

Kapitola 5

Tvorba vlastního vzhledu

Proč se vůbec zabývat vytvořením vlastního motivu vzhledu, když pro Drupal existuje celá řada vynikajících témat, která jejich autoři uvolnili k volnému použití – jednoduše si je stáhnete z drupal.org, nainstalujete a ihned můžete začít používat? Odpověď je snadná – nikdo nechce mít web, který lze při pohledu snadno zaměnit s nějakým jiným. U mnoha blogů se samozřejmě setkáte s tím, že používají standardní motiv s mírně upravenými barvami pomocí administračního rozhraní. Ale vytvořit si úplně vlastní vzhled webu asi časem napadne každého.

Začít lze s úpravou obrázků, které jsou součástí většiny témat. Pokud se budete držet rozměrů předloh, můžete obrázky jednoduše zaměnit při zachování stejného názvu souboru. Postupně můžete začít s úpravami barev v kaskádových stylech, změnami rozměrů jednotlivých sloupců, rozmístěním atd.

Součástí práce na novém vzhledu webu je také úprava bloků, jejich rozmístění, změna patičky a další. To vše můžete jednoduše dělat přes administrační rozhraní Drupalu, aniž byste museli cokoli programovat nebo měnit v šabloně vzhledu. Od verze 5 vám pomáhá také barevná změna výchozích témat prostřednictvím administračního rozhraní. K oživení webu přispěje anketa, do patičky můžete umístit malý obrázek, pohrajte si s bloky. Kromě výchozích bloků s žebříčky oblíbených článků, přihlašovacím formulářem nebo agregátorem RSS máte možnost vytvořit bloky s vlastním obsahem.



The image shows a screenshot of the official website of Britney Spears. At the top left, the 'Blackout' album logo is displayed with the text 'It's Britney.com, b*tch!'. To the right, there is a banner for 'Happy Holidays From G1FT-BOT-2007' featuring a robot and the text 'See the holiday collection.' and the Puma logo. Below this is a navigation menu with links: HOME, NEWS + BLOG, MUSIC, PHOTOS, VIDEOS, EVENTS, EXTRAS, MOBILE, BIO, FORUMS, HELP. A search bar is located on the right side of the menu. Below the menu, there is a large banner for 'GIVE US MORE!' with the text 'Enter the wallpaper DESIGN CONTEST NOW!' and an image of Britney Spears wearing a hat. To the right of this banner, there are two smaller promotional boxes. The top one is for 'BUY THIS NOW!' with links to iTunes, Amazon.com, and Sony Music Store, and a 'LISTEN NOW' button. The bottom one is for the Puma holiday collection, featuring a robot and the text 'See the holiday collection.' and the Puma logo.

Obrázek 5.1: Drupal používá také oficiální web Britney Spears

Využijte toho pro vkládání reklamy, obrázků nebo krátkých kousků kódu v PHP, které vám zajistí například provázanost s jinými servery.

Při tvorbě svého prvního motivu vzhledu vycházejte s již existující šablony. Na drupal.org si najdete takovou, která bude nejvíce odpovídat vašim představám. Chcete-li například dvousloupcový layout, nestahujte si šablonu, která má vedle hlavního obsahu sloupce po obou stranách. Pozornost věnujte grafické úpravě navigační nabídky. Zkušenost praví, že pro méně zkušené uživatele je tvorba efektivně zaoblených tlačítek v navigaci problém, takže se vyplatí použít téma, které tyto prvky používá, a následně pak v grafickém editoru pouze přebarvit obrázky tvořící pozadí tlačítek na požadovaný barevný odstín.

Zkušení uživatelé začínají s tvorbou témat vzhledu od úplného začátku. Praktickou ukázkou toho, co může při dostatečné kreativitě grafika, vhodné volbě modulů a při jejich vhodném uspořádání vzniknout, je web *The New York Observer* (www.observer.com).



Obrázek 5.2: Zajímavě upravený vzhled Drupalu

Témata vzhledu v Drupalu

Každé téma vzhledu v redakčním systému Drupal je složeno z několika souborů PHP, ve kterých je naznačena kostra stránky. V současné době Drupal využívá systém témat *PHPTemplate*. Ten definuje pět základních regionů, ve kterých se zobrazuje obsah – levý sloupec, pravý sloupec, obsah, hlavička, patička. Tyto regiony můžete vidět například v administraci bloků, kde tak snadno poznáte, do které části webu blok umísťujete.

PHPTemplate má tu výhodu, že veškerý obsah generovaný Drupalem je v hlavní kostře uzavřen do pěti základních proměnných. Jakýkoli další obsah můžete přidat sami. Může to být jiný kód v PHP, část (X)HTML, JavaScript, kaskádový styl... Na druhou stranu to neznamená, že vypsáním obsahu pěti proměnných do

hlavní kostry stránky končí vaše možnosti, jak jednotlivé části webu ovlivnit. Naopak. Projděte si strukturu souborů *TPL.PHP*, které tvoří téma, a uvidíte, jak bohaté možnosti úprav se vám otevírají.

Struktura stránky (page.tpl.php)

Základem každého tématu redakčního systému Drupal je soubor *page.tpl.php*. Ten je spolu s dalšími uložen ve složce s názvem tématu umístěné v */themes*. V tomto souboru najdete směs kódu (X)HTML, odkazů na kaskádové styly, JavaScript a hlavně části PHP, které zajistí vypsání kódu Drupalu do kostry (X)HTML. Rozebereme si tento soubor po jednotlivých částech. Pro ukázkou jsem zvolil šablonu Garland nastavenou jako výchozí téma v Drupalu.

```

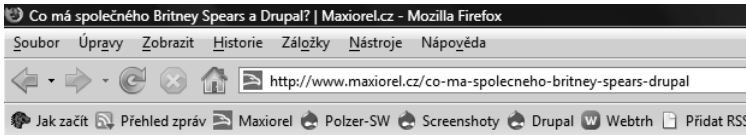
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="<?php print $language ??"
  lang="<?php print $language ??">
  <head>
    <title><?php print $head_title ?></title>
    <?php print $head ?>
    <?php print $styles ?>
    <?php print $scripts ?>
    <style type="text/css" media="print">@import "<?php print base_path()
      . path_to_theme() ?>/print.css";</style>
    <!--[if lt IE 7]>
    <style type="text/css" media="all">@import "<?php print base_path()
      . path_to_theme() ?>/fix-ie.css";</style>
    <![endif]-->
  </head>
  <body<?php print phptemplate_body_class($sidebar_left, $sidebar_right); ?>>

```

Jelikož je soubor *page.tpl.php* základní kostrou stránky, musí obsahovat všechny náležitosti kódu (X)HTML, který bude stránku tvořit. Základem každého tématu by měla být deklarace typu použitého dokumentu. V případě tématu Garland je to poměrně přísný XHTML 1.0 Strict. Následuje začátek struktury HTML, ve kterém se poprvé setkáváte s výpisem nějaké proměnné. Kód PHP je zde běžným způsobem vkládán pomocí značek *<?php a ?>* do XHTML. Jako první můžete vidět vypsání jazykové verze dokumentu, která odpovídá jazyku webu, který jste nastavili v jeho administračním rozhraní. Podobným způsobem je zajištěno vypsání názvu stránky do titulku pomocí funkce *print \$head_title*. Titulek je u webů s Drupelem tvořen názvem stránky svislítkem a názvem webu, například *Váš web a pozice ve vyhledávačích/Maxiorel.cz*.

Jak vidíte na obrázku, následuje trojice příkazů pro vypsání metaznaček v hlavičce, odkazů na kaskádové styly používané jádrem Drupalu a jeho doplňkovými moduly a pro vypsání odkazů na použité soubory se skripty. Ihned poté následuje vypsání odkazu na kaskádový styl určený pro tiskovou verzi. Vzhledem ke konstrukci si tento příkaz zaslouží bližší vysvětlení.

Část *<style type="text/css" media="print">@import* „, je běžným prvkem v XHTML, pomocí kterého importujete soubor s kaskádovými styly. Následující kód v jazyku PHP *<?php print base_path() . path_to_theme() ?>* vloží do stránky základní adresu webu a název složky s aktuálním tématem vzhledu. Pomocí */print.css">* celý výpis dokončíte. Podobně je vložena cesta ke kaskádovému stylu pro Internet Explorer. Méně zkušeného webdesignéra možná překvapí konstrukce *<!--[if lt IE 7]>* a *<![endif]-->*. Vězte, že nejde o specialitu Drupalu, ale o běžnou formulaci požadavku, aby se kód mezi těmito dvěma značkami aplikoval pouze v Internet Exploreru 7. Ostatní prohlížeče jej ignorují, neboť je jakoby v poznámce uvozené a zakončené pomocí *<!-- a -->*.



Obrázek 5.3: Titulek webu v prohlížeči

Hlavní soubor s kaskádovými styly, který je určen pro zobrazování stránky na monitoru počítače, je připojován automaticky pomocí `<?php print $styles ?>` za předpokladu, že je nazván `style.css` a leží ve složce s tématem. Poslední položkou ve výpisu hlavičky webu je značka `<body>`, do které je přidána třída z kaskádových stylů. V tomto případě autor tématu využil pokročilé techniky s definicí vlastní funkce, u vašich témat postačí, když uvedete prostě `<body>`.

Hlavička stránky v prohlížeči

Jak vypadá kód hlavičky poté, co je návštěvníkovi stránky odeslán do prohlížeče, si prohlédněte na následujícím výpisu. V titulku není žádné svislítko, jelikož jde o kód z úvodní stránky webu. V metainformacích můžete vidět informaci o použitém kódování, dva odkazy směřují na zdroj RSS a ikonku webu, následuje sada importovaných kaskádových stylů.

```
<!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" xml:lang="en" lang="en">
<head>
  <title>Drupal</title>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
  <link rel="alternate" type="application/rss+xml" title="Drupal RSS"
    href="http://localhost/drupal6/rss.xml" />
  <link rel="shortcut icon" href="/drupal6/misc/favicon.ico" type="image/x-icon" />
  <style type="text/css" media="all">@import "/drupal6/modules/node/node.css";</style>
  <style type="text/css" media="all">@import
    "/drupal6/modules/system/defaults.css";</style>
  <style type="text/css" media="all">@import
    "/drupal6/modules/system/system.css";</style>
  <style type="text/css" media="all">@import
    "/drupal6/modules/user/user.css";</style>
  <style type="text/css" media="all">@import
    "/drupal6/themes/garland/style.css";</style>
  <style type="text/css" media="print">@import
    "/drupal6/themes/garland/print.css";</style>
  <!--[if lt IE 7]>
  <style type="text/css" media="all">@import "/drupal6/themes/garland/fix-ie.css";</style>
  <![endif]-->
</head>
<body class="sidebar-left">
```

Hlavička webu

Jestliže předchozí část má většina témat vzhledu společnou, další část kostry souboru `page.tpl.php` již z velké části záleží na konkrétním motivu. V další ukázce kódu si všimněte vypsání hlavičky webu. Pomocí sady proměnných se vykreslí pozadí, zobrazí se slogan, logo stránek a titulek. Tedy vše, co máte možnost zadat v administrační části webu v položce **Administrace** → **Nastavení** → **Informace o webu**.

```

<!-- Layout -->
<div id="header-region" class="clear-block"><?php print $header; ?></div>
  <div id="wrapper">
    <div id="container" class="clear-block">
      <div id="header">
        <div id="logo-floater">
          <?php
            // Prepare header
            $site_fields = array();
            if ($site_name) {
              $site_fields[] = check_plain($site_name);
            }
            if ($site_slogan) {
              $site_fields[] = check_plain($site_slogan);
            }
            $site_title = implode(' ', $site_fields);
            $site_fields[0] = '<span>'. $site_fields[0] . '</span>';
            $site_html = implode(' ', $site_fields);
            if ($logo || $site_title) {
              print '<h1><a href="'. check_url($base_path) .'>' title="'. $site_title .'>';
              if ($logo) {
                print '' id="logo" />';
              }
              print $site_html . '</a></h1>';
            }
          ?>
        </div>

```

Rozmístění jednotlivých proměnných, tedy místo, kde je na stránce necháte zobrazit, záleží na vašich potřebách. Zajímat vás bude především význam jednotlivých položek z příkladu. Vybral jsem ty, které pro vás budou významově nejdůležitější.

- ◆ Proměnná *\$site_name* obsahuje informaci o tom, zda je nastaven název webu.
- ◆ Podobně proměnná *\$site_slogan* vám umožní zjistit, zda bylo v administraci vyplněno políčko se sloganem webu.
- ◆ Pozornost věnujte podmínce *if (\$logo || \$site_title)*, ve které autor tématu zjišťuje, zda existuje cesta k obrázku s logem a je nastaven titulek webu. Pokud tomu tak je, dojde k vypsání hlavičky *H1* obsahující název webu (proměnné *\$site_title* a *\$site_html*), který zároveň slouží jako hypertextový odkaz na úvodní stránku. Ten je zajišťován značkou HTML *<a href=""*, do které je odkaz vypsán pomocí kontrolovaného obsahu proměnné *check_url(\$base_path)*. Jestliže existuje cesta k obrázku s logem, je tento obrázek následně vytvořen.

Zobrazení navigace

Hlavička webu je zakončena vypsáním položek primárního a sekundárního menu. Navigaci lze samozřejmě vložit přímo ve formě odkazů HTML, avšak důrazně to nedoporučuji. Správa odkazů v navigaci prostřednictvím administrace je jednodušší a nebudete muset při případných změnách opět upravovat kód šablony.

V ukázce vidíte, že je nejprve ověřeno, zda položky v menu existují, resp. zda je navigace nastavena. Následuje speciální konstrukce, pomocí které jsou z pole vypsány všechny odkazy konkrétního menu, a je jim přiřazena třída kaskádového stylu.

```

<?php if (isset($primary_links)) : ?>
    <?php print theme('links', $primary_links, array('class' =>
        'links primary-links')) ?>
<?php endif; ?>
<?php if (isset($secondary_links)) : ?>
    <?php print theme('links', $secondary_links, array('class' =>
        'links secondary-links')) ?>
<?php endif; ?>
</div> <!-- /header -->

```

Práce s položkami menu však může vypadat i naprosto odlišně, přičemž se dosáhne stejného výsledku:

```

<?php
print '<ul>';
foreach ($primary_links as $link) {
    print '<li>'. l($link['title'], $link['href'],
        $link['attributes'], $link['query'],
        $link['fragment'], FALSE, $link['html']) .'</li>';
}
print '</ul>';
?>

```

Menu je uzavřeno v nečíslovaném seznamu ``, pro každou položku z menu je vytvořena značka ``, do které je vložen odkaz s popisem, adresou URL a názvem. Vše probíhá pomocí smyčky `foreach`. Jak vidíte na tomto příkladu, lze velmi jednoduše přistupovat nejenom k jednotlivým položkám menu, ale také k vlastnostem těchto položek. To je ostatně pro Drupal typické. Můžete použít buď jednoduchou cestu, nebo složitější, kde však máte možnost ovlivnit chování částí webu do detailu.

Zobrazení sloupce

Po vypsání menu následuje v tématu Garland zobrazení levého sloupce. Opět byste měli nejprve ověřit, zda levý sloupec existuje, tj. zda obsahuje nějaké položky. Příkazem `print $sidebar_left` jej pak na stránku vypíšete. Autor tématu Garland přidal značky `div`, které mu ve spojení s kaskádovými styly pomohou levý sloupec na stránce správně umístit. Také přidal podmínku pro zobrazení vyhledávacího políčka. Pokud je aktivováno v nastavení tématu, zobrazí se ještě před samotným obsahem levého sloupce.

```

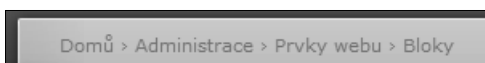
<?php if ($sidebar_left): ?>
    <div id="sidebar-left" class="sidebar">
        <?php if ($search_box): ?><div class="block block-theme"><?php
            print $search_box ?></div><?php endif; ?>
        <?php print $sidebar_left ?>
    </div>
<?php endif; ?>

```

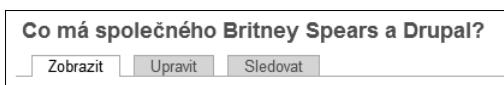
Hlavní část stránky

Vytváření hlavní části webu vidíte na následujícím výpisu. Autor tématu použil několik značek `div` pro rozmístění obsahu velkého středního sloupce a grafické zpracování jeho okrajů. Jak se jednotlivé části webu vypisují do konstrukce HTML jste již jistě pochopili. Dále tedy najdete stručný popis toho, co která proměnná znamená a obsahuje:

- ◆ *\$breadcrumb* slouží k zobrazení tzv. drobečkové navigace. Návštěvník, popřípadě administrátor stránky díky této navigaci ihned vidí, jak hluboko je ve stránce zanořen a případně ji může použít k přesunu o jednu nebo více úrovní nahoru. Drobečková navigace může vypadat například takto: *Administrace > Logy > Nejnovější záznamy v logu*.
- ◆ *\$mission* lze použít k zobrazení tzv. cíle webu. Obsah této proměnné nastavujete ve stejnojmenném políčku v administrační části Informace o webu. V praxi se setkáte většinou s tím, že tato funkce není provozovateli stránek využívána. Pěknou ukázkou jejího použití je záhlaví webu www.drupal.cz s vysvětlením toho, co Drupal vlastně je.
- ◆ *\$tabs* jsou záložky. V běžném článku záložky zřejmě jako návštěvník webu nevidíte, nicméně stačí, abyste se přepnuli do režimu správce, a záložky se zobrazí. Slouží například k zobrazení odkazů pro úpravy článků, přepínání informací v modulech a v nastavení témat a podobně. Ačkoli to na první pohled vypadá, že se je autor tématu Garland pokouší vypsát dvakrát, není tomu tak. V prvním případě jde pouze o přípravu elementu div.



Obrázek 5.4: Ukázka drobečkové navigace



Obrázek 5.5: Záložky v administraci článku

\$title obsahuje název stránky. V tomto případě nejde o nadpis článku. Zde tato proměnná obsahuje například název výpisu kategorií, název položky v administraci atd.



Tip: Řada standardních i doplňkových modulů v Drupalu obsahuje nápovědu k ovládní, která se uživateli zobrazí na místě vypsání obsahu proměnné *\$help*. Nepodceňujte její důležitost. Mnohdy obsahuje cenné informace k nastavení modulů, které jinde nenajdete.

Je-li povoleno zobrazování hlášení na obrazovku, ověříte v proměnné *\$show_messages*. Obsah samotných hlášení je pak dostupný pomocí proměnné *\$messages*. V těchto hlášeních se nachází případné informace o chybách ve skriptech PHP, ale také třeba informace o tom, že návštěvníkovi stránky bylo odesláno potvrzení registrace. Pamatujte na to, až budete tvořit nové téma vzhledu. Tato informace by neměla být přehlédnutelná. Návštěvníkovi je potřeba jasně sdělit, že registrace proběhla. Malý nenápadný textík zobrazovaný většinou témat vzhledu není to pravé řešení. V proměnné *\$messages* se mimo jiné nachází informace o tom, že vkládaný článek byl uložen, smazán nebo upraven.

Proměnná *\$content* obsahuje hlavní obsah webu. Jeho vzhled lze měnit pomocí dalších souborů šablon, jak se dozvíte v další kapitole. Obsahem je výpis článků, tělo konkrétního článku, galerie obrázků, zkrátka to, co čtenáře na webu nejvíce zajímá.



Tip: Přitáhněte další čtenáře tím, že obsah poskytnete ve formě RSS. Výpisem proměnné *\$feed_icons* zajistíte zobrazení odkazu RSS na příslušnou kategorii článků.

Proměnná *\$footer_message* obsahuje obsah patičky stránky a uzavírá výpis hlavní části obsahu.

```

<div id="center"><div id="squeeze"><div class="right-corner"><div class="left-corner">
  <?php if ($breadcrumb): print $breadcrumb; endif; ?>
  <?php if ($mission): print '<div id="mission">'. $mission .'</div>'; endif; ?>
  <?php if ($tabs): print '<div id="tabs-wrapper" class="clear-block">'; endif; ?>
  <?php if ($title): print '<h2'. ($tabs ? ' class="with-tabs" : '') .'>'. $title
    .'</h2>'; endif; ?>
  <?php if ($tabs): print $tabs .'</div>'; endif; ?>
  <?php if (isset($tabs2)): print $tabs2; endif; ?>
  <?php if ($help): print $help; endif; ?>
  <?php if ($show_messages && $messages): print $messages; endif; ?>
  <?php print $content ?>
  <span class="clear"></span>
  <?php print $feed_icons ?>
  <div id="footer"><?php print $footer_message ?></div>
</div></div></div></div> <!-- /.left-corner, /.right-corner, /#squeeze, /#center -->

```

Konec stránky

Zbylá část základní šablony zjišťuje zobrazení pravého sloupce. Všimnete si podmínky, ve které autor sleduje, jestli je zobrazen levý sloupec. Pokud není a nastavení webu zároveň velí zobrazit vyhledávací políčko, je toto zobrazeno na pravé straně.

```

<?php if ($sidebar_right): ?>
  <div id="sidebar-right" class="sidebar">
    <?php if (!$sidebar_left && $search_box): ?>
      <div class="block block-theme"><?php print $search_box ?></div>
    <?php endif; ?>
    <?php print $sidebar_right ?>
  </div>
<?php endif; ?>
</div> <!-- /container -->
</div>
<!-- /layout -->
<?php print $closure ?>
</body>
</html>

```

K pochopení toho, jak základní kostra šablony tématu v Drupalu funguje, máte již dostatek informací. Pamatujte na to, že veškeré prvky generované Drupalem je třeba umístit do nějaké kostry HTML, které dáte tvář pomocí kaskádových stylů.

Ostatní soubory v balíčku vzhledu

Pro základní nastavení šablony vzhledu postačuje samotný soubor *page.tpl.php*. Většina témat vzhledu pro redakční systém Drupal však obsahuje také další soubory s příponou *.tpl.php*. S jejich pomocí můžete upravit zobrazování jednotlivých prvků Drupalu, jako je článek, boxík, přihlašovací formulář, menu a podobně.

Šablona článku

Nejčastěji se vyskytujícím souborem se šablonou je hned po *page.tpl.php* téma vzhledu článku/obsahu. Najdete jej v souboru *node.tpl.php*. Díky uspořádání prvků v tomto souboru je možné změnit

způsob zobrazování jak samotného obsahu článku, tak věcí okolo něj: informací o datu publikování a autorovi, seznam kategorií, do kterých článek patří, stylů článku a podobně.

```
<div id="node-<?php print $node->nid; ?>" class="node<?php if ($sticky) {
print ' sticky'; } ?><?php if (!$status) { print '
node-unpublished'; } ?>">
<?php print $picture ?>
<?php if ($page == 0): ?>
  <h2><a href="<?php print $node_url ?>" title="<?php print $title ?>"><?php
print $title ?></a></h2>
<?php endif; ?>
  <?php if ($submitted): ?>
    <span class="submitted"><?php print $submitted; ?></span>
  <?php endif; ?>
  <div class="content">
    <?php print $content ?>
  </div>
  <div class="clear-block clear">
    <div class="meta">
      <?php if ($taxonomy): ?>
        <div class="terms"><?php print $terms ?></div>
      <?php endif;?>
    </div>
    <?php if ($links): ?>
      <div class="links"><?php print $links; ?></div>
    <?php endif; ?>
  </div>
</div>
```

V ukázkovém tématu vzhledu Garland najdete základní element DIV, ve kterém je celý článek umístěn. Tento element je označen identifikátorem začínajícím znaky *node-* a pokračujícím identifikačním číslem samotného článku (prvku obsahu). Téma vzhledu by mělo počítat také s možností umístění článku nahoře v seznamu, tedy přednostně a nezávisle na datu vydání. Tuto volbu můžete zapnout při vkládání nebo úpravách článku ve skupině **Možnosti vydání** pod volbou Přilepit nahoru v seznamu. Autor tématu Garland s tím počítá v nastavení třídy základního elementu DIV, kdy třídu *class=„node“* doplňuje v případě potřeby na *class=„node sticky“*. Administrátor webu má možnost zobrazovat i články, které nejsou publikovány pro veřejnost a jsou v administraci jakoby připraveny k dalšímu zpracování. V tématu Garland je na to pamatováno doplněním třídy *node-unpublished* v základním elementu DIV.



Poznámka: Názvy tříd nejsou pevně dány, záleží na vás, jak je budete chtít s ohledem na použité kaskádové styly nazvat. Rovněž vás nikdo nenutí odlišovat články přilepené nahoře v seznamu od ostatních.

Následuje zobrazení obrázku uživatele. Jestliže v nastavení **Administrace** → **Uživatelé** → **Uživatelské nastavení** povolíte každému uživateli vložení jeho obrázku (avatar), může se tento obrázek zobrazovat u každého článku, který tento uživatel bude publikovat. Pomocí *print \$picture* zobrazíte potřebný kód HTML pro vypsání tohoto obrázku.

Další část je již o maličko složitější. Je tvořena podmínkou, jejíž tvar je pro šablony vzhledu v Drupalu typický. Zažitým zvykem je používat co nejméně kódu v PHP a co nejvíce HTML. Proto uživatelský kód HTML není vypisován přes příkaz PHP *print*, ale naopak části PHP jsou pomocí značek `<?php` a `?>` vkládány do HTML. Přehlednost šablony je tak mnohem větší. První podmínka v ukázkovém příkladu slouží k rozhodnutí, zda je zobrazena stránka se seznamem článků. Pokud proměnná `$page` obsahuje nulu, pak může být zobrazena nejenom standardní výchozí stránka se seznamem článků, ale například i seznam příspěvků v určité kategorii a podobně. Budete-li potřebovat ověření, zda je zobrazena úvodní stránka webu, použijte logickou proměnnou `$is_front`.

V případě, že je zobrazena stránka s výpisem článků, vykreslí Garland nadpis článku do značky `H2`, přičemž použije zároveň odkaz s URL článku, do plovoucího popisku vloží nadpis článku a do názvu odkazu totéž.

Podmínka ověřující informace o autorovi a datu vydání článku je zde kvůli tomu, že tyto informace lze přes **Administraci** → **Prvky webu** → **Témata vzhledu** → **Konfigurace** vypnout a nezobrazovat. V tématu Garland jsou tyto informace zobrazovány na jednom řádku.



Obrázek 5.6: Ukázka standardního zobrazení informací o datu a autorovi článku

Pomocí vhodné úpravy šablony a kaskádových stylů lze docílit zobrazení například ve formě malého obdélníčku u článku, ve kterém bude pouze datum:



Obrázek 5.7: Upravené informace na Maxiorel.cz

Tato technika je použita například na webu www.maxiorel.cz. V tématu je potlačeno ověřování, zda se má informace zobrazit či nemá, namísto toho je vykreslena pokaždé, když je typ článku odlišný od statické stránky (page). Informace jsou uzavřeny v elementu DIV s třídou `submitted`, na třech řádcích je postupně zobrazen den vyjádřený slovem, datum a rok publikace článku.

```
<?php if($node->type != "page"): ?>
  <div class="submitted">
    <?php print format_date($node->created, 'custom', 'l')."<br/>"; ?>
    <?php print "<span class=\"datum\">".format_date($node->created, 'custom',
      'j. n.')."</span><br />"; ?>
    <?php print format_date($node->created, 'custom', 'Y'); ?>
  </div>
<?php endif; ?>
```

Vraťme se zpět k tématu Garland. Důležitou položkou je zobrazení obsahu článku pomocí `print $content`. Drupal sám rozhodne, jestli je nutné zobrazit pouze úvodník, či zda se návštěvníkovi stránky odkryje celý obsah článku.

Za tělem článku následuje zobrazení proměnné `$terms`. V tématu se nejprve ověří, zda je aktivováno třídění článků do kategorií, a pokud tomu tak je, zobrazí se seznam kategorií, do kterých články

patří. Poslední skupina v šabloně *node.tpl.php* slouží k vypsání odkazů na pokračování článku (je-li zobrazen jen úvodník), přidání komentářů a podobně.

Šablona bloku

Důležitým prvkem, který Drupal vytváří, je blok. Každý blok opět může být vykreslován jinak, než je standardní podoba daná jádrem Drupalu. Úpravu provedete v souboru *block.tpl.php*. Pro ilustraci, jak může obsah této šablony vypadat, se opět koukněte do tématu Garland:

```
<div id="block-<?php print $block->module .'-'. $block->delta; ??"
    class="clear-block block block-<?php print $block->module
        ?>">
<?php if (!empty($block->subject)): ?>
    <h2><?php print $block->subject ?></h2>
<?php endif;?>
    <div class="content"><?php print $block->content ?></div>
</div>
```

Podobně jako u výpisu článku je i zde obsah bloku uzavřen do elementu DIV s identifikátorem *block-*. Součástí identifikátoru je také označení modulu, který blok poskytuje, a hodnoty delta, která je jedinečná pro každý blok vytvořený určitým modulem. Typ modulu vytvářejícího blok lze použít také u nastavení třídy. Jestliže je povoleno zobrazení nadpisu bloku, resp. obsahuje-li nadpis bloku nějakou hodnotu, měl by být zobrazen. To zajistíte podmínkou a výpisem pomocí příkazu *print \$block->subject*. Máte-li s PHP trochu více zkušeností, jistě jste si všimli objektového přístupu k *\$block*. Dále nezapomeňte vypsát samotný obsah bloku pomocí příkazu *print \$block->content*.

Přehled šablon a dostupných proměnných

Při tvorbě témat využijete následující přehled, který shrnuje většinu dostupných proměnných PHP Template Engine, které lze v jednotlivých šablonách pro téma vzhledu používat. Ne každé z témat obsahuje všechny tyto proměnné, takže se s nimi seznámte na těchto stránkách. Třeba tu najdete inspiraci pro vylepšení informací na svém webu. Aktualizovaný seznam těchto proměnných najdete na adrese <http://drupal.org/node/11816>.

Soubor page.tpl.php

Soubor *page.tpl.php* je základní kostrou tématu vzhledu a určuje celý vzhled webu poháněného Drupalem.

- ◆ **\$base_path** – základní adresa URL k instalaci Drupalu. Ve většině případů obsahuje */*.
- ◆ **\$breadcrumb** – kód HTML k zobrazení drobečkové navigace, nejčastěji v horní části webu
- ◆ **\$closure** – proměnnou je nutno vypsát na konci každé stránky, obsahuje například javascriptové kódy, které je nutno zavolat po kompletním načtení stránky
- ◆ **\$content** – obsah HTML generovaný Drupalem
- ◆ **\$css** – pole se soubory kaskádových stylů pro aktuální stránku
- ◆ **\$directory** - složka s aktuálním tématem vzhledu, např. *themes/garland*
- ◆ **\$feed_icons** – proměnná sloužící k zobrazení ikonky zdrojů RSS pro aktuální stránku
- ◆ **\$footer_message** – řetězec s textem pro patičku stránky, obsah nastavujete v administraci
- ◆ **\$head** – kód HTML vytvářený Drupalem pomocí funkce *_get_html_head()*

- ◆ **\$head_title** – text, který má být zobrazen v titulku stránky
- ◆ **\$help** – nápověda, zpravidla se zobrazuje v administrační sekci při práci s konkrétními moduly
- ◆ **\$is_front** – logická proměnná s informací o tom, zda je zobrazena titulní stránka webu. V `page.tpl.php` se zpravidla používá k rozhodnutí, zda zobrazit pole **Cíle webu**, které vyplňujete v administraci a mělo by se zobrazit na titulní straně.
- ◆ **\$language** – jazyková verze, ve které je web s Drupalem zobrazen
- ◆ **\$layout** – pomocí tohoto nastavení je možné nastýlovat různé typy layoutu webu, v závislosti na tom, zda jsou zobrazeny boční pruhy. `$layout` může obsahovat hodnoty *none*, *left*, *right* a *both* informující o zobrazení žádného, levého, pravého nebo obou bočních sloupců.
- ◆ **\$logo** – cesta k obrázku s logem, závisí na nastavení v konfiguraci témat
- ◆ **\$messages** – kód HTML obsahující chybová hlášení, případně informace o tom, že byl vložen článek, že byl smazán nějaký obsah a podobně.
- ◆ **\$mission** – text obsahující **Cíle webu**, nastavuje se v administraci a je zamýšlen pro zobrazení na titulní straně webu. Jeho zobrazení lze potlačit v nastavení tématu vzhledu.
- ◆ **\$node** – počínaje Drupalem 5.x můžete `$node` použít k nastavení zobrazení celého článku přímo v `page.tpl.php`
- ◆ **\$onload_attribute** – proměnná používaná do Drupalu 4.7. Obsahuje značky přidávané do hlavičky HTML, nejčastěji příložený JavaScript
- ◆ **\$primary_links** – pole obsahující odkazy definované ve specifickém bloku a nastavované pomocí administrace
- ◆ **\$scripts** – počínaje Drupalem 5.x obsahuje tato proměnná kód HTML sloužící k zobrazení odkazů pro načtení JavaScriptových souborů. Není tedy nutné je ručně vypisovat do kódu v `page.tpl.php`.
- ◆ **\$search_box** – je-li umožněno zobrazování vyhledávacího políčka, obsahuje proměnná hodnotu
- ◆ **\$search_button_text** – lokalizovaný text tlačítka ve vyhledávacím formuláři, používáno do Drupalu 4.7
- ◆ **\$search_description** – lokalizovaný popis tlačítka ve vyhledávacím formuláři, používáno do Drupalu 4.7
- ◆ **\$search_url** – adresa URL, na kterou byly do Drupalu 4.7 předávány údaje z vyhledávacího formuláře
- ◆ **\$secondary_links** – pole obsahující tzv. sekundární odkazy, tak jak jsou definovány prostřednictvím administrace
- ◆ **\$sidebar_left** – kód HTML sloužící k zobrazení levého postranního sloupce
- ◆ **\$sidebar_right** – kód HTML sloužící k zobrazení pravého postranního sloupce
- ◆ **\$site_name** – název webu, jak je nastaven v administraci; hodnota je prázdná, je-li zobrazení potlačeno v nastavení tématu vzhledu
- ◆ **\$site_slogan** – slogan webu, jak je nastaven v administraci; hodnota je prázdná, je-li zobrazení potlačeno v nastavení tématu vzhledu
- ◆ **\$styles** – kód obsahující odkazy na styly připojované ke stránce
- ◆ **\$tabs** – kód HTML zajišťující případné zobrazení záložek v horní části stránky, typicky v administrační části
- ◆ **\$title** – titulek stránky. Nezaměňujte jej s `$head_title`. `$title` je ve většině případů název článku, případně název kategorie ve výpisu článků.

Soubor `node.tpl.php`

Důležitá šablona ovlivňující zobrazení celého článku a seznamu článků. Umožňuje ovlivnit výstup proměnné `$content`.

- ◆ **`$content`** – obsah článku, v případě zobrazení seznamu je zde pouze úvodník
- ◆ **`$date`** – datum vytvoření článku
- ◆ **`$directory`** – složka s tématem vzhledu, např. `themes/garland`
- ◆ **`$id`** – identifikační číslo pořadí článku v seznamu
- ◆ **`$is_front`** – logická proměnná s informací o tom, zda je zobrazena titulní stránka webu
- ◆ **`$links`** – kontextové odkazy k článku, například k přidání komentáře, zobrazení pokračování úvodníku atd.
- ◆ **`$main`** – proměnná obsahující hodnotu `true`, je-li článek zobrazen v kontextu, například ve výpisu článků na titulní stránce. Tato proměnná se v současnosti již nevyužívá (od Drupalu 4.7) a narazit na ni můžete jenom v některých starších tématech vzhledu.
- ◆ **`$name`** – jméno autora článku
- ◆ **`$node`** – objekt reprezentující článek. Vzhledem k tomu, že obsah objektu `$node` je závislý na typu článku, musíte si konkrétní vlastnosti zobrazit sami. Stačí do šablony `node.tpl.php` vložit následující kód: `<pre><?php print_r($node); ?></pre>`.
- ◆ **`$node_url`** – adresa URL s odkazem na článek
- ◆ **`$page`** – je-li článek zobrazen jako statická stránka (page), obsahuje proměnná hodnotu `true`
- ◆ **`$picture`** – odkaz a obrázek komentujícího uživatele. Musí být povoleno zobrazování avatarů a obrázek musí být nastaven v uživatelském profilu.
- ◆ **`$sticky`** – je-li článek nastaven pro „Přilepení nahoru v seznamu“, obsahuje proměnná hodnotu `true`
- ◆ **`$submitted`** – jméno autora článku a datum vytvoření, podmínkou je povolení zobrazení těchto informací v administraci webu pro konkrétní typ obsahu
- ◆ **`$taxonomy`** – pole obsahující kategorie, kam článek patří
- ◆ **`$teaser`** – úvodník článku
- ◆ **`$terms`** – HTML pro zobrazení odkazů na kategorie, do kterých článek patří
- ◆ **`$title`** – nadpis článku
- ◆ **`$zebra`** – střídání odd a even k využití sudého a lichého pořadí v seznamu článků

Vybrané prvky objektu `$node` a jejich význam

Význam prvků objektu `$node` je následující:

- ◆ **[`nid`]** – identifikační číslo článku, typicky pořadové číslo článku, ze kterého vidíte, kolikátým již je tento článek na vašem webu
- ◆ **[`type`]** – typ článku, například `story` nebo `page`
- ◆ **[`status`]** – status článku
- ◆ **[`created`]** – datum vytvoření článku v unixovém formátu, např. `1188810677`
- ◆ **[`changed`]** – datum poslední změny článku v unixovém formátu, např. `1188810677`
- ◆ **[`comment`]** – nastavení komentářů, číselné hodnoty odpovídají pořadí voleb při zadávání článku
- ◆ **[`promote`]** – obsahuje hodnotu `1`, je-li článek nastaven k zobrazování na titulní straně

- ◆ **[sticky]** – obsahuje hodnotu 1, je-li článek nastaven jako „Přilepit nahoru v seznamu“
- ◆ **[revision_timestamp]** – datum poslední revize článku v unixovém formátu, např. 1188810677
- ◆ **[title]** – nadpis článku
- ◆ **[teaser]** – úvodník článku
- ◆ **[log]** – poznámka ke článku, kterou nevidí čtenáři
- ◆ **[format]** – pořadové číslo zvoleného formátování článku, typicky 1 odpovídá Filtered HTML, dvojka značí PHP Code a trojka formátování Full HTML
- ◆ **[uid]** – identifikační číslo uživatele, který článek vytvořil
- ◆ **[name]** – jméno uživatele, který článek vytvořil
- ◆ **[picture]** – avatar (obrázek) uživatele, který článek vytvořil
- ◆ **[path]** – cesta k článku, tedy adresa URL bez základní cesty k webu
- ◆ **[last_comment_timestamp]** – datum posledního komentáře k článku v unixovém formátu, např. 1188823108
- ◆ **[last_comment_name]** – jméno naposledy komentujícího uživatele
- ◆ **[comment_count]** – počet komentářů k článku
- ◆ **[taxonomy]** – pole s odkazy na kategorie článku
- ◆ **[readmore]** – obsahuje jedničku, je-li zobrazen odkaz Číst dále
- ◆ **[content] [body] [#value]** – obsah článku
- ◆ **[content] [links]** – pole s odkazy pod článkem

Přehled těchto prvků není vyčerpávající, velice záleží na konkrétním nastavení vašeho webu, na typu článku, na instalovaných modulech, které přidávají nebo mění obsah článku, a na samotném obsahu a jeho nastavení. Vybraný seznam prvků objektu `$node` by vám však měl zásadně ulehčit složitější návrhy témat vzhledu pro Drupal. Připomínám, že k objektu `$node` přistupujete konstrukcí `$node->prvek`, například `$node->teaser`.

Soubor `block.tpl.php`

Tento soubor slouží k nastavení zobrazování bloků umístěných nejčastěji po stranách hlavního obsahu webu. Šablona není povinná.

- ◆ **`$block->module`** – jméno modulu, který vygeneroval konkrétní blok
- ◆ **`$block->delta`** – číslo bloku pro odpovídající typ modulu
- ◆ **`$block->subject`** – nadpis bloku
- ◆ **`$block->content`** – obsah/tělo bloku v HTML
- ◆ **`$block->status`** – stav bloku obsahující hodnoty 0 nebo 1
- ◆ **`$block->region`** – poloha umístění bloku typicky obsahující hodnoty `left`, `right`, `header` a `footer` (vlevo, vpravo, v hlavičce a v patičce)
- ◆ **`$block->throttle`** – nastavení pro odlehčení webu při zátěži – zde znamená, že se blok nemá zobrazovat
- ◆ **`$directory`** – složka, ve které je umístěno aktuální téma, například `themes/garland`
- ◆ **`$is_front`** – logická proměnná s informací o tom, zda je zobrazena titulní stránka webu
- ◆ **`$id`** – identifikátor pořadí bloku, v jakém je zobrazen: první blok obsahuje jedničku, druhý dvojku atd.

- ◆ **\$block_id** – stejně jako předchozí, nicméně hodnoty jsou samostatné pro pravý a levý boční pruh
- ◆ **\$zebra** – obsahuje hodnoty *odd* nebo *even*, tedy sudá a lichá. Praktické využití najde při odlišování jednotlivých položek v řadě pomocí kaskádových stylů.
- ◆ **\$block_zebra** – stejně jako předchozí, nicméně tato hodnota je samostatná pro pravý a levý boční pruh

Soubor box.tpl.php

Nepovinná šablona zajišťující zobrazení HTML boxíku okolo prvku stránky. Příkladem budiž formulář pro nastavení způsobu zobrazení komentářů.

- ◆ **\$title** – nadpis boxíku
- ◆ **\$content** – obsah boxíku
- ◆ **\$region** – poloha umístění boxíku, tedy *left*, *right*, *header* nebo *footer*

Soubor comment.tpl.php

Pomocí této šablony ovlivníte zobrazení komentáře. Vztahuje se k jedné položce, nikoli k výpisu komentářů na stránce a jeho grafickému ztvárnění.

- ◆ **\$author** – odkaz na stránku s profilem autora
- ◆ **\$comment** – objekt reprezentující konkrétní komentář
- ◆ **\$content** – tělo komentáře
- ◆ **\$date** – datum vložení komentáře
- ◆ **\$directory** – složka s umístěním aktuálního tématu, např. *themes/garland*
- ◆ **\$id** – pořadové číslo zobrazeného komentáře
- ◆ **\$is_front** – logická proměnná s informací o tom, zda je zobrazena titulní stránka webu
- ◆ **\$links** – kontextové odkazy pod komentářem
- ◆ **\$new** – lokalizovaný text pro označení nového komentáře
- ◆ **\$picture** – odkaz a obrázek komentujícího uživatele. Musí být povoleno zobrazování avatarů a obrázek musí být nastaven v uživatelském profilu.
- ◆ **\$submitted** – informace o autorovi a datu vložení komentáře
- ◆ **\$title** – nadpis komentáře
- ◆ **\$zebra** – střídání *odd* a *even* k využití sudého a lichého pořadí v seznamu komentářů

Soubor template.php

Součástí mnoha témat vzhledu je také nepovinný soubor *template.php*. Práce s ním je už záležitostí zkušených tvůrců témat pro redakční systém Drupal. V tomto souboru můžete definovat funkce, které poslouží k jinému zobrazování standardních prvků v Drupalu a které nelze ovlivnit pomocí běžných šablon *tpl.php*.

Úpravu standardních funkcí Drupalu provedete tak, že si najdete původní kód funkce a vložíte jej do souboru *template.php*. Nezapomeňte na to, že soubor musí začínat značkou `<?php` a končit `?>`. V názvu funkce zaměňte úvodní předponou *theme_* za výraz *phptemplate_*. Je také dobrým zvykem doplnit před funkcí komentář v podobě poznámky o tom, k čemu konkrétní úprava slouží. Nezapom-

meňte na to, že každá předefinovaná funkce pomocí `template.php` musí mít návratovou hodnotu. Pro ilustraci uvádím příklad z průvodce tvorbou témat na adrese drupal.org/node/11811:

Původní funkce:

```
<?php
function theme_item_list($items = array(), $title = NULL) {
?>
```

A její upravená varianta, kde je v návratové hodnotě uveden původní název funkce a pole s předávanými hodnotami. Tělo upravené ukázkové funkce je prázdné, vývojář by na tomto místě nejspíše upravil hodnoty pole `$items` a titulku:

```
<?php
/**
 * Catch the theme_item_list function, and redirect through the template api
 */
function phptemplate_item_list($items = array(), $title = NULL) {
    return _phptemplate_callback('item_list', array('items' => $items, 'title' => $title));
}
?>
```

Úprava udržovací stránky

Čas od času je potřeba v redakčním systému provést úpravy, u kterých je vhodné, aby během nich neměli návštěvníci přístup k webu. Typickou situací je aktualizace redakčního systému na novější verzi, během níž probíhá změna struktury v databázových tabulkách, a je tedy nežádoucí, aby kdokoli cokoli během této operace zobrazoval. Při zobrazení servisní stránky můžete zjistit, že její vzhled vám příliš nevyhovuje – většinou obsahuje ikonku Drupalu a standardní text o nedostupnosti webových služeb. Přestože ne každé téma vzhledu nabízí šablonu pro úpravu servisní stránky, není její vytvoření nic složitého. Zvládnete je j sami i bez předlohy.

Prvním předpokladem pro vytvoření vlastní udržovací stránky je úprava souboru `template.php` v aktuálně používaném tématu vzhledu. Nemáte-li tento soubor ve složce s tématem přítomen, tak jej vytvořte; pamatujte na to, že jeho kódování by mělo být UTF-8. Následně do tohoto souboru doplňte tuto funkci:

```
function phptemplate_maintenance_page($content) {
    return _phptemplate_callback('maintenance_page', array('content' => $content));
}
```

Funkce zajistí odchyčení standardní funkce `theme_maintenance_page` a pomocí správce témat ji přesměruje na použití šablony v souboru `maintenance_page.tpl.php`. Tento soubor pak musíte ve složce s tématem vzhledu vytvořit. Zde je ukáзка, jak může takový soubor vypadat (převzato a upraveno z tématu *internet_jobs*):

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="cs" xml:lang="cs">
<head>
<title>Maxiorel.cz</title>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<link rel="Shortcut Icon" type="image/x-icon" href="favicon.ico" />
<meta name="generator" content="Drupal" />
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
```



```

<base href="<?php global $base_url; print $base_url;?>" />
<style type="text/css" media="all">@import "themes/internet_jobs/maintenance.css";</style>
</head>
<body>
<div id="message-box">
  <?php if ($site_name != ""): ?>
  <h2><?php print $site_name ?><br />Údržba stránek</h2>
  <?php else: ?>
  <h2>503 - Service Unavailable<br />Údržba stránek</h2>
  <?php endif; ?>
  <div id="message"><?php print $content ?></div>
</div>
<div>
  
</div>
</body>
</html>

```

Jak vidíte, obsahem souboru *maintenance_page.tpl.php* je běžná struktura HTML, ve které vůbec nemusí být použit kód z PHP. Obsah souboru je zcela na vašem uvážení, můžete využít univerzálnosti šablony a některé věci, jako je název webu a logo, vypisovat pomocí běžných proměnných a PHP. Stejně tak lze vše zvládnout čistě s použitím (X)HTML.

Vykreslování bloků

Tvorba témat vzhledu umožňuje zkušeným uživatelům přetvořit základní vzhled Drupalu do podoby, která se mu ani trochu nepodobá. S použitím modulu *Views* lze vytvářet bloky s novým obsahem, který pomocí pokročilých technik zobrazíte na kterémkoli místě webu, nejenom na standardních pozicích jako je pravý a levý sloupec, hlavička či patička nebo obsah. Jednoduše tak můžete vytvořit panel s novinkami, který na úvodní stránce zobrazíte nad běžným výpisem článků. Uživatelské vykreslování bloků mimo standardní pozice lze využít v praxi například k vkládání reklamy na web. Jistě jste si někde všimli širokého reklamního banneru, který byl zobrazen úplně na začátku stránky, ještě před jejím obsahem. Můžete jej samozřejmě umístit do šablony *page.tpl.php*, takové řešení by však nebylo příliš praktické a vyžadovalo by neustálé zásahy do šablony pokaždé, když by se změnil inzerent nebo by nastala potřeba reklamní obrázek a odkaz vyměnit.

Místo toho je lepší vytvořit pomocí **Administrace** → **Prvky webu** → **Bloky** nový blok, do jehož těla vložíte kód pro zobrazování reklamy. Poté otevřete v editoru šablonu *page.tpl.php* a do místa, které není standardně obsluhováno pozicí v nabídce bloků, například před ostatní obsah, vložte tento kód:

```

<?php
$block = module_invoke('block', 'block', 'view', 1);
if ($block['content']) {
  $output = "<div class=\"content\">".$block['content']. "</div>\n";
  print $output;
}
?>

```

S malým vysvětlením vám bude struktura kódu i tvorba dalších výpisů konkrétních bloků připadat jasná a srozumitelná. Jako první musíte přiřadit konkrétní blok do nějaké proměnné. To provedete použitím funkce *module_invoke()*, která obsahuje čtyři důležité parametry.

- ◆ Na prvním místě je název modulu, který blok vytváří, v první ukázce je to tedy přímo modul *block*.
- ◆ Druhý a třetí parametr zůstanou stejné, tedy *block* a *view*.
- ◆ Poslední, čtvrtý parametr určuje hodnotu *\$delta* konkrétního bloku, který chcete zobrazovat. Tuto hodnotu zjistíte nejlépe v administraci.

Otevřete si **Administrace** → **Prvky webu** → **Bloky**. Přejedete-li myší nad odkazem Konfigurace u každého z bloků v seznamu, webový prohlížeč by vám měl ukázat ve stavovém řádku adresu k nastavení tohoto každého bloku, přičemž na konci adresy je uvedeno číslo. To je právě hodnotou *\$delta*. Pokud nic takového v prohlížeči nevidíte, jednoduše na odkaz **Konfigurace** klepněte a číslo si přečtěte z adresního řádku webového prohlížeče.

Pro ilustraci takto může vypadat vykreslení bloku poskytovaného modulem *Views*:

```

<?php
  $blocknovinky = module_invoke('views', 'block', 'view', 3);
  if ($blocknovinky['content']) {
    $output = "<div id=\"novinky\">\n\t<h2>\".$block['subject']\".
              \"</h2>\".$blocknovinky['content']\".</div>\n";
    print $output;
  }
  ?>

```



Tip: Všimněte si, že v uvedeném bloku je kromě vypsání těla bloku (`$blocknovinky['content']`) použito i zobrazení nadpisu bloku definovaného v administraci. Autor kódu k tomu využívá `$block['subject']`.

The screenshot shows a website layout with the following elements:

- Navigation Bar:** Links for 'Levné letenky', 'Polzer-SW.com', 'Recepty, kuchařka', 'Spamy CZ - antispamový portál', 'Vánoční dárky pro muže', and 'DoBazaru.cz - inzerce ZDARMA a bazar'.
- Header:** Logo for 'maxiorel' and text 'software, poradenství a webové stránky do velikosti XL'.
- Search Bar:** A search input field with a 'Hledat' button.
- Navigation:** 'Úvodní strana', 'Videokázky softwaru', 'Nabídka služeb', 'Ceník', 'Záloha webů, souborů, FTP a databází jedním klepnutím', 'Fórum'.
- Content Blocks:**
 - Rozdělení článků:** 'Zajímavé weby a programy', 'Programování, tvorba webu', 'Videokázky softwaru', 'Různé, ze života', 'Maxiorel – novinky', 'Tiskové zprávy'.
 - Online nástroje:** 'MD5 online generátor', 'Whois - info o doménách'.
 - maxiorel:** 'Google PageRank', 'Můj účet', 'Můj blog', 'Poslední příspěvky', 'TextLinkAds stats', 'vytvořit obsah'.
 - Letem IT světem:** 'Plná verze nahrávače obrazovky Camtasia Studio zadarmo', '23. 11. 2007 Rádi byste občas pořídili video s obsahem obrazovky? U obrázků je to snadné, stačí použít klávesu PrintScreen, případně specializovaný program. V případě videa se nabízí využití Camtasia Studio uvolněného zadarmo...', 'Drupal 6.0 beta 3 k dispozici', '22. 11. 2007 Před několika hodinami (tedy včera...) byla uvolněna další betaverze open source redakčního systému Drupal. Od druhé betaverze se vyskytla potřeba zanést 180 oprav do kódu Drupalu.', 'Video: WordPress, Windows Server 2008 a IIS7', '18. 11. 2007 Opět je tu další video týkající se spolupráce nového IIS7 na Windows Server 2008 s aplikacemi psanými v PHP, konkrétně s blogovacím systémem WordPress. Co vše na tomto videu z Channel 9 uvidíte?'
 - Tiskové zprávy z IT:** 'Microsoft vyhláší tvůrce nejlepších her', 'IGNUM snižuje cenu generických domén', 'IGNUM ohlašuje akci 100% sleva na webhosting', 'Češi propadli blogování', 'Microsoft nabízí podnikům a organizacím vyhledávač zdarma', 'Photoshop CS3 Extended za pouhých 1 100 Kč', 'Silverlight Streaming k dispozici na Windows Live'.
 - Reklama:** 'odkazy adFOX', 'Data Recovery: obnova dat', 'Expresní záchrana a obnova dat z HDD, RAID aj. Specializovaná laboratoř s nonstop provozem. www.datarecovery.cz', 'Orbix - letenky pro firmy', 'Kompletní řešení služebních cest. Letenky, víza, ubytování, pojištění. www.orbix.cz', 'Dovolená letecky -'.




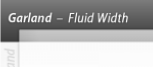
Obrázek 5.8: Ukázka ručně vložených bloků do tématu vzhledu v podobě boxíků s novinkami a tiskovými zprávami

Kaskádové styly, obrázky a další soubory v tématu

Častou otázkou začínajících tvůrců témat vzhledu pro redakční systém Drupal je: „Kam umístit obrázky tvořící součást vzhledu?“ Na to je velmi jednoduchá odpověď. Kamkoli do složky s tématem. Samozřejmě nejlepší praxí je dodržovat ve složce s tématem vzhledu pořádek a vše pěkně třídit. Soubory šablon tedy přijdou přímo do složky s konkrétním tématem. Tamtéž by měly přijít soubory *screenshot.png* a *screenshot_large.png*. Ty se zobrazují v administraci a umožňují lepší rozhodnutí o volbě vzhledu webu.

Témata vzhledu **Seznam** Konfigurace

Vyberte, která témata vzhledu jsou uživatelům k dispozici, a zvolte výchozí téma. Pro nastavení zobrazení pro celý web, klikněte na záložku "konfigurace". Případně můžete toto nastavení přebít pro určité téma (klikněte na odkaz "konfigurace" u daného tématu). Poznámka: různá témata vzhledu mohou mít různé rozložení prvky obsahu.

Ukázka	Jméno	Povoleno	Výchozí	Operace
	arcateria: themes/arcateria	<input checked="" type="checkbox"/>	<input type="radio"/>	konfigurace
	blumarine: themes/blumarine	<input checked="" type="checkbox"/>	<input type="radio"/>	konfigurace
	chameleon: themes/chameleon	<input type="checkbox"/>	<input type="radio"/>	
	garland: themes/garland	<input checked="" type="checkbox"/>	<input checked="" type="radio"/>	konfigurace

Obrázek 5.9: Screenshots se zobrazují při výběru témat

Ikonky na web, obrázky tématu a další doplňující grafiku můžete umístit do nějaké podložky v tématu, například *images*. Myslete na to, že v kaskádových stylech byste měli uvádět relativní cesty k těmto souborům. Potřebujete-li uvést cesty v souborech se šablonami, vždy použijte funkce *base_path()* a *path_to_theme()*, které zajistí vypsání plné cesty k webu a složce s tématem. Vaše nové téma se tak stane univerzálnější a případný přesun do jiné složky a přetvoření v jiné téma bude bez problémů způsobených špatně uvedenými cestami k souborům.