

# بسم الله الرحمن الرحيم

## آموزشگاه تحلیل داده

تخصصی ترین مرکز برنامه نویسی و دیتابیس در ایران

توابع مرتبط با فایل

مدرس : مهندس افشین رفوآ

## توابع مرتبط با فایل

### آموزش برنامه نویسی : آموزش برنامه نویسی PHP

این مبحث درباره ی تابع های زیر {function} مرتبط با فایل ها این گونه شرح می دهد.

- باز کردن فایل
- خواندن فایل
- نوشتن فایل
- بستن فایل

### باز کردن و بستن فایل ها

**PHP fopen() function** برای باز کردن فایل ها به کار می رود. به دو **argument** { آرگومان } احتیاج دارد که اولی اسم فایل {file name} را مشخص کند و دیگری حالتی {mode} که در آن عمل و فعالیت می کند را تعیین کند.

**file mode** {حالت و نوع فایل ها} از شش **option** یا گزینه ی انتخابی زیر خارج نیست.

| mode | Purpose {هدف}   |
|------|---|
| r    | فایل ها را فقط برای خواندن باز می کند.<br>File pointer یا نشانه رو فایل را، در ابتدای فایل قرار می دهد.   |
| r+   | فایل ها را به منظور خواندن و {write} نوشتن باز می کند.<br>File pointer یا نشانه رو فایل را، در ابتدای فایل قرار می دهد.   |
| w    | فایل ها را فقط به منظور نوشتن باز می کند.<br>File pointer یا نشانه رو فایل را، در ابتدای فایل قرار می دهد.<br>طول فایل را تا صفر کوتاه می کند یا فایل را تا طول zero برش می دهد. در صورتی که فایلی وجود نداشته باشد، یک فایل می سازد. |
| a    | فایل ها را فقط برای خواندن باز می کند.<br>File pointer را در انتهای فایل قرار می دهد. در صورتی که فایلی وجود نداشته باشد، سعی می کند یک فایل بسازد.   |
| a+   | فایل ها را تنها به منظور خواندن و {write} نوشتن باز می کند. File pointer  |

|  |
|--|
| را در انتهای فایل قرار می دهد. در صورتی که فایلی وجود نداشته باشد، سعی می کند یک فایل بسازد. |
|--|

اگر تلاش سیستم برای باز کردن یک فایل به شکست بیانجامد، `fopen` مقدار `{value}` غلط `{false}` می دهد، در غیر این صورت یک `file pointer` برگردانده می شود که از آن می توان برای خواندن و نوشتن بیشتر این فایل استفاده کرد.

پس از وارد آوردن تغییرات به فایل باز شده، باید حتما فایل را با `fclose function` {عملکرد `fclose`} ببندید.

`fclose()` تابع `fclose` به `file pointer` به عنوان آرگومان `{argument}` نیاز دارد. در صورتی که عملیات بسته شدن `{closure}` با موفقیت صورت پذیرد، نتیجه ی `true` یا درست می دهد و اگر شکست بخورد `false` برمی گرداند.

### خواندن فایل

زمانی که فایلی را با `fopen` باز می کنید، برای خواندن آن باید از `fread function` استفاده کنید. این `function` به دو `argument` نیاز دارد. یکی از این دو تابع `file pointer` هست و دیگری طول و `length` فایل است که در واحد `byte` نمایش داده شده باشد.

طول یک فایل را می توان به کمک `filesize() function` پیدا کرد، که این خود از اسم فایل به عنوان `argument` استفاده می کند و سایز یا اندازه ی فایل را با واحد `byte` در جواب نمایش می دهد.

مراحلی که برای خواندن فایل باید طی کرد به شرح زیر است.

فایل را با `fopen()` باز کنید.

طول فایل را با استفاده از `function()filesize` به دست آورید.

محتوا و `content` فایل را به کمک `fread ()function` مطالعه کنید.

فایل را با `fclose()` ببندید.

مثال زیر محتوا `{content}` یک فایل متنی `{textfile}` را به یک متغیر `{variable}` اختصاص `assign` می کند { می دهد، سپس آن محتویات را بر روی صفحه به نمایش می گذارد.

```
<html>
<head>
<title>Reading a file using PHP</title>
```

```

</head>
<body>

    <?php
    $filename = "/home/user/guest/tmp.txt";
    $file = fopen( $filename , "r" );
    if( $file == false )
    {
        echo ( "Error in opening file" );
        exit();
    }
    $filesize = filesize( $filename );
    $filetext = fread( $file , $filesize );

    fclose( $file );

    echo ( "File size : $filesize bytes" );
    echo ( "<pre>$filetext</pre>" );
    ?>

</body>

```

## نوشتن {write کردن} فایل

با تابع **{fwrite function}** در PHP یک فایل را می توان نوشت یا یک متن **{text}** را می توان به یک فایل آماده و موجود پیوند داد. این **function** به دو **argument** نیاز دارد، که یک **file pointer** و یک رشته اطلاعات **{string of data}** که قرار است نوشته شود را مشخص کند. در صورت تمایل می توان یک **argument integer** سومی که طول اطلاعات را مشخص می کند منظور **{include}** کنید. در صورت لحاظ کردن **argument** سوم، پس از این که طول مورد نظر به دست آید نوشتن به پایان می رسد.

مثال زیر **text file** جدیدی پدید می آورد، سپس **text heading** {تیتیر متن} کوتاهی در داخل آن می نویسد و قرار می دهد. پس از بسته شدن این فایل، **file\_exist()**function وجود **{existence}** این فایل را تایید می کند. این تابع از **file name** {اسم فایل} به عنوان **argument** کمک می گیرد.

```

<?php
$filename = "/home/user/guest/newfile.txt";
$file = fopen( $filename , "w" );
if( $file == false )
{
    echo ( "Error in opening new file" );
    exit();
}
fwrite( $file , "This is a simple test\n" );
fclose( $file );
?>

<html>
<head>
    <title>Writing a file using PHP</title>

```

```
</head>
<body>

    <?php
    if( file_exist( $filename ) )
    {
        $filesize = filesize( $filename );
        $msg = "File created with name $filename ";
        $msg.= "containing $filesize bytes";
        echo ($msg );
    }
    else
    {
        echo ("File $filename does not exit" );
    }
    ?>
</body>
</html>
```

در فصل **PHP File System Function** تمام مسائل و **function** های مربوط به **file input** و **output** را به طور کامل توضیح دادیم.