

Using cookies

```
bool setcookie ( string name [, string value [, int expire [, string path [, string domain [,  
int secure]]]]])
```

Taking the example of sorting a messageboard index, a cookie would need to be placed that holds the user's preference on message sorting - whether they want it newest first, oldest first, or sorted alphabetically. Take a look at this piece of code:

```
<?php  
    if (!isset($_COOKIE['Ordering'])) {  
        setcookie("Ordering", $_POST['ChangeOrdering'], time() + 31536000);  
    }  
?>
```

```
<form method="post" action="mbprefs.php"> Reorder messages:  
<select name="ChangeOrdering">  
    <option value="DateAdded ASC">Oldest first</option>  
    <option value="DateAdded DESC">Newest first</option>  
    <option value="Title ASC">By Title, A-Z</option>  
    <option value="Title DESC">By Title, Z-A</option>  
</select>  
<input type="submit" value=" Save Settings " />  
</form>
```

Starting a session

```
bool session_start ( void )
```

Adding session data

To set a session variable, use syntax like this:

```
$_SESSION['var'] = $val;  
$_SESSION['FirstName'] = "Jim";
```

Reading session data

```
<?php  
    $_SESSION['foo'] = 'bar';  
    print $_SESSION['foo'];  
?>
```

Removing session data

To extend the previous script to remove data, use this:

```
<?php  
    $_SESSION['foo'] = 'bar';  
    print $_SESSION['foo'];  
    unset($_SESSION['foo']);  
?>
```

Ending a session

bool session_destroy (void)

To end a session and clear up its data, use this code:

```
<?php
    session_start();
    $_SESSION = array();
    session_destroy();
?>
```

Checking session data

```
<?php
    session_start();

    if (isset($_SESSION['FirstName'])) {
        /// your code here
    }
?>
```