

# Cookies e sessões

- HTTP é um protocolo que não mantém estado, isto é **não tem memória**.
- Cada pedido que um browser faz ao servidor web é independente do pedido anterior.
- Muitas aplicações necessitam de manter o estado ao saltar de página em página.
- Exemplo: carrinho de compras num site de comércio electrónico.

## **Aplicações que mantêm estado**

- guardam informação de estado no browser do cliente utilizando cookies.
- guardam informação de estado no servidor web utilizando sessões).

# Cookies

- Cookies são variáveis enviadas pelo servidor web para o browser através do protocolo HTTP.
- Cookies ficam guardados no browser.
- Cookies são enviados em futuros pedidos do browser para esse servidor web.

## **Cookies (cont.)**

- o Cookie pode ter parametros que:
  - façam com que expire a uma determinada data/hora.
  - restringe o seu envio a um determinado dominio.
  - só permitem o seu envio no acesso a determinados directorios dentro do site.

## **Limitações dos cookies**

- Browser só pode guardar até 20 cookies enviadas de um determinado domínio.
- Cookies não podem ter mais do que 4KB.
- Utilizadores podem desactivar cookies no browser.

## Cookies em PHP

- PHP tem a função `setcookie` que gera o cabeçalho `HTTP_COOKIE =`
- os cookies ficam disponíveis em PHP através do array associativo `$_COOKIE`

## cookie - example.php

```
<?php
    // set a cookie called count. cookie expires after 600 seconds.
    $count = 0;
    setcookie("count", $count, time()+600);
?>

<html>
<head>
    <title>cookie-example.php</title>
</head>

<body>
<h1>Cookie 'count' created with value <?=$count?></h1>

<ul>
    <li>Click here to <a href="increment-count.php">increment count</a></li>
    <li>Click here to <a href="delcookie.php">destroy the count
cookie</a></li>
</ul>

</body>
</html>
```

## código HTTP enviado pelo servidor

```
HTTP/1.0 200 OK
Date: Tue, 24 Apr 2007 15:19:25 GMT
Server: Apache/1.3.33 (Debian GNU/Linux) PHP/5.2.1-0.dotdeb.1 with
Suhosin-Patch mod_ssl/2.8.22 OpenSSL/0.9.7e mod_python/2.7.10 Python/2.3.4
X-Powered-By: PHP/5.2.1-0.dotdeb.1
Set-Cookie: count=0; expires=Tue, 24-Apr-2007 15:29:25 GMT
Content-Type: text/html; charset=iso-8859-1
X-Cache: MISS from proxy1.si.ualg.pt
X-Cache-Lookup: MISS from proxy1.si.ualg.pt:8080
Proxy-Connection: close
```

```
<html>
<head>
  <title>Cookies</title>
</head>

<body>
<h1>Cookie 'count' created with value 0</h1>
<ul>
<li>Click here to <a href="increment-count.php">increment count</a></li>
<li>Click here to <a href="delcookie.php">destroy the count
cookie</a></li>
</ul></body>
```



## **código HTTP enviado pelo browser quando se clica em "increment count"**

```
GET http://www.deei.fct.ualg.pt/~figo/cookies/increment-count.php
HTTP/1.1
Host: www.deei.fct.ualg.pt
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.0; en-US;
rv:1.7.12) Gecko/20050915 Firefox/1.0.7
Accept:
text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,
text/plain;q=0.8,image/png,*/*;q=0.5
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 300
Proxy-Connection: keep-alive
Referer: http://www.deei.fct.ualg.pt/~figo/cookies/cookie-
example.php
Cookie: count=0
```

## increment-count.php

```
<?php
// Check if cookie count is enabled
if(isset($_COOKIE["count"])) {
    // if yes, increment its value and display it
    $count = $_COOKIE["count"];
    $count++;
    setcookie("count", $count, time()+600);
}
?>
<html>
<head>
    <title>increment-count.php</title>
</head>

<body>
<h1>Cookie 'count' incremented. Value is <?=$count?></h1>
<ul>
    <li>Click here to <a href="increment-count.php">increment count</a></li>
    <li>Click here to <a href="delcookie.php">destroy the count
cookie</a></li>
</ul>

</body>
</html>
```

# Sessões

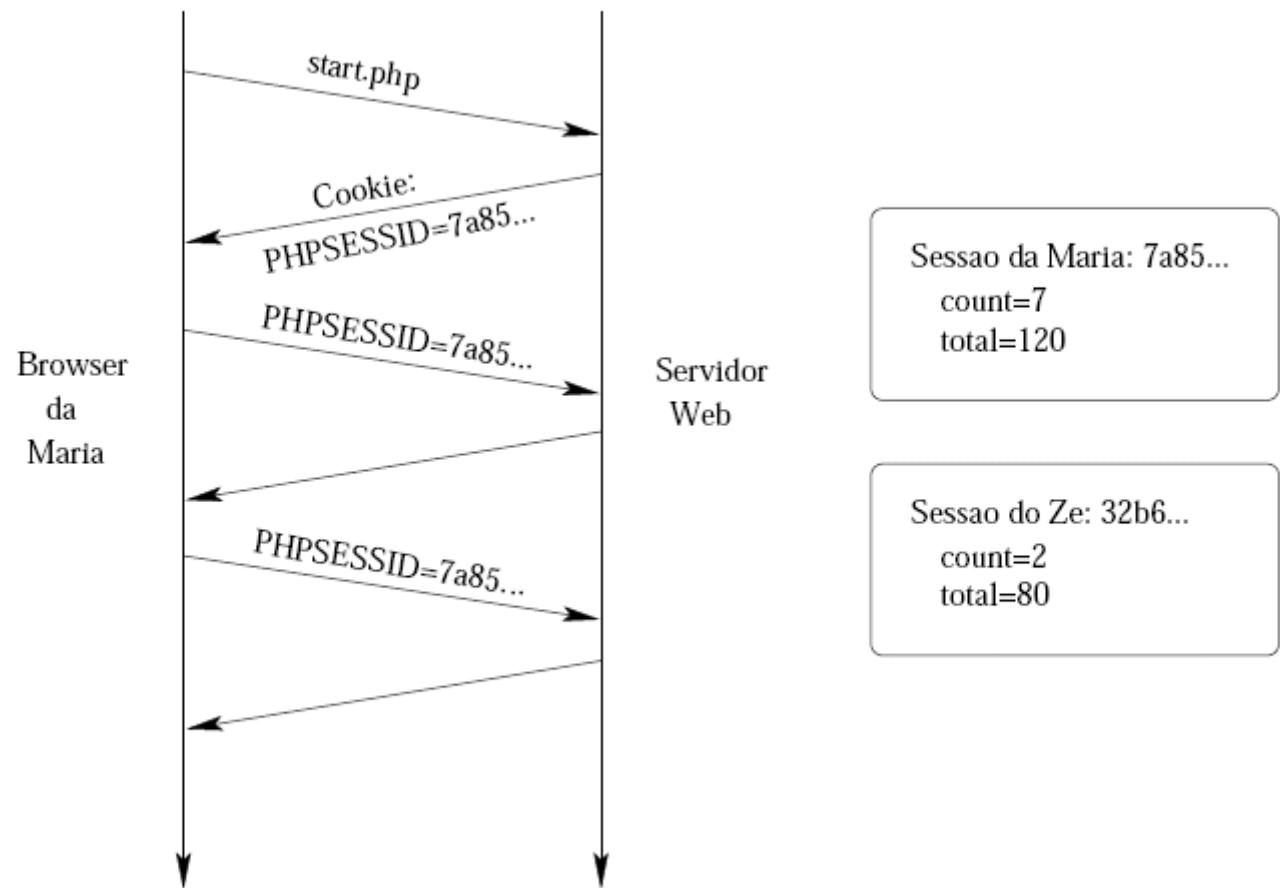
- Permitem guardar informação de estado (variáveis) no servidor.
- Em vez do browser guardar informacao sobre várias variáveis, guarda apenas um "session id".
- Este "session id" permite aceder às variáveis associadas à sessão.

## **Sessoes (cont.)**

- Variaveis de sessao têm de ser guardadas no servidor web (por exemplo, em ficheiros de texto ou numa base de dados).
- Sessões necessitam de ter um "timeout". Caso contrario, o servidor web poderá nao saber se uma sessao terminou ou nao.

## Sessoes com PHP

- função `session_start()` cria uma nova sessao ou identifica uma sessao já estabelecida.
- Quando um script PHP chama `session start()` pela primeira vez, é gerado um "session id".
- Por defeito, este "session id" é enviado como cookie para o browser com o nome de PHPSESSID
- o "session id" é uma string aleatoria de 32 digitos hexadecimais, tal como  
fcc17f071bca934ae2f24f290343c3b7



## Sessoes com PHP (cont.)

- Depois de se fazer `session_start()` podemos guardar ou aceder a variaveis de sessao disponiveis no array associativo  
`$_SESSION["equipa"] = "Benfica";`
- Uma vez registadas, as variaveis ficam disponiveis para todos os scripts que chamem `session_start()`
- No exemplo, `equipa` é uma variavel de sessao que fica guardada no servidor web com o valor "Benfica".

## Sessoes com PHP (cont.)

- Variaveis podem ser removidas da sessao atribuindo-lhes o valor FALSE  
`$_SESSION["equipa"] = FALSE;`
- a função `session_id()` devolve o "session id" da sessão.
- `session_destroy()` elimina todas as variaveis associadas à sessao.



## Exemplo 1 – registo em vários passos: page1.html

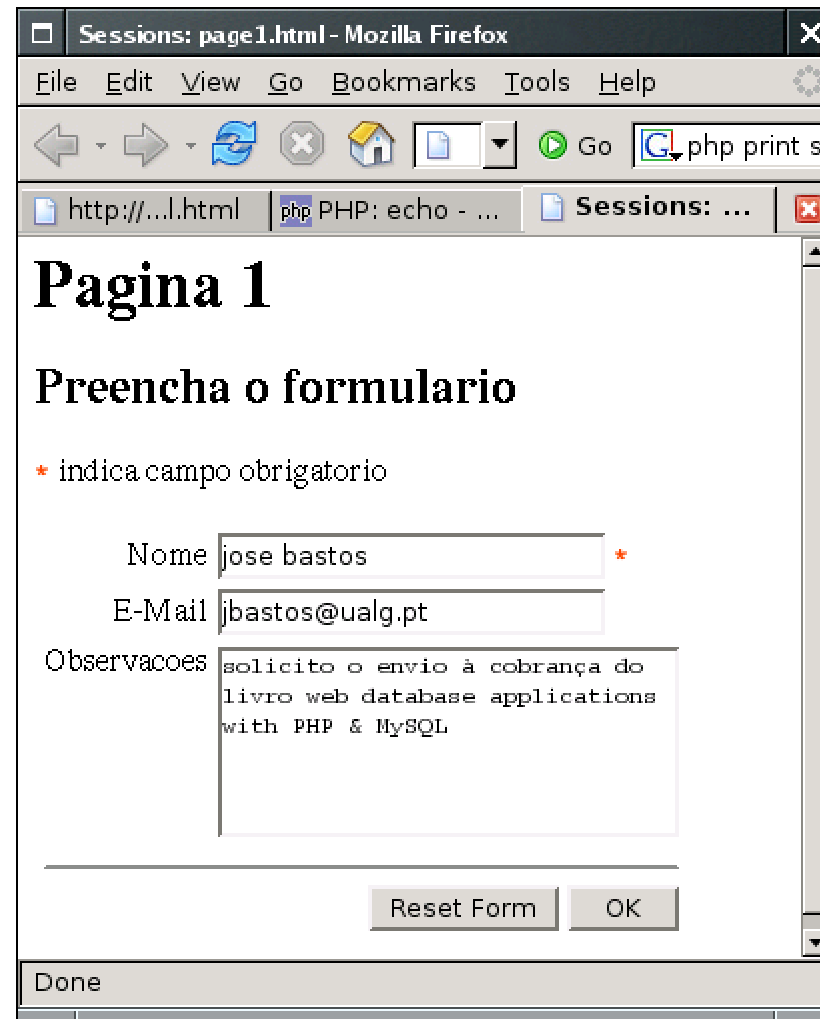
```
<html>
<head>
  <title>Sessions: page1.html</title>
</head>
<body>

<h1>Pagina 1</h1>
<h2>Preencha o formulario</h2>
<p><font color="orangered" size="+1"><tt><b>*</b></tt></font>
  indica campo obrigatorio</p>
<form method="post" action="page2.php">
<table border="0" cellpadding="0" cellspacing="5">
  <tr>
    <td align="right">Nome</td>
    <td><input name="nome" type="text" size="25" />
      <font color="orangered"
size="+1"><tt><b>*</b></tt></font>
    </td>
  </tr>
  <tr>
```

```

        <td align="right">E-Mail</td>
        <td><input name="email" type="text" size="25" /></td>
    </tr>
    <tr valign="top">
        <td align="right">Observacoes</td>
        <td>
            <textarea wrap="soft" name="obs" rows="5"
            cols="30">Enter your request here</textarea>
        </td>
    </tr>
    <tr>
        <td align="right" colspan="2">
            <hr noshade="noshade" />
            <input type="reset" value="Reset Form" />
            <input type="submit" name="submitok" value="    OK    "
        </td>
    </tr>
</table>
</form>
</body>
</html>

```



## registo em vários passos: page2.php

```
<?php
```

```
session_start();
```

```
if($_POST['nome']==='') {  
    header("Location: page1.html");  
    exit;  
}
```

```
$_SESSION['nome'] = $_POST['nome'];  
$_SESSION['email'] = $_POST['email'];  
$_SESSION['obs'] = $_POST['obs'];  
?>
```

```
<h1>Pagina 2</h1>
```

```
<p>
```

```
Ola <b><?=$_SESSION['nome']?></b>. Bemvindo a este site.  
Preencha mais estes campos.
```

```
</p>
```

```

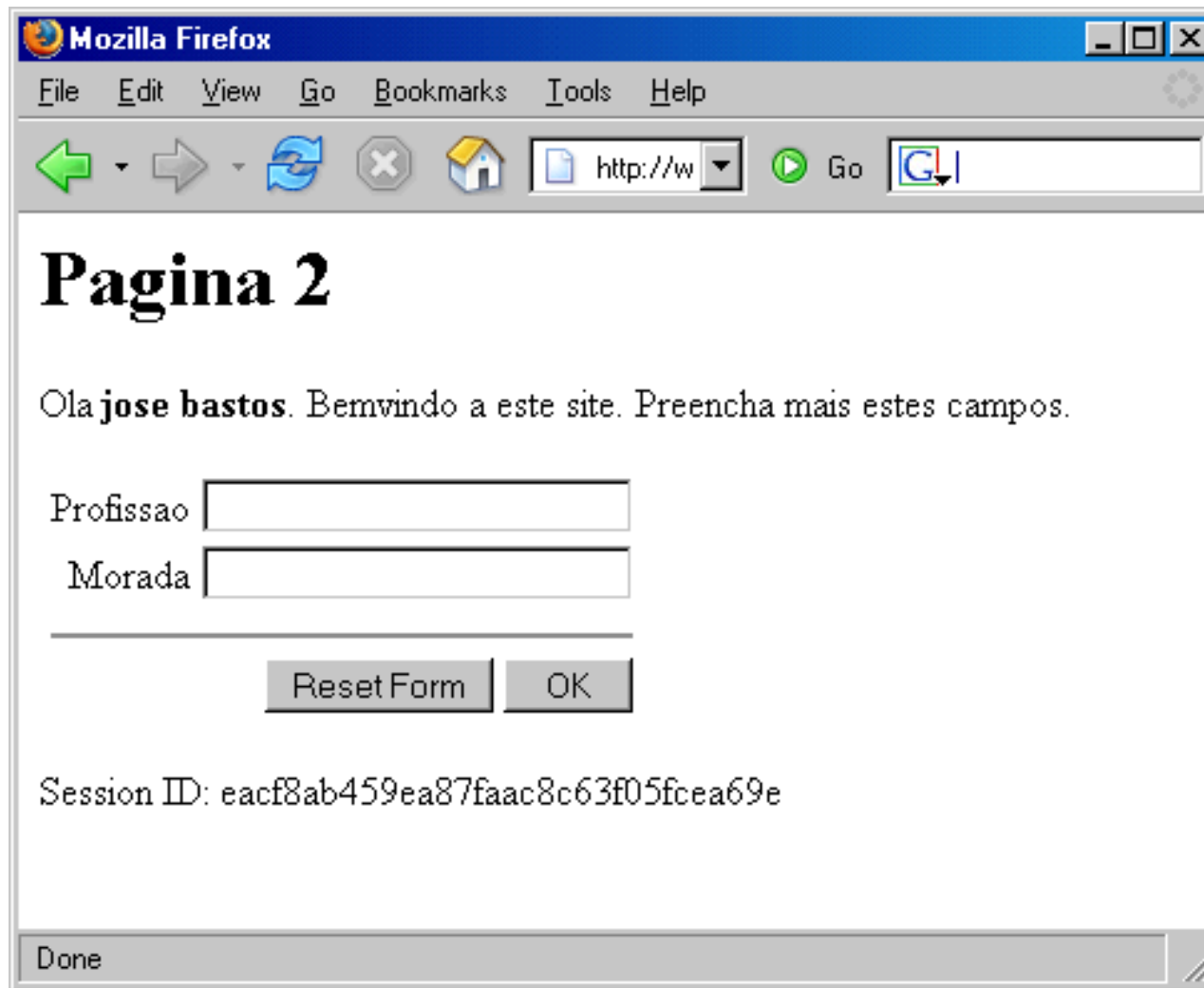
<form method="post" action="page3.php">
<table border="0" cellpadding="0" cellspacing="5">
  <tr>
    <td align="right">Profissao</td>
    <td><input name="profissao" type="text" size="25" /></td>
  </tr>
  <tr>
    <td align="right">Morada</td>
    <td><input name="morada" type="text" size="25" /></td>
  </tr>
  <tr>
    <td align="right" colspan="2">
      <hr noshade="noshade" />
      <input type="reset" value="Reset Form" />
      <input type="submit" name="submitok" value="    OK    "
    />
    </td>
  </tr>
</table>
</form>

```

## código HTTP enviado pelo servidor

```
HTTP/1.0 200 OK
Date: Tue, 24 Apr 2007 16:14:37 GMT
Server: Apache/1.3.33 (Debian GNU/Linux) PHP/5.2.1-0.dotdeb.1 with
Suhosin-Patch mod_ssl/2.8.22 OpenSSL/0.9.7e mod_python/2.7.10
Python/2.3.4
X-Powered-By: PHP/5.2.1-0.dotdeb.1
Set-Cookie: PHPSESSID=eacf8ab459ea87faac8c63f05fcea69e; path=/
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate, post-check=0,
pre-check=0
Pragma: no-cache
Content-Type: text/html; charset=iso-8859-1
X-Cache: MISS from proxy1.si.ualg.pt
X-Cache-Lookup: MISS from proxy1.si.ualg.pt:8080
Proxy-Connection: close
```

(código html)



## registo em vários passos: page3.php

```
<?php
```

```
session_start();
```

```
if($_SESSION['nome']=='' ) {  
    header("Location: page1.html");  
    exit;  
}
```

```
$_SESSION['profissao'] = $_POST['profissao'];  
$_SESSION['morada'] = $_POST['morada'];  
?>
```

```
<h1>Pagina 3</h1>
```

```
<p>
```

```
Ola <b><?=$_SESSION['nome']?></b>.
```

```
Aqui estao os seus dados:
```

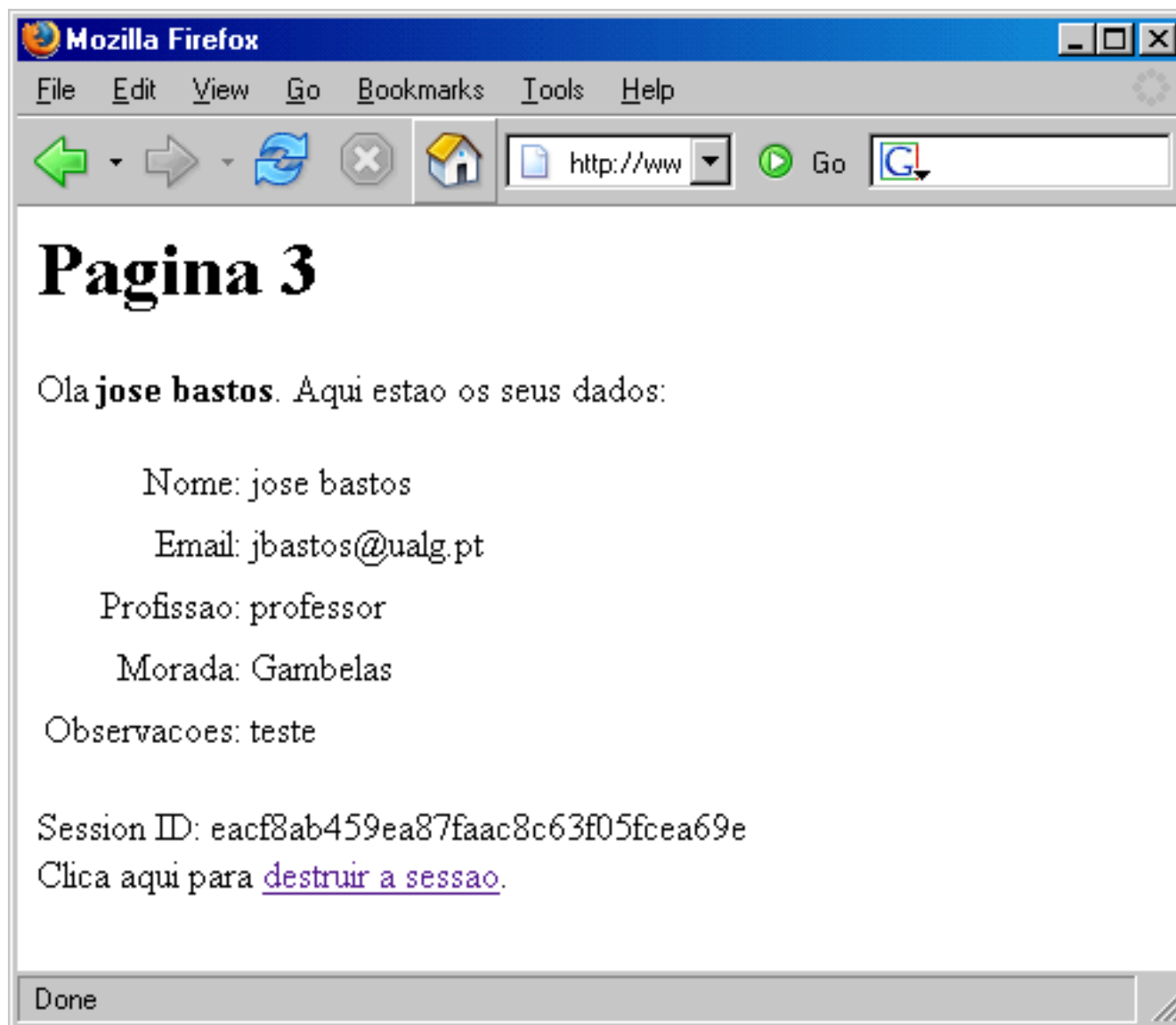
```
</p>
```

```
<table>
```



```
<tr><td align="right">Nome:</td><td><?=$_SESSION['nome']?></td></tr>
<tr><td align="right">Email:</td><td><?=$_SESSION['email']?></td></tr>
<tr><td align="right">Profissao:</td><td><?=$_SESSION['profissao']?></td></tr>
<tr><td align="right">Morada:</td><td><?=$_SESSION['morada']?></td></tr>
<tr><td align="right">Observacoes:</td><td><?=$_SESSION['obs']?></td></tr>
</table>
```

```
<p>
Session ID: <?=session_id()?><br />
Clica aqui para <a href="session-end.php">destruir a sessao</a>.
</p>
```



## Exemplo 2 – validação de dados utilizando sessões: entrada de dados

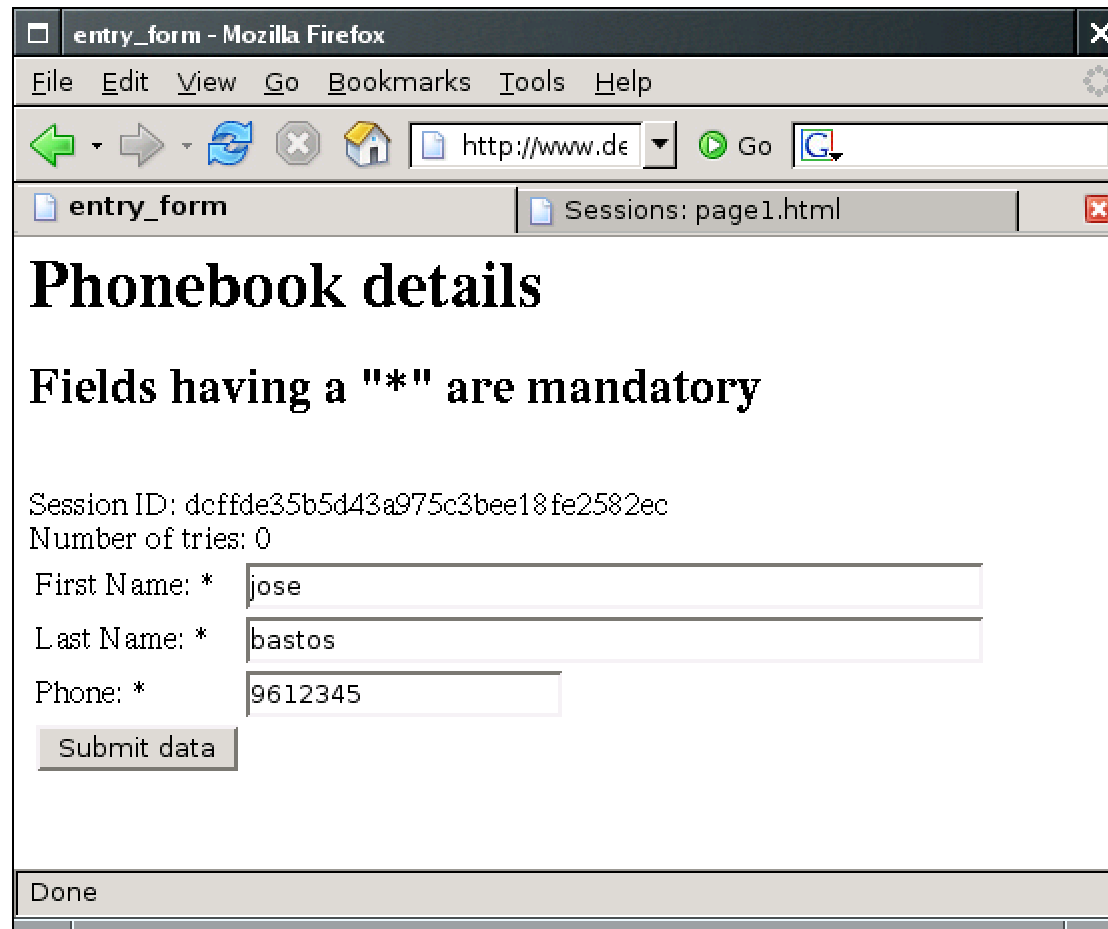
```
<?php
session_start();
?>

<html>
<head>
  <title>entry_form</title>
</head>
<body>
<h1>Phonebook details</h1>
<h2>Fields having a "*" are mandatory</h2>

<?php
printf ("</br>Session ID:  %s \n",session_id());
printf ("</br>Number of tries:  %u \n",$_SESSION['try']);

foreach ($_SESSION['errors'] as $message)
  echo("</br> <font color=\"red\"> $message </font>\n");
?>
```

```
<form method="post" action="input_validation.php">
<table>
  <tr><td>First Name: *</td><td><input type="text"
name="firstname" value="<?=$_SESSION['firstname']
?>"size=50></td></tr>
  <tr><td>Last Name: *</td><td><input type="text" name="lastname"
value="<?=$_SESSION['lastname'] ?>"size=50></td></tr>
  <tr><td>Phone: *</td><td><input type="text" name="phone"
value="<?=$_SESSION['phone'] ?>"size=20></td></tr>
  <tr><td><input type="submit" value="Submit data"> </td></tr>
</table>
</form>
</body>
</html>
```



## validação dos dados

```
<?php
session_start();
$_SESSION['firstname'] = $_POST['firstname'];
$_SESSION['lastname'] = $_POST['lastname'];
$_SESSION['phone'] = $_POST['phone'];
$_SESSION['try']++;

//set-up an empty array to hold errors
$errors=array();

//validate data
if (empty($_POST['firstname']))
    $errors['firstname'] = 'the first name field can not be blank';

if (empty($_POST['lastname']))
    $errors['lastname'] = 'the last name field can not be blank';

if (!ereg('^([0-9]){9}$', $_POST['phone']))
    $errors['phone'] = 'the phone number must have 9 digits';
```

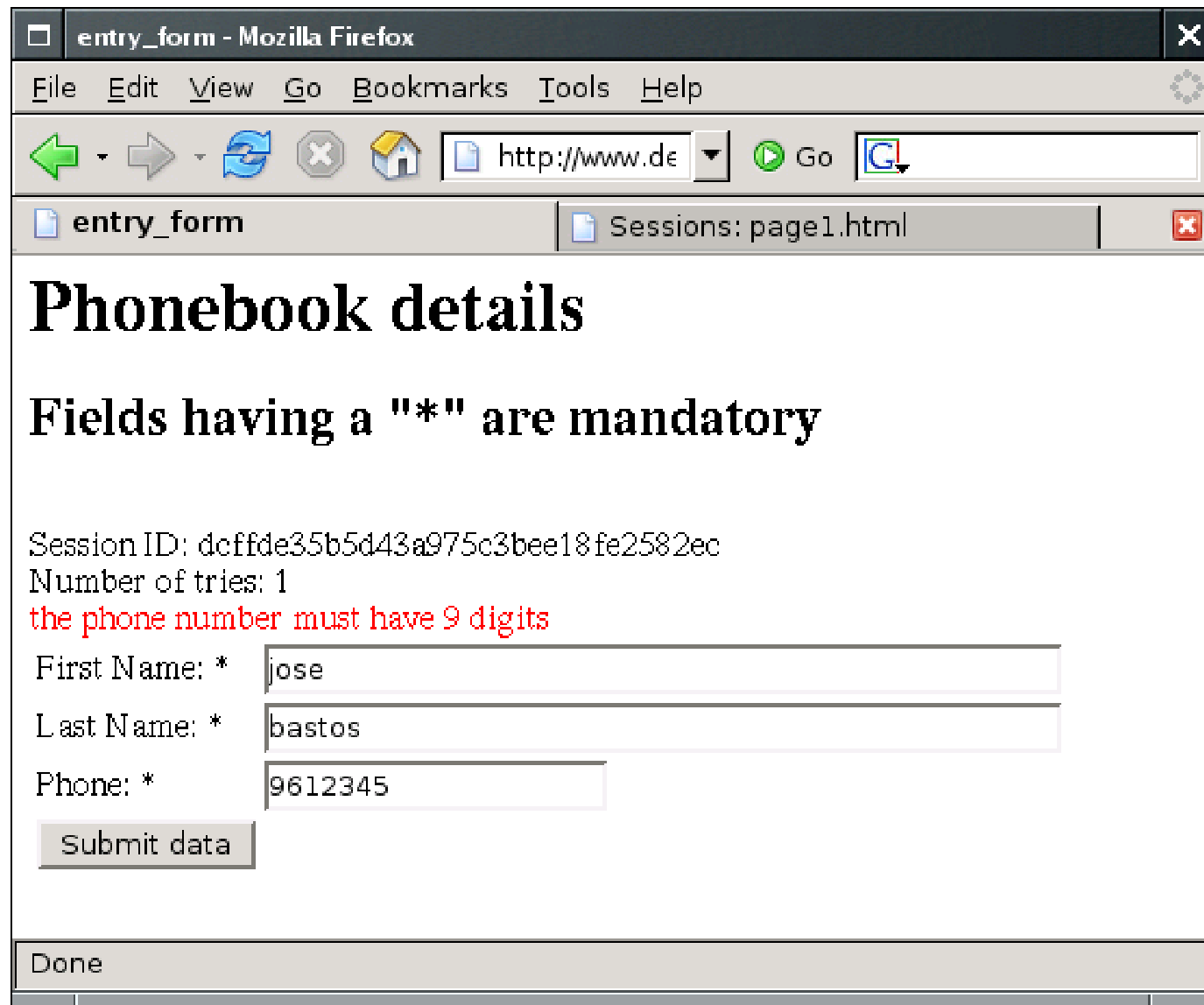
```
//store the errors as a session variable
$_SESSION['errors'] = $errors;

if(!empty($errors)) {
    // there are errors: try again
    header('Location: entry_form.php');
    exit;
}
else {
    //there are no errors: life goes on
    //destroy session
    session_destroy();

    printf("<html>\n");
    printf("<head>\n");
    printf("<title>Success</title>\n");
    printf("</head>\n");
    printf("<body>\n");
    printf("<p>Your data is valid and is ready for further
processing</p>\n");
    printf("<p>A sessao terminou. Clique aqui para <a
href=\"entry_form.php\">recomecar de novo</a>.</p>\n");
}
```

```
printf( "</body>\n" );  
printf( "</html>\n" );  
  
}  
  
?>
```





## Sessoes com PHP com cookies desactivados

- Se o browser desactivar as cookies, tem de se passar PHPSESSID no URL.
- Para ver se o cookie está activo, pode-se verificar o valor de `$COOKIE["PHPSESSID"]`
- Se nao estiver activo, temos de passar o 'session id' no URL.

Exemplo:

```
$url = "page.php?PHPSESSID=" . session_id();
```

## Sessões com PHP com cookies desactivados (cont.)

- Uma outra alternativa consiste em configurar o motor de PHP para incluir "PHPSESSID=" . session id() automaticamente em todos os links.
- Chama-se a isto URL rewriting
- Tem de se editar o ficheiro `php.ini` e colocar `session.use_trans_sid = 1`
- Obviamente que só poderão fazer isto se forem administradores do servidor web...