

# فصل ۶

## به کارگیری کوکی ها (Cookies) و جلسه ها (sessions)

هدف های رفتاری:

پس از آموزش این فصل هنرجو می تواند:

- ۱- مفهوم و کاربرد cookie را توضیح دهد.
- ۲- تنظیمات cookie را در برنامه انجام دهد.
- ۳- مفهوم و کاربرد session را شرح دهد.
- ۴- متغیرهای session را نام برده و از آنها در برنامه استفاده کند.
- ۵- توابع مهم برای کار با session ها را به کار ببرد.

در یک وب سایت اطلاعات مختلفی از طرف کاربران به سمت سرویس دهنده ارسال می شود. تنها بخشی از این اطلاعات به طور مستقیم و با اطلاع کاربر، توسط سرویس گیرنده ارسال می شوند. قسمت دیگری از آنها بدون اینکه کاربر به طور آگاهانه اقدام به ذخیره و ارسال آنها کرده باشد، ذخیره و ارسال می شوند.

اغلب این گونه داده ها توسط انواعی از امکانات زبان های برنامه نویسی به نام cookie ها و session ها ایجاد می شوند که در این فصل با ماهیت، کاربرد و نحوه مدیریت آنها آشنا خواهید شد.

### ۱-۶- cookie

cookie فایل متنی کوچکی است که از طرف سرویس دهنده و توسط مرورگر روی سیستم کاربر ایجاد می شود. این فایل می تواند در موارد مختلفی به سرویس دهنده وب و سرویس دهنده برنامه کاربردی کمک کند. به عنوان مثال می تواند به سرویس دهنده وب کمک کند تا کاربرانی را که قبلاً به سایت آنها مراجعه کرده اند بشناسد، یا اینکه به یادآوری رمز عبور کاربر در مراجعات بعدی کمک کند. یک کاربر می تواند سیستم خود را طوری تنظیم کند که سرویس دهنده اجازه تولید cookie را نداشته باشد و یا اینکه کاربر را هنگام تولید cookie مطلع کند. cookie می تواند اطلاعات مهمی را درون خود نگهداری کند و می تواند توسط سرویس دهندگان غیر مجاز اطلاعات مهمی از هویت شما را فاش کند. به عنوان مثال اگر در یک سایت عضو شده باشید، چنانچه یک cookie برای آن روی هارد سیستم شما ایجاد شده باشد، یک سایت غیر مجاز می تواند با اهداف سوء از طریق خواندن cookie به رمز عبور شما دسترسی پیدا کند.

**نکته:** هر فایل cookie به تنهایی نمی تواند برای سیستم مشکلات امنیتی ایجاد کند اما می تواند در آسیب رساندن به امنیت سیستم شما دخیل باشد. به طور مثال هکرها و افرادی که قصد سوء استفاده از اطلاعات موجود بر روی سیستم شما را دارند، با خواندن محتویات cookie، به اطلاعات محرمانه شما دسترسی پیدا کنند.

حداکثر اندازه فایل cookie ، ۴KB (کیلوبایت) است.

یکی از ویژگی های فایل های cookie این است که پس از آنکه توسط یک مرورگر روی یک سیستم ایجاد شدند، فقط روی همان سیستم می توانند خوانده شوند. هر cookie یک آدرس URL منحصر به فرد دارد و سرویس گیرنده از روی این آدرس تعیین می کند که cookie مربوط به هر کاربر کدام است.

استفاده از cookie در سرویس گیرنده‌ها و مرورگرهای مختلف محدودیت‌هایی دارد. به عنوان مثال حداکثر تعداد فایل‌های cookie مرورگر Internet explorer، ۳۰۰ فایل است. PHP قادر است عملیات ایجاد، مدیریت و حذف cookie را به خوبی انجام دهد.

### ۱-۶-۱- ایجاد cookie

ایجاد cookie در فایل‌ها و اسناد PHP توسط دستور setcookie انجام می‌شود. ساختار کلی این دستور به شکل زیر است:

setcookie (Name, Value, Expire, Path, Domain, Secure);

setcookie(cookie نام، مقدار cookie، تاریخ انقضای cookie، مسیر، اطلاعات دامنه، مسیریابی);  
درج پارامترهای cookie اختیاری است و به دلخواه می‌توانید تعدادی از آنها را از لیست پارامترها حذف کنید.

- پارامتر Name، نام کوکی و value مقدار آن را مشخص می‌کند. این دو پارامتر اجباری هستند.
- در پارامترهای تابع فوق Path مسیری را در سمت سرویس دهنده تعیین می‌کند که در آن کوکی قابل دسترسی است و در حالت پیش فرض، کوکی می‌تواند توسط کل دامنه در سمت سرویس دهنده، مورد دستیابی قرار گیرد.

- Domain دامنه مجاز را تعریف می‌کند که به طور پیش فرض همان دامنه سرویس دهنده‌ای است که آن را ایجاد کرده است. اطلاعات دامنه مشخص می‌کند که کوکی توسط چه دامنه‌ای قابل دسترسی است.

- گزینه Secure سطح دسترسی به cookie را تعیین می‌کند و به صورت پیش فرض غیرفعال است. اگر این پارامتر فعال شده باشد، کوکی فقط مجاز است از طریق کانال‌های امن، مانند https منتقل شود. مقدار صفر برای غیرفعال و ۱ برای فعال کردن آن است.

- توسط تاریخ انقضا می‌توان اعتبار یا عدم اعتبار cookie را سنجید، اگر تاریخ انقضای یک cookie تمام شده باشد، مرورگر دیگر آن را برای سرویس دهنده ارسال نمی‌کند (تاریخ انقضا با قالب استاندارد GMT (گرنویچ) تعیین می‌شود). این پارامتر برحسب ثانیه نوشته می‌شود.

پارامترهای cookie توسط سرویس گیرنده تعیین می‌شود و می‌تواند علاوه بر پارامترهای ساختار فوق، پارامترهای دیگری را شامل شود، علاوه بر این ممکن است یک Cookie فقط شامل نام و یک پارامتر دیگر باشد.

در ادامه در مثال ۱ به محض اجرای صفحه در مرورگر، یک cookie با نام user برای کاربر

وبسایت ساخته می‌شود. این cookie با مقدار meisam مقداردهی شده و برای آن تاریخ انقضای یک ساعت در نظر گرفته شده است.

### مثال ۱ :

```
<?php  
Setcookie ("user", "meisam", time ( ) + 3600);  
?>
```

اگر با نسخه 6 Internet explorer یا قبل از آن کار می‌کنید برای مشاهده cookie در مسیر نصب ویندوز، Documents and Settings، نام User و سپس پوشه s cookie را باز کنید، مشاهده خواهید کرد که فایل مربوط به آن محتوای مشابه شکل ۶-۱ دارد.



شکل ۶-۱

برای اینکه cookie ایجاد شده را راحت‌تر پیدا کنید، بهتر است ابتدا تمام فایل‌های cookie موجود بر روی سیستم را پاک کنید.

در مثال ۲ یک cookie با زمان انقضای یک ماه ایجاد شده است.

**نکته :** چنانچه ۷ Internet Explorer و یا بالاتر از آن استفاده می‌کنید برای مشاهده کوکی‌ها از منوی Tools گزینه Internet Options را انتخاب نموده، در قسمت Browsing history روی دکمه Setting کلیک کرده و در کادر باز شده روی دکمه view files کلیک نمایید.

### مثال ۲ :

```
<?php  
$expire=time()*60*60*24*30*;  
setcookie ("user", "meisam", $expire);  
?>
```

## ۲-۱-۶- دسترسی به اطلاعات cookie

Cookie های ارسال شده از طرف هر سرویس دهنده در یک آرایه به نام `$_COOKIE` ذخیره شده‌اند و دسترسی به آنها مانند سایر آرایه‌هاست.

به عنوان مثال اگر یک Cookie با نام `username` داشته باشیم می‌توانیم توسط `$_COOKIE ["user name"]` به محتوای آن دسترسی پیدا کنیم.

**نکته:** دستور `Set cookie` در برنامه باید قبل از هر خروجی (مثل `echo`) نوشته شود. علاوه بر این باید در ابتدای برنامه و قبل از پرچسب `<html>` نوشته شود.

### مثال:

می‌خواهیم یک cookie ایجاد کنیم که تعداد بازدیدهای کاربر را از سایت تعیین کند. نام cookie را `visit` می‌گذاریم.

همچنین یک متغیر به نام `$Count` تعریف می‌کنیم که با هر بار ملاقات توسط کاربر، به مقدار آن یک واحد افزوده شود.

```
<?php
if isset ($_COOKIE ["visit"])
{
    $count=$_COOKIE ["visit"] + 1;
    setcookie ("visit",$count,time( )+3600);
    echo $count. "<br/>";
}
else
{
    $count=1;
    setcookie ("visit",$count,time( )+3600);
    echo $count. "<br/>";
}
?>
```

با اجرای فایل فوق در مرورگر مشاهده خواهید کرد که با هر بار اجرای صفحه (و حتی فشردن دکمه F5 به منظور بازنشانی مجدد صفحه) مقدار شمارنده یک واحد افزایش پیدا می کند.

به ساختار کد فوق دقت کنید :

متغیر \$Count را در نظر بگیرید. در صورتی که اولین بار باشد که این برنامه اجرا می شود چون هنوز کوکی ایجاد نشده این متغیر با عدد یک مقداردهی می شود و سپس cookie ایجاد شده و مقدار متغیر count در آن ذخیره می شود.

حال اگر بار اول اجرای این برنامه نباشد، به مقدار قبلی count یک واحد افزوده شده و به همین ترتیب در هر بار ملاقات صفحه توسط کاربر، یک واحد به شمارنده افزوده می شود.

**سؤال:** تاریخ انقضای cookie در مثال فوق چه مدت است؟

**تمرین:** با استفاده از تابع isset() تعداد بازدیدهای کاربر از سایت را به دست آورید.

### ۳-۱-۶- حذف یک cookie

• برای حذف یک cookie کافی است زمان انقضای آن را در گذشته قرار دهیم. اغلب مرورگرها کوکی هایی را که تاریخ انقضای آنها گذشته باشد، نادیده می گیرند و پس از بستن پنجره مرورگر، آنها را حذف می کنند.

#### مثال:

در مثال زیر زمان انقضای یک cookie با نام user، یک ساعت قبل از زمان اجرای برنامه تنظیم شده است.

```
setcookie ("user"," ", time()-3600);
```

• همچنین برای حذف cookie می توان از درج زمان انقضاء خودداری نمود. در این صورت پس از بستن پنجره فعلی مرورگر، cookie نیز حذف خواهد شد. در این صورت شکل کلی تابع به شکل زیر استفاده می شود :

```
setcookie ("cookie نام");
```

### ۲-۶- جلسه session

session یکی از مباحث اصلی شبکه و اینترنت است و توسط آن می توان اطلاعات کاربران را نگهداری کرد. تفاوت session با cookie در این است که اطلاعاتی که توسط session نگهداری می شوند همگی در سمت سرور بوده و هیچ اطلاعاتی در سمت سرویس گیرنده ذخیره نمی شود. (برخلاف کوکی).

هر کاربر که به سایتی وارد می‌شود، با اجرای اسکریپت PHP، یک جلسه ایجاد می‌شود و در صورتی که اولین بار باشد که وارد آن سایت می‌شود یک شناسه منحصر به فرد به او اختصاص داده می‌شود. این شناسه می‌تواند در تمام صفحات وب سایت، برای اهداف مختلف از قبیل شناسایی کاربر، تعداد دفعات مشاهده صفحات، زمان مشاهده و ... به کار برده شود. شناسه اختصاصی توسط دستور ( session\_id ) تعیین می‌شود. این شناسه در اینترنت با شکستن آدرس IP و به صورت اعداد هگزا دسیمال تعیین می‌شود.

مثال زیر یک نمونه از این شناسه را نشان می‌دهد.

### مثال:

88dccc2a835ae4cf04f95ce2347fbcf

### ۱-۲-۶- ایجاد جلسه

**نکته:** یکی از مهم‌ترین کاربردهای جلسه زمانی است که یک کاربر پس از وارد کردن مشخصات کاربری اش، به صفحات دیگر وب سایت می‌رود، در این صورت با استفاده از جلسه، می‌توان برای دسترسی به مشخصات او، در همه جای برنامه اقدام نمود.

برای شروع جلسه دستور ( session\_start ) به کار می‌رود. این دستور معمولاً در ابتدای صفحه و قبل از شروع سایر برچسب‌ها حتی DOCTYPE و html نوشته می‌شود.

### مثال:

با ذخیره و اجرای کد زیر در هر بار اجرای صفحه توسط مرورگر، جلسه شروع می‌شود و مشاهده خواهید کرد که شناسه اختصاص داده شده در هر بار اجرا تغییر پیدا می‌کند.

```
<?php
```

```
session_start();
```

```
?>
```

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
```

```
"http://www.w3.org
```

```
TR/xhtml1/DTD/xhtml1-strict.dtd">
```

```
<html>
```

```
<head>
```

```

<title>start a session</title>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
</head>
<body>
<?php
echo"your session id is : " . session_id();
?>
</body>
</html>

```

در فایل `ini` PHP یک شناسه با عنوان `auto_start` وجود دارد. این شناسه به طور پیش فرض مقدار صفر دارد. اگر این مقدار را به یک تغییر دهید، برای هر سند PHP به طور خودکار یک جلسه ایجاد می‌شود. اما اگر مقدار آن صفر باشد باید با استفاده از دستورات برنامه نویسی برای ایجاد جلسه اقدام شود.

## ۲-۶- متغیرهای جلسه

هر جلسه می‌تواند تعداد زیادی متغیر داشته باشد و توسط آنها می‌توان اطلاعات کاربران را در صفحات مختلف مدیریت کرد.

این متغیرها در یک آرایه فوق سراسری به نام `$_SESSION` نگهداری می‌شوند. تعریف متغیرهای جلسه به صورت زیر است:

مقدار متغیر = ['نام متغیر'] `$_SESSION`;

مثال: `$_SESSION['number'] = ۱۲۳۹;`

مثال: `$_SESSION['name'] = "zahra";`

پس از تعریف متغیرهای جلسه در یک صفحه، در صفحات دیگر می‌توان به آنها دسترسی پیدا کرد.

دسترسی به متغیرهای جلسه با همان نامی صورت می‌گیرد که هنگام تعریف متغیرها به کار رفته بود.

## مثال:

دستورات زیر را در یک فایل به نام `set-session.php` نوشته و ذخیره کنید.

```

<?php
session_start();

```



```
$_SESSION['username']="Ali";
echo ($_SESSION['username']);
?>
```

همان‌طور که اشاره شد، در صفحات و فایل‌های دیگر نیز می‌توان به محتوای جلسه دسترسی پیدا کرد، به‌منظور دسترسی به محتوای جلسه مثال قبل، کدهای زیر را در فایل دیگری به نام Access\_session.php نوشته و آن را اجرا کنید :

```
<?php
session_start();
echo ($_SESSION['username']);
?>
```

پس از اجرای کد فوق، مشاهده خواهید کرد نام 'Ali' که در فایل قبل به عنوان محتوای جلسه مقداردهی شده بود، نمایش پیدا می‌کند.

### مثال:

می‌خواهیم با استفاده از متغیرهای جلسه، تعداد دفعات مشاهده صفحه وب را توسط هر کاربر محاسبه کنیم :

```
<?php
session_start();
if(isset($_SESSION['views']))
$_SESSION['views']=$_SESSION['views']+1;
else
$_SESSION['views']=1;
echo "Views=". $_SESSION['views']. "<br/>";
?>
```

در مثال زیر شکل به کارگیری متغیرهای جلسه به‌طور کاربردی بررسی شده است.

### مثال:

می‌خواهیم در فایلی به نام session1.php، یک جلسه ایجاد کنیم، سپس در فایل دیگری به نام session2.php آن را بخوانیم.

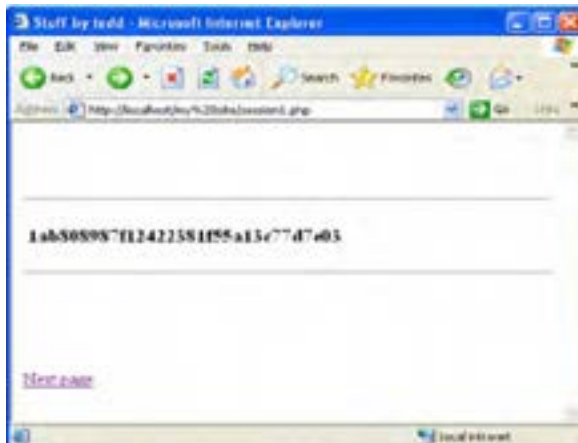
کد زیر را نوشته، سپس در فایلی به نام session1.php ذخیره کنید :

```
<?php
session_start();
?>
<h3> <?php echo session_id(); ?> </h3> <hr/>
<a href="session2.php">Next page</a>;
```

کد زیر را نیز در فایلی با نام session2.php ذخیره کنید :

```
<?php
session_start();
isset($_SESSION['count']) ? $_SESSION['count']++ : $_SESSION['count'] = 1;
?>
<?php echo( $_SESSION['count'] ); ?> times in this session
```

با اجرای فایل session1.php در مرورگر شناسه جلسه را مشاهده خواهید کرد.



شکل ۲-۶- فایل session1.php

چنانچه روی متن Next Page کلیک کنید، session2.php اجرا خواهد شد. حال با هر بار رفت و برگشت بین این دو صفحه و یا refresh کردن صفحه session2.php مشاهده خواهید کرد که شمارنده تعداد بازدیدهای صفحه، یک واحد افزایش می‌یابد.

### ۳-۲-۶- حذف جلسه و متغیرهای آن

حذف جلسه از طریق تابع `session_destroy()` صورت می‌گیرد.

اگر این تابع در یک جلسه استفاده شود، تمام متغیرها و شناسه‌های جلسه نیز حذف می‌شوند.

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

چنانچه قصد داشته باشید بدون از بین بردن جلسه، تنها یک یا چند متغیر آن را از بین ببرید، می‌توانید از تابع `unset()` استفاده کنید.

**مثال:**

```
<?php  
unset($_SESSION['views']);  
?>
```

### ۴-۲-۶- نگهداری تنظیمات شخصی کاربر

همان‌طور که گفته شد جلسه‌ها می‌توانند متغیرهای زیادی را در صفحه ایجاد کنند، این متغیرها می‌توانند حتی شامل اطلاعات محرمانه کاربر باشند و به دلیل این که اطلاعات جلسه به صورت یک آرایه فوق سراسری ذخیره می‌شود، می‌تواند مورد استفاده هکرها قرار گیرد. به همین دلیل باید به نوعی به دنبال مخفی کردن این اطلاعات بود.

رمزگذاری اطلاعات جلسه روشی است که توسط PHP قابل پشتیبانی است، این عمل توسط دستور `session_encode()` صورت می‌گیرد، درمقابل اگر اطلاعات `session` بخواند مورد استفاده فایل PHP قرار گیرد باید از حالت رمز خارج شود و این عمل توسط دستور `session_decode()` صورت می‌گیرد.

**مثال:**

```
<?php  
session_start();  
$_SESSION["login_ok"] = true;  
$_SESSION["name"] = "sina";  
$_SESSION["age"] = 34;
```

```
$_SESSION["data"]= session_encode();  
echo $_SESSION["data"];  
?>
```

سپس در صفحات دیگر برای رمز گشایی و مشاهده متغیرهای جلسه می‌توان دستورات زیر را به کار برد :

```
session_decode($_SESSION['data']);  
echo "Decoded session data for session"  
. session_id() . " : " . $_SESSION['login_ok'] . "&" . $_SESSION["name"] ;
```

Cookie فایل کوچک متنی است که از طرف سرویس دهنده و توسط مرورگر روی سیستم کاربر ایجاد و ذخیره می‌شود.

هر فایل cookie به تنهایی نمی‌تواند برای سیستم مشکلات امنیتی ایجاد کند اما می‌تواند در آسیب رساندن به امنیت سیستم شما دخیل باشد به طور مثال هکرها و افرادی که قصد سوءاستفاده از اطلاعات موجود بر روی سیستم شما را دارند، با خواندن محتویات cookie، به اطلاعات محرمانه شما دسترسی پیدا می‌کنند.

حداکثر اندازه فایل cookie، ۴KB، (کیلوبایت) است.

ایجاد cookie در PHP توسط دستور set cookie انجام می‌شود.

session یکی از مباحث اصلی شبکه و اینترنت است و توسط آن می‌توان اطلاعات کاربران را نگهداری کرد. هر کاربر که به سایتی وارد می‌شود، با اجرای اسکریپت PHP یک جلسه ایجاد می‌شود و در صورتی که اولین بار باشد که وارد آن سایت می‌شود، یک شناسه منحصر به فرد به او اختصاص داده می‌شود. این شناسه می‌تواند در تمام صفحات وب سایت برای اهداف مختلف از قبیل شناسایی کاربر، تعداد دفعات مشاهده صفحات، زمان مشاهده و ... به کار برده شود.

حذف جلسه از طریق کد() session\_destroy صورت می‌گیرد.

- ۱- مفهوم و کاربرد cookie را توضیح دهید.
  - ۲- مفهوم و کاربرد session را توضیح دهید.
  - ۳- در یک صفحه وب، فرمی برای دریافت مشخصات کاربر طراحی کنید و آن را برای صفحه‌ای با نام session\_c.php ارسال کنید، سپس در صفحه session\_c.php یک cookie برای ذخیره شناسه کاربر طراحی کنید که به مدت یک ماه انقضا داشته باشد.
  - ۴- در سؤال ۳، شناسه کاربر را در یک جلسه ذخیره کنید.
-