

# 5

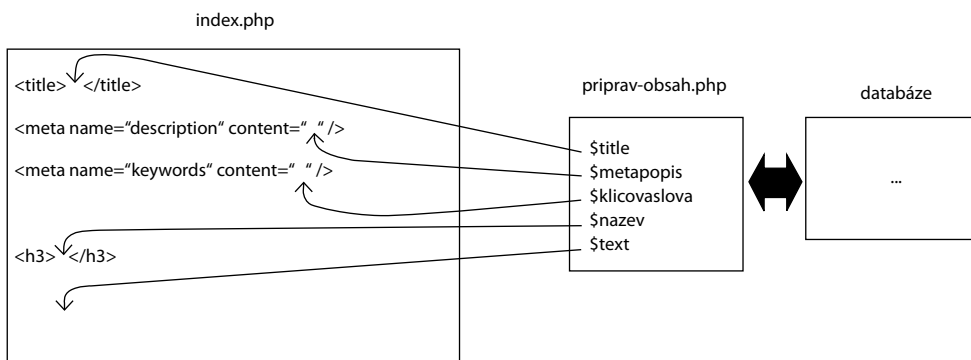
## Kapitola 5 Dynamický obsah načítaný z databáze

Když už víme, jak vytvořit novou databázi a jak v ní vytvořit tabulky pro ukládání informací, můžeme přistoupit k dalšímu kroku. Vytvoříme novou databázi, v které prozatím vytvoříme tabulku, kde budeme uchovávat obsah naší stránky – jednotlivé texty a další informace, jež k nim patří.

K dynamickému zobrazování obsahu na našich stránkách budeme používat následující postup:

1. Hlavním souborem našich stránek bude i nadále soubor *index.php*. Jednotlivé stránky našeho testovacího webu budou mít stále formát *index.php?page=nazev-textu*.
2. Soubor *index.php* zavolá soubor *nacti-obsah.php*.
3. Tento soubor se připojí k databázi a nechá si z ní poslat záznam s požadovaným textem. Který text chce uživatel zobrazit, zjistí z parametru *page* z URL adresy aktuální stránky.
4. Soubor *nacti-obsah.php* připraví celou obsahovou část (texty), která se na stránce bude používat – nadpis stránky (ve značce `<title>`), texty pro metaznačky pro klíčová slova a metapopis a samotný text. Všechny tyto obsahové části stránky se uloží do příslušných proměnných.
5. V souboru *index.php* pak tyto proměnné pouze dosadíme na odpovídající místa a dynamické načítání obsahu z databáze je hotové.

Pro lepší pochopení celého systému fungování si pořádně prohlédneme obrázek 5.1.



**Obrázek 5.1** Schéma fungování dynamického zobrazování obsahu braného z databáze

## Vytvoření databáze a tabulky pro texty

Nejdříve musíme připravit novou databázi, ve které budeme mít všechny tabulky naší testovací dynamické webové aplikace. Zobrazme si prostředí PHPMyAdmin pro správu databází načtením stránky <http://localhost/phpmyadmin>.

Vytvoříme novou databázi se jménem **dynamicka**. V ní postupně vytvoříme všechny další tabulky, které budeme používat.

Vytvořme si novou tabulku se jménem **texty** a s následujícími sloupci:

- **idtextu**: INT, UNSIGNED, AUTO\_INCREMENT,
- **nazev**: VARCHAR(100), utf8\_czech\_ci nebo utf8\_slovak\_ci,
- **urlnazev**: VARCHAR(100), utf8\_czech\_ci nebo utf8\_slovak\_ci, UNIQUE,
- **klicovaslova**: VARCHAR(100), utf8\_czech\_ci nebo utf8\_slovak\_ci,
- **metapopis**: VARCHAR(255), utf8\_czech\_ci nebo utf8\_slovak\_ci,
- **text**: TEXT, utf8\_czech\_ci nebo utf8\_slovak\_ci.



**Poznámka:** U pole **urlnazev** jste použili možnost **UNIQUE**, protože každý text musí mít přiřazen unikátní název, který se bude používat jako atribut page v URL adrese stránky.

Do této tabulky teď vložíme následující záznamy:

**Tabulka 5.1** Nejdříve si musíme připravit tabulku s daty, která budeme brát z databáze

idtextu	nazev	urlnazev	klicovaslova	metapopis	text
1	Úvodní stránka	index	úvod	Úvodní stránka testovacích stránek	Vítejte na naší úvodní stránce!
2	O nás	o-nas	testovací aplikace	Podstránka, kde se dočtete něco o nás	Předěláváme statické stránky na dynamické.
3	Kontakt	kontakt	kontakt	Kontaktujte nás prostřednictvím formuláře, který jsme sami vytvořili	Můžete nás kontaktovat přes tento formulář:

## Skript pro přípravu textů

Tabulku, ze které budeme brát požadované texty, máme připravenou. Teď musíme připravit skript, který se připojí k naší databázi. Použijeme prakticky stejný skript *pripojeni.php* jako v předešlé kapitole. Změníme pouze název databáze, ke které se budeme připojovat, na **dynamicka**.

```
$databaza='dynamicke';
```

Skript *pripojeni.php* najdeme v adresáři *C:\xampp\htdocs\kniha\05\*.

Teď se podívejme přímo na celý skript *priprav-obsah.php* (v adresáři *C:\xampp\htdocs\kniha\05\*), který připraví texty, jež pak jednoduše zobrazíme na stránce v souboru *index.php*:

```
<?php  
require "pripojeni.php";
```

Nejdříve se standardně připojíme k databázi.

```
$php=trim(htmlspecialchars(strip_tags($_GET['page'])));
```

Do proměnné `$php` si uložíme bezpečnostně ošetřený parametr `page` z URL adresy.

```
if($php=="") {  
    $php="index";  
};
```

Pokud má URL adresa stránky tvar *http://localhost/kniha/05/index.php?page=o-nas*, je vše v pořádku a v proměnné `$php` máme uloženu hodnotu, kterou budeme hledat v tabulce `text` ve sloupci **urlnazev**.

Co když se však zobrazí úvodní stránka, která nemá parametr `page` nastaven? Tuto stránku bychom mohli přeměřovat na adresu *http://localhost/kniha/05/index.php?page=index*, ale my použijeme elegantnější řešení. Pokud parametr `page` není nastaven, nastavíme proměnné `$php` hodnotu **index**, která se pak bude hledat v tabulce **texty**.

```
$dotaz='SELECT * FROM texty WHERE urlnazev="'. $php. "'';
```

Z databáze si necháme vybrat všechna data patřící k požadovanému záznamu.

```
$vysledek=mysql_query($dotaz);  
$zaznam=mysql_fetch_array($vysledek);
```

V proměnné `$zaznam` teď máme uloženy všechny informace o požadovaném záznamu.

```
$nazev=$zaznam['nazev'];  
$text=$zaznam['text'];  
$klicovaslova=$zaznam['klicovaslova'];  
$title=$zaznam['nazev'].' | Programování WWW stránek pro začátečníky';  
$metapopis=$zaznam['metapopis'];  
?>
```

Nakonec naplníme všechny potřebné proměnné hodnotami z databáze.

Podívejme se teď na skript *index.php*. Budeme používat pouze jeden skript, protože v případě našich testovacích stránek bude menu i hlavička vypadat vždy stejně, proto nemusí být uloženy v samostatných souborech. Stačí když změnu provedeme v souboru *index.php* a změny se projeví na všech podstránkách webu.

Teď se podíváme na soubor *index.php*. Vycházíme v něm ze souboru *index.php*, který jsme použili v první kapitole. Proto máme prakticky celou stránku v jediném souboru a nevoláme soubory *hlavicka.php*, *menu.php* ani *paticka.php*. Jak uvidíme, při dynamickém načítání obsahu z databází to není potřeba.

```
<?php
session_start();
require "priprav-obsah.php";
?>
```

Nejdříve si připravíme texty, které později použijeme na stránce. Již jsme si řekli, co přesně dělá a jak funguje soubor *priprav-obsah.php*.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
```

Tato část kódu je pro všechny podstránky stejná. Definujeme v ní pouze formální záležitosti jako kódování apod.

```
<title><?php echo $title; ?></title>
```

Do značky pro název stránky vypíšeme hodnotu proměnné `$title`, kterou jsme si připravili již v souboru *priprav-obsah.php*.

```
<style type="text/css" media="all">@import "css/style.css";</style>
<link rel="shortcut icon" href="images/favicon.ico" />
<meta name="robots" content="index, follow" />
<meta name="description" content="<?php echo $metapopis; ?>" />
<meta name="keywords" content="<?php echo $klicovaslova; ?>" />
```

Stejným způsobem naplníme i obsah metaznaček pro klíčová slova a metapopis stránky.

```
</head>
<body>
  <div id="obalStranky">
    <div id="hlavicka">
      <div id="logo">
        <a href="index.php" /></a>
```

Jak vidíme, odkaz z loga stránek odkazuje na stránku *index.php* bez parametru `page`. I proto jsme v souboru *priprav-obsah.php* použili podmínku, která nastavila proměnné `$php` hodnotu **index** v případě, kdy nebyl parametr `page` nastaven.

```
</div>
<div id="nadpisy">
  <h1>Programování WWW stránek pro úplné začátečníky</h1>
  <h2>...a statické stránky se změni na dynamické</h2>
</div>
</div>
<div id="stred">
  <div id="levySloupec">
    <div id="menu">
      <a href="index.php"><div class="menuLink">
        Domů</div></a>
```

```

<a href="index.php?page=o-nas">
  <div class="menuLink">0 nás</div></a>
<a href="index.php?page=kontakt">
  <div class="menuLink">Kontakt</div></a>

```

Menu používáme na každé podstránce našich ukázkových stránek stejné, takže stačí, když případné změny provedeme zde. Jak vidíme, na jednotlivé podstránky (s výjimkou úvodní, hlavní stránky) odkazujeme prostřednictvím URL adresy s parametrem `page`.

```

</div>
</div>
<div id="obsah">
  <h3><?php echo $nazev; ?></h3>

```

Mezi značky `<h3>` a `</h3>` vypíšeme název aktuálního textu.

```

<p><?php echo $text; ?></p>

```

Poté vypíšeme samotný text.

```

<?php
if($_GET['page']=="kontakt"){
  require "kontakt.php";
};
?>

```

Nakonec (pokud jsme na stránce s kontakty) zavoláme skript *kontakt.php*, který již velmi dobře známe z třetí kapitoly.



**Poznámka:** Když chcete, aby byl skript funkční a reálně odesílal e-maily, musíte změnit definice přístupů k e-mailovému účtu, z něhož budete e-maily odesílat. Definice musíte změnit v souboru *kontakt.php* na řádcích 60, 61, 63 a 65. Pokud chcete odesílat e-maily z jiného hostitele, jako je gmail.com, změňte i jeho definici na řádku 59. Pak už bude formulář plně funkční.

```

</div>
</div>
<div id="paticka">
  Ukázka ke knize: <strong>Programování WWW stránek
  pro úplné začátečníky</strong>
</div>
</div>
</body>
</html>

```

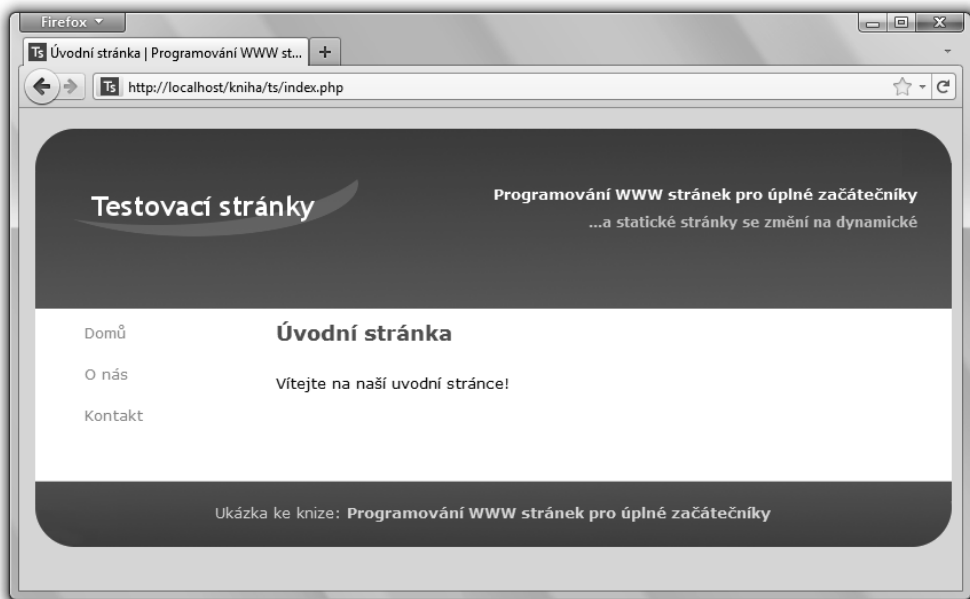
Konec souboru *index.php* obsahuje pouze patičku, která se také nikdy nemění a je pro každou stránku stejná.

## Jak to funguje?

Přejdeme do adresáře `C:\xampp\htdocs\kniha\ts\` a smažeme všechny soubory v tomto adresáři. Neporušené ponecháme pouze adresáře *css* a *obrazky* a jejich obsah.

Teď do tohoto adresáře (*C:\xampp\htdocs\kniha\ts\*) nakopírujme soubory *index.php*, *pripojeni.php*, *priprav-obsah.php*, *kontakt.php*, *class.phpmailer.php* a *class.smtp.php* z adresáře *C:\xampp\htdocs\kniha\05\*. Díky těmto krokům se naše testovací stránky dostanou na novou úroveň.

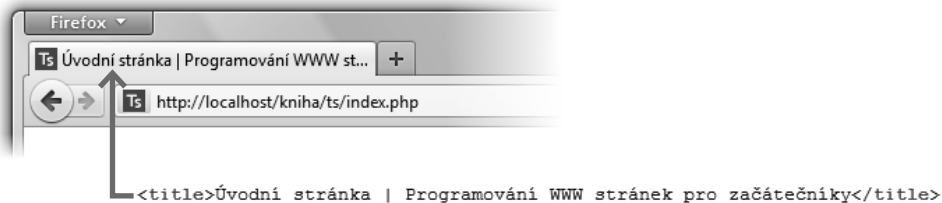
Nyní si zobrazíme úvodní stránku našeho testovacího webu – *http://localhost/kniha/ts/*. Jak vidíme na obrázku 5.2, zobrazila se nám už dobře známá stránka. Obsah však již bereme z databáze.



Obrázek 5.2 Úvodní stránka testovacího webu – texty bereme z databáze

Podívejme se však na pár detailů, které jsme v této kapitole zakomponovali do kódu:

- Titulek stránky je vyplněn. Má formát **Název aktuální stránky | Programování WWW stránek pro začátečníky**. Titulek se vyplňuje automaticky a pro každou stránku je unikátní.



Obrázek 5.3 Automaticky vyplněný unikátní titulek pro každou stránku

- Když si zobrazíme zdrojový kód, uvidíme, že máme vyplněny i značky s klíčovými slovy a metapopisem stránky.

```
<meta name="description" content="Úvodní stránka testovacích stránek" />
<meta name="keywords" content="úvod" />
```

**Obrázek 5.4** Automaticky vyplníme i značky pro klíčová slova a metapopis

- Nadpis hlavního textu a samotný text je také vypsán automaticky.

Proklepejme i zbylé dvě podstránky našeho testovacího webu a všimněme si, že všechny výše uvedené texty se mění. Je to velmi vhodné a nyní si řekneme proč.

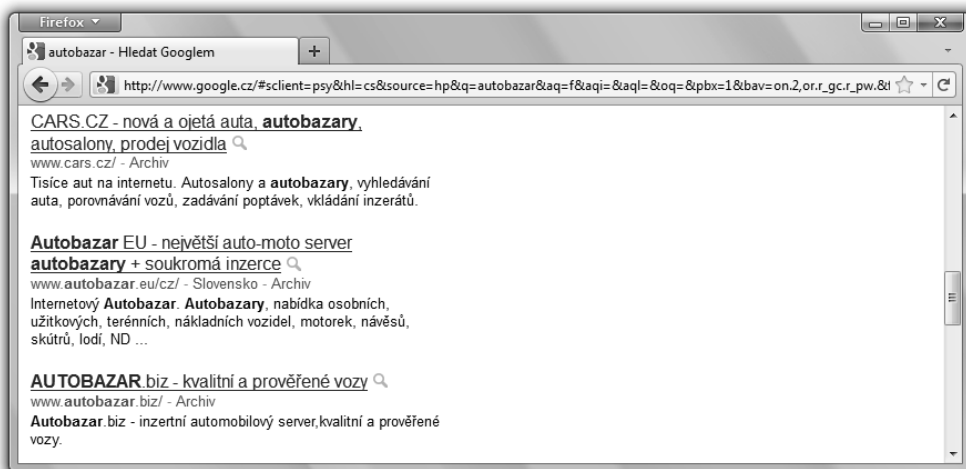
Můžete také otestovat funkčnost kontaktního formuláře – nezapomeňte ale upravit řádky 60, 61, 63 a 65 pro přihlášení k vaší e-mailové adrese (pokud je na Gmailu; jinak změňte i definici hostitele na řádku 59).

## Proč vyplňovat značky unikátními texty

Teď se soustředíme na značku `<title></title>` a značku pro metapopis stránky. Představme si, že jako uživatel hledáme nějakou informaci přes vyhledávač. Nezáleží na tom, jestli máme na mysli Google, Seznam nebo jiný – všechny totiž mají něco společného.

Když v nich chceme hledat nějakou informaci, zadáme jim dotaz – např. **jak vyrobit zmrzlinu, autobazary, last minute dovolená** apod. Tyto vyhledávače pak prohledají své databáze (s informacemi o milionech stránek) a podle svých algoritmů seřadí výsledky tak, abychom my, jako uživatelé, dostali na náš dotaz co nejrelevantnější odpovědi (nebo odkazy na stránky, kde tyto odpovědi nalezneme).

A nyní se dostáváme k otázce z nadpisu této podkapitoly. Jak máme docílit toho, aby uživatel klepl právě na odkaz na naši stránku? Musíme ho něčím zaujmout a k tomu se velmi hodí právě nadpis našich stránek – tedy text, který je mezi značkami `<title>` a `</title>`. Proto ho generujeme na našich testovacích stránkách dynamicky a právě proto do něj vkládáme název textu z každé konkrétní podstránky.



**Obrázek 5.5** Text ze značky `<title>` je první, čeho si uživatelé při vyhledávání všimnou, proto musí být originální, zajímavý a musí uživatele informovat o tom, co na dané stránce naleznou

Ve výsledcích vyhledávání se obvykle zobrazuje také krátký text z dané stránky. Záleží už na vyhledávači, jestli zvolí část samotného textu ze stránky, nebo použije text z metaznačky pro popis stránky, který píšeme my. Je však vhodné napsat tento text tak, aby obsahoval klíčová slova, která jsou relevantní k textu na dané stránce. Zvýšíme tak pravděpodobnost, že vyhledávač použije právě text ze značky pro metapopis.

Čistý kód | knihy - Computer Press 🔍 - [ Preložit tuto stránku ]  
knihy.cpress.cz/cisty-kod.html - V paměti  
14. září 2009 – Poznáte špatný kód od dobrého? Naučte se tvořit správný a srozumitelný kód nejen pro efektivní týmovou spolupráci.

↑

```
<meta name="description" content="Poznáte špatný kód od dobrého? Naučte se tvořit správný a srozumitelný kód nejen pro efektivní týmovou spolupráci."
```

**Obrázek 5.6** Vyhledávač pro danou stránku použil text z její značky s metapopisem

Když se nám podaří napsat přitažlivý a unikátní nadpis ve značce `<title>` a k tomu přidat zajímavý metapopis, můžeme zaujmout více lidí, kteří hledají to, čemu se naše stránky věnují, a zvýšit tak návštěvnost našich stránek.

V našem případě vyplníme také značku pro klíčová slova. Měla význam v minulosti, kdy vyhledávače hledaly a řadily výsledky právě podle slov, která tato značka obsahovala. Dnes už pro vyhledávače velký význam nemá, protože jsou natolik chytré, že umí zjistit samy, o čem daná stránka je. Nic však nepokazíme, pokud značku vyplníme.

**Důležité:** I zde však platí – všeho s mírou. Nesnažte se do značek pro nadpis stránky, metapopis nebo klíčová slova vtěsnat příliš mnoho klíčových slov. Vyhledávače to mohou považovat za spamování a docílíte tak přesně opačného efektu, než jste chtěli.

**Poznámka:** Psaní textů pro internetové stránky a zlepšení návštěvnosti webových stránek jsou velmi obšírná témata, a proto se zde jimi nebudeme detailněji zabývat. Pokud o se o nich chcete dozvědět více, existuje mnoho knih na dané téma; další informace naleznete také na Internetu.

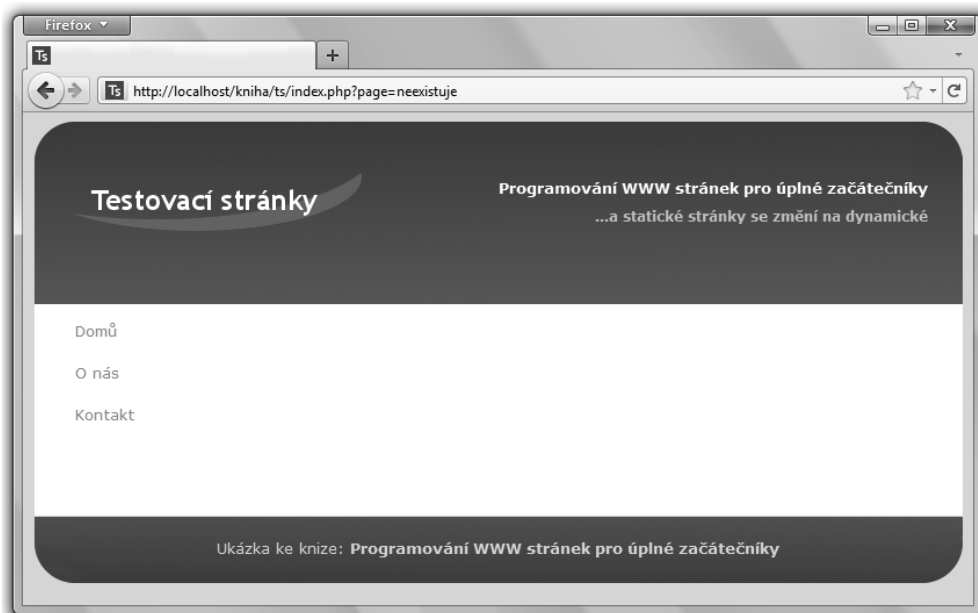
## Co když požadovaná stránka neexistuje?

Vraťme se zpět k naší testovací stránce. Zkusme si zobrazit stránku, která neexistuje – třeba `http://localhost/kniha/ts/index.php?page=neexistuje`. Uvidíme to, co na obrázku 5.7.

Uvidíme pouze prázdnou šablonu stránky – to, co je napevno dáno v souboru `index.php`. Chybí však nadpis stránky, název textu i samotný text (což však není překvapivé, protože požadovaný text s URL názvem **neexistuje** v databázi nemáme).

Takové problémy se obvykle řeší přesměrováním na nám již známou stránku **Error 404**. My si ukážeme pouze řešení, kdy se uživatel pokouší zobrazit stránku s neexistujícím parametrem `page` v URL adrese stránky.





**Obrázek 5.7** Pokud stránka neexistuje, zobrazí se pouze prázdná šablona

Nejdříve si do databáze vložíme nový záznam, ve kterém budou texty právě pro tuto chybovou stránku.

V prostředí PHPMyAdmin na adrese <http://localhost/phpmyadmin> klepneme na název databáze **dynamicka** a na tabulku **texty**. Dále klepneme na záložku **Vložit** a přidáme následující záznam:

idtextu	nazev	urlnazev	klicovaslova	metapopis	text
4	Error 404	error404	neexistuje	Požadovaná stránka neexistuje.	Litujeme, ale požadovaná stránka neexistuje.

Na tuto stránku přesměrujeme všechny neexistující stránky, které se uživatelé pokusí zobrazit. Přesměrování vykonáme (pokud to bude potřeba) v souboru *priprav-obsah.php*. Otevřeme si tento soubor (verzi v adresáři `C:\xampp\htdocs\kniha\ts\`) a na jeho konec (ještě před řádek s `??`) dopíšeme:

```
if($nazev=="") { echo '<html><meta http-equiv="REFRESH" content="0;index.php?page=error404"></html>'; die(); };
```

Nejdříve zkontrolujeme, zda má proměnná `$nazev` nastavenou nějakou hodnotu. Pokud ano, je vše v pořádku (požadovaný text v databázi je) a můžeme pokračovat v běhu skriptu *index.php*.

Pokud však tato proměnná nemá nastavenou žádnou hodnotu, text v databázi neexistuje (vyloučíme možnost, že v databázi máme text, který nemá zvolen nadpis). Proto vypíšeme velmi jednoduchý HTML kód, který nás přesměruje na stránku `index.php?page=error404`. Jelikož nemá smysl pokračovat v dalším běhu skriptu, použijeme také funkci `die()`, která jeho běh ukončí.

Podívejme se, jak naše přesměrování funguje. Pokusíme se zobrazit např. stránku `http://localhost/kniha/ts/index.php?page=123`. Hned po stisku klávesy Enter jsme přesměrováni na stránku `http://localhost/kniha/ts/index.php?page=error404`. To stejné se stane při pokusu o zobrazení jakékoliv neexistující stránky.



**Obrázek 5.8** Při pokusu o zobrazení neexistující stránky jsme přesměrováni na stránku s chybovou hláškou Error 404



**Poznámka:** Přesměrování se dá udělat mnoha různými způsoby. Jedním z nich je třeba použití souboru `.htaccess` nebo přesměrování přímo pomocí PHP a změnou hlavičky.

Pokud se vám nepodařilo „rozchodit“ testovací stránky do podoby, v jaké by teď měly být, můžete obsah adresáře `C:\xampp\htdocs\ts\` nahradit obsahem adresáře `C:\xampp\htdocs\kniha\05\ts\`. Před přechodem na další kapitolu tak budeme opět na stejné „startovní“ pozici.