**PHP Dosya Açma, Okuma, Yazma – PHP Dosya İşlemleri**

**PHP dosya açma**, okuma ve yazma gibi işlemler PHP dosya ve dizin fonksiyonları kullanılarak birçok yöntemle yapılabilmektedir. Satır satır, harf harf dosya içeriklerinin okunması, tüm içeriklerin elde edilebilmesi ve içeriği dosyaya yazdırma gibi çeşitli dosya işlemleri birçok yöntemle gerçekleştirilebilmekte. PHP dosya açma, okuma ve yazma gibi **PHP dosya işlemleri** ile alakalı fonksiyonları aşağıdaki başlıklarda görelim.

**fopen() fonksiyonu**

Fopen fonksiyonu ile parametre olarak belirtilen adreste bulunan dosyayı açma işlemini gerçekleştirir. Dosyadan veri okuma veya veri yazma işlemi için ilk olarak dosyanın açılması gerekiyor. 2 parametre ile kullanılabilir. İlk parametre dosyanın yolunun belirtildiği string türündeki değer, ikinci parametre ise dosyanın okunmak için yada yazmak için veya hem okumak hem de yazmak için açılması gerektiği gibi dosyanın kullanılacağı amacı belirten dosya kip değeri.

Aşağıda ikinci parametrede kullanılabilecek dosyanın amaca uygun kullanılabilmesini sağlayan dosya erişim modları listesini görebilirsiniz.

**Dosya Erişim Modları -Dosya Kipleri**

* **r** : Read işlemi. Dosyanın sadece içeriğinin okunması gerektiğini belirtir.
* **r+** : Dosyanın içeriğinin he okunabilmesi hem de yazılabilmesinin gerektiğini belirtir.
* **w** : Dosyaya sadece yazma işleminin yapılabilmesinin gerektiğini belirtir. Dosyanın içeriği tamamen silinir ve baştan itibaren yazmaya başlanır. Dosya belirtilen adreste yok ise yeniden oluşturulur.
* **w+** : Dosyaya hem yazma işleminin yapılabilmesi hem de dosya içeriğinin okunabilmesinin gerektiğini belirtir. Dosyanın içeriği silinir ve baştan veri yazma işlemi gerçekleştirilir. Dosya belirtilen adreste yok ise yeniden oluşturulur.
* **a** : Dosya içerisine sadece veri eklenebilmesi gerektiğini belirtir. Dosyanın içeriği silinmez, içeriğin sonuna veri eklenir. Dosya belirtilen adreste bulunmuyorsa yeniden oluşturulur.
* **a+** : Dosyaya hem veri eklenebilmesini hem de verinin okunabilmesinin gerektiğini belirtir. Dosyanın içeriği silinmez, içeriğin sonuna veri eklenir. Dosya belirtilen adreste yok ise yeniden oluşturulur.
* **x** : Dosyanın oluşturulması sağlanır ve oluşturulan dosyanın içerisine veri yazmak için açılması gerektiğini belirtir. Aynı isimde bir dosya belirtilen adreste var ise fopen fonksiyonundan ‘false’ değeri döner.
* **x+** : Dosyanın oluşturulması sağlanır ve içerisine hem veri yazma hem de verileri okunması gerektiğini belirtir. Dosya belirtilen adreste aynı isimde zaten var ise fopen fonksiyonundan ‘false’ değeri döner ve hata oluşur.

Fopen fonksiyonu içerisine girilen parametreler doğrultusunda dosyayı açar ve dosyayı temsil eden, işaret eden değeri bir değişkene atar. Dosyayı belirten bu değişken üzerinden dosya okuma ve yazma işlemleri yapılabilecektir. PHP dosya açma işlemini gerçekleştiren fopen fonksiyonu kullanımı aşağıdaki gibidir.

$dosya = fopen('dosyalar/metin.txt','w+');

Yukarıdaki örnekte dosyalar klasörü içerisindeki metin.txt dosyası hem okunmak hem de içerisine veri yazmak için PHP üzerinde fopen fonksiyonu ile dosya açma işlemi yapılıyor ve dosyayı belirten değer dosya değişkenine aktarılıyor.

Fopen açılıp dosya işlemleri gerçekleştirildikten sonra dosyanın kapanması için fclose fonksiyonu kullanılır. Fclose fonksiyonu kullanımı aşağıdaki gibidir.

fclose($dosya);

**fgets() fonksiyonu**

Fgets fonksiyonu ile dosya okuma işlemi satır satır gerçekleşmektedir. Fonksiyon her çalıştığında bir sonraki satırdaki veriler elde edilecektir. Döngü içerisinde fgets fonksiyonu kullanılır ise dosya içerisindeki bütün veriler satır satır okunacaktır. Satır satır dosya okumak için PHP fgets fonksiyonu kullanımı aşağıdaki gibidir.

$dosya = fopen('dosyalar/metin.txt','r');

echo fgets($dosya);

fclose($dosya);

Yukarıdaki örnekte metin.txt dosyasının içerisindeki verilerden ilk satırının okunup ekrana yazdırılması sağlanıyor.

Aşağıdaki örnekte ise döngü içerisinde kullanılarak dosya içerisindeki bütün satırlar okunarak ekrana alt alta yazdırılıyor.

$dosya = fopen('dosyalar/metin.txt','r');

while($oku = fgets($dosya)){

echo $oku.'<br>';

}

fclose($dosya);

**fgetc() fonksiyonu**

Fgetc fonksiyonu ile **dosya okuma** işlemi harf harf gerçekleşmekte. Fonksiyon her çalıştığında dosya içerisindeki veriler harf harf teker teker okunarak elde edilir. Böylelikle harfleri teker teker okuyarak dosyadaki bütün veriler elde edilebilir. Harf harf dosya okumak için PHP fgetc fonksiyonu kullanımı aşağıdaki gibidir.

$dosya = fopen('dosyalar/metin.txt','r');

while($oku = fgetc($dosya)){

echo $oku.'<br>';

}

fclose($dosya);

**fread() fonksiyonu**

Fread fonksiyonu ile dosya okuma işlemi ikinci parametresine girilen ‘byte’ cinsinden sayısal değere göre ele alınır. İkinci parametreye girilen sayısal değer, dosyadan kaç bayt veri okunması gerektiğini belirtir. PHP fread fonksiyonu kullanımı aşağıdaki gibidir.

$dosya = fopen('dosyalar/metin.txt','r');

$dosya\_boyutu = filesize('dosyalar/metin.txt');

$oku = fread($dosya, $dosya\_boyutu);

fclose($dosya);

echo $oku;

[**PHP filesize fonksiyonu**](https://muhammedyaman.com/php-dosya-boyutu-klasor-boyutu/) ile dosyanın boyutu bayt olarak elde ediliyor. Bu değeri fread fonksiyonunun ikinci parametresine tanıtarak dosyanın bütün verilerini okuma işlemi sağlanmış oluyor.

**Fwrite() fonksiyonu**

Fwrite fonksiyonu ile dosyaya yazma işlemi gerçekleştirilir. 3 parametre ile kullanılır. İlk parametre **fopen fonksiyonu** ile açılan dosyayı belirten değişken değeri, ikinci parametre dosyaya eklenecek veri, üçüncü parametre ise dosyaya eklenecek verinin sınırlandırılması için girilen bayt türünden sayısal değer. Eklenecek verinin kaç bayt ile sınırlandırılması gerektiği belirtilir. Üçüncü parametreyi kullanmak zorunlu değildir. Fonksiyon başarılı bir şekilde çalışır ise dosyaya eklenen verinin bayt uzunluğunu döndürür, aksi halde ‘false’ değeri döner. Dosyaya yazmak için PHP fwrite fonksiyonu kullanımı aşağıdaki gibidir.

$dosya = fopen('dosyalar/metin.txt','a');

$sonuc = fwrite($dosya, 'yeniveri');

fclose($dosya);

echo $sonuc;

Yukarıdaki örnekte fopen fonksiyonunun ikinci parametresine dosya erişim kiplerinden biri olan “a” değeri yazıldı. Böylelikle dosyadaki verilerin sonuna yeni veri ekleme yapılacak. Eklenecek veri “yeniveri”, bayt uzunluğu 8, fonksiyon başarılı olur ise 8 değeri dönecek ve bu değer ekrana yazdırılacak.

**Fputs() fonksiyonu**

Fwrite fonksiyonu ile aynıdır, hiçbir farkı yok. Uygulamada fwrite yerine fputs yazılır ise proğramın akışında değişiklik olmayacak.

**feof() fonksiyonu**

Dosya okuma işlemleri sırasında fgets, fgetc ve fread gibi dosya okuma fonksiyonların kullanıldığı bir uygulamada bu fonksiyonlar her seferinde fonksiyon içerisinde tanımlandıkları değer kadar dosyadaki bir sonraki veri kümesini getirir. Böylelikle sürekli bir sonraki veri getirilebilir ve dosyadaki bütün veriler okunarak elde edilebilir. Aslında [**dizi işaretçileri**](https://muhammedyaman.com/php-dizi-isaretcileri-konusu-ve-each-fonksiyonu/) gibi dosya işaretçileri mevcuttur ve böylelikle dosyadaki veriler her okunduğunda dosya işaretçisi buna göre konumlandırılır.

Peki PHP’de feof nedir ne işe yarar? Feof fonksiyonu ile bu fonksiyonların **dosyadaki verileri okuma** sırasında konumları elde edilerek dosya sonuna gelindi mi kontrolü yapılarak bütün verilerin elde edilebilmesi sağlanabilir. PHP feof fonksiyonu kullanımı aşağıdaki gibidir.

$dosya = fopen('dosyalar/metin.txt','r');

while(!feof($dosya)){

echo fgets($dosya);

}

Fgets fonksiyonu her çalıştığında **dosya işaretçisi** konumlanır ve bu konuma göre feof fonksiyonu dosyanın sonuna gelinip gelinmediğini kontrol eder. Gelinmediyse döngü devam eder ve dosyadaki diğer satırlarında okunarak elde edilebilmesi ve ekrana yazılması sağlanır. Bu mantıkla diğer dosya okuma fonksiyonları için de aşağıdaki gibi bir kodlama yapılabilir.

Feof fonksiyonu ve fgetc fonksiyonu ile dosyadaki bütün verilerin okunması.

$dosya = fopen('dosyalar/metin.txt','r');

while(!feof($dosya)){

echo fgetc($dosya);

}

Feof fonksiyonu ve fread fonksiyonu ile dosyadaki bütün verilerin okunması

$dosya = fopen('dosyalar/metin.txt','r');

while(!feof($dosya)){

echo fread($dosya,35);

}