

به کارگیری Cookies ها و جلسه ها (sessions)

فصل ششم

هدف‌های رفتاری:

پس از آموزش این فصل هنرجو می‌تواند:

- 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 ببرید.

مقدمه

در یک وب سایت اطلاعات مختلفی از طرف کاربران به سمت سرور ارسال می‌شود. تنها بخشی از این اطلاعات به طور مستقیم و با اطلاع کاربر توسط سرور گیرنده قابل دریافت هستند و قسمت دیگری از آن‌ها بدون اینکه کاربر به طور آگاهانه اقدام به ارسال آن‌ها کرده باشد ذخیره می‌شوند.

اغلب این گونه داده‌ها توسط انواعی از امکانات زبان‌های برنامه‌نویسی به نام cookie ها و session ها ایجاد می‌شوند که در این فصل با ماهیت، کاربرد و نحوه مدیریت آن‌ها آشنا خواهید شد.

۱-۶ cookie

cookie ها فایل‌های کوچک متنی هستند که از طرف سرور ارسال شده و توسط مرورگر روی سیستم کاربر ایجاد می‌شوند. این فایل‌ها می‌توانند در موارد مختلفی به سرور ارسال شده و سرور دهنده برنامه کاربردی کمک کنند. به عنوان مثال می‌توانند به سرور دهنده وب کمک کنند تا کاربرانی را که قبلاً به سایت آن‌ها مراجعه کرده‌اند بشناسد، یا اینکه به یادآوری رمز عبور کاربر در مراجعات بعدی کمک کنند.

یک کاربر می‌تواند سیستم خود را طوری تنظیم کند که سرور دهنده اجازه تولید cookie را نداشته باشد و یا اینکه کاربر را هنگام تولید cookie مطلع کند. cookie ها می‌توانند اطلاعات مهمی را درون خود نگهداری کنند و می‌توانند توسط سرور دهنده‌ها غیر مجاز اطلاعات مهمی از هویت شما را فاش کنند. 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 می‌تواند با اهداف سوء از طریق خواندن cookie به رمز عبور شما دسترسی پیدا کند.

نکته: 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 سوء استفاده از اطلاعات موجود بر روی سیستم شما دارند، با خواندن محتویات cookie ها، به اطلاعات محرمانه شما دسترسی پیدا کنند.

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

یکی از ویژگی‌های cookie ها این است که پس از آنکه توسط یک مرورگر روی یک سیستم ایجاد شدند، فقط روی همان سیستم می‌توانند خوانده شوند. هر cookie یک آدرس URL منحصر به فرد دارد و سرور گیرنده از روی این آدرس تعیین می‌کند که cookie مربوط

به هر کاربر کدام است.

استفاده از cookie ها در سرویس گیرنده ها و مرورگرهای مختلف محدودیت هایی دارد. به عنوان مثال حداکثر cookie های مرورگر Internet explorer، ۳۰۰ فایل است. PHP قادر است عملیات ایجاد، مدیریت و حذف cookie ها را به خوبی انجام دهد.

۱-۱-۶ ایجاد cookie

ایجاد cookie در فایل ها و اسناد PHP توسط دستور setcookie انجام می شود. ساختار کلی این دستور به شکل زیر است:

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

؛ (اطلاعات دامنه، مسیر، تاریخ انقضای cookie، مقدار cookie، نام cookie) setcookie درج پارامترهای cookie اختیاری است و به دلخواه می توانید تعدادی از آن ها را از لیست پارامترها حذف کنید.

در پارامترهای تابع فوق Path برای تعیین مسیر به کاربر می رود و مقدار پیش فرض آن تمام پوشه های سرویس دهنده است.

Domain دامنه مجاز را تعریف می کند که به طور پیش فرض همان دامنه سرویس دهنده ای است که آن را ایجاد کرده است.

گزینه Secure سطح دسترسی به cookie را تعیین می کند و به صورت پیش فرض غیرفعال است.

در مثال زیر به محض اجرای صفحه در مرورگر یک cookie با نام user برای کاربر وب سایت ساخته می شود، این cookie با مقدار meisam مقداردهی شده و برای آن تاریخ انقضای یک ساعت در نظر گرفته شده است.

مثال ۱:



```
<?php
```

اگر با نسخه 6 Internet explorer یا قبل از آن کار می کنید 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 محتویاتی مشابه شکل زیر دارد.

```
setcookie ("user", "meisam", time()+3600);
?>
```

Create cookie !



شکل ۱-۶

برای اینکه cookie ایجاد شده را راحت‌تر پیدا کنید، بهتر است ابتدا تمام cookie های موجود بر روی سیستم را پاک کنید. در مثال ۲ یک cookie با زمان انقضای یک ماه ایجاد شده است.

نکته: چنانچه **Internet Explorer ۷** ویا بالاتر از آن استفاده می کنید برای مشاهده کوکی ها از منوی **Tools** گزینه **Options Interne** 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);
?>
.....
```

توسط تاریخ انقضا می توان اعتبار یا عدم اعتبار cookie را سنجید، اگر تاریخ انقضای یک cookie تمام شده باشد، مرورگر دیگر آن را برای سرویس دهنده ارسال نمی کند (تاریخ انقضا با قالب استاندارد GMT (گرینویچ) تعیین می شود). پارامترهای cookie ها توسط سرویس گیرنده تعیین می شود و می تواند علاوه بر پارامترهای ساختار فوق، پارامترهای دیگری را شامل شود، علاوه براین ممکن است یک

Cookie فقط شامل نام و یک پارامتر دیگر باشد.

اطلاعات دامنه تعیین می‌کند که Cookie توسط چه دامنه‌های قابل دسترسی است. اگر پارامتر امن نیز در خصوصیات Cookie تنظیم شده باشد، Cookie فقط مجاز است از طریق کانال‌های امن مانند https منتقل شود.

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



مثال:

می‌خواهیم یک Cookie ایجاد کنیم که تعداد بازدیدهای کاربر را از سایت تعیین کند. برای این کار ابتدا یک متغیر تعریف می‌کنیم که با هر بار ملاقات توسط کاربر، به مقدار آن یک واحد افزوده شود.

```
<?php
```

```
$count=$_COOKIE ["count"];
$count++;
setcookie ("count",$count,time()+36000);
echo "Counter is: " .$_COOKIE ["count"];
?>
```

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 می‌کند. به ساختار کد فوق دقت کنید:

در سطر دوم یک متغیر جدید تعریف شده است که در صورتی که اولین بار باشد که cookie اجرا می‌شود این متغیر با عدد صفر مقداردهی می‌شود و در سطر سوم به آن یک واحد اضافه می‌شود و در خط بعد cookie ایجاد شده و مقدار متغیر count در آن ذخیره می‌شود. 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 می‌شود.

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

نکته: دریک cookie اگر تاریخ انقضا مشخص نشود، cookie تا زمانی زنده است که پنجره IE باز باشد و به محض بستن این پنجره زمان انقضای cookie نیز به پایان خواهد رسید.

نکته: هر تابع یا دستوری که قبل از تابع setcookie در برنامه نوشته شود، اجرا نمی‌شود، لذا دستورات چاپ و... را هرگز قبل از این دستور قرار ندهید و همیشه ساختار cookie را در ابتدای صفحه پیاده سازی کنید.

۳-۱-۶ تابع isset()

یکی از توابع مهم و کاربردی در PHP تابع isset() است، توسط این تابع می‌توان از معرفی شدن متغیرها اطمینان حاصل کرد.

اگر متغیر مورد نظر قبلاً تعریف شده باشد، مقدار true و در غیر این صورت مقدار false را بر می‌گرداند.


فرض کنید در برنامه‌ای به مقدار یک متغیر نیاز داریم، اما نمی‌دانیم که این متغیر قبلاً معرفی و مقدار شده است یا خیر، در این صورت می‌توانیم از تابع isset() برای بررسی این مسأله استفاده می‌شود.



```
if (isset($first_name))
```

```
{
    print "$first_name is set" ;
}
```

در PHP علاوه بر دستور echo می‌توان از دستور print برای نمایش در خروجی استفاده نمود.

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

۴-۱-۶ حذف یک cookie

برای حذف یک `cookie` فقط کافی است زمان انقضای آن را در گذشته قرار دهیم. اغلب مرورگرها کوکی‌هایی را که تاریخ انقضای آن‌ها گذشته باشد نادیده می‌گیرند و پس از بستن پنجره مرورگر، آن‌ها را حذف می‌کنند.

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);
```

برای حذف `cookie` می‌توان از درج زمان و کشیدن آن به عقب 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 می‌شود:

```
setcookie ("cookie نام");
```

۲-۶ جلسه session

`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 کاربرده 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:



```
88dced2a835ae4cf04f95ce2347fbcf
```

برای شروع جلسه دستور `session_start()` به کار می‌رود. این دستور معمولاً در ابتدای صفحه و قبل از شروع سایر برچسب‌ها 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 PHP. ini یک شناسه باعنوان 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.

زبان PHP اطلاعات جلسه را در قالب فایل ذخیره می‌کند، مسیر ذخیره این فایل‌ها از طریق دستور session_save_path تعیین می‌شود.

مثال:



```
<?php
session_save_path("/home/sessions");
?>
```

۱-۲-۶ متغیرهای جلسه

هر جلسه می تواند تعداد زیادی متغیر داشته باشد و توسط آن ها می توان اطلاعات کاربران را در صفحات مختلف مدیریت کرد.

این متغیرها در یک آرایه فوق سراسری نگهداری می شوند.

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 می توان به آن ها دسترسی پیدا کرد، دسترسی به متغیرهای جلسه با همان نامی صورت می گیرد که هنگام تعریف متغیرها به کار رفته بود.

مثال:



```
<?php
session_start();
$_SESSION[<username>]=»Ali»;
echo ($_SESSION[<username>]);
?>
```

همانطور که اشاره شد، در صفحات و فایل های دیگر نیز می توان به محتوای جلسه دسترسی پیدا کرد، به منظور دسترسی به محتوای جلسه مثال قبل، کدهای زیر را در فایل دیگری 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 مقادردهی شده بود، نمایش پیدا می کند.

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

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

مثال:



می‌خواهیم درفایلی به نام session1.php، یک جلسه ایجاد کنیم، سپس درفایل دیگری به نام session2.php آن را بخوانیم. کد زیر را نوشته، سپس در فایلی به نام session1.php ذخیره کنید:

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

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

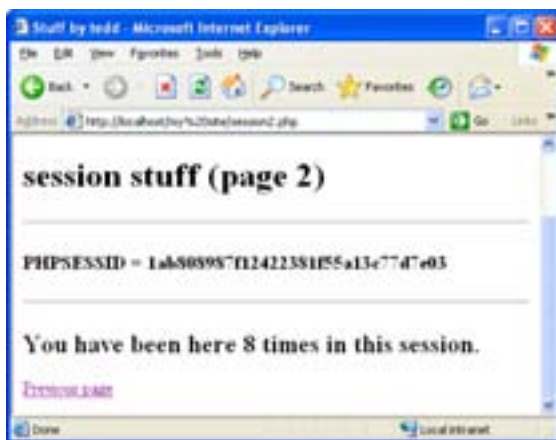
```
<?php
session_start();
($_SESSION['count']) ? $_SESSION['count']++ : $_SESSION['count'] = 1;
?>
<?php echo( $_SESSION['count'] ); ?> times in this session
```

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



شکل ۲-۶ فایل session1. php

چنانچه روی متن Page Next کلیک کنید، session2. php اجرا خواهد شد که در آن نیز همان شناسه صفحه قبل تکرار شده است، 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.



شکل ۳-۶ فایل session2. php

۲-۲-۶ حذف جلسه و متغیرهای آن

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

اگر این تابع در یک جلسه استفاده شود، تمام متغیرها و شناسه های 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']);
?>
```

۳-۲-۶ نگهداری تنظیمات شخصی کاربر

همان‌طور که گفته شد جلسه‌ها می‌توانند متغیرهای زیادی 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` بنخواهد 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 می‌وان دستورات زیر را به کار برد:

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

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.

چکیده فصل:

Cookie ها فایل‌های کوچک متنی هستند که از طرف سرویس دهنده و توسط مرورگر روی سیستم کاربر ایجاد می‌شوند.

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 دارند، با خواندن محتویات cookie ها، به اطلاعات محرمانه شما دسترسی پیدا کنند حداکثر اندازه فایل cookie ۴۰KB (کیلوبایت) است.

ایجاد cookie در PHP توسط دستور `setcookie` 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 کاربرده شود.

زبان PHP اطلاعات جلسه را در قالب فایل ذخیره می‌کند، مسیر ذخیره این فایل‌ها از طریق دستور `session_save_path` تعیین می‌شود.

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

خودآزمایی:

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 سوال ۳، شناسه کاربر را در یک جلسه ذخیره کنید.