
WEB TABANLI PROGRAMLAMA

BÖLÜM 13

PHP İLE ÇEREZ VE OTURUMLAR :
OTURUM KAVRAMI VE ÇEREZ
İŞLEMLERİ

Prof. Dr. Turgay Tugay Bilgin

turgay.bilgin@btu.edu.tr



GENEL BAKIŞ...

13.1) Çerez Kavramı

13.2) PHP’de Çerez Kullanımı

13.3) PHP’de Çerezlere Erişim

13.4) PHP’de Çerezleri Silmek

13.5) PHP’de Session(Oturum) Yönetimi

13.6) PHP’de Session(Oturum) Başlatma

13.7) PHP’de Session(Oturum) Sonlandırma

13.8) Özet

-
- İnternet ortamında en büyük problem, aynı site içerisindeki farklı sayfalar ziyaret edildiğinde “*bu ziyaret aynı kişi tarafından mı yapılıyor yoksa farklı kişiler tarafından mı yapılıyor*” sorusunun cevabını bulmaktır. Çünkü İnternet ortamında her bir sayfa için yapılan İstek-yanıt süreci birbirinden bağımsızdır. Bu problemin çözümü için PHP dilinde oturum ve oturumları kontrol edebilmek için çerez kavramı geliştirilmiştir.
 - Oturum, kullanıcının bir web sitesine girmesinden itibaren o siteden çıkmasına kadar geçen süreci ifade eder. Oturum kavramı olmasaydı kullanıcı sitenize şifre ile erişiyorsa her bir sayfaya girdiğinde tekrar şifre girmesi gerekecekti. Oysa ki oturum kavramı sayesinde sadece ilk girişte şifre isteniyor, şifre doğru ise buna ait bilgi diğer sayfalara taşınarak kullanıcının her sayfada şifre girmesinin önüne geçilmektedir. Bir sayfadaki ziyaretçi kişi ve onunla ilgili diğer bilgilerin sitenizdeki diğer sayfalara taşınması birçok açıdan hayatı kolaylaştıracaktır. Ayrıca sitenizi ziyaret eden kişiyi tanımlayabilmeniz durumunda kişiye özel içerik sunabilme imkânı doğacaktır.
 - PHP’de oturum yönetimi için iki temel kavram kullanılmaktadır. Bunlar:
 - Çerez (cookie) kullanarak oturum yönetimi
 - Oturum (session) fonksiyonu kullanarak oturum yönetimi
 - Bu slaytta her iki kavramın çalışma yöntemi örneklerle açıklanmıştır.

(13.1) ÇEREZ KAVRAMI

- Türkçe olarak çerez, İngilizce olarak “cookie” olarak ifade edilen kavram aslında web tarayıcı tarafından sizin bilgisayarınızda oluşturulan küçük metin dosyalardan ibarettir. Bu dosyalar web tarayıcının kullandığı “Geçici İnternet Dosyaları (Temporary Internet Files)” klasörü altında konumlandırılırlar. Her bir web tarayıcının çerezleri kaydettiği konum birbirinden farklı olabilmektedir. Firefox gibi bazı web tarayıcılar çerezleri yönetmek için bir görsel arayüze de sahiptirler.
- Çerezler, web sayfanıza giren bir kullanıcıyı algılamak ve diğerlerinden ayırt etmek için kullanılır. Bir web sitesinin ilk sayfasına girdiğinizde PHP ile o kullanıcıya bir kimlik numarası atayarak bu bilgiyi bir çerez içerisinde kullanıcının bilgisayarına depolarsanız, aynı kullanıcı web sitenizde başka bir sayfayı ziyaret ederken bu çerezi okuyarak ziyaret eden kişinin aynı kişi olduğunu tespit edebilirsiniz. Birçok internet sitesi ziyaretçileri hakkında bir takım bilgiler toplarlar. Kimi zaman bu bilgileri veri tabanına kimi zaman ise kullanıcıların kendi bilgisayarlarına kaydederler. Bu işleme kullanıcıyı izleme (*user tracking*) adı verilir.

Dikkat ⚠ İzlenmekten kaçınmak için ne yapabilirim?

Bazı web siteleri kullanıcıları izleme amaçlı kullandıkları çerezleri başka web sitelerinin erişimine de açabilmektedirler. Bu web siteleri, sizin hangi web sitelerini gezdiğinizi içeren bilgileri başka firmalara pazarlayabilmektedirler. Bu bilgileri satın alan firmalar bu bilgileri size özel reklamlar görüntülemekte kullanabilirler. Kişisel gezinme bilgilerinizin başka firmalara pazarlanmasını istemiyorsanız Google Chrome, Opera veya Firefox için “**Privacy Badger**” adlı tarayıcı eklentisini kullanabilirsiniz. Eklenti ile ilgili aşağıdaki linkten bilgi alınabilir.

<https://www.eff.org/tr/privacybadger>

Bu eklenti reklamları engellemeyi sağlamaz, çerezler aracılığıyla gezinme davranışınızın toplanmasını engeller.

(13.2) PHP'DE ÇEREZ KULLANIMI

- PHP kullanarak kullanıcının web tarayıcısında çerez oluşturabilirsiniz. Çerezler, istemcinin tarayıcısında veri saklamayı ve böylece tekrar gelen kullanıcılar hakkında izleme yapmayı sağlar. Çerezler, `setcookie()` fonksiyonu ile oluşturulur. Bu fonksiyonun düzgün çalışabilmesi için `<html>` başlama etiketinden daha önce kullanılmalıdır.
- `setcookie()` fonksiyonu altı adet parametreye ihtiyaç duyar. Her bir çerez için bu fonksiyon ayrı ayrı çağırılmalıdır. Genel kullanımı aşağıdaki gibidir:

```
setcookie(name, value, expire, path, domain, security);
```

- Parametrelerin tanımı ise şöyledir:

-
- **name:** Buraya çerezin adı yazılır. Çerez oluştuktan sonra erişmek için bu ad kullanılacaktır.
 - **value:** Buraya çerez içinde depolamak istediğiniz veriyi yazmalısınız. Fakat şunu unutmayın ki çerezler kullanıcının bilgisayarında ve kullanıcının kolaylıkla erişebileceği konumda depolandığından çerezin içine kullanıcının şifresi gibi önemli bilgileri depolamayınız veya şifreleyerek depolayınız.
 - **Expire:** Çerezin ömrünü, diğer bir ifade ile son kullanma tarihini belirtir. Genellikle çerezin oluşturulma anına göre bağıl olarak tanımlanır. Örneğin oluştuktan sonra 1 ay geçerli bir çerez oluşturmak istendiğinden oluşma anı `time()` fonksiyonu ile alınır ve buna 1 ay tutacak kadar saniye ömür eklenir. Diğer bir ifadeyle `time()` fonksiyonundan dönen değere, çereze verilecek ömür saniye cinsinden eklenerek bulunur. Örneğin

`time() + 60 * 60 * 24 * 30`

ifadesiyle oluşturulan çerez olduğu andan itibaren 30 gün ömre sahip olur. Ömrü biten çerez web tarayıcı tarafından silinir. Çerez ömrü olarak 0 atanır veya bir şey belirtilmezse, çerez oturumun sonunda (tarayıcı kapandığında) ömrünün sonuna gelir ve silinir.

-
- **Path:** Çerezler site üzerinde sadece belirli konumlarda etkin olacak şekilde ayarlanabilir. Örneğin bir çerez sadece

`http://www.birwebsitesi.com/geziler/`

şeklinde bir URL adresinde geçerli olsun isteniyorsa path (yol) olarak “/geziler/” ifadesi girilmelidir. “/” şeklinde ifade girilirse çerez sitenin tamamında kullanılabilir olacaktır. Eğer “/deneme/” şeklinde belirtilirse, çerez sadece “/deneme/” dizininde geçerli olur. “/deneme/bir/iki/” gibi alt dizinler de kullanılabilir. Çerezin varsayılan etki alanı php kodunun çalışmakta olduğu klasördür.

Domain: Çerezler site üzerinde sadece belirli alan adlarında (domain) etkin olacak şekilde ayarlanabilir. Oluşturduğunuz çerezi **deneme.com** ve alt alanlarında kullanılabilir yapmak için değer olarak “**.deneme.com**” girilmelidir. Değer olarak, “**posta.deneme.com**” belirtilirse çerez sadece posta alt alanında kullanılabilir olacaktır.

-
- **Security:** Çerezin istemciye ulaştırılma şeklinin güvenli veya güvensiz olup olmama durumunu belirlemeyi sağlar. Güvenli HTTPS bağlantısı üzerinden aktarılması isteniyorsa “1”, normal HTTP protokolü ile aktarılması isteniyorsa “0” bilgisi girilmelidir. Varsayılan aktarım biçimi HTTP’dir.

PHP’de çerez kullanımı ile ilgili aşağıdaki Kod 13.1’i inceleyiniz. Örnekte kişi adı ve yaşı bilgisi iki ayrı çerez içinde depolanmaktadır. Çerezin ömrü oluştuğu andan itibaren 3600 saniye yani 1 saat olarak belirlenmiştir.

Dikkat ✍ *Çerezlerin kod içinde kullanım sırası önemlidir.*

Çerezler, HTTP protokolünün başlığının bir parçasıdır, bu yüzden `setcookie()` fonksiyonu aşağıdaki örnekte olduğu gibi tarayıcıya herhangi bir veri göndermeden önce kullanılmalıdır. Aksi halde HTML belge başladıktan sonra çerez oluşturmaya çalışırsanız PHP derleyicisi hata mesajı görüntüleyecektir.

Kod 13.1. setcookie () kullanımı

```
<?php
    setcookie("adsoyad", "Ali YILMAZ", time()+3600, "/", "", 0);
    setcookie("yasi", "36", time()+3600, "/", "", 0);
?>
<html>
<body>
<?php echo "Çerez oluşturuldu."?>
</body>
</html>
```

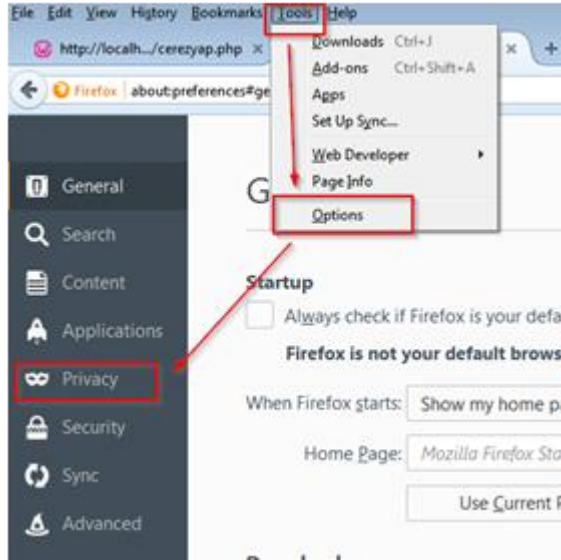
- Yukarıdaki kodu “c:\wamp\www” klasörüne **cerezyaz.php** adıyla kaydediniz. Web tarayıcınızı açıp aşağıdaki gibi

http://localhost/cerezyap.php

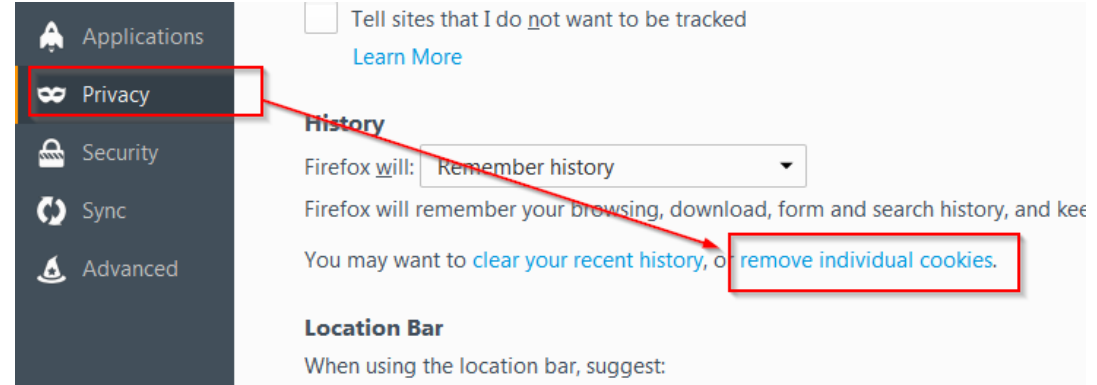
-
- Çalıştırdığınızda kullandığınız web tarayıcı kendi özel geçici internet dosyaları klasöründe “**adsoyad**” ve “**yasi**” adlı iki çerez oluşturacaktır. Oluşturulan çerezler ile ilgili bilgiler web tarayıcıların ayarlarının yapıldığı bölümden görülebilir. Çerezlerin yönetimini sağlayan görsel arayüzler her web tarayıcısında birbirinden farklı olduğu için her biri ayrı ayrı anlatılmayacaktır. Çerez yönetim ara yüzü en gelişmiş olan Mozilla Firefox tarayıcısı örnek olarak gösterilecektir. Diğer web tarayıcılarında çerez yönetimi için web tarayıcı üreticisinin yardım dosyalarına veya ilgili internet sitesine başvurabilirsiniz.
 - Firefox tarayıcısında öncelikle aşağıdaki Şekil 13.1’de görüldüğü gibi

- İngilizce: Tools -> Options -> Privacy
 - Türkçe : Araçlar -> Seçenekler -> Gizlilik

ayarlarına girilir. Privacy (*Gizlilik*) menüsünde “remove individual cookies” (*çerezleri tek tek silin*) seçeneği aracılığıyla çerez yönetimi bölümüne ulaşılır (Şekil 13.2).

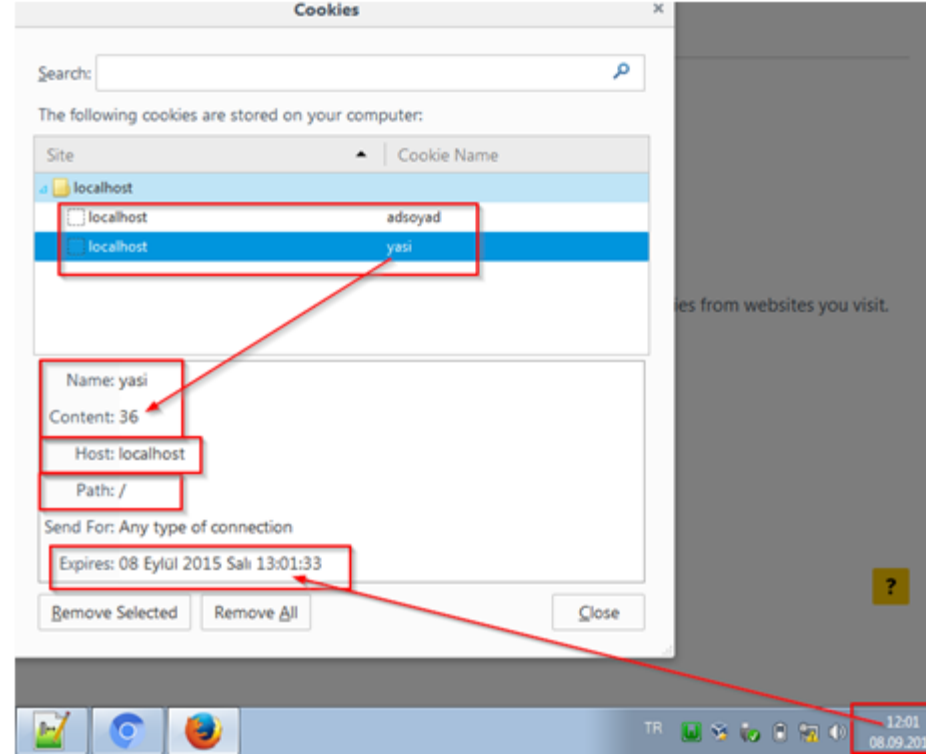


Şekil 13.1. Firefox tarayıcısında çerezleri görüntüleme: 1. Adım



Şekil 13.2. Firefox tarayıcısında çerezleri görüntüleme: 2. Adım

- Şekil 13.3’de görünen Cookies (çerezler) penceresinde yukarıdaki örnekte oluşturulan “adsoyad” ve “yasi” adlı çerezler listelenmektedir. “yasi” adlı çerez tıklandığında içeriği olan 36 değeri şekilde ok ve dikdörtgen ile işaretlenmiştir. Çerezin ömrü ise bilgisayarın güncel saat bilgisine göre (Şekil 13.3’de saat 12:01 olarak görüldüğü) 1 saat ileride (13:01) sona ereceği ok ve dikdörtgen ile işaretli olarak görülmektedir.



Şekil 13.3. Firefox tarayıcısında çerezleri görüntüleme: 3. Adım

(13.3) PHP'DE ÇEREZLERE ERİŞİM

- PHP ile çerezlere erişip içindeki bilgiyi okuyabilmek için daha önce oluşturulmuş çerezin adını biliyor olmamız gereklidir. Adını bildiğiniz çerezi

`$_COOKIE["..."]`

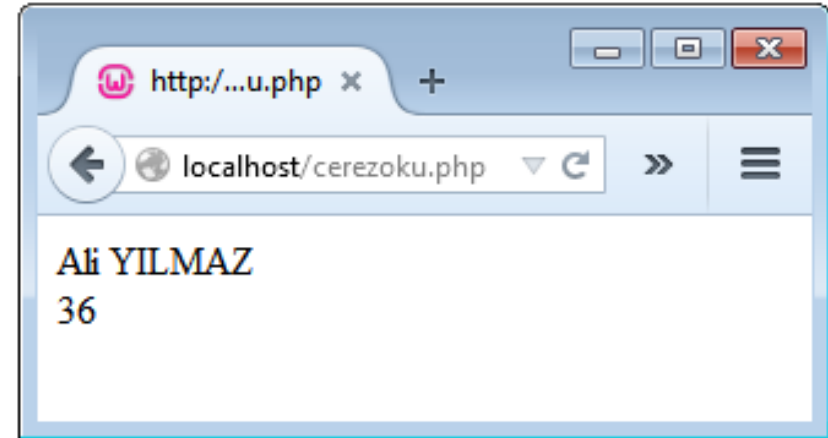
ifadesi ile okuyabilirsiniz. Çerez içeriğini doğrudan ekrana yazabildiğiniz gibi bir değişkene de aktarabilirsiniz.

- Aşağıdaki kodu “c : \wamp\www” klasörüne **cerezoku.php** adıyla kaydediniz. Web tarayıcınızı açıp aşağıdaki gibi

<http://localhost/cerezoku.php> çalıştırdığınızda Şekil 13.4.’deki gibi çerez içerikleri ekranda görüntülenir.

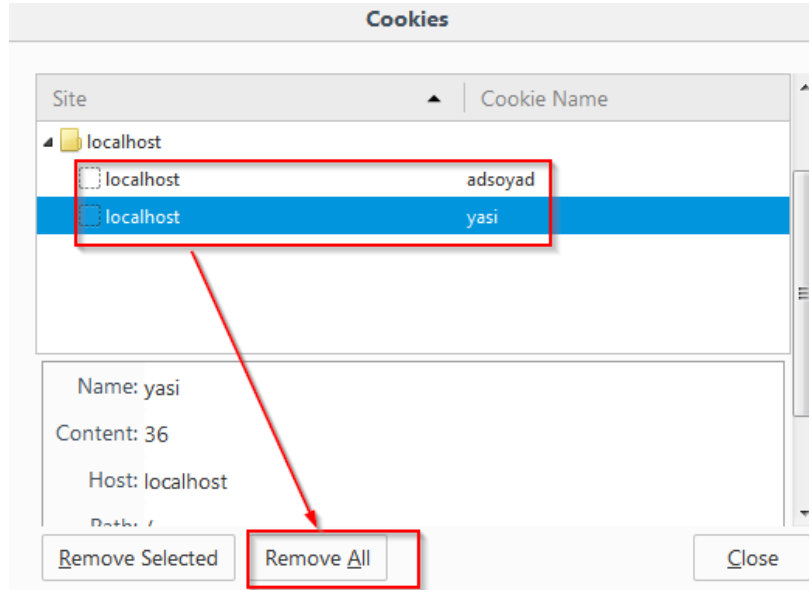
```
<html>
<body>
<?php
echo $_COOKIE["adsoyad"]. "<br />";
echo $_COOKIE["yasi"];
?>
</body>
</html>
```

Kod 13.2. Çerezlere erişim

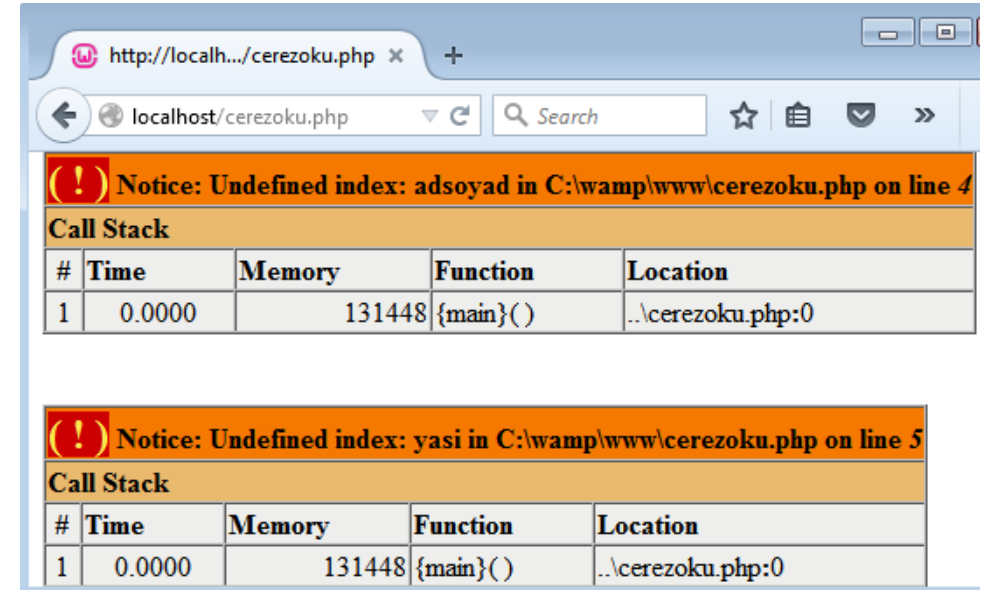


Şekil 13.4. PHP’de çerezleri okuma

- Firefox tarayıcısı üzerinden çerezleri silmek için önceki bölümde anlatıldığı şekilde çerez görüntüleme penceresine erişiniz. Bu pencerede Şekil 13.5.’de ok ve dikdörtgen ile işaretlenmiş “Remove All” butonu ile çerezleri silebilirsiniz. Eğer bu pencerede çerez görünmüyorsa, çerezin ömrü bitip kendiliğinden silinmiş de olabilir.
- Bir çerezi sildiğinizde veya ömrü bitmiş bir çerezi okumaya kalkıştığınızda Şekil 13.6.’daki gibi bir hata mesajı ile karşılaşılır ve PHP kodu sonlandırılır.



Şekil 13.5. Firefox tarayıcısında çerezleri silme



Şekil 13.6. Ömrü bitmiş veya silinmiş çereze erişmek istendiğinde oluşan hata

- Şekil 13.6'daki gibi bir hata mesajı ile karşılaşmamak için çerezin var olduğunu ve ömrünün sürmekte olup olmadığını kontrol etmek için `isset()` fonksiyonunu kullanabilirsiniz. Bu durumda eğer çerez yok ise veya ömrü bitmiş ise sayfada oluşacak hataların önüne geçilecektir. `isset()` fonksiyonunun kullanımı için aşağıdaki örneği inceleyiniz.

```
<html>
<body>
<?php
if( isset($_COOKIE["adsoyad"]))
    echo $_COOKIE["adsoyad"]. "<br />";

if( isset($_COOKIE["yasi"]))
    echo $_COOKIE["yasi"];
?>
</body>
</html>
```


Kod 13.3. Çerezlerde `isset()` kullanımı

- Bölüm 13.3'deki çerez okuma konusunda oluşturduğunuz **cerezoku.php** dosyasını bu örnekteki gibi değiştirerek “c:\wamp\www” klasörüne kaydediniz. Bu dosyayı web tarayıcınızda aşağıdaki adresi yazarak çalıştırdığınızda, silinmiş veya ömrü bitmiş çerezlerde hata oluşmadığı görülecektir.

`http://localhost/cerezoku.php`

(13.4) PHP'DE ÇEREZLERİ SİLMEK

- Normalde bir çerezi silmek için `setcookie()` fonksiyonu sadece çerezin adı verilerek kullanılır. Bu durumda çerezin web tarayıcı tarafından silinmesi gerekir. Fakat bazı web tarayıcıları bu şekilde çerez silmeyi tam olarak gerçekleştiremezler. Bu sebeple bir çerezi silmek için en iyi yol çerezin ömrünü geçmiş bir tarih olarak belirlemektir. Bunun için `time()` fonksiyonundan herhangi bir sayısal değer çıkarılarak geçmiş bir zaman dilimi elde edilebilir. Aşağıdaki örnekte **adsoyad** ve **yasi** adlı çerezlerin ömrü şu anki zamandan 60 saniye gerisi olarak belirlendiği için web tarayıcı tarafından kullanılamaz hale getirileceklerdir.

Hatırlatma  Aşağıdaki örneği çalıştırmadan önce çerezleri silmişseniz Bölüm 13.2'deki gibi tekrar oluşturunuz ve çerezlerin var olduğunu Firefox tarayıcısının çerez yönetim arayüzünde gözlemleyiniz.

Kod 13.4. Çerezleri silmek

```
<?php
    setcookie("adsoyad", "", time()-60, "/", "", 0);
    setcookie("yasi", "", time()-60, "/", "", 0);
?>
<html>
<body>
<?php echo "Çerez silindi."?>
</body>
</html>
```

Yukarıdaki kodu “c : \wamp\www” klasörüne **cerezsil.php** adıyla kaydediniz. Web tarayıcınızı açıp aşağıdaki gibi

http://localhost/cerezsil.php

çalıştırdığınızda çerezleri silinmiş olacaktır. Firefox tarayıcısında çerez yönetim arayüzüne girerek çerezlerin silinmiş olduğunu gözlemleyiniz.

(13.5) PHP'DE SESSION(OTURUM) YÖNETİMİ

- PHP'de çerezlerin dışında oturum yönetimi ve sayfalar arasında veri geçişi için kullanılabilecek diğer bir yapı ise session (oturum) sistemidir. Session sistemi çerezler ile aynı mantıkta çalışır, aralarındaki tek fark çerezler ve içindeki tüm veriler istemcinin web tarayıcısı tarafından istemcinin hard diskinde oluşturulurken Session'da ise çerezlerin içindeki bilgi sunucu yazılımı tarafından sunucunun hard diskinde tutulur, istemcide sadece ilgili çerezi belirten bir numara (PHPSESSID) tutulur. Böylece çerezlerin içindeki bilgiye istemci bilgisayar üzerinden erişmek mümkün olmadığından çok daha güvenli bir bilgi saklama yöntemidir.
- Bir oturum başlatıldığında aşağıdaki süreçler sırasıyla gerçekleşir:

-
- 1) PHP yorumlayıcısı `session_start()` oturum açma komutunu aldığı anda 16'lık sayı sisteminde (hexadecimal) benzersiz bir kimlik numarası üretir. Bu numara `jsck1i1sse05i4j6s2adoeo574` gibi bir değer olabilir.
 - 2) Adı `PHPSESSID` olan bir çerez üretilir ve bu çerez kullanıcının web tarayıcısına gönderilir, fakat bu çerez yalnızca 1 nolu maddede belirtilen kimlik numarasını depolar. Bunun dışında herhangi bir bilgi depolamaz.
 - 3) Sunucunun hard diskinde geçici bir klasörde bu kimlik numarasını içeren `sess_jsck1i1sse05i4j6s2adoeo574` adlı bir metin dosya oluşturulur. Kullanıcı session içinde herhangi bir bilgi depolamak istediğinde bilgiler bu dosyada depolanır. Kullanıcının aynı sitede başka bir sayfayı ziyaret etmesi durumunda sunucu yazılım istemciadaki çerezden `PHPSESSID` değerini öğrenir ve kendi diskindeki metin dosyayı bulur içindeki bilgiyi kullanır.

-
- Bu yöntem sayesinde veriler kullanıcının diskinde tutulmadığı için son derece güvenli bir oturum mekanizması üretilmiş olur. Kullanıcı web tarayıcısını kapattığında veya başka bir siteyi gezmek üzere sitenizden ayrıldığında sunucu yazılımı sunucu diskindeki çerezi siler. Bunun yanında sunucudaki oturum `session_destroy()` fonksiyonu ile sonlandırılabilir veya herhangi bir işlem yapılmadan 30 dakika geçtiğinde sunucu yazılımı oturum çerezini siler.

(13.6) PHP'DE SESSION(OTURUM) BAŞLATMA

- Bir `session` başlatmak için `session_start()` fonksiyonu parametresiz olarak çağırılır. Bu fonksiyon öncelikle mevcut bir `session` bulunup bulunmadığına bakar. Bunu, sayfaya bağlanan kullanıcının web tarayıcısında bu sunucudan gönderilmiş bir `PHPSESSID` çerezinin olup olmadığına bakarak gerçekleştirir. Başlamış bir `session` varsa bunun sunucudaki değişkenlerine ulaşarak bunun üzerinden devam eder. Eğer başlamış bir `session` yok ise yeni bir `session` oluşturur. Herhangi bir hata oluşmaması için `session_start()` fonksiyonunun sayfanın ilk satırında çağırılması gereklidir.
- `session` değişkenlerinin her biri `$_SESSION['...']` komutu ile oluşturulur. Noktalı yerlere çerezlerde olduğu gibi oturum değişkeninin adı yazılır. `session` ömrü devam ettiği sürece değişkenlere bu şekilde erişilebilir veya değer atanabilir.

```
<?php
    session_start();
    $_SESSION['adsoyad']="Ali YILMAZ";
    $_SESSION['yasi']="36";
?>
<html>
<body>
<?php echo "Session Başlatıldı."?>
</body>
</html>
```

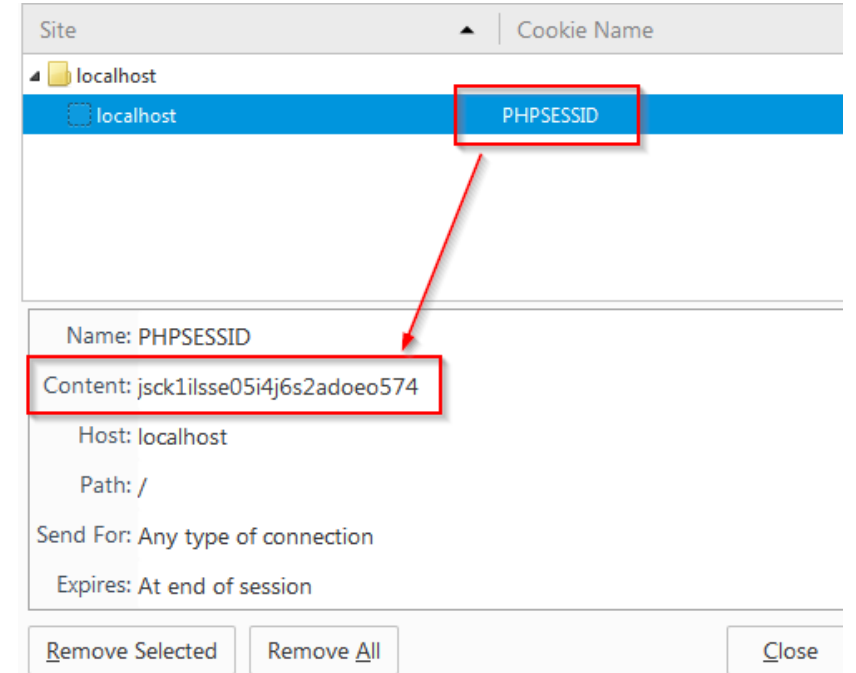
Kod 13.5. Oturum başlatmak

- Kod 13.5'i "c : \wamp\www" klasörüne **sessionyap.php** adıyla kaydediniz. Web tarayıcınızı açıp aşağıdaki gibi

`http://localhost/sessionyap.php`

çalıştırdığınızda web tarayıcınızda bir PHPSESSID çerezi oluşacak ve sunucuda oturum çerezi üretilecektir.

- Kod 13.5'te session başlatılmış ve 2 adet session değişkeni oluşturulmuştur. Bu kod çalıştırıldıktan sonra Firefox tarayıcısında çerez yönetim penceresi açıldığında PHPSESSID adlı çerez görülecektir (Şekil 13.7.). Bu çerezin içinde sadece kimlik numarası bulunduğu, "Ali Yılmaz" ve "36" değerlerinin bulunmadığı görülmektedir.



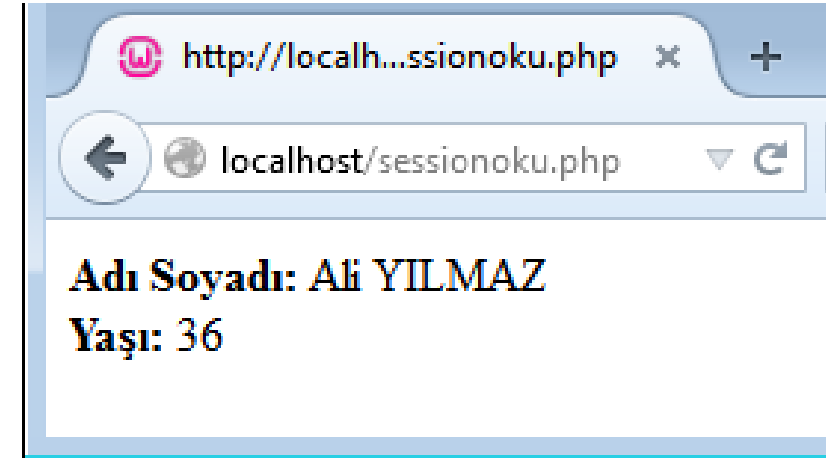
Şekil 13.7. Firefox'ta Session çerezinin içeriğine ulaşmak

- Oturum değişkenlerine **sessionoku.php** adlı aşağıdaki kod ile erişilebilir.

```
<?php
    session_start();
    echo "<b>Adı Soyadı:</b> ".
$_SESSION['adsoyad'];
    echo "<br />";
    echo "<b>Yaşı:</b> ". $_SESSION['yasi'];
?>
```

Kod 13.6. Oturum değişkenlerine erişim

- Yandaki kod parçasının çıktısı Şekil 13.8'deki gibi olacaktır.



Şekil 13.8. sessionoku.php dosyası çalıştığında oluşan ekran görüntüsü

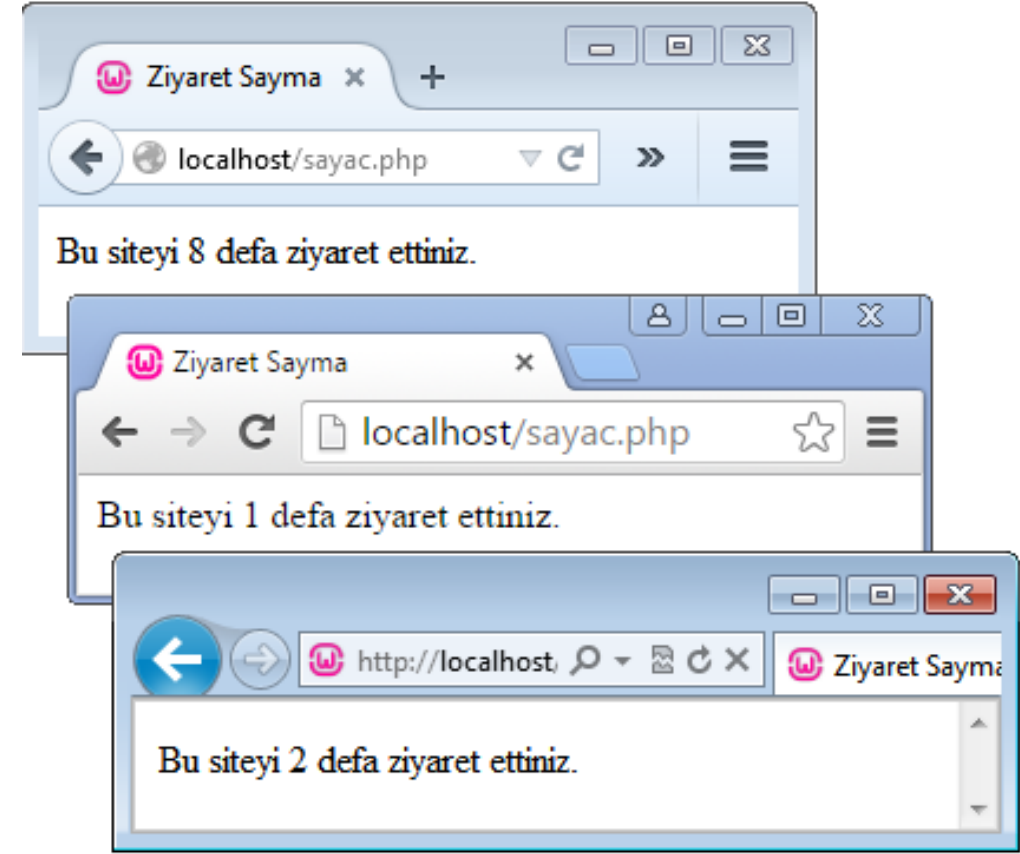
- Böylece **sessionyaz.php** adlı dosyada oluşturduğumuz değişkenlere **sessionoku.php** adlı başka bir php dosyadan erişmiş olduk. Fakat **sessionyaz.php** ile **sessionoku.php** dosyalarını farklı tarayıcılarda çalıştırsaydık bir tarayıcıda oluşturulan çerez diğer tarayıcıda erişilebilir olmayacağından Şekil 13.6'daki gibi bir hata ile karşılaşacaktık.

- Session ile ilgili diğer bir örnek ise ziyaretçi sayacı uygulaması olacaktır. Web sitelerinde ziyaretçi sayaçları görmüşsünüzdür. Bunun basit bir uygulamasını yapacağız. Kullanıcı sayfaya her girdiğinde session değişkeni olarak belirlediğimiz **counter** değeri 1 artacak böylece aynı kişinin kaç defa sayfayı ziyaret ettiği görüntülenecektir. Bu kod toplam ziyaretçi sayısını değil, aynı kişinin sayfayı kaç kere ziyaret ettiğini gösterecektir. Örnekte **isset()** fonksiyonu aracılığıyla session değişkeninin daha önce var olup olmadığı kontrol ediliyor. Böylece, eğer **counter** değişkeni yok ise 1 değeri ile oluşturulmaktadır.

```
<?php
    session_start();
    if( isset( $_SESSION['counter'] ) )
    {
        $_SESSION['counter'] += 1;
    }
    else
    {
        $_SESSION['counter'] = 1;
    }
    $mesaj = "Bu siteyi  ".  $_SESSION['counter']. " defa
ziyaret ettiniz.";
?>
<html>
<head>
<title>Ziyaret Sayma</title>
</head>
<body>
<?php echo ( $mesaj ); ?>
</body>
</html>
```

Kod 13.7. Sayaç uygulaması

- Kod 13.7 Firefox tarayıcısında çalıştırılıp sayfa 8 defa yenilendiğinde (refresh) ekranda 8 defa ziyaret edildi bilgisi görülmektedir. Fakat daha önce bahsedildiği gibi session oluşurken ziyaretçinin bilgisayarına değil “web tarayıcısına” çerez gönderilir. Bir web tarayıcı başka bir tarayıcının çerezini göremediği için sayfa Google Chrome ile ziyaret edildiğinde 1 defa ziyaret edilmiştir şeklinde mesaj görünmektedir (Şekil 13.9). Kısacası her tarayıcıdaki oturum birbirinden bağımsızdır.



Şekil 13.9. sayac.php dosyasının farklı web tarayıcılarda ekran görüntüsü

(13.7) PHP'DE SESSION(OTURUM) SONLANDIRMA

- Bir PHP oturumunu sonlandırmak için `session_destroy()` fonksiyonu kullanılır. Bu fonksiyon herhangi bir argümana ihtiyaç duymaz. Oturumu ve tüm oturum değişkenlerini yok eder. Aşağıdaki örnekte oturum değişkenlerine erişebilmek için oturum başlatılmış ve hemen arkasından sonlandırılmıştır.

Dikkat ☞ *Oturum sonlandırma hakkında*

Oturum sonlandırılacak olsa bile önce `session_start()` ile başlatılmalıdır. `session_start()` fonksiyonu çağırılmadan `session_destroy()` kullanılmamalıdır. Başlatılmayan `session` sonlandırılmaz.

Kod 13.8. Oturum sonlandırma

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

    echo "Session sonlandırıldı.";
?>
```

- Böylece çerezler ve session yapıları tamamlanmıştır. Bundan sonraki ünite de veritabanı bağlantıları anlatılacaktır. Veritabanı bağlantılı uygulamalarda çerezler ve oturum kavramı yoğun olarak kullanıldığından son ünite de hem veritabanı hem oturum ve çerez içeren uygulama projesi gerçekleştirilecektir.

(13.8) ÖZET

- Oturum, kullanıcının bir web sitesine girmesinden itibaren o siteden çıkmasına kadar geçen süreci ifade eder. PHP’de oturum yönetimi için iki temel kavram kullanılmaktadır. Bunlar: Çerez ve Oturumdur. Türkçe olarak çerez, İngilizce olarak “cookie” olarak ifade edilen kavram aslında web tarayıcı tarafından sizin bilgisayarınızda oluşturulan küçük metin dosyalardan ibarettir. Çerezler, web sayfanıza giren bir kullanıcıyı algılamak ve diğerlerinden ayırt etmek için kullanılır. Çerezler, `setcookie()` fonksiyonu ile oluşturulur. PHP ile çerezlere erişip içindeki bilgiyi okuyabilmek için daha önce oluşturulmuş çerezin adını biliyor olmamız gereklidir. Bir çerezi silmek için en iyi yol çerezin ömrünü geçmiş bir tarih olarak belirlemektir. Oturum sistemi çerezler ile aynı mantıkta çalışır, aralarındaki tek fark çerezler ve içindeki tüm veriler istemcinin web tarayıcısı tarafından istemcinin hard diskinde oluşturulurken oturumda ise çerezlerin içindeki bilgi sunucu yazılımı tarafından sunucunun hard diskinde tutulur, istemcide sadece ilgili çerezi belirten bir numara (PHPSESSID) tutulur. Bir oturum başlatmak için `session_start()` fonksiyonu parametresiz olarak çağırılır. Oturumunu sonlandırmak için `session_destroy()` fonksiyonu kullanılır.