

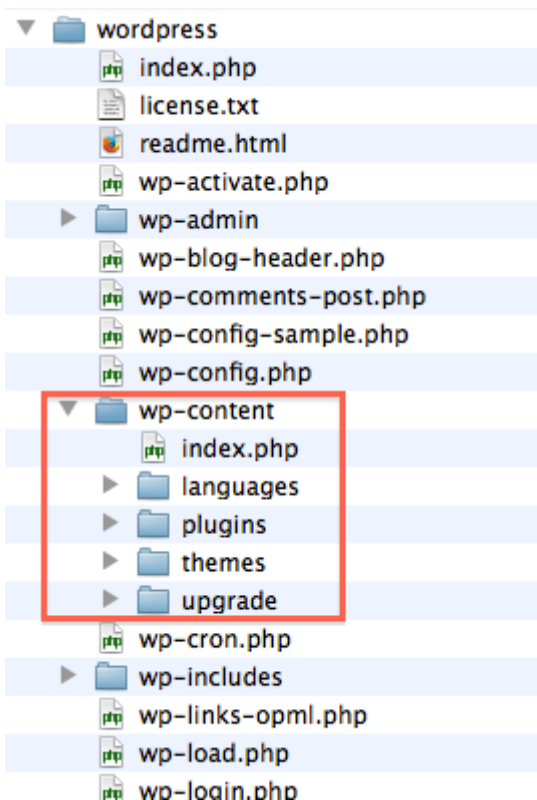
Šablony ve Wordpressu

Co se v modulu naučíte?

- Jak fungují šablony
- Jak vytvořit vlastní šablonu
- Co vše musí obsahovat šablona, aby využívala dynamické možnosti Wordpressu

Jak fungují šablony ve Wordpressu

Instalace Wordpressu obsahuje v adresáři wp-content veškerý obsah, zatímco mimo tuto složku je samotný kód redakčního systému, do kterého není vhodné zasahovat. Zásahy do kódu samotného Wordpressu by totiž způsobily problémy při aktualizaci na novější verzi, což není žádoucí.



Šablony jsou umístěny v adresáři themes. Každá šablona má svůj vlastní adresář, ve kterém jsou všechny soubory té šablony. Naprosto **minimální šablona** musí obsahovat dva soubory:

- soubor `style.css`, který definuje šablonu pro administrátorské rozhraní Wordpressu, a
- soubor `index.php`, který nemusí obsahovat vůbec nic (ale pak také nic nedělá); pro reálné použití musí obsahovat alespoň základní kód pro výpis příspěvků.

Základní definice šablony: soubory `style.css` a `index.php`

Soubor `style.css` musí v poznámce na začátku obsahovat alespoň základní údaje:

```

1  /*
2     Theme Name: První šablona
3     Author: Jméno Příjmení
4     Description: První testovací šablona
5     Version: 0.1
6  */

```

V rozšířené verzi je daleko více položek, lze se inspirovat např. u šablony twentythirteen, která je součástí standardní instalace Wordpressu:

```

1  /*
2  Theme Name: Twenty Thirteen
3  Theme URI: http://wordpress.org/themes/twentythirteen
4  Author: the WordPress team
5  Author URI: http://wordpress.org/
6  Description: The 2013 theme for WordPress takes us back to the blog, featuring a full range of post
7  Version: 1.1
8  License: GNU General Public License v2 or later
9  License URI: http://www.gnu.org/licenses/gpl-2.0.html
10 Tags: black, brown, orange, tan, white, yellow, light, one-column, two-columns, right-sidebar, flex
11 Text Domain: twentythirteen
12
13 This theme, like WordPress, is licensed under the GPL.
14 Use it to make something cool, have fun, and share what you've learned with others.
15 */

```

Doplníme-li takto vytvořený soubor style.css o prázdný index.php, máme vytvořenou první šablonu. Zatím nebude funkční, ale už se zobrazí v administraci Wordpressu v položce **Appearance - Themes**.

The screenshot shows the WordPress Admin Dashboard. On the left is a sidebar menu with 'Appearance' selected. The main content area has two tabs: 'Manage Themes' (active) and 'Install Themes'. Under 'Manage Themes', the 'Current Theme' is 'Twenty Thirteen' by the WordPress team, version 1.1. A preview of the theme is shown on the left. Below this, the 'Available Themes' section displays a grid of theme cards. The first card is 'První šablona' by Jméno Příjmení, with 'Activate', 'Live Preview', and 'Details' links, and a red 'Delete' button. The second card is 'Twenty Twelve' by the WordPress team, also with 'Activate', 'Live Preview', 'Details' links, and a red 'Delete' button.

Chcete-li šablonu aktivovat, je třeba kliknout na odkaz **Activate** uvedený u příslušné šablony.

Funkční šablona - doplnění obsahu souboru index.php

```

1 <?php
2
3 get_header();
4
5 if ( have_posts() ) {
6     while ( have_posts() ) {
7         the_post();
8         the_content();
9     }
10 }
11
12 get_sidebar();
13 get_footer();
14
15 ?>

```

V kódu vidíte velmi jednoduchý soubor `index.php`, který vypisuje obsah příspěvků uložených ve Wordpressu. Projdeme si teď jednotlivé řádky a vysvětlíme si, co dělají.

Řádek 3, 12 a 13: vkládají do šablony hlavičku, sidebar a patičku. Kromě funkcí `get_header()`, `get_sidebar()` a `get_footer()` je možné použít také obecnější funkci `get_template_part()`, která umožňuje vložit libovolný soubor ze šablony určením částí jeho názvu, viz níže.

Řádky 5 - 10 tvoří základ zobrazování ve Wordpressu, takzvaný **the loop** (smyčka). Vypisují podle parametrů v URL příspěvky, tj. např.

- když se má vypisovat kategorie, vypíše všechny příspěvky z kategorie,
- když je v URL konkrétní příspěvek, vypíše jen jeden příspěvek,
- když je v URL nějaký štítek (tag), vypíšou se všechny příspěvky označené tímto štítkem atd.

Řádek 5 a 10: podmínka `if` jestli jsou pro dané parametry k dispozici nějaké příspěvky.

Řádek 6 a 9: pomocí cyklu `while` projdeme jednotlivé příspěvky.

Řádek 8: funkce `the_post()` musí být obsažena ve smyčce, protože se stará o nasměrování wordpressu na ten správný příspěvek. Nic sama nevypisuje.

Řádek 9: funkce `the_content()` vypisuje obsah příspěvku.

Tentýž kód lze zapsat i s použitím jiné syntaxe v PHP:

```

1 // začátek smyčky
2 <?php if ( have_posts() ) : while ( have_posts() ) : the_post(); ?>
3
4 <?php the_content(); ?>
5
6 // konec smyčky
7 <?php endwhile; else: ?>
8
9 // v případě, že nejsou žádné příspěvky, oznámíme to uživateli.
10 <p><?php _e('Sorry, no posts matched your criteria.');

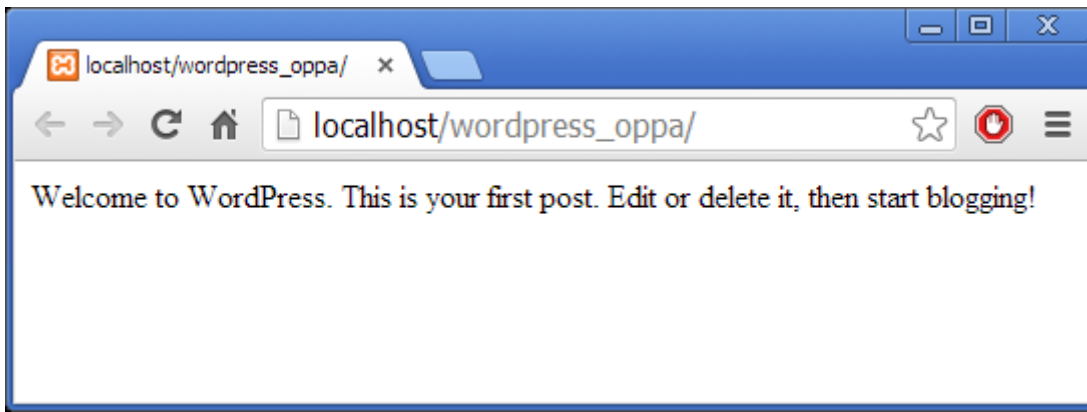
```

Funkce `_e()` na **řádku 8** se používá k vypsání lokalizovaných textů. Lokalizace je uložena v adresáři `wp-content/languages`.

Pokud neexistují soubory `header.php`, `footer.php` a `sidebar.php`, Wordpress automaticky použije svůj přednastavený obsah těchto souborů. Výsledný web tedy vypadá zhruba takto:



V dalším kroku doplníme chybějící soubory `header.php`, `footer.php` a `sidebar.php` a naplníme je vlastním obsahem. Pokud bychom je pouze vytvořili a nechali prázdné, nebudou se už zobrazovat přednastavené obsahy, jak je vidět na předchozím obrázku, ale jen obsah, který jsme definovali v souboru `index.php`, tedy pouze obsah příspěvků:



Funkce `get_template_part()`

```
1 | <?php get_template_part( $slug, $name ); ?>
```

Funkce `get_template_part()` má dva parametry, určující obecnou a specifickou část souboru šablony. Například lze mít samostatný soubor pro zobrazování navigace, přičemž můžeme (specifickou částí) určit, že se bude vypisovat jiná navigace pro příspěvky s videem.

```
1 | get_template_part('navigace');           // vloží soubor navigace.php
2 | get_template_part('navigace','video');   // vloží soubor navigace-video.php
```

Touto funkcí je možné oddělit části kódu ze šablony, aby celá šablona byla přehlednější, například je možné vkládat samostatně smyčku pro vypisování příspěvků:

```
1 | get_template_part('loop');               // vloží soubor loop.php
```

Zajímavé využití této funkce se nabízí u odvozených šablon (tzv. **child themes**). Soubor se totiž hledá nejprve v odvozené šabloně a poté v nadřazené, ze které je použita šablona odvozená. Pokud použijeme i druhý parametr, např. `get_template_part('navigace', 'video')`, hledají se soubory v šablonách v tomto pořadí:

1. `navigace-video.php` v odvozené šabloně (child)
2. `navigace-video.php` v nadřazené šabloně (parent)
3. `navigace.php` v odvozené šabloně a nakonec
4. `navigace.php` v nadřazené šabloně.

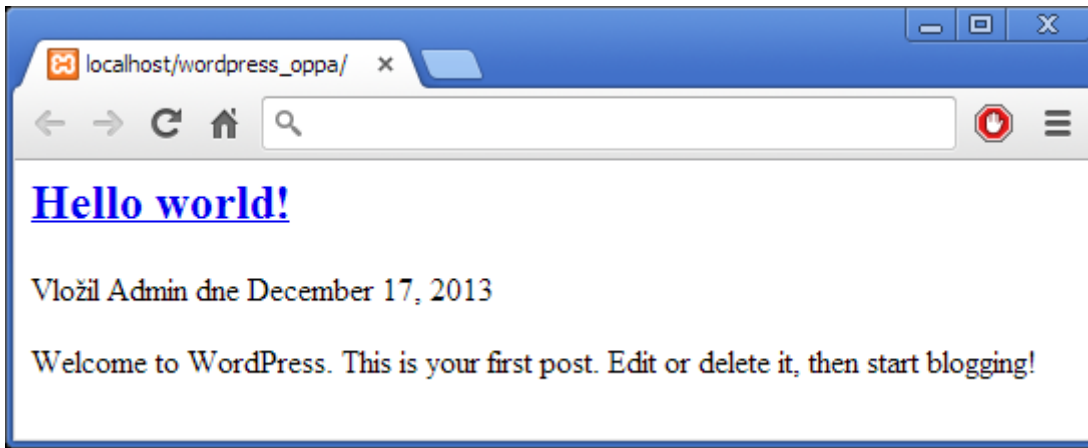
Další rady k využití této funkce naleznete buď v [codexu](#) nebo na stránce [Understanding get_template_part](#).

Úpravy uvnitř smyčky - výpis příspěvků

Do souboru `index.php` můžeme přidat i další funkce, které vypisují další části příspěvku, například název, datum nebo autora:

```
1 | <?php
2 | if ( have_posts() ) {
3 |     while ( have_posts() ) {
4 |         the_post();
5 |     }
6 |     <div class="clanek">
7 |     <a href="<?php the_permalink(); ?>"><h2><?php the_title(); ?></h2></a>
8 |     <p>Vložil <?php the_author(); ?>
9 |     dne <?php echo get_the_date(); ?>
10 |    <?php the_content(); ?>
11 |    </div>
12 |    <?php
13 |    }
14 | }
15 | ?>
```

Výsledek:



Funkce mohou mít různé parametry. Tyto funkce, které vypisují části příspěvku, mívají parametry pro řetězce, které se mají vypsat před a za samotným textem. Např.

```
1 | <?php the_title( '<h3>', '</h3>' ); ?>
```

přidá před titulek příspěvku HTML tag pro začátek nadpisu třetí úrovně a za titulkem příspěvku tento tag opět uzavře.

Všimněte si, že ve Wordpressu existují dva různé typy funkcí, jedny, které rovnou vypisují (např. `the_title()`) a druhé, které pouze vracejí hodnotu a výpis je třeba zajistit pomocí příkazu `echo` (např. `get_the_date()`). Některé funkce mají parametr pro vypisování, tj. lze určit, zda pouze vrátí hodnotu nebo se rovnou vypisují, např. třetí parametr `the_title()`.

Formát zobrazovaného jména a data se nastavuje v administraci Wordpressu:

- jméno nastavíte v nabídce Uživatelé (Users) při úpravách konkrétního uživatele. Hledejte položku Veřejně zobrazovat jako (Display name publicly as).
- datum nastavíte v nabídce Nastavení (Settings), formát data lze vybrat z připravených šablon nebo doplnit vlastní. Pro datum ve tvaru den. měsíc. rok (např. 23. 8. 2013) se použije nastavení j. n. Y.

Doplnění souborů `header.php` a `footer.php`

Hlavička a patička jsou normální php soubory, které obsahují běžné HTML značky používané ve všech dokumentech. Části kódu jsou dynamicky doplňované z Wordpressu.

Pro hlavičku (soubor `header.php`) využijeme především funkci `bloginfo()`, která zobrazuje různé parametry webu, například jeho název nebo URL adresu složky, ve které jsou uloženy kaskádové styly.

Z následujícího příkladu je zřejmé, že využití těchto funkcí (a tedy dynamické doplňování údajů do záhlaví stránek) je poměrně jednoduché. Používáme funkci `bloginfo()` pro čtyři různé parametry:

- `bloginfo('name')` pro název webu
- `bloginfo('stylesheet_url')` obsahuje umístění souboru `style.css` z aktuální šablony
- `bloginfo('stylesheet_directory')` je adresář se styly, v tomto případě totožný s adresářem šablony; pozor, že nekončí lomítkem, tj. pokud chceme připojit soubor `vzhled.css`, je třeba před jeho název dopsat chybějící lomítko

- `bloginfo('description')` obsahuje popis webů (slavné "Just another WordPress site"), které můžete nahradit v nastavení webu v administraci Wordpressu

```

1 <!doctype html>
2 <html lang="cs">
3 <head>
4     <meta charset="UTF-8">
5     <title><?php bloginfo('name'); ?></title>
6     <link rel="stylesheet" href="<?php bloginfo('stylesheet_url'); ?>" type="text/css">
7     <link rel="stylesheet"
8     href="<?php bloginfo('stylesheet_directory'); ?>/vzhled.css" type="text/css">
9     <?php wp_head(); ?>
10 </head>
11 <body>
12
13 <h1><?php bloginfo('name'); ?></h1>
14 <p><em><?php bloginfo('description'); ?></em></p>

```

Výsledek:



Zbývá vysvětlit **řádek 9**: na funkci `wp_head()` jsou navázané různé funkce spojené s hlavičkou, je proto třeba ji v souboru `header.php` zavolat. Mimo jiné ji vyžaduje lišta, která se zobrazuje, když jste přihlášení do administrace Wordpressu. Ta ale ještě potřebuje zavolat funkci `wp_footer()` ze souboru `footer.php`. Do souboru `footer.php` vložíme následující kód:

```

1 <?php
2 wp_footer();
3 ?>
4 <p>... obsah patičky ...</p>
5 </body>
6 </html>

```

Výsledek vypadá takto:



Dynamické části stránky - widgety

V další části se podíváme na dynamicky vkládané části stránky, ať už jde o různé kalendáře, RSS kanály nebo vyhledávací pole. Všechny tyto prvky je možné vložit prostřednictvím volby **widgety** v nastavení šablony - ovšem za předpokladu, že šablona takovou funkčnost nabízí.

Abychom mohli tento nástroj využívat, je třeba nejprve registrovat bloky, do kterých se potom přetáhnou v administraci Wordpressu jednotlivé widgety. Toho dosáhneme vytvořením souboru `functions.php` v adresáři aktuální šablony a přidáním tohoto kódu:

```

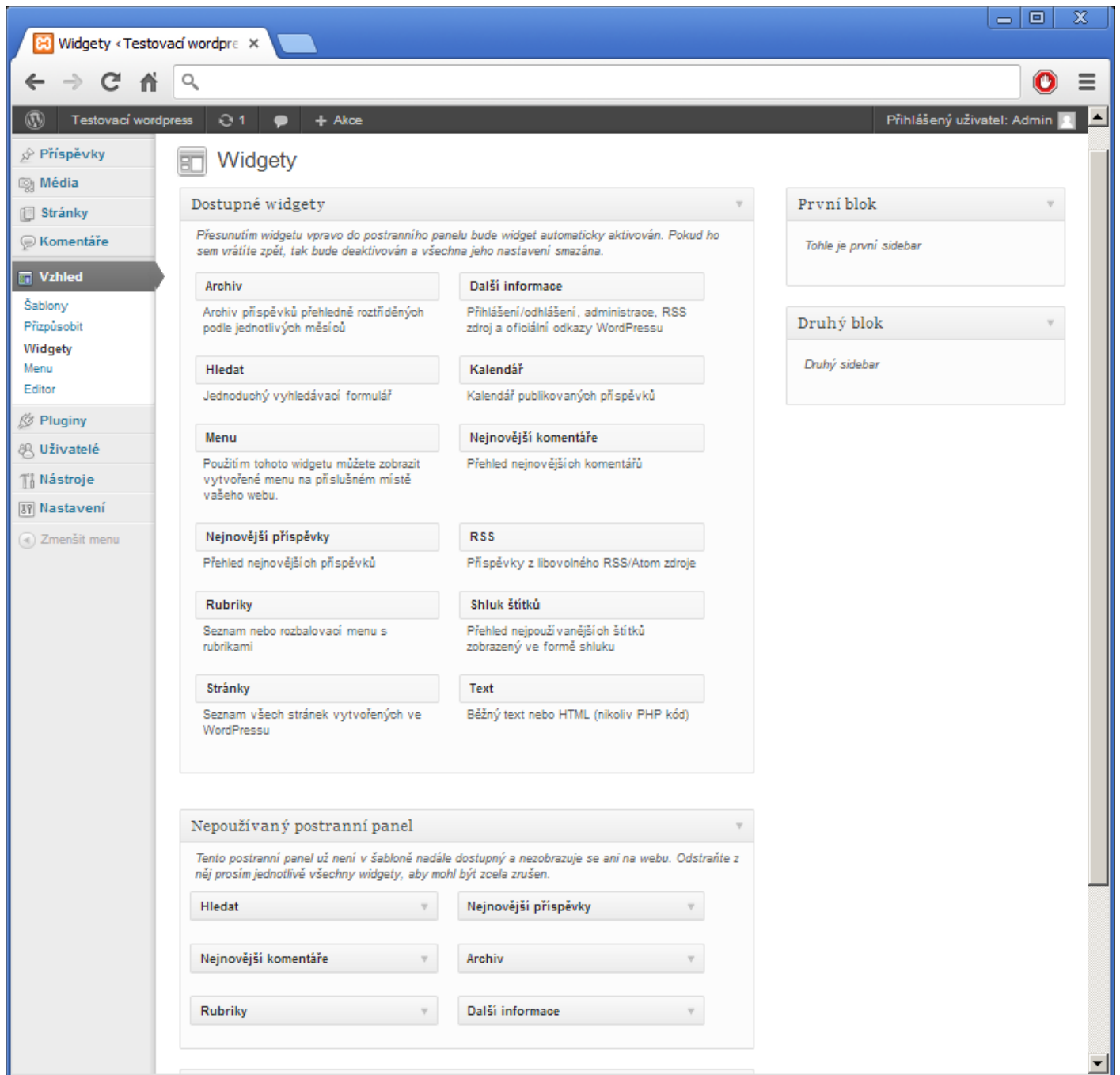
1  <?php
2
3  function my_widgets_init() {
4      register_sidebar( array(
5          'name'         => 'První blok',
6          'id'           => 'sidebar-1',
7          'description' => 'Tohle je první sidebar',
8          'before_widget' => '<aside id="%1$s" class="widget %2$s">',
9          'after_widget'  => '</aside>',
10         'before_title'  => '<h3 class="widget-title">',
11         'after_title'   => '</h3>',
12     ) );
13
14     register_sidebar( array(
15         'name'         => 'Druhý blok',
16         'id'           => 'sidebar-2',
17         'description' => 'Druhý sidebar',
18         'before_widget' => '<aside id="%1$s" class="widget %2$s">',
19         'after_widget'  => '</aside>',
20         'before_title'  => '<h3 class="widget-title">',
21         'after_title'   => '</h3>',
22     ) );
23 }
24 add_action( 'widgets_init', 'my_widgets_init' );
25
26 ?>

```

Kód je rozdělen na dvě části. V první části (řádek 3 až 23) definujeme funkci `my_widgets_init()`. Funkce se může jmenovat libovolně, pokud nebude v konfliktu s již existujícími funkcemi

Wordpressu. To si snadno ověříte na stránkách [codexu](#).

Na **řádku 24** použijeme funkci `add_action()`, kterou provedeme právě nadefinovanou funkci `my_widgets_init()` a tím zpřístupníme widgety v administrátorském rozhraní. Zároveň se nám v administrátorském rozhraní zobrazí vpravo dva bloky, do nichž můžeme přetahovat jednotlivé widgety. Najdeme je pod volbou Vzhled - Widgety.



Teď můžeme do jednotlivých bloků přetáhnout požadované widgety:

Zbývá je zobrazit, což provedeme v souboru `sidebar.php`, který jsme vkládali funkcí `get_sidebar()` do souboru `index.php`. Soubor `sidebar.php` může pak vypadat třeba takto:

```

1 <div class="sidebar">
2 <?php dynamic_sidebar('sidebar-1'); ?>
3 </div>
4
5 <div class="sidebar">
6 <?php dynamic_sidebar('sidebar-2'); ?>
7 </div>

```

Bloky vkládáme pomocí funkce `dynamic_sidebar()`, kde jako parametr uvedeme identifikátor sidebaru (viz [řádky 6 a 16](#) v předchozí ukázce kódu). Výsledkem je zobrazení obou bloků jako samostatných `div`ů ve stránce, přesně jak jsme to určili v souboru `sidebar.php`.

Testovací wordpress

Just another WordPress site

Testovací wordpress

Hello world!

Vložil Admin dne 17. 12. 2013

Welcome to WordPress. This is your first post. Edit or delete it, then start blogging!

Vyhledávání

Vyhledávání

Rubriky

Novinky v knihovně VŠE

- [Naši panovníci a prezidenti : od Sámovy říše po Miloše Zemana / Roman Čilek](#)
- [Trénujte si mozek 2 / Philip J. Carter, Kenneth A. Russell : \[z anglického originálu přeložila](#)
- [Trénujte si mozek / Philip J. Carter, Kenneth A. Russell : \[z anglického originálu přeložila He](#)
- [Hypermoderní doba : od požitku k úzkosti / Gilles Lipovetsky, Sébastien Charles : přeložila Ba](#)
- [Business models for sustainability / Peter E. Wells](#)
- [Zápas o východní Středomoří : zahraniční politika Spojených států amerických vůči Řec](#)
- [Posvícení bezdomovců : úvod do sociologie domova / Jan Keller](#)
- [Česká politika a média po roce 1989 / Petr Žantovský](#)
- [Šedivější hodnoty? : aktivita jako dominantní způsob stárnutí / Marcela Petrová Kafková](#)
- [Zelínář a jeho televize : kultura komunismu po pražském jaru 1968 / Paulína Bren : \[z anglickéh](#)

... obsah patičky ...



Tyto materiály vznikly v rámci projektu CZ.2.17/3.1.00/34129
 Rozvoj oboru Multimédia v ekonomické praxi pro lepší uplatnění absolventů v praxi
 Evropský sociální fond - Praha & EU: Investujeme do vaší budoucnosti