

PHP Sunum 5

Çerez(Cookie) Kullanımı
Oturum Yönetimi

Çerez Kullanımı

- Girilen sitelerin tarayıcılar(browserlar) üzerinde bıraktığı izlere **cookie**, Türkçesiyle de **çerez** denir. Bu izler siteye tekrar ziyaret edildiğinde yeniden gönderilerek bazı şeylerin yeniden hatırlanmasını sağlar.
- Mesela giriş formlarında kullanıcının şifresini ve kullanıcı adını cookieler sayesinde tarayıcıya kaydederek bir sonraki gelişinde bilgilerini yeniden girmek zorunda kalmadan direkt giriş yapabilir.
- Temel olarak kullanıcı bilgilerini hatırlanmasında kullanılan çerezleri hayal gücünüzün el verdiği kadar bir çok alanda da kullanmak mümkün.

Cookie Oluřturma

- Cookie oluřturma iřlemi **setcookie()** fonksiyonu ile yapılır.

Genel kullanımını řöyledir;

setcookie("isim", "deęer", "zaman ařımı");

Cookie Oluřturma

- `setcookie("erezim", "selam dnya");`
- Yukarıdaki rnekke **erezim** adında bir cookie oluřturduk ve deęerinde **selam dnya** yaptık. Ve 3. parametre olarak hi bir Őey girmediđimiz iin varsayılan olarak tarayıcı kapatıldıđında erez silinecek.

Cookie Oluřturma

- Eęer erezin bizim istedięimiz bir gelecek tarihte silinmesini istiyorsakta 3. parametreyi devreye sokacaęız.
- *setcookie("erezim", "selam dnya", time() + (60*60*24));*
- Őimdi 3. olarak girdięimiz **time() + (60*60*24)** bu parametre Őu anki zamana saniye cinsinden **60*60*24** tam olarak **24** saat yani 1 gn eklemiř olduk.
- Bu oluřturduęumuz erezin son kullanma tarihi Őuandan itibaren + 24 saat sonra demek oluyor. 1 gn sonrasında bu erez siteye girildięinde **gnderilmiyor olacak.**

Cookie Okuma

- Oluşturulmuş cookie yi görüntülemek için **`$_COOKIE`** komutu kullanılır.
- Kullanımı şu şekildedir:
- *`echo $_COOKIE["çerez adı"];`*

Cookie Oluřturma ve Okuma

- Alttaki örnekte setcookie komutu ile emyo isimli bir cookie oluřturduk. Cookie'nin deęerini 1234 olarak tanımladık. Zaman ařımı olarak time() fonksiyonunu kullandık ve 3600 sn. yani 1 saat geęerli olacak řekilde belirttik. Son olarak echo komutu ile emyo isimli cookie'nin deęerini ekrana yazdırdık. Bu örneęi test ettięimizde ekrandaki çıktı 1234 olacaktır.

```
<?php  
setcookie("emyo","1234",time()+3600);  
echo $_COOKIE["emyo"];  
?>
```

Bu deęişkenler internet explorer tarayıcısında řu klasörde C:/Documents and Settings/isminiz/Cookies ięerisinde bir yazı dosyasında tutulur.

Cookie Silme

- Oluşturduğumuz bir çerezi silmek için yine normal bir çerez tanımlarmış gibi **setcookie()** fonksiyonunu kullanıyoruz. Tek fark 3. son kullanma tarihi paramtremize geçmiş bir zaman giriyoruz. Bu sayede çerezin tarihi dolmuş varsayıлып hemen siliniyor. Örnek olarak bu çereze **time() - 3600** diyerek 1 saat öncesini göstereceğiz ve çerez silinmiş olacak:
- ***setcookie("çerezim", "selam dünya", time() - 3600);***

Cookie Deęerini Deęiřtirme

- erezleri oluřturmakta ve silmekte olduęu gibi deęerini deęiřtirmekte de **setcookie()** fonksiyonunu kullanıyoruz. Ve oluřturmakta olduęu gibi deęerini deęiřtirmekte aynıdır. Sadece erez adına nceden tanımladıęımız adı yazmamız yeterli. Eęer erez yoksa oluřturulur varsa deęeri deęiřtirilir.
- `setcookie("erezim", "selam dnya");`
- `setcookie("erezim", "merhaba dnya");`
- `echo $_COOKIE['erezim'];`

Cookie Kontrolü

- Cookie'nin tanımlı olup olmadığını kontrol etmek için isset() fonksiyonunu kullanabilirsiniz.

```
<?php
if(isset($_COOKIE["çerezim"]))
{
echo "cookie tanımlanmış";
}
else
{
echo "cookie tanımlanmamış!";
}
?>
```

Oturum Yönetimi

- İnternet üzerinde SESSION olarak bildiğimiz terimin Türkçe karşılığı oturumdur. Siz, kullanıcı olarak bir web sitesine girdiğinizde, o domain üzerinde sizin için bir oturum açılmış olur. Oturumun açıldığı sayfadan itibaren sizin bir session id'niz yani oturum numaranız olur ve sunucu sizi bu numara ile tanır. Sitenin programatik altyapısını hazırlayan kişi de sizin siteye verdiğiniz bazı bilgileri session'a kaydederek bu bilgileri sayfalar arasında aktarır. Buna en güzel örnek olarak bu sitedeki kullanıcı adını gösterebiliriz. Gördüğünüz gibi her sayfada sağ üstte kullanıcı adınız session yardımıyla sayfalara taşı ve sitenin üyesi olarak bu sayede işlem yaparsınız.

Session için yapabileceğimiz 3 işlem var.

- 1- Session'ı başlatmak
- 2- Session kaydetmek
- 3- Session'ı bitirmek

Session oluřturma

- Session kullanımına ya da oluřturulmasına bařlamadan önce bir kere her sayfanın en bařında **session_start()** fonksiyonunu yazmalıyız, bu fonksiyon session olayını kullanıma hazırlar ve olmazsa olmazdır.
- Bazı sunucularda session kullanımını otomatik olarak hep açıktır, açık olan sunucularda **session_start()** yazmaya çalışırsanız **hata verir**. Bu durumda bu fonksiyonu kullanmaya gerek yoktur.

Session_start()

```
<?php  
session_start() ;  
$_SESSION['mesaj'] = 'Selam dünya!';  
?>
```

- Yukarıda ilk önce session kullanımını başlattık ve ardından aynı bir dizi tanımlarmış gibi \$_SESSION global dizisine **mesaj** anahtarına **Selam dünya!** değerini verdik. Yani mesaj adında bir session oluşturduk ve değerini Selam dünya! yaptık.
- Bu oluşturduğumuz değer giren kişinin tarayıcısı kapatılmadığı sürece gezdiği her sayfaya tekrar geri gönderilecek ve bir önceki sayfada yaptığı işlemleri, kullanıcı adı ya da şifre gibi verileri bu şekilde yeniden hatırlanmasını sağlayacak.

Session silme

- Session silme işleminde **unset()** fonksiyonunu kullanıyoruz.
- `session_start();`
- `$_SESSION['gezegen'] = 'dünya';`
`unset($_SESSION['gezegen']);`
- Yukarıda önce bir session oluşturduk ve hemen ardından da onu sildik.

Tüm sessionları silmek

- Bu işlem için `session_destroy()` fonksiyonunu kullanılır. Bu fonksiyon tanımlanmış tüm sessionları yok eder.
- `session_start();`
- `session_destroy();`

Session var mı diye kontrol etmek

- Yine diğer çalışmalarda varlığını kontrol etmekte kullandığımız **isset()** fonksiyonunu kullanacağız. Bu fonksiyon eğer içine yazdığımız session varsa olumlu yoksa olumsuz döner.
- `session_start();`
- `echo isset($_SESSION['deneme']);`

örnek

- Öncelikle ilk.php adında bir php dosyası oluşturalım ve içinde bir session oluşturup bir başka php sayfasında o veriyi okutmayı deneyelim.
- İlk.php
- <?php
- session_start();
- \$_SESSION['isim'] = 'emyo';
- echo 'Sessionumuzu oluşturduk.';
- ?>

Örnek-Devam

- Bu sayfayı oluşturup çalıştırdıktan sonra 2. php sayfamızıda yapalım. Bunun adıda **son.php** olsun
- Son.php

```
<?php  
session_start();  
echo 'Daha önceden kaydedilmiş isim: ' . $_SESSION['isim'];  
?>
```

ilk.php yi ziyaret ettikten sonra bu sayfaya geldiğimizde ekranda **Daha önceden kaydedilmiş isim: emyo** yazacaktır. Eğer ilk.php ye uğramadan yani sessionumuzu oluşturmadan son.php ye girmeye çalışırsak böyle bir session oluşturulmadığı için kod hata verecektir.

Örnek Üyelik Sistemi

- Giriş bilgilerini içinde tutacağımız **ayarlar.php** dosyamızı oluşturuyoruz.
- **Ayarlar.php**

```
<?php  
$kullanici = "admin";  
$parola = "1234";  
?>
```

Örnek Üyelik Sistemi

- Şimdi de ziyaretçinin giriş yapacağı **index.php** dosyamızı oluşturalım. Ziyaretçi bu sayfadan giriş yapacak.

```
<form action="denetim.php" method="POST">
```

Kullanıcı Adı:

```
<input type="text" name="kullanici"><br />
```

Parola:

```
<input type="password" name="parola"><br />
```

```
<input type="submit" value="Giriş Yap">
```

```
</form>
```

Örnek Üyelik Sistemi

- Şimdi de giriş bilgilerinin denetleneceği ve sonucuna göre işlem yapacağımız **denetim.php** dosyasını oluşturuyoruz.

```
<?php
include("ayarlar.php");
session_start();

if(($_POST["kullanici"]==$kullanici) and ($_POST["parola"]==$parola))
{

    $_SESSION["giris"] = true;
    $_SESSION["kullanici"] = $kullanici;
    $_SESSION["parola"] = $parola;

    header("Location:yonetim.php");

}
else
{

    echo "Kullanıcı adı veya Şifre Yanlış.<br>";
    echo "<a href=index.php>Geri dön</a>";

}

?>
```

Örnek Üyelik Sistemi

- **denetim.php** dosyasında \$_SESSION özelliğini kullanarak 3 yeni session tanımladık. Bunlardan giriş isimli olan true değerini taşır ve diğer sayfalarda bu değişken yardımıyla giriş yapılıp yapılmadığını kontrol ederiz. kullanıcı ve parola session'ları da giriş yapan ziyaretçinin kullanıcı adı ve şifre bilgisini tutacak, gerektiğinde yine bu bilgiler kullanılacaktır.

Örnek Üyelik Sistemi

- Şimdi **yonetim.php** isimli yönetim sayfasını oluşturalım

```
<?php
include("ayarlar.php");
session_start();

if(!isset($_SESSION["giris"]))
{
echo "Bu sayfayı görüntüleme yetkiniz yoktur.<br>";
echo "<a href=index.php>Giriş sayfası</a>";

}
else
{
echo "Admin Paneli<br>";
echo "<a href=\"cikis.php\">Çıkış Yap</a>";
}
?>
```

Örnek Üyelik Sistemi

- **yonetim.php** dosyasında kullandığımız **isset()** fonksiyonu, içerisine gönderilen değişkenlerin var olup olmadığını kontrol eder. Yani bir değişken daha önceden tanımlanmışsa **TRUE** tanımlanmamışsa **FALSE** değerini döndürür. Biz giriş yapan ziyaretçi için **giris** isimli bir session tanımlamıştık, bu sayfada onun var olup olmadığını kontrol ettik.

Örnek Üyelik Sistemi

- Şimdi giriş yapan kullanıcının çıkış yapması için **cikis.php** dosyasını oluşturalım

```
<?php  
session_start();  
session_destroy();
```

```
echo "Çıkış işlemi tamamlandı."  
echo "<br><a href=index.php>Anasayfa</a>";  
?>
```

- **cikis.php** dosyasında kullandığımız **session_destroy()** komutu açılmış olan oturumun tüm bilgilerini silmek için kullanılır.