Server side SESSION

So, what is the SESSION?
Session in term of server-side scripting concern is the connection between a client (browser) and the server that hosting the web application. Each time a client connects to a web server, there is a connection happens. The connection can be registered in the server and there are a lot of information regarding the connection that can be recorded in the web server. For example the web browser being used or the IP number.

Starting a session
Each unique connection provides a unique session id. To enable the session management of a page, first the *session_start*() function must be executed. This function can only work if there is no information from the page was sent to the client, not even a space or a single bit.

For example, try the following code.

```
<html>
<head>
<title>Session Testing</title>
</head>

<body>
<?php
     session_start();
     //trying to start session management
?>
The session management in this page won't work, because there are a
lot  of  characters  has  been  sent  to  the  client  before  the
session_start() is executed.
</body>
</html>
```
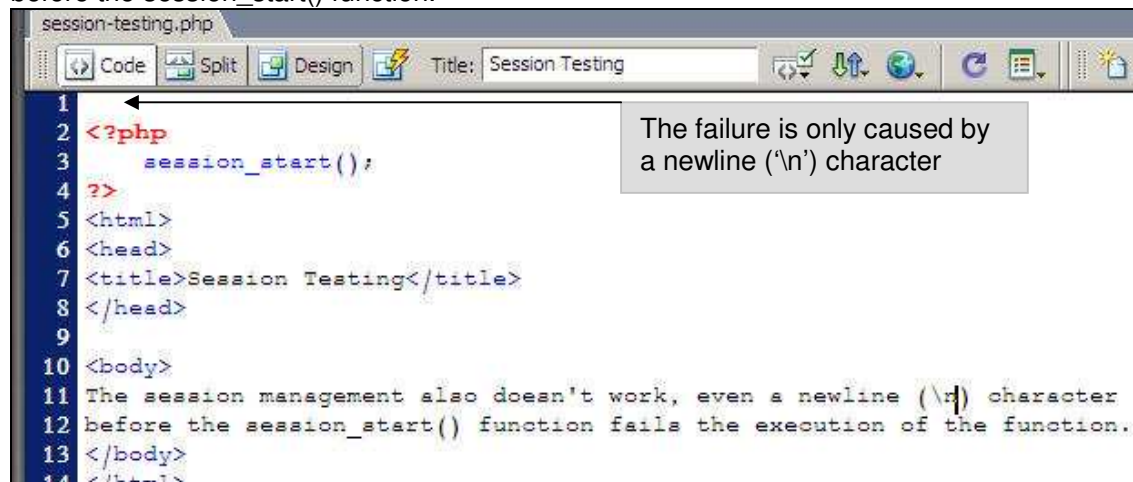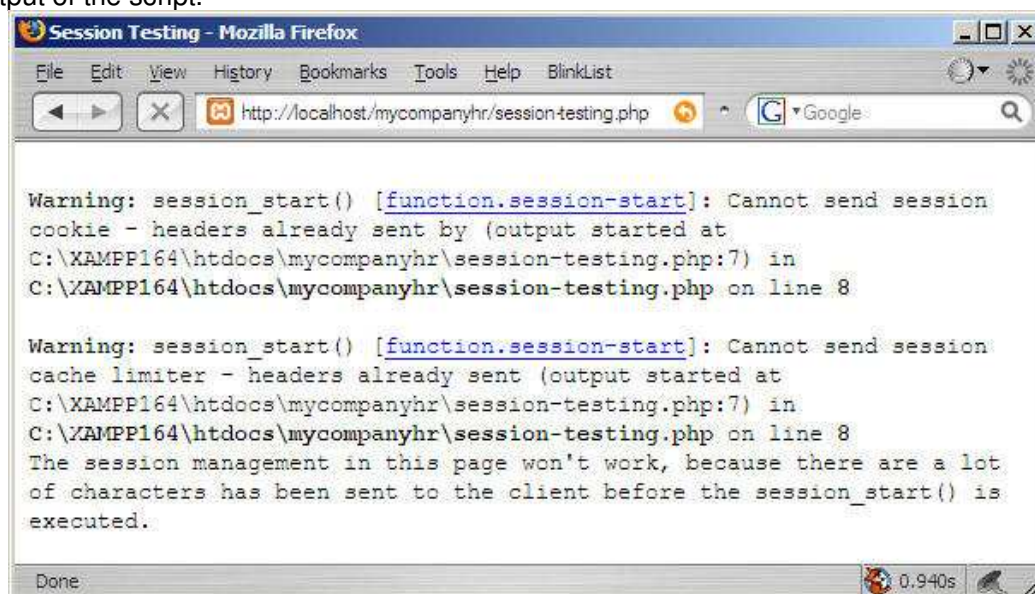
> This information has been sent to the client. It is the reason why the sesion_start() function fail to executed.

Output of the script.

This example also doesn't work. The problem is caused by only a newline ('\n') character before the session_start() function.

```
session-testing.php
  Code    Split    Design        Title: Session Testing

1
2  <?php
3      session_start();
4  ?>
5  <html>
6  <head>
7  <title>Session Testing</title>
8  </head>
9
10 <body>
11 The session management also doesn't work, even a newline (\n) character
12 before the session_start() function fails the execution of the function.
13 </body>
14 </html>
```
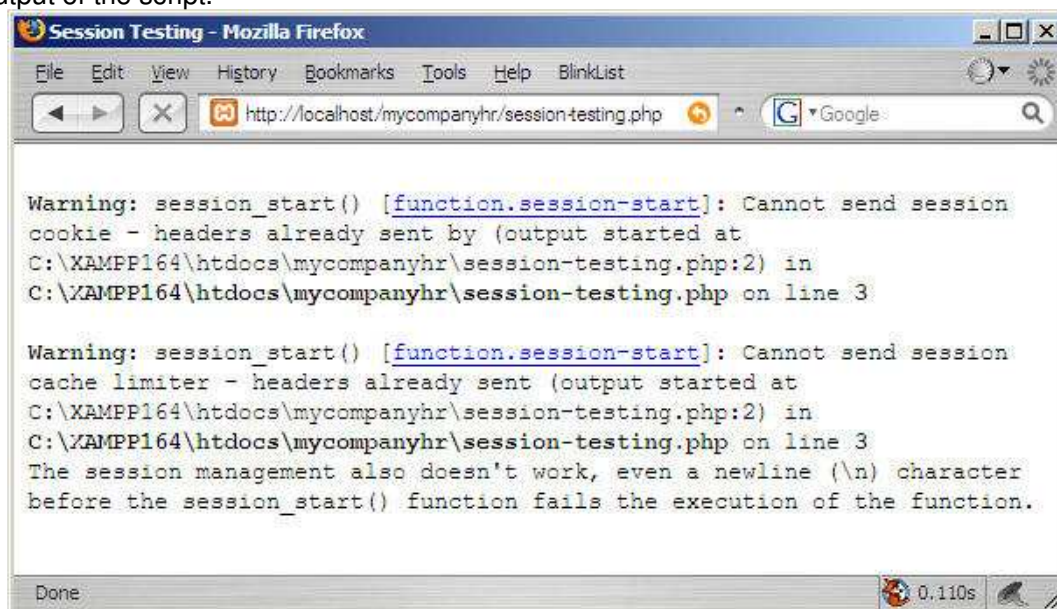
> The failure is only caused by a newline ('\n') character

Output of the script.

```
Session Testing - Mozilla Firefox

File  Edit  View  History  Bookmarks  Tools  Help  BlinkList

http://localhost/mycompanyhr/session-testing.php     Google

Warning: session_start() [function.session-start]: Cannot send session
cookie - headers already sent by (output started at
C:\XAMPP164\htdocs\mycompanyhr\session-testing.php:2) in
C:\XAMPP164\htdocs\mycompanyhr\session-testing.php on line 3

Warning: session_start() [function.session-start]: Cannot send session
cache limiter - headers already sent (output started at
C:\XAMPP164\htdocs\mycompanyhr\session-testing.php:2) in
C:\XAMPP164\htdocs\mycompanyhr\session-testing.php on line 3
The session management also doesn't work, even a newline (\n) character
before the session_start() function fails the execution of the function.

Done                                                    0.110s
```
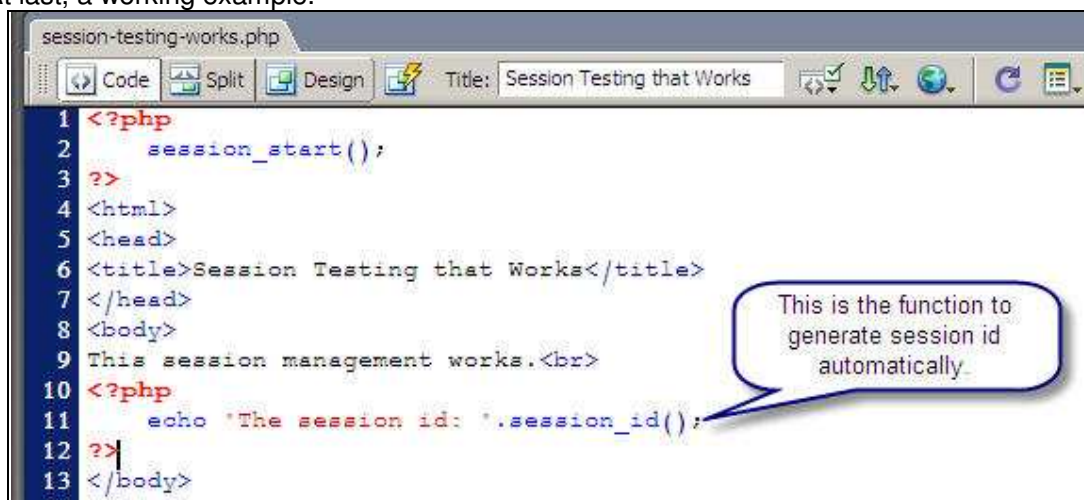
At last, a working example.

```
session-testing-works.php
  Code    Split    Design        Title: Session Testing that Works

1  <?php
2      session_start();
3  ?>
4  <html>
5  <head>
6  <title>Session Testing that Works</title>
7  </head>
8  <body>
9  This session management works.<br>
10 <?php
11     echo 'The session id: '.session_id();
12 ?>
13 </body>
14 </html>
```
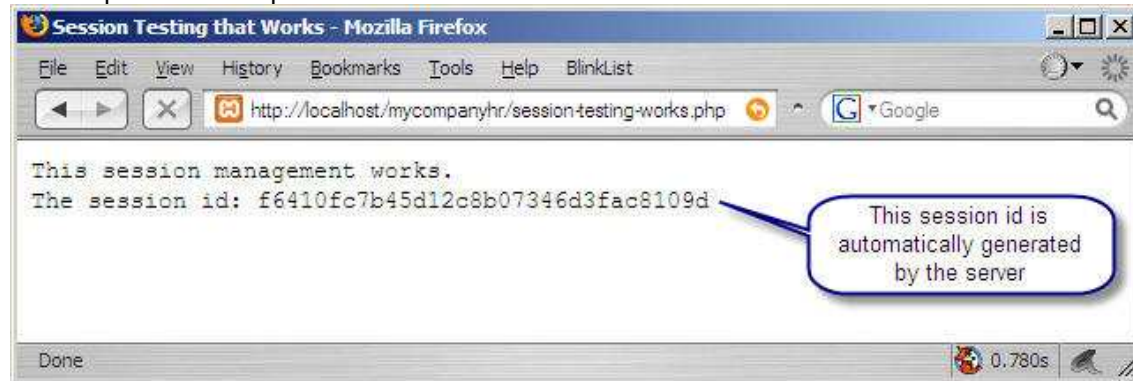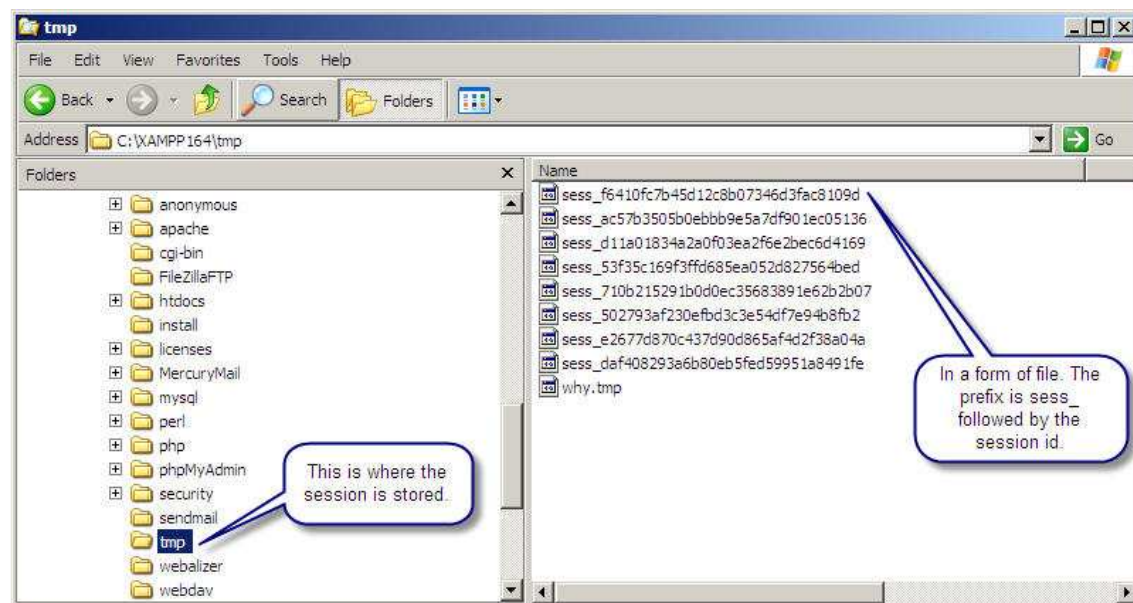
> This is the function to generate session id automatically.

The output of the script.

So where is this session information is stored?
Currently we are using XAMPP. By default the Apache web server store the session information in the tmp directory. Each session will generate a file with the sess_ as the prefix, and followed by the session id.

Try to open the file and you will nothing is stored in the file where the latest session was created using the script in previous page. This is because there is no session variable is registered.

Session variable registration need to be done in order to store values to be used as long as the session (or the connection) is established. For example, in a web application there are few files connected to each other. In order to maintain a certain amount of value to be used for every pages in the system, we need to store it in the session variable. For example the username of the user who are using the system.

Registering a session variable
To store the username (so that all the pages in the system will be displaying the same username) we need to register a session variable. The following script is to register few session variable.

```php
<?php  session_start();  ?>
<html>
<head>
<title>Session Registration</title>
</head>
<body>
This page is to register user's information.<br>
<?php
if(!isset($_SESSION['sessionid'])){
      $_SESSION['sessionid']=session_id();//session id
      $_SESSION['browser']=$_SERVER['HTTP_USER_AGENT'];//browser
      $_SESSION['ipnumber']=$_SERVER['REMOTE_ADDR'];//client's ip
      $_SESSION['username']="kerul";//the username
      $_SESSION['name']="Khirulnizam Abd Rahman";//full name
      $_SESSION['level']=1;//user access level
}
?>
</body>
</html>
```

The script provide the registration of four session variable namely; *sessionid*, *username*, *name* and *level*, with their respective value.

Now open the file where the server store the session id in notepad.
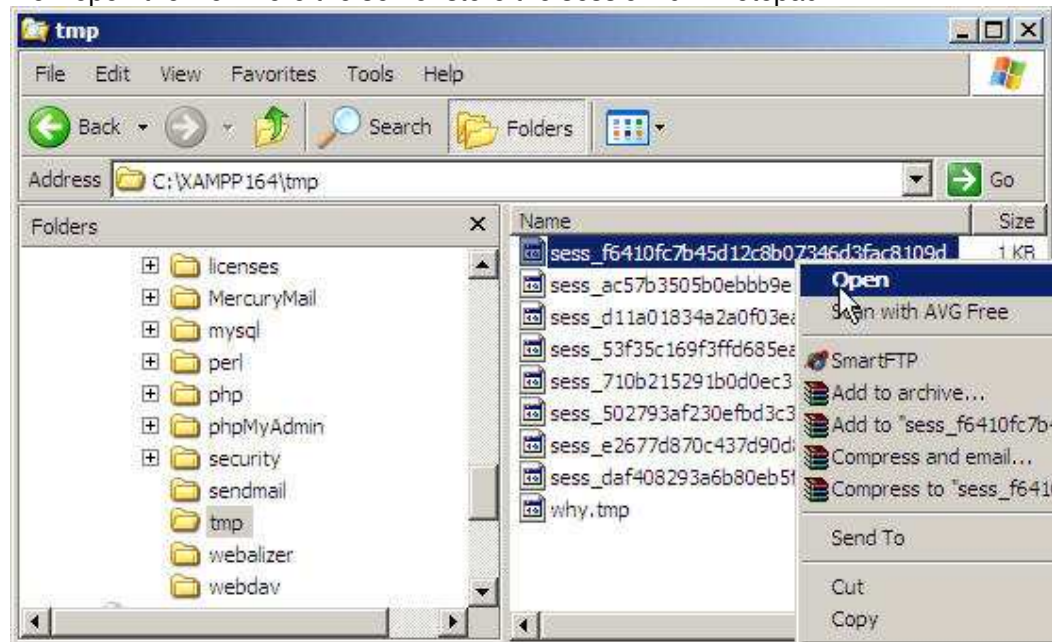
This is the content of the session file.

```
sess_f6410fc7b45d12c8b07346d3fac8109d - Notepad
File  Edit  Format  View  Help
sessionid|s:32:"f6410fc7b45d12c8b07346d3fac8109d";
browser|s:92:"Mozilla/5.0 (Windows; U; Windows NT
5.1; en-US; rv:1.8.1.12) Gecko/20080201
Firefox/2.0.0.12";ipnumber|s:9:"127.0.0.1";usernam
e|s:5:"kerul";name|s:22:"Khirulnizam Abd
Rahman";level|i:1;
```

Magnified:

```
sessionid|s:32:"f6410fc7b45d12c8b07346d3fac8109d";
```

session variable name   value

data type
s for string   string length

Using the session variable.

```php
<?php  session_start(); ?>
<html>
<head>
<title>Session Usage</title>
</head>
<body>
This page is to use user's information stored in session.<br>
<?php
      echo $_SESSION['sessionid'].'<br>';
      echo $_SESSION['browser'].'<br>';
      echo $_SESSION['ipnumber'].'<br>';
      echo $_SESSION['username'] .'<br>';
      echo $_SESSION['name'] .'<br>';
      echo $_SESSION['level'] .'<br>';
?>
</body>
</html>
```

Checking the session variable.

```php
<?php
      session_start();
?>
<html>
<head>
<title>Session Checking</title>
</head>
<body>
This page is to check whether user's information are stored in the
session.<br>
```

```php
<?php
     if (isset($_SESSION['sessionid'])){
          echo $_SESSION['sessionid'] .'<br>';
          echo $_SESSION['browser'].'<br>';
          echo $_SESSION['ipnumber'].'<br>';
          echo $_SESSION['username'] .'<br>';
          echo $_SESSION['name'] .'<br>';
          echo $_SESSION['level'] .'<br>';
     }
     else{
          $_SESSION['sessionid']=session_id();
          $_SESSION['browser']=$_SERVER['HTTP_USER_AGENT'];
          $_SESSION['ipnumber']=$_SERVER['REMOTE_ADDR'];
          $_SESSION['username']="kerul";
          $_SESSION['name']="Khirulnizam Abd Rahman";
          $_SESSION['level']=1;
     }
?>
</body>
</html>
```

Destroy the session variable.

```php
<?php
     session_start();
?>
<html>
<head>
<title>Session Checking</title>
</head>
<body>
This page is to check whether user's information are stored in the
session.<br>
<?php
     if (isset($_SESSION['sessionid'])){
          session_destroy();//this to destroy all session info
     }
?>
</body>
</html>
```