
| <code>diskfreespace ()</code> | این تابع ، میزان فضای خالی در یک پوشه که آدرس آن داده شده را بر حسب bytes بر می گرداند . عملکرد این تابع همانند تابع <code>disk_free_space()</code> است . |