برای تبادل اطلاعات بین صفحات مختلف در وب چند روش وجود دارد:

۱. **ارسال مقادیر از روشهای GET یا POST**. ساده ترین روش تبادل داده است. اما حجم ترافیک شبکه را بالا می برد، به این دلیل که همه مقادیر در هر درخواست باید مجددا ارسال شوند. همچنین از امنیت پایینی برخوردار است و برای ذخیره بلند مدت اطلاعات کاربرد ندارد.

۲. **استفاده از مفهوم کوکی (cookie)** . کوکی (cookie) ها فایل های کوچکی هستند که برای ذخیره اطلاعات روی ماشین کاربر به کار می روند. برای هر کوکی یک نام، مقدار و تاریخ انقضا تعریف می شود. کوکی ها با نامشان قابل دسترسی هستند و با رسیدن زمان انقضا، از بین می روند. کوکی ها معمولا برای ذخیره بلند مدت اطلاعات روی ماشین کاربر به کار می روند. به عنوان مثال کاربر یک بار اطلاعات کاربری و رمز عبورش را وارد کرده و در کوکی ذخیره می شود. سپس در مراجعات بعدی، این اطلاعات از کوکی خوانده شده و نیاز نیست دوباره وارد شود.

در php از این دستور برای ایجاد کوکی استفاده می شود:

setcookie(نام کوکی , مقدار کوکی , زمان انقضا);

و برای دسترسی به مقادیر ثبت شده در کوکی از آرایه $_COOKIE استفاده می شود. مثال:

```
setcookie('c1','value1',time()+5*60);
```

یک کوکی به نام c1 با مقدار value1 با زمان انقضای ۵*۶۰ ثانیه بعلاوه زمان فعلی یعنی ۵ دقیقه بعد، تعریف می کند. سپس در برنامه بعدی بررسی می کند اگر کوکی با این نام موجود است، مقدار آن را نمایش می دهد:

```
if(isset($_COOKIE["c1"]))
  echo $_COOKIE["c1"];
```

۳. **نشست – جلسه – session**

پروتکل http یک پروتکل بدون حالت (stateless) است. تعامل کاربر با سرور از طریق درخواستهایی (Request) است که کاربر برای سرور ارسال می کند و پاسخهایی (Response) که سرور به کاربر برمی گرداند. هر درخواستی یک پاسخ دارد و درخواست ها مجزا از هم در نظر گرفته می شوند. در ارسال هر درخواست جدید، تاریخچه ای از درخواستهای قبلی وجود ندارد. به عبارت دیگر، در صفحه وب که در حال حاضر در حال اجراست، هیچ اطلاعاتی از صفحات قبلی و تعاملات قبلی که کاربر با سرور داشته، موجود نیست. اما مواقعی هست که نیاز داریم بدانیم کاربر قبلا چه صفحاتی را باز نموده و چه اطلاعاتی برای سرور ارسال نموده و در حال حاضر در چه وضعیتی قرار دارد. به عنوان مثال کاربر یک بار اطلاعات کاربری و رمز عبور خود را برای سرور ارسال نموده و احراز هویت شده است. در صفحه بعدی که کاربر باز می کند باید بدانیم این همان کاربری است که قبلا احراز هویت شده و مجاز به دسترسی به منابع مورد نظر می باشد. برای این منظور، مفهوم نشست یا جلسه (session) مطرح می شود. نشست بین سرور

و یک کاربر برقرار می شود. از یک زمان مشخص شروع شده و مدت زمان مشخصی معتبر است و بعد از گذشت مدت زمان مشخص، منقضی می شود. در یک نشست، متغیرهایی می تواند تعریف شود و به آنها مقادیری اختصاص داده شود. متغیرهای تعریف شده و مقادیر آنها در هر درخواست کاربر قابل دسترسی است. از این طریق می توان مقادیری را بین صفحات مختلف به اشتراک گذاشت. نشست در مدل OSI به عنوان لایه پنجم شبکه و در مدل TCP/IP به عنوان بخشی از لایه کاربرد (Application) پیاده سازی می شود.

در زبان php برای کار با یک نشست ابتدا دستور session_start() را فراخوانی می کنیم که باعث می شود یک نشست جدید ایجاد شده یا اگر نشستی قبلا ایجاد شده بود، ادامه یابد. مقادیر مورد نظر در آرایه ای به نام $_SESSION ذخیره و استفاده می شوند. به عنوان مثال:

```
<?php
session_start();
$user=$_POST["username"];
$pass=$_POST["password"];
$_SESSION["username"]=$user;
$_SESSION["password"]=$pass;
.
.
.
?>
```

در برنامه فوق مقادیر نام کاربری و رمز عبور از کاربر دریافت شده و در نشست ذخیره می شود و در برنامه دیگر می توانیم از آنها استفاده کنیم:

```
<?php
session_start();
if(isset($_SESSION["username"]) && isset($_SESSION["password"]))
{
 $user=$_SESSION["username"];
 $pass=$_SESSION["password"];
 .
 .
 .
}
else
 echo "Athentication failed";
?>
```

اطلاعات نشست می تواند در حافظه سرور یا حافظه کاربر (client) ذخیره شود. معمولا اطلاعات متغیرها روی سرور ذخیره شده و شماره شناسه نشست (session id) در قالب یک کوکی (cookie) روی ماشین کاربر ذخیره می شود. کاربر در هر درخواست، شماره شناسه نشست خود را برای سرور ارسال می کند.

یکی از تفاوتهای کوکی با نشست در این است که کوکی برای ذخیره بلند مدت اطلاعات روی ماشین کاربر به کار می رود و اطلاعات بعد از قطع ارتباط کاربر هم باقی مانده و قابل دسترسی هستند. اما نشست برای نگهداری کوتاه مدت اطلاعات به کار می رود و معمولا با قطع ارتباط کاربر با سرور، از بین می رود. مزیت استفاده از نشست نسبت به کوکی در این است که ذخیره اطلاعات نشست می تواند در سرور انجام شده و در اینصورت نیازی به تبادل داده بین

کاربر و سرور در هر درخواست نمی باشد که ضمن کاهش حجم ترافیک، از امنیت بالاتری هم برخوردار می باشد. کوکی ها زمانی استفاده می شوند که اطلاعات برای مدت طولانی قرار است باقی بماند.

برای پایان دادن به یک نشست در php از دستور session_setroy() استفاده می شود و برای حذف یک متغیر از نشست:

```
unset($_SESSION["نام متغیر"])
```

برای تعیین مدت زمان اعتبار یک نشست، از تابع زیر استفاده می شود:

```
session_set_cookie_params(زمان بر حسب ثانیه);
```