



بزرگترین مرجع تخصصی آموزشی و دانلود در افغانستان

نهایت لذت دانلود را ب ما تجربه کنید!

WWW.P30AF.COM

به نام خدا

چندی قبل در تاریخ سوم دی سال ۱۳۸۷ فرصتی شد تا توفیق این را بیابم که نحوه‌ی ساخت پوسته‌های وردپرس را طی یک سری آموزشی برای شما آماده کنم. هرچند که در ابتدا پیش بینی می‌شد که جلسات این سری ۱۲ یا ۱۴ درس بیشتر نباشند ولی به هر صورت تا ۲۱ جلسه ادامه پیدا کردند. در این جلسات سعی شده تا اندازه‌ی ممکن بتوانیم وبلگ نویسان عزیز را با ساختار وردپرس آشنا کنیم. البته هر که مایل بود، همین سری آموزشی می‌تواند برایش پله‌ای برای حرکت به سوی وردپرس پیشرفت‌های باشد. با توجه به اینکه وردپرس حقی بزرگ به گردن ما دارد، شاید این آموزش کمی از محبت‌های او را نسبت ما جبران کند....

سخن را دراز نمی‌کنم....

امیدوارم که این سری آموزشی مورد توجه شما دوستان وردپرسی قرار گیرد.
پیروز و سر بلند باشید.....

مهدی رادی

۱۳۸۸/۷/۱

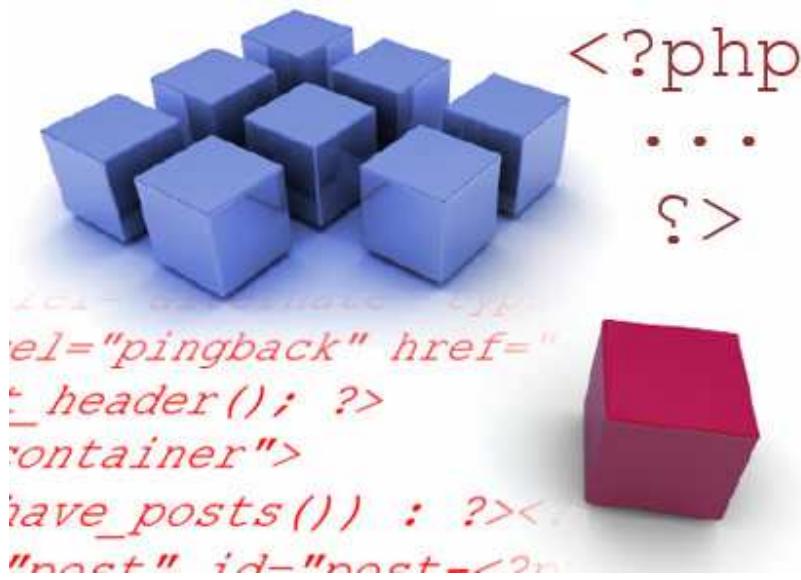


این سری آموزش همانگونه که بروی سایت [کمک رایانه \(p30help\)](#) قرار گرفته، با اندکی تغییر در اینجا نیز نمایش داده شده است. در انتهای هر پست، آدرس منبع مطلب نیز گذاشته شده است تا در صورتی که با مشکلی برخوردید یا سوالی داشتید، به کمک آن آدرس، براحتی بتوانید به صفحه اصلی مطلب دسترسی پیدا کرده و مشکل خود را مطرح کنید....

چگونه برای وردپرس پوسته بسازیم؟

نوشته شده توسط [مهندی رادی](#) در تاریخ ۱۰ / ۰۳ / ۱۳۸۷

مدتی است که قصد داشتم آموزش نحوه‌ی ساخت پوسته برای وردپرس را بدهم، امروز تصمیم خود را عملی کرده و اولین درس از این سری آموزشی را به استحضار بینندگان عزیز می‌رسانم. این سری آموزشی احتمالاً شامل ۱۲ الی ۱۴ درس می‌باشد. مطمئناً با خواندن این سری آموزشی قرار نیست که یک شبه قالب‌های آنچنانی بسازید، اما به شما قول می‌دهم که اگر این سری را به صورت کامل پی‌بگیرید و مطالب گفته شده را آنگونه که من می‌خواهم دریافت کنید، علاوه بر آشنایی با بخش‌های اصلی وردپرس (البته در بخش پوسته‌ها) قادر به رفع بسیاری از مسائل موجود در پوسته تان نیز خواهید بود. البته به غیر از اینکه این سری آموزشی شاید پله‌ای باشد برای حرفه‌ای شدتان. به هر حال، به امید خدا، این سری را شروع می‌کنیم، به امید اینکه مورد استقبال تان قرار گیرد.



در پایان این سری آموزشی شما با کمی تلاش قادر هستید حتی پوسته های بسیار قشنگ و مقبولی را بسازید ، اما باید کمی هم کار کنید و مسلماً از دانش بکار رفته در سایر پوسته ها نیز استفاده کنید . شما قادرید تا در پایان این سری آموزشی قالبی همچون قالب رادیان بسازید (و البته خیلی بهتر از این) . من برای ساخت این قالب از معلوماتی در حد همین سری آموزشی استفاده کرده ام.

در اولین درس از این سری آموزش به آموزش مفاهیم اولیه و قوانین اصلی در کد نویسی را خواهیم پرداخت. مطمئناً اگر کمی php و xhtml کار کرده باشد فهم بسیاری از مطالب برایتان راحت تر خواهد بود ، ولی اگر هیچ گونه آشنایی با این دو زبان ندارید ، حتماً به قوانین گفته شده در این مباحث توجه کنید.

قبل از اینکه درس را به صورت جدی شروع کنیم ، دو کار را باید انجام داده باشید.

۱ - نصب یک سرور خانگی بروی رایانه تان با استفاده از برنامه هایی مانند WinLAMP یا Xampp

۲ - نصب وردپرس بروی سرور خانگی

توجه ۱ : پیشنهاد می کنم که از برنامه Xampp استفاده کنید . برای اینکه بتوانید هر دو مورد بالا را پوشش بدھید حتماً این مطلب را مشاهده کنید.

توجه ۲: برای اینکه از این سری آموزشی عقب نمانید ، در اسرع وقت دو عمل بالا را انجام دهید.

در این سری قرار نیست که همه چیز را به صورت یکجا به شما بیاموزیم ، زیرا با اینکار تنها شما را گیج تر خواهیم کرد. آموزشی که به شما خواهیم داد ، اگرچه بر اساس سورس نیست ، ولی کاملاً پله به پله خواهد بود. در صورتیکه نیازمند یک رفرنس برای آموزش ساخت پوسته وردپرس هستید صفحه‌ی زیر را مشاهده کنید:

[WordPress.org's documentations, Design and Layout](#)

در غیر اینصورت با ما همراه باشید. فقط ۲ نکته‌ی آخر را نیز بگوییم و درس را شروع کنیم . حتماً با این سری آموزشی همراه باشید و هر عملی که انجام دادیم ، شما نیز آنرا انجام دهید. و دیگر من در این سری آموزشی اصطلاحات و کلمات را بیشتر به زبان انگلیسی بکار می‌برم ، زیرا بسیاری از مفاهیم ، با معنی کردن ، هویت خود را از دست می‌دهند و آنچه را که من می‌خواهم را به خواننده القا نمی‌کنند. البته سعی کرده ام که تا حدودی فارسی را نیز پاس بدارم.

از آنجایی که هر درس شامل عکس و متن می‌باشد ، به خاطر طولانی شدن پست به ادامه مطلب مراجعه کنید.

عنوان درس : معرفی مفاهیم

● شروع درس :

در این درس مباحث زیر را تحت پوشش خود قرار خواهیم داد.

- « قوانین اصلی
- « زبان ویژه / اصطلاحات مخصوص
- « سلسه مراتبی بودن

« قوانین اصلی: سه قانون اصلی که باید به یاد بسپارید:

قانون اول : هر تگی را که باز کرده اید ، باید بیندید.

```
<ul>
    <?php wp_register(
        </li>
        <?php wp_login_c
        </li>
        <?php wp_meta(); ?>
</ul>
```

نکته : تگ چیست ؟ هر تگ بوسیله‌ی دو کاراکتر < و > ساخته می‌شود. هر تگی را که زمانی باز کرده ایم باید در سرانجام ببندیم. برای بستن تگ علاوه بر استفاده از کاراکترهای < و > باید از / نیز کمک گرفت. در مثال بالا تگ الا اینگونه <الا> باز شده است و اینگونه <الا/> نیز بسته شده است . اگر دقیق کنید در بین تگ باز و بسته الا یکسری کدهای دیگر آمده است. این کدها چون بین تگ الا قرار دارند بنابراین خاصیت تگ الا را نیز به خود می‌گیرند. در ادامه با نحوه‌ی کار و خاصیت این تگها آشنا خواهید شد.

قانون دوم : تگها نباید بصورت تو در تو در یکدیگر ایجاد شوند.

برای مثال به عکس زیر توجه کنید:

The right way to close:

```
<ul>
  <li>
    </li>
</ul>
```



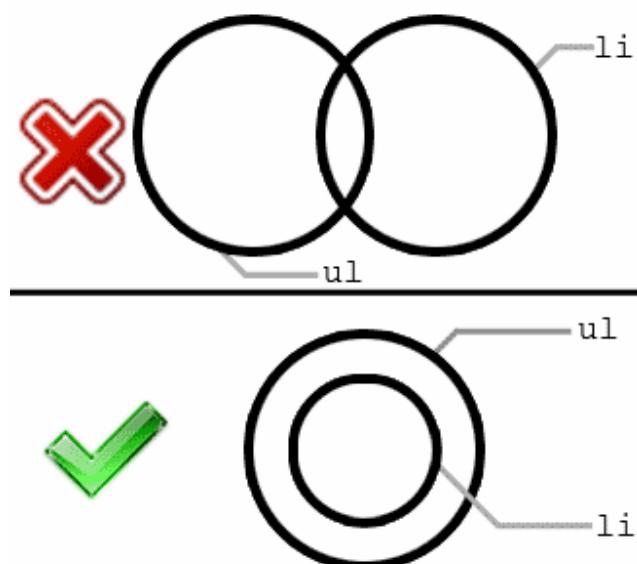
The wrong way to close:

```
<ul>
  <li>
    </ul>
</li>
```



اگر یک تگ باز و بسته را دایرہ ای در نظر بگیریم ، هیچ دایرہ ای نباید همدیگر را قطع کند ، بلکه باید دایرہ ها یا کاملاً جدا از یکدیگر باشند و یا داخل یکدیگر قرار گیرند .

باز هم برای فهم بیشتر به عکس زیر توجه کنید:



قانون سوم : هر پوسته ای حداقل نیازمند دو فایل است : یکی Style.css و دیگری index.php . البته مسلمان شما پوسته هایی را که تا بحال مشاهده کرده اید ، شامل فایلهای بیشتری بوده است.

معمولاً یک پوسته از فایل‌های زیر تشکیل می‌شود.

style.css	•
index.php	•
home.php	•
single.php	•
page.php	•
archive.php	•
category.php	•
search.php	•
404.php	•
comments.php	•
comments-popup.php	•
author.php	•
date.php	•

از لیست بالا ترسیم ، در درس‌های بعدی خواهید فهمید که هر فایل برای چیست و اینکه ترستان بی مورد بوده است.

«**زبان ویژه / اصطلاحات مخصوص (Lingo)**»

مسلماً در هر مبحثی اصطلاحات و کلمات ویژه‌ای وجود دارد ، در زیر با برخی از این کلمات آشنا خواهیم شد:

۱ – Template : یک گروه از کدها که شما می‌توانید از این کدها بدون نوشتن چندباره‌ی آنها در مکان‌های مختلف استفاده کنید.

۲ – Template file : یک فایل که شامل یک یا چند گروه از کدها (Template) می‌شود. هر صفحه‌ی وبی شامل چندین Template file می‌شود ، مانند فایل‌های sidebar.php ، style.css ، index.php

۳ – Theme or WordPress theme : تمام فایل‌هایی که شما استفاده می‌کنید مانند : عکس ، کد ، نوشته و ...

توجه : پوسته‌های وردپرس (WordPress theme) و template وردپرس دو چیز متمایز هستند . با این وجود خیلی از مردم آنها را یک چیز به حساب می‌آورند.

۴ – Post : هم اکنون ، شما در حال خواندن یک post هستید . بعلاوه ، Post ها یک

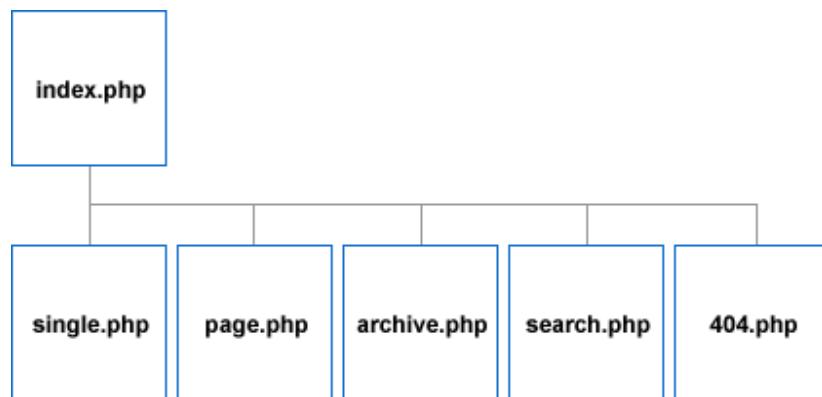
ورودی ساده برای و بلاگ شما و یا هر و بلاگ دیگری هستند . برای مثال یک صفحه از دفترچه‌ی خاطرات شما هم یک Post محسوب می‌شود.

Page – ۵ : یک نوع خاص از Post که در زیر مجموعه‌ی دسته‌ها (Category) نیز قرار نمی‌گیرد. Page ها از گروه Post سایت شما مجزا هستند. در زبان فارسی(بهتر است بگوییم در وردپرس فارسی) Page را همان برگه معنی کرده‌اند.

توجه : در وردپرس Page و page دو چیز مختلف هستند. یک P و p بین آنها تفاوت بزرگی را ایجاد می‌کند ، که در آینده بیشتر با این تفاوت آشنا می‌شویم.

« سلسله مراتبی بودن:

شكل زیر به راحتی به شما نشان می‌دهد که ، سیستم وردپرس به جستجوی چه فایلهایی می‌گردد. البته در عکس زیر بعضی از فایلها آورده نشده‌اند . من تنها ۶ فایل از ۱۳ فایل موجود را برایتان لیست کرده‌ام. با این فایلها در درس‌های آینده بیشتر آشنا خواهیم شد.



سلسله مراتبی بودن مربوط به template file ها می‌شود و اهمیت مراحل و فایلها را برای ما نمایان می‌کند. در عکس بالا فایل index.php دارای بیشترین اهمیت می‌باشد و سپس در ردیف دوم از سمت چپ به راست اهمیت فایلها کاهش می‌یابد. برای مثال در وردپرس فایل archive.php مربوط به تنظیمات صفحات آرشیو می‌شود. حال اگر به هر دلیلی فایل archive.php وجود نداشته باشد، وردپرس این امتیاز را به فایل index.php می‌دهد ، که در این صورت تنظیمات صفحات آرشیو از طرف فایل index.php کنترل می‌شود. به این می‌گویند سلسله مراتبی بودن.

اگر از مجموعه‌ی template file ها فایل single.php از دست برود، کدام فایل جایگزین single.php برای نشان دادن محتوای صفحات تنها خواهد شد؟ درست حدس زدید. باز هم فایل index.php این مسئولیت را تقبل خواهد کرد.

پس اگر اشتباه نکنم باید متوجه شده باشید که فایل index.php دارای بیشترین اهمیت می‌باشد و در صورت نبود هر یک از فایلهای موجود، فایل index.php مسئولیت آنرا قبول می‌کند تا محتویات آن صفحه را نشان دهد.

درس امروز ما به پایان رسید، در جلسه‌ی بعدی با مطالب بیشتری آشنا شده و دست شما به کدها نیز آلوده خواهد گشت.

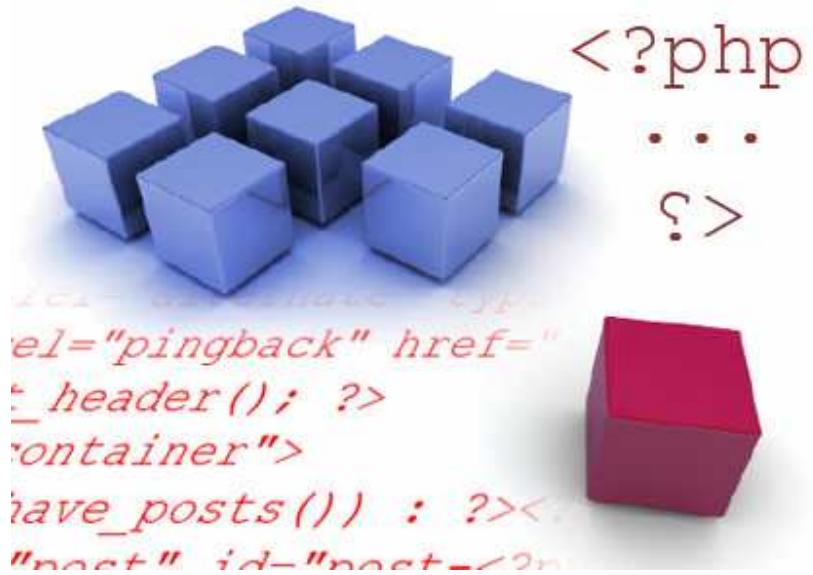
منبع :

<http://www.p30help.ir/how-make-wordpress-theme>

چگونه برای وردپرس پوسته بسازیم؟ (بخش ۲)

نوشته شده توسط مهندی رادی در تاریخ ۱۰ / ۰۶ / ۱۳۸۷

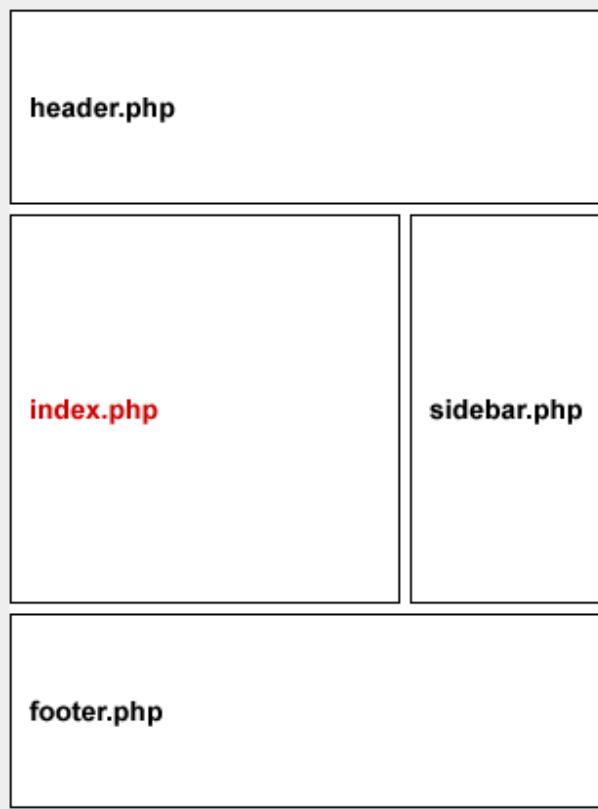
در دومین درس از سری آموزشی ساخت پوسته‌ی وردپرس باز هم به توضیح و معرفی برخی دیگر از مفاهیم می‌پردازیم. و از جلسه‌ی بعد به طور رسمی کار با فایل index.php را شروع خواهیم کرد. اگر شما درس اول از این سری را نخوانده اید، این بخش را ادامه ندهید و در ابتدا درس اول را مطالعه کنید، در غیر این صورت برخی از کلمات و توضیحات گفته شده در این بخش برای شما نا آشنا خواهند بود.



در این درس قصد داریم تا با ساختار **templates** و **files template** در صفحه بیشتر آشنا شویم. درس امروز ما کوتاه است.

بعلت طولانی شدن صفحه ، بقیه ی پست را در ادامه ی مطلب آوردم.

بیاد داشته باشید که هر صفحه ای از وبلاگ شما از چندین **template files** تشکیل شده است ، مانند عکس زیر که یک مثال است:



در مثال بالا ، صفحه‌ی مورد نظر از چهار template files تشکیل شده است که عبارتند از:

footer و sidebar ، index ، header

« و اما Header Template File چیست؟

عنوان وبلاگ شما (*blog's title*) و توضیحات وبلاگ تان (*description*) در این فایل قرار می‌گیرد. معمولاً Header Template File در تمامی صفحات وبلاگ به یک صورت و بدون تغییر باقی می‌ماند.



« حال Index Template File چیست؟

این فایل یکی از مهمترین بخش ها محسوب می گردد و شامل عنوان مطالبتان (*post title*) ، محتوای مطالبتان (*content*) (مانند نوشته و عکس هر *post*) و اطلاعات هر پست (*post meta data*) (مانند اسم نویسنده مطالب ، تاریخ انتشار ، زیر دسته ، تعداد نظرات و ..) می شود.



جیست؟ Sidebar Template File «

توسط این بخش ، برگه ها (*Category*) ، دسته ها (*Page links listing*) ، آرشیو (*archive links listing*) و بلاغاتان کنترل می شود . پیوند ها (*blogroll listing*) در مثال گفته شده سایدبار (*sidebar*) شما در سمت راست قرار گرفته دارد ، در حالی که این تنها یک مثال است ، سایدبار شما هر جایی می تواند باشد .

توجه : ابزارک ها (Widgets) Sidebar Template File (شما نیز توسط کنترل می شوند).

Categories

[Uncategorized](#)

[Tutorials](#)

[Ramblings](#)

[Personal](#)

[News and Updates](#)

[Incoherent Speed Linking](#)

Archives

[February 2007](#)

[January 2007](#)

[December 2006](#)

[November 2006](#)

[October 2006](#)

[September 2006](#)

[August 2006](#)

« و در آخر Footer Template File چیست؟

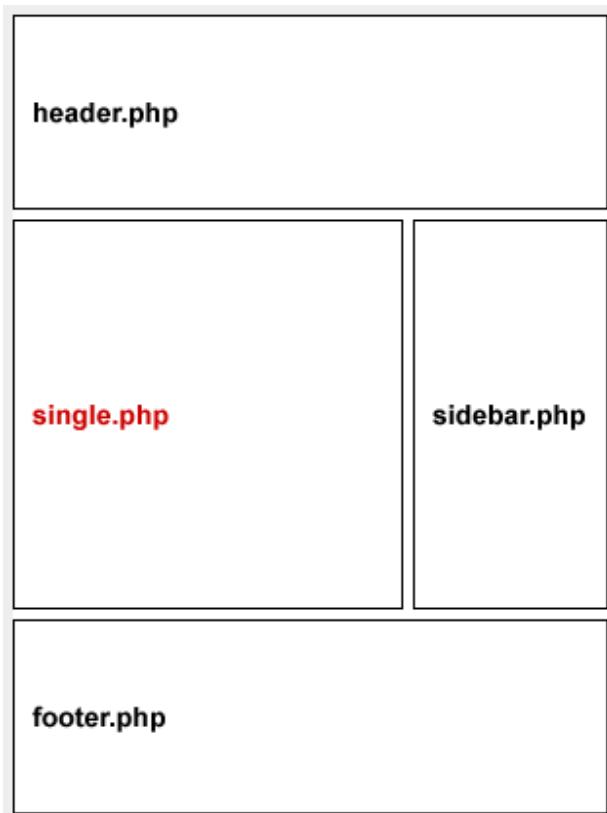
این Header Template File شبیه بخش template file و یا همان header.php می باشد. فوتر (Footer) معمولاً از صفحه ای به صفحه ای دیگر ثابت می ماند . شما هر چیزی را می توانید در این فایل قرار دهید ، اما معمولاً داخل این فایل اطلاعات کپی رایت نوشته می شود.

Copyright 2007 [YourBlog.com](#)

حالا به من اجازه دهید تا توضیح بدhem که چرا فایل index.php در عکس بالا با رنگ **قرمز** مشخص شده است . در مثال بالا index.php با رنگ **قرمز** مشخص شده است تا به شما نشان دهد که این قسمت یک بخش متغیر است و به هر صفحه از وبلاگ شما بستگی دارد. پس باید متوجه شده باشید که بخش‌های Header و Footer و sidebar که با رنگ سیاه مشخص شده اند معمولاً قسمتهای ثابتی هستند.

اگر شما به داخل صفحات تنها سایت خود بروید (single post) چهار files خواهید داشت ، شامل : header , sidebar , single , footer

اما اگر نمی دانید که صفحات تنها (single) چه هستند باید بگوییم ، همان صفحاتی که ادامه‌ی مطلب در آن نوشته می‌شود، برای مثال شما که همین الان در حال خواندن این مطلب هستید در یک صفحه‌ی تنها (single) قرار دارید . در صفحات تنها امکان نظر گذاشتن نیز وجود دارد.



درس کوتاه امروز ما به پایان رسید . [سری آموزشی ساخت پوسته‌ی وردپرس](#) را دنبال کنید.

منبع :

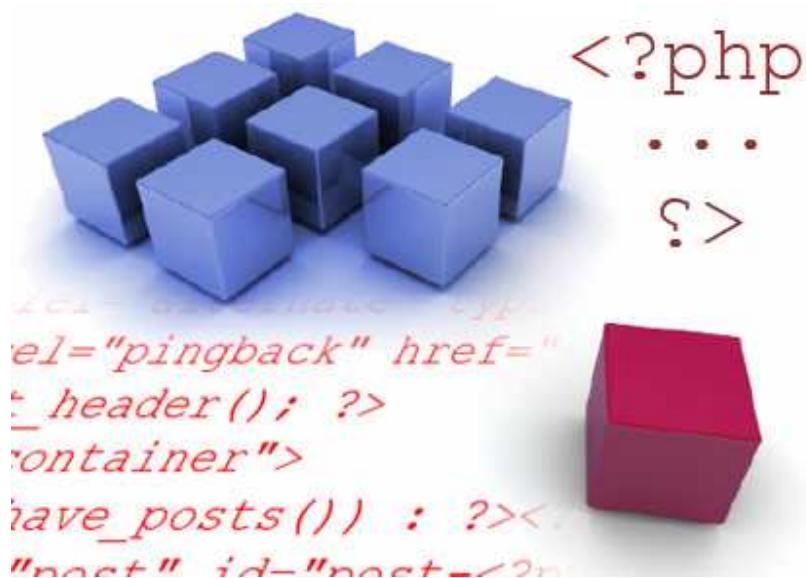
<http://www.p30help.ir/how-make-wordpress-theme-lesson-2>

چگونه برای وردپرس پوسته بسازیم؟ (بخش ۳)

نوشته شده توسط مهندی رادی در تاریخ ۱۱ / ۱۰ / ۱۳۸۷ در دسته HTML, Css, PHP, وبمستر, وردپرس, پوسته

شروع کار با index.php

همانگونه که قول داده بودیم در این جلسه کار با فایل **index.php** را شروع خواهیم کرد که شامل ۳ درس از این سری آموزشی ساخت پوسته‌ی وردپرس نیز خواهد شد. اگر شما درسهای اول و دوم را نخوانده‌اید و تازه کار نیز هستید، برای اینکه سرنخی دست تان بباید، همین الان از مطلوب اول شروع کرده و به جلو حرکت کنید.



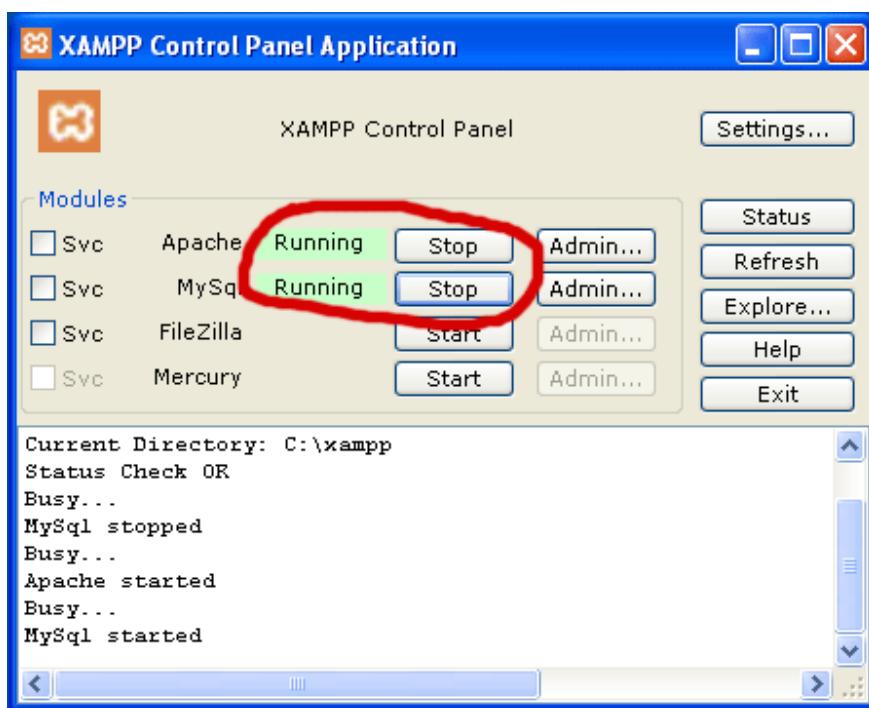
حال وقت آن رسیده است که خواندن محض را رها کرده و بصورت عملی دست به ساخت پوسته‌ی وردپرس بزنید. در این درس، شما دست هایتان به کدهای وردپرس کمی آشنا خواهد شد. قبل از شروع درس باید حتماً یک وبلاگ وردپرس، بروی رایانه تان نصب کنید (نه بصورت آنلاین، بلکه آفلاین زیرا بسیار آسانتر است بویژه برای ما ابرانیان) که در این باره در جلسه‌ی اول توضیح دادیم.

بعلت طولانی شدن صفحه به ادامه مطلب مراجعه کنید.

• مرحله‌ی اول: باز و فعال کردن برنامه‌ی Xampp

به فolder اصلی این برنامه بروید. *xampp < My Computer* (یا هر جای دیگری که این برنامه را نصب کرده‌اید).

بروی **xampp-control.exe** کلیک کنید. پنجره‌ای با حالت pop up باز خواهد شد. آپاچی (Apache) و MySQL را اجرا کنید. برای اینکار شکل برنامه باید مانند تصویر زیر شده باشد:



حال موتور آپاچی و MySQL روشن شده و آماده برای کار هستند. برنامه را کوچک کنید.

• مرحله‌ی دوم: ساخت یک فolder برای پوسته

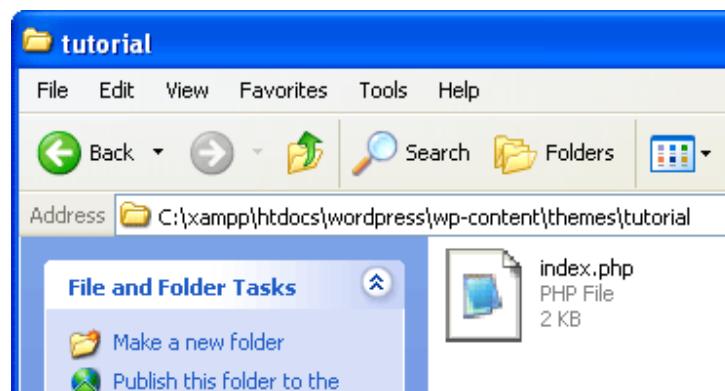
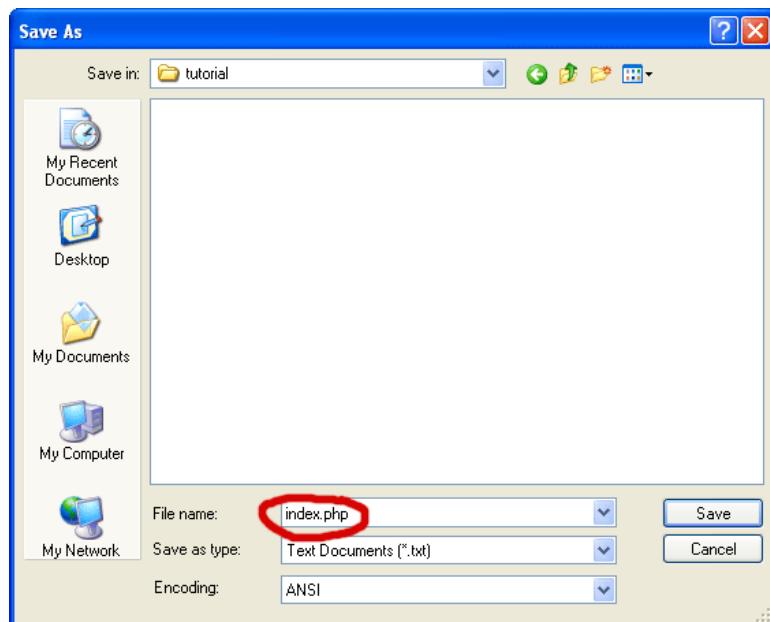
به بخش پوسته‌های وردپرس بروید، یعنی آدرس زیر:
xampp/htdocs/wordpress/wp-content/themes
در این مکان فolderی بنام **tutorial** بسازید. به همین سادگی...

• مرحله‌ی سوم: ساخته فایل **index.php** و **style.css**

برنامه‌ی notepad یا هر ویرایشگر متن مورد علاقه‌ی خود را باز کنید.

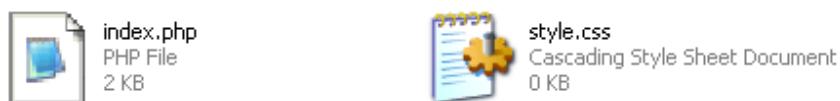
فایل **index.txt** را دانلود کرده و هرآنچه داخل این فایل هست را کپی کرده و در برنامه‌ی notepad بچسبانید (paste کنید).

بعد از اینکار فایل **index.php** را درون پوشه‌ی **tutorial** بناه **index.php** ذخیره کنید.



دوباره‌ی یک notepad جدید باز کنید (باید کاملاً خالی باشد). در همان حالت کاملاً خالی فایل notepad را با نام **Style.css** درون فolder **tutorial** ذخیره کنید. حال پنجره‌های notepad را ببندید.

بعد از این اعمال شما درون این فolder باید فایلهایی با نام‌های **style.css** و **index.php** داشته باشید.



▪ توضیحات : **index.php**

بروی عکس زیر کلیک کنید تا آنرا بصورت کامل و بزرگ مشاهده کنید تا من برای شما قلمرو و ناحیه‌های هر کد را شرح دهم. (برای بزرگ‌دیدن عکس بروی آن کلیک کنید).

```
<!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">
<head profile="http://gmpg.org/xfn/11">
<title><?php bloginfo('name'); ?><?php wp_title(); ?></title>
<meta http-equiv="Content-Type" content="<?php bloginfo('html_type'); ?>; charset=<?php bloginfo('charset'); ?>" />
<meta name="generator" content="WordPress <?php bloginfo('version'); ?>" /> <!-- leave this for s
please -->
<link rel="stylesheet" href="<?php bloginfo('stylesheet_url'); ?>" type="text/css" media="screen" />
<link rel="alternate" type="application/rss+xml" title="RSS 2.0" href="<?php bloginfo('rss2_url'); ?>" />
<link rel="alternate" type="text/xml" title="RSS .92" href="<?php bloginfo('rss_url'); ?>" />
<link rel="alternate" type="application/atom+xml" title="Atom 0.3" href="<?php bloginfo('atom_url'); ?>" />
<link rel="pingback" href="<?php bloginfo('pingback_url'); ?>" />
<?php wp_get_archives('type=monthly&format=link'); ?>
<?php //comments_popup_script(); // off by default ?>
<?php wp_head(); ?>
</head>
<body>
</body>
</html>
```

— این عبارت نشان می‌دهد که شما از چه نوع کدی در پوسته تان Doctype استفاده کرده‌اید. فهم Doctype زیاد اهمیت ندارد. من خودم ویژگیهای را تعیین کردم، بنابراین شما نیازی نیست که در این مورد زیاد فکر کنید.

— این تگ نشان می‌دهد که صفحه‌ی وب شما از کجا شروع می‌شود.

- این تگ به ما نشان می دهد که سر **head**) صفحه‌ی وب ما از کجا شروع می شود. هر صفحه‌ی وب شامل یک **body** و **head** می باشد. </head> تگ به شما نشان می دهد که تگ **head** کجا تمام می شود.

<?php bloginfo('stylesheet_url'); ?> یک تابع (**function**) هست که صدا می زند یا فرا می خواند برای مکانی که فایل style.css در آن قرار دارد (البته در همان فolder که در اینجا فolder می شود) بنابراین پوسته‌ی ما می تواند به آن و یا هر استایل دیگری برای صفحه‌ی وب مان لینک بدهد . هرگاه کدی قرار بگیرد بین عبارت‌های <?php و ?> این قسمت از کد زبان **PHP** محسوب می گردد که با گروه دیگر کدهای من تفاوت دارد . در زبان php عبارت <?php ?> نشان از شروع و عبارت <?> نشان دهنده‌ی پایان کدهای پی اچ پی می باشد.

بنابراین :

<?php > - یعنی شروع کدهای php «
- صدا می زند برای مکانی که **bloginfo('stylesheet_url')** قرار دارد .
« ; - صدا زدن یا فراخواندن برای Style.css را متوقف می کند . سی می کالون(;) یک راه برای بستن گروهی از کدهای **php** هست . بنابراین هرگاه که عبارت ; را مشاهده کردید ، بدانید که یک دستوری از پی اچ پی پایان یافته است.
? > - یعنی پایان کدهای php «

توجه: دقت کنید که هرگاه عبارت ?> بباید یعنی کدهای PHP تمام شده است . در صورتی که اگر ; باید یعنی اینکه یک خط از دستورات PHP پایان یافته است. برای پایان دادن به هر دستور از PHP یک سی می کالون می آید در حالی که ?> برای کل دستورها بکار می رود.

[ادامه بدهید...](#)

<body> - این تگ نشان می دهد که بدنه‌ی اصلی صفحه‌ی وب ما از کجا شروع می شود. بدنه شامل هر چیزی می شود که ما بروی صفحه‌ی وب می بینم و می خوانم . شما باید که در حال خواندن این آموزش هستید یعنی در حال نگاه کردن به بدنه‌ی

صفحه‌ی وب این صفحه هستید . تگ </body> هم نشان دهنده‌ی این است که بدن‌ه صفحه‌ی وب پایان یافته است.(مشخص است دیگر)

«</html> - این تگ نشان دهنده‌ی این است که صفحه‌ی وب من تمام شده است و هیچ چیز دیگری بعد از آن نیست.

● مرحله‌ی چهارم : هر چیزی که داخل فایل **style.txt** هست را به درون فایل save که در فolder tutorial قرار دارد ، کپی و پیست کنید و تغییرات را نموده و صفحه را ببینید.

● مرحله‌ی پنجم : پوسته تان را منتشر کنید برای اینکار یک مرورگر را باز کنید

درون آدرس بار مرورگرتان تایپ کنید عبارت :

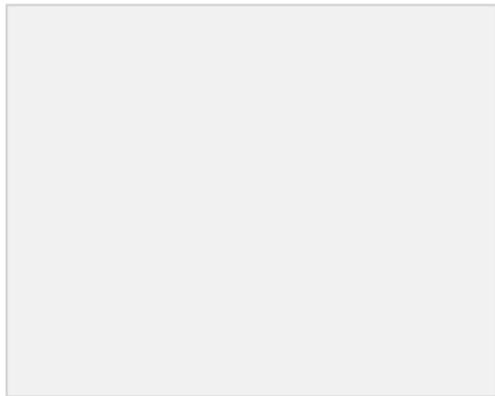
[wp-admin/wordpress,http://localhost](http://localhost/wp-admin/wordpress)

و سپس به داخل بخش مدیریت وردپرس بروید.
که با رنگ قرمز مشخص شده است نشان دهنده‌ی این است که این عبارت فقط یک مثال است و بستگی دارد به نامی که شما برای فolder وردپرس گذاشته اید.)

هنگامی که داخل بخش وردپرس شدید ، به بخش مدیریت پوسته‌ها بروید و پوسته‌ی **Tutorial** را فعال نمایید.

توجه کنید که پوسته‌ی شما هنوز هیچگونه عکس **بند انگشتی (thumbnail)** ندارد.
و باکس آن خالی از عکس می‌باشد . عکس آن به صورت زیر باید باشد:

Tutorial 1.0

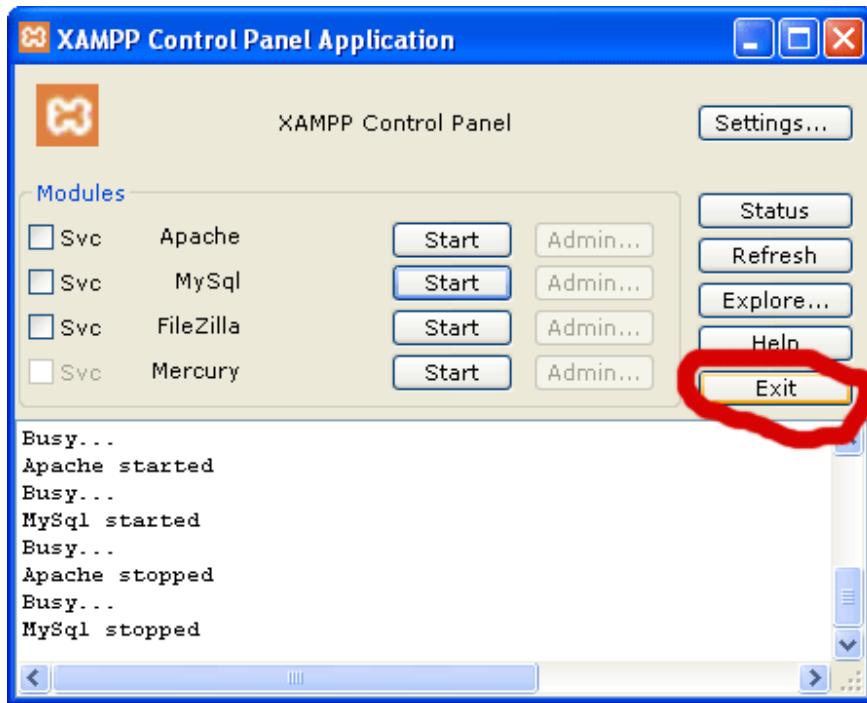


This is my theme for a tutorial.

حال یک صفحه‌ی جدید در مرورگر باز کنید و بروید به آدرس [wordpress/http://localhost](http://localhost) خواهید دید . منظورم یک صفحه‌ی کامل‌خالی است. اگر صفحه‌ی باز شده کامل‌خالی نیست ، شما صفحه‌ی اشتباهی را گشوده اید و یا اشتباهی در کارتان بوده است. توجه : که با رنگ قرمز مشخص شده است نشان دهنده‌ی این است که این عبارت فقط یک مثال است و بستگی دارد به نامی که شما برای فolder وردپرس انتخاب کرده اید.

حال پوسته‌ی شما منتشر شده است. فکر کنم که برای این جلسه کافیست . در جلسه‌ی بعدی ، ما کار با **header template** را شروع خواهیم کرد.

فراموش نکنید که بخش کنترل **Xampp** را ببندید. برای اینکار بروی آیکون آن در بخش System tray دوبار کلیک کنید و **Apache** و **Mysql** را متوقف کنید ، سپس بروی Exit کلیک نمایید.
مانند عکس زیر:



منتظر درس بعدی از این سری باشید.

منبع:

<http://www.p30help.ir/how-make-wordpress-theme-lesson-4>

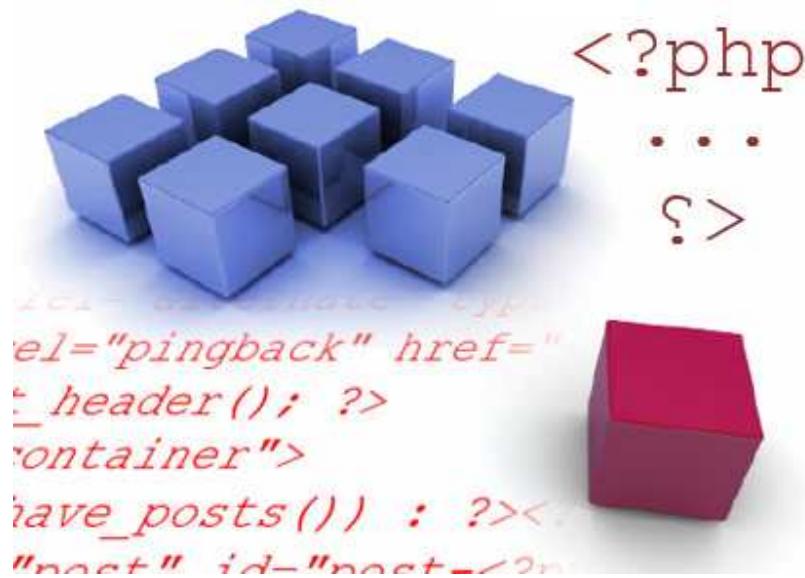
چگونه برای وردپرس پوسته بسازیم؟ (بخش ۴)

نوشته شده توسط مهندی رادی در تاریخ ۱۶ / ۱۰ / ۱۳۸۷

کار با Header Template

اگر با این سایت همراه بوده اید، حتماً مطلع هستید که مدتی پیش سری آموزشی ساخت پوسته‌ی برای وردپرس را شروع کردیم، حال در این پست چهارمین بخش از این سری آموزشی را به شما یاد خواهیم داد. باز لازم است تا این جملات تکراری را به کار ببرم، که اگر شما دروس قبل را مطالعه نکرده اید، همینجا دست نگه داشته و از

طريق [این لینک](#) اولین درس را خوانده و بعد از این، با ما همراه شوید. فرصت زیاد است.



در جلسه‌ی قبل، من به شما نشان دادم که چگونه در بخش کنترل برنامه Xampp، آپاچی و MySQL را فعال کنید. پوسته‌تان را نصب نمایید، و در آخر شما را کمی با PHP آشنا کردیم. امروز ما با PHP ادامه داده و به شما یاد خواهیم داد که چگونه عنوان وبلاگ‌تان (**blog's title**) را صدا بزنید یا فرآخوانی کنید.

بعلت طولانی شدن صفحه برای خواندن پست به ادامه مطلب مراجعه کنید.

نکته مهم: بیاد داشته باشید ، هر کدی را تایپ کنید، و از انجام کپی و پیست پرهیز نمایید. این مسئله باعث می شود تا هر آنچه را که فرا گرفته اید، در خاطرتان بماند.

خوب درس را شروع می کنیم:

• مرحله اول:

از طريق کنترل برنامه‌ی Xampp، آپاچی و MySQL را فعال کنید. سپس، فolder پوسته هایتان را که در آدرس زیر: **xampp/htdocs/wordpress/wp-content/themes/tutorial** قرار

دارند را باز کنید. شما هم اکنون باید ۲ فایلی که جلسه‌ی قبل ساخته اید را مشاهده کنید. فایل‌های **Style.css** و **index.php**



index.php
PHP File
2 KB

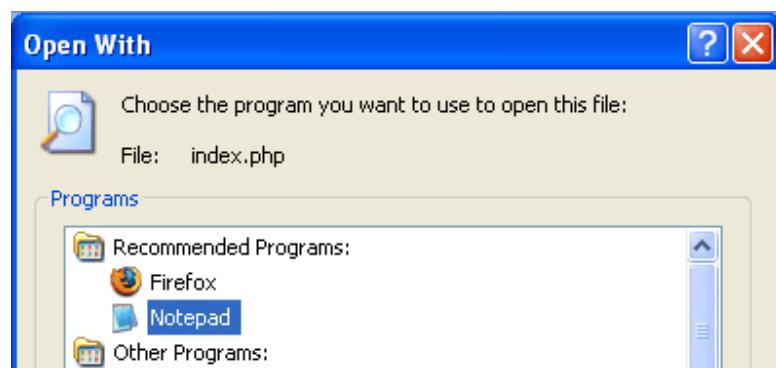
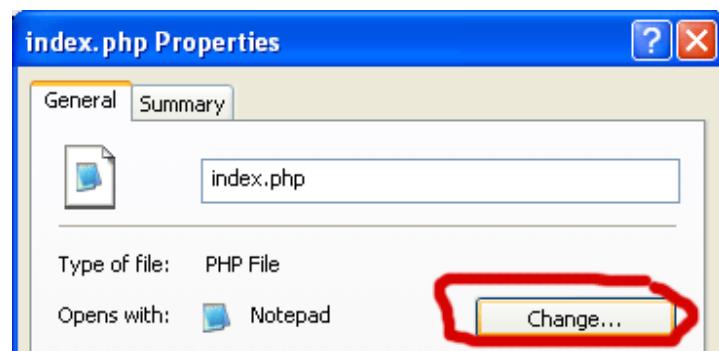
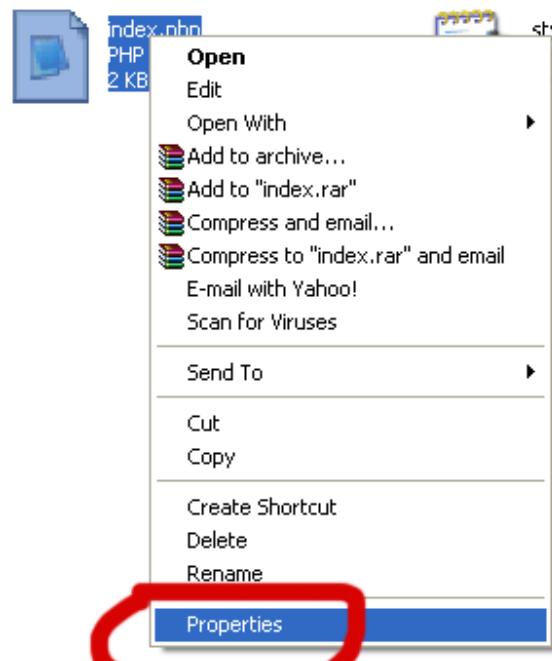


style.css
Cascading Style Sheet Document
0 KB

فایل‌های **style.txt** و **index.txt** باید با فایل‌های **style.css** و **index.php** پر شده باشند (یعنی باید محتویات داخل آنها یکی باشد).

• مرحله دوم:

- یک پنجره از مرورگر وب باز کنید. سپس به آدرس [wordpress/http://localhost](http://localhost) بروید. شما باید یک صفحه‌ی خالی مشاهده کنید، چون شما یک پوسته‌ی خالی را در جلسه‌ی قبل نصب کردید.
- که با رنگ قرمز مشخص شده است نشان دهنده‌ی این است که این عبارت فقط یک مثال است و بستگی دارد به نامی که شما برای فolder وردپرس انتخاب کرده‌اید.
- به فolder پوسته‌هایتان باز گردید و فایل index.php در notepad را انتخاب نموده و بروی اینکار بروی properties راست کلیک کرده، Index.php را انتخاب نموده و بروی change کلیک کنید، سپس notepad یا ویرایشگر متن دلخواه تان را انتخاب کنید. من به شخصه notepad را انتخاب کردم. در آخر نیز بروی Ok کلیک کنید.



تا اینجا، شما، فolder پوسته هایتان، یک مرورگر وب و فایل index.php را باز کرده اید. (notepad)

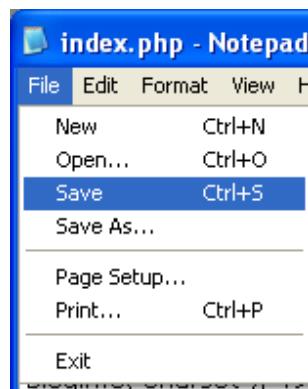


• مرحله‌ی سوم:

شما باید به محتویات داخل فایل index.php دقت کنید. حال برای شروع:

عبارت <body> و <?php bloginfo('name'); ?> را در بین تگ‌های </body> تایپ کنید و تغییرات را Save نمایید.

```
<body>
<?php bloginfo('name'); ?>
</body>
```



سپس بازگردید به صفحه‌ی مرورگر وب‌tan و آن صفحه را Refresh کنید(بوسیله‌ی دکمه‌ی F5 هم می‌توانید اینکار را انجام دهید). شما باید عنوان وبلاگتان را مشاهده کنید. عنوان وبلاگ من این هست : **Development Demo Theme**



❖ حال توضیح و آنچه اتفاق افتاده است؟

شما یک خط از کدهای PHP را درون فایل index.php و درون تگهای body از صفحه ی وب تان اضافه کردید. این یک خط اطلاعات و بلاگتان را صدا می زند. در حقیقت دستور **(bloginfo()** وظیفه ی صدا زدن و یا فراخوانی اطلاعات و بلاگ شما را دارد و از آنجایی که داخل پرانتز مقدار **“name”** را وارد کرده ایم ، پس این دستور صرفاً نام و بلاگتان را صدا می زند.

نکته ۱: نامی که شما بعنوان عنوان و بلاگتان قرار دادید، در بخش مدیریت وردپرس قابل تنظیم و تغییر است.

نکته ۲: چون **(bloginfo()** یک دستور PHP هست بنابراین باید بین عبارت `<?php` و `?>` قرار گیرد.

« **php** - شروع کدهای **<?php** «
« **bloginfo('name')** - اطلاعات و بلاگتان را فراخوانی می کند البته در اینجا عنوان و بلاگتان، فراخوانی می شود.
« ; - سیمی کولون صدا زدن(فراخوانی) اطلاعات و بلاگ را متوقف می کند.
« **php** - پایان کدهای **?>** «

در صورتی که شما هرگونه تغییری در فایل index.php ایجاد کردید، آنرا save کنید و در آخر هم صفحه ی مرورگر وب را refresh کنید تا تغییرات را مشاهده نمایید.

● مرحله چهارم:

اگر دقت کرده باشید عنوان و بلاگتان (**title blog's**) قابل دیدن است، ولی قدرت لینک ندارد، یعنی قادر نیستم که بر روی عنوان و بلاگ کلیک کنیم تا به صفحه ی اول سایت برویم.

برای فعال کردن لینک نوشته عنوان و بلاگ، باید از یک تگ XHTML کمک بگیریم. برای اینکار به فایل index.php بروید.

تگهای **** و **** را به همان خط اضافه کنید. خط جدید شما بایستی به صورت زیر در بیاید:

```
<a href="#"><?php bloginfo( 'name' ) ; ?></a>
```

سپس به مرورگر تان بازگردید و صفحه را refresh کنید. هم اکنون باید امکان لینک را برای عنوان و بلاگتان مشاهده کنید. جالب است نه....



حالا، یک لینک تشکیل داده اید، اما لینکی هست که به هیچ کجا نمی رود، زیرا بجای آدرس درست آن، از # استفاده کرده اید. از این پس اگر بخواهید برای عنوان و بلاگتان، یک لینک بسازید، این روش را بکار بگیرد. یعنی:

عبارت < a href="<?php bloginfo('url'); ?>"><?php bloginfo('name') ; ?> قرار دهید.

تغییرات را save کنید، الان باید خطی مانند زیر را داشته باشید:

```
<a href="<?php bloginfo('url') ; ?>"><?php  
bloginfo('name') ; ?></a>
```

برگردید به مرورگر و صفحه را refresh کنید، وقتی که بروی عنوان سایت بروید، باید به شما آدرس [wordpress/http://localhost](http://localhost) را به شما نشان می دهد(دلیل اینکه Wordpress با رنگ **قرمز** مشخص شده است را دیگر نمی گوییم).



وقتی که بروی لینک مورد نظر ما یعنی همان عنوان و بلاگ کلیک کنید، شما به صفحه اول و بلاگ تان فرستاده خواهید شد.

از آنجایی که در صفحه‌ی اول قرار دارید با کلیک بروی عنوان وبلاگ نیز ، همان صفحه‌ی اول را مشاهده خواهید کرد. اما بدانید که یک تفاوت عظیم بین این دو عبارت # و وجود دارد. شما با این نکات، در درس‌های <http://localhost/wordpress> بعدی بیشتر آشنا خواهید شد.

❖ حال توضیح و آنچه اتفاق افتاده است؟

شما عنوان وبلاگتان را داخل یک لینک فعال کردید و به صفحه‌ی اول تان یا همان صفحه‌ی خانگی وبلاگتان لینک دادید:

« **bloginfo('url')** - اطلاعات وبلاگ شما را فراخوانی می‌کند. مخصوصاً در اینجا که آدرس یا URL صفحه‌ی اصلی تان را فراخوانی می‌کند.

« **<a>** - یک تگ از XHTML هست برای لینکدار کردن یک عبارت مخصوص (عبارتی که بین تگ‌های **<a>** و **** قرار می‌گیرد)

« **** - تگی است که امکان لینک دار شدن یک نوشته را می‌بندد . در صورت عدم استفاده از این تگ ، صفحه‌ی وب شما نمی‌داند که نوشته‌ای که لینکدار کرده اید کجا پایان می‌یابد و در این صورت یک لینک بزرگ خواهیم داشت، یعنی تمام نوشته‌های وبلاگ به یک آدرس لینک دار خواهد شد. **همچنین قانون اول** را بیاد بیاورد، هرچیزی را که باز کرده اید، باید ببندید.

« **href=""** - این عبارت مخفف کلمه‌ی **hypertext value** هست. هر چیزی که در بین علامت نقل قول ("") این دستور بعنوان مقدار قرار بگیرید، آدرس لینک ما خواهد بود.

توجه: شاید از خودتان بپرسید که چرا بجای عبارت **(bloginfo('url'))** ، آدرس دلیلش کاملاً واضح است، بدین خاطر که در صورتی که شما بروی یک وب سایت قرار بگیرد. هنگامی که بروی عنوان وبلاگ آن سایت برویم ، بجای آوردن آدرس صفحه‌ی اصلی آن سایت، آدرس <http://localhost/wordpress> را بخواهد داد.

حال از این توضیحات بگذریم، خط کد شما به صورت زیر باید باشد:

```
<a href="<?php bloginfo('url'); ?>"><?php  
bloginfo('name'); ?></a>
```

یعنی: من با آوردن تگ **a** امکان لینک را آغاز کردم. سپس آدرس لینک را صفحه‌ی اصلی و بلاگم قرار دادم. برای اینکار من از تابع PHP یعنی **bloginfo('url')** استفاده کردم، تا آدرس و یا URL و بلاگم را صدا بزنم. در آخر هم اسم لینک را با آوردن **?php bloginfo('name');** همان عنوان و بلاگم قرار دادم. تگ لینک را نیز با عبارت **** بستم.

شما امروز کمی با XHTML نیز آشنا شدید. خوب فکر کنم، برای درس امروزمان کافی باشد. اگر نگرانیید که چرا اینقدر کند پیش می‌رویم باید بگوییم که شما در ابتدای کار هستید و هنگامی که بیشتر آشنا شدید، سرعت درس مان نیز بیشتر خواهد شد.

با این سری آموزشی در جلسات بعد همراه باشید.

منبع:

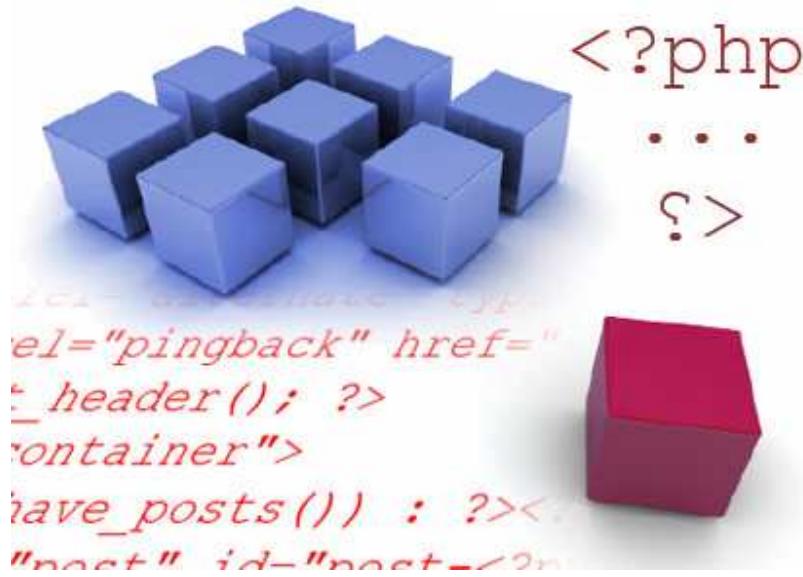
<http://www.p30help.ir/how-make-wordpress-theme-lesson-4>

چگونه برای وردپرس پوسته بسازیم؟ (بخش ۵)

نوشته شده توسط مهندی رادی در تاریخ ۲۵ / ۱۰ / ۱۳۸۷

کار با Header Template (بخش ۲)

در این پست بخش پنجم از این سری آموزشی را پی‌خواهیم گرفت. این درس ادامه‌ی درس قبلی است. باز هم می‌گوییم که اگر این سری را از ابتدا دنبال نکرده‌اید، از همین الان از طریق این پست درس اول و تمامی دروس قبلی را مطالعه کرده و سپس با ما همراه شوید. در ضمن از خودتان مطمئن شوید که دروس قبلی را خوانده و فهمیده‌اید (اگر کمی نیز کار کرده باشید که دیگر عالی است)



درس امروز ما آنچنان طولانی نیست ولی با بعضی از مهمترین تگ های بکار رفته در پوسته های وردپرس آشنا خواهید شد. با پایان پست امروز بخش **header** و **template** به پایان می رسد، و در جلسات بعدی با بدنه اصلی و ساختار حلقه بیشتر آشنا خواهیم شد . اگر آماده اید درس را شروع کنیم....

از آنجایی که صفحه کمی طولانی می شود ، متن کامل پست را در ادامه مطلب مطالعه فرمایید.

نکته: به هیچ عنوان کدهایی که به شما گفته می شود را کپی و Paste نکنید و خودتان آنرا را **تایپ** کنید.

● مرحله‌ی اول:

- ۱- برنامه‌ی Xampp را فعال کنید.
- ۲- فolder پوسته هایتان ، که نامش را Tutorial گذاشته اید را باز کنید
- ۳- یک مرور گر وب باز کنید و در آن آدرس [wordpress/http://localhost](http://localhost) را تایپ کنید.(دلیل قرمز بودن **wordpress** را در جلسات گذشته گفته ایم).
- ۴- به فolder پوسته بازگردید و فایل **index.php** را در **notepad** باز کنید.

بعد از انجام مقدمات کار مرحله‌ی بعدی و یا اصل درس را شروع می کنیم:

● مرحله دوم :

درون فایل **index.php** باید عبارت زیر را که از جلسه‌ی قبل کار کرده‌ایم، وجود داشته باشد:

```
<a href="<?php bloginfo('url'); ?>><?php bloginfo('name'); ?></a>
```

حال به دور عبارت بالا تگ‌های **</h1>** و **<h1>** را اضافه کنید. تگ **H1** به معنی **H4, H5, ,H1, H2, H3** heading بطور کلی وجود دارند: **heading 1** **H6**. بطور پیش فرض، **H1** **بزرگترین سایز** و **H6** **کوچکترین سایز** می‌باشد.

نکته: اگر بخواهیم به زبان ساده تگ **H1** را توصیف کنم، باید بگوییم که در واقع تگ **H1**، یک سایز پیش فرض از فونت می‌باشد. یعنی اگر این تگ بدور یک نوشته‌ی ما قرار گیرد، همانگونه که در بالا گفته شد، نوشته‌ی ما بزرگترین سایز پیش فرض را می‌گیرد و اگر تگ **H6** قرار بگیرد، نوشته‌ی ما کوچکترین سایز پیش فرض را به خود می‌گیرد. دلیل اینکه می‌گوییم سایز پیش فرض این است که می‌توانیم در فایل **Style.css** این تنظیمات را تغییر داده و اندازه را با توجه به خواسته‌ی خودمان تغییر دهیم. که البته در جلسات بعد با اینگونه اعمال بیشتر آشنا می‌شویم.

بعد از توضیحات گفته شده باید الان کد شما بصورت زیر باشد:

```
<h1><a href="<?php bloginfo('url'); ?>><?php bloginfo('name'); ?></a></h1>
```

تغییرات را save کرده و به مرورگر برگردید و آن را refresh کنید. مسلماً عنوان وبلاگ (**blog's title**) شما بزرگتر شده است. جالب است نه....

● مرحله سوم:

در این مرحله قصد داریم تا توضیحات وبلاگمان (**description**) را صدا بزنیم و یا فراخوانی کنیم، برای اینکار عبارت **<?php bloginfo('description'); ?>** را درست در زیر عبارت عنوان وبلاگ (**blog's title**) تایپ کنید، حال کد شما باید بصورت زیر باشد:

```
<h1><a href="<?php bloginfo('url'); ?>"><?php bloginfo('name'); ?></a></h1>
<?php bloginfo('description'); ?>
```

تغییرات را ثبت کرده و صفحه را refresh کنید، مشاهده خواهید کرد که توضیح و بلاگتان در زیر لینک عنوان و بلاگتان (**blog's title**) اضافه شده است. این نکته را نیز بگوییم که شما می‌توانید توضیحات و بلاگتان (**description**) را در بخش مدیریت وردپرس تنظیم کنید.

❖ توضیح آنچه اتفاق افتاده است:

– شروع کدهای **php** «
– اطلاعات و بلاگتان را (بویژه در اینجا **توضیحات** و بلاگتان (**description**) را) صدا می‌زند. (به دلیل اینکه مقدار ("") را قرار داده ایم، فقط **توضیحات و بلاگtan** (**description**) فراخوانی می‌شود).
– صدا زدن **توضیحات و بلاگ** (**description**) را متوقف می‌کند.
– پایان کدهای **php** »
»

● مرحله‌ی چهار:

در این مرحله برای شما یک تگ جدید و البته بسیار مهم را به نام **div** را معرفی می‌کنیم:

بدور کدهایی که تاکنون نوشته اید تگ‌های **<div>** و **</div>** را تایپ کنید، در نتیجه خواهیم داشت:

```
<div>
<h1><a href="?php bloginfo('url'); ?>"><?php bloginfo('name'); ?></a></h1>
<?php bloginfo('description'); ?>
</div>
```

تغییرات را ثبت کرده و صفحه را دوباره refresh کنید. نباید هیچ تغییری را مشاهده کنید.

شاید خیال کنید که تگ **div** یک تگ بلا مصرف است، اما در جلسات بعد علاوه بر اینکه بر روح سازنده‌ی آن درود خواهید فرستاد، کاربرد آنرا نیز بیشتر خواهید فهمید. ولی به طور اجمالی در اینجا در مورد این تگ توضیحاتی خواهیم داد: ما این تگ را قرار دادیم تا لینک عنوان و بلاگ (**blog's title**) و توضیحات و بلاگ (**description**) را از هر چیز دیگر موجود در صفحه جدا کنیم، حال اگر شما از این تگ استفاده نکنید. هیچ محتوایی از پوسته‌ی شما مجزا نخواهد بود و در حقیقت همه چیز تو در تو خواهد بود. شما با قرار دادن تگ **div** به دور کدهای بالا در حقیقت یک باکس مخفی بدور آن ایجاد کرده اید که علاوه بر جدا کردن محتوای داخل آن از سایر عناصر موجود در صفحه، می‌توانید بعد‌ها در فایل **style.css** برای باکس‌های مخفی (تگ **div**) ویژگی تعریف کنید و در حقیقت سایت تان را بسازید. کارهایی نظیر حاشیه دادن، لایه‌گذاری، لبه گذاشتن، عکس پس زمینه، رنگ پس زمینه و غیره.. را می‌توانید بروی این تگها اعمال کنید.

ممکن است این سوال برای شما پیش بیاید که با توجه به اینکه تگ **baekس مخفی** یا همان **div** را برای بسیاری از اجزای صفحه بکار می‌بریم، چگونه می‌توان بعدها در فایل **Style.css** یک تگ **div** مخصوص را مورد تغییر قرار داد. جواب شما در مرحله‌ی پنجم قرار دارد...

• مرحله‌ی پنجم:

مقدار "header" را به تگ **div** اضافه کنید مانند زیر:

```
<div id="header">
```

تغییرات را ثبت کرده و صفحه را refresh کنید.

البته الان، باز هم هیچ تغییر را مشاهده نخواهید کرد. ما یک **id** را به تگ **div** اضافه کردیم. دلیل اینکه ما به این تگ **id** یک **div** اضافه کردیم این است که بتوانیم بعداً در فایل **Style.css** تگی با مقدار آی دی **header** را مورد تغییر قرار دهیم، زیرا

تگ های **div** زیادی وجود خواهند داشت و شما باید به گونه ای آنها را از یکدیگر متمایز کنید.

نکته: اگر از این به بعد واژه **i** باکس مخفی را بکار بردیم بدانید که منظورمان همین تگ **div** است.

به گمانم درس امروز کوتاه بود، ولی با تگ بسیار مهمی آشنا شدید. اگر حرف من را باور نمی کنید، فایل **index.php** یک پوسته (نه پوسته ای که می خواهیم بسازیم) را گشوده و مشاهده کنید که این تگ چقدر زیاد بکار رفته است.

با سری آموزشی ساخت پوسته **i** وردپرس در جلسات بعد همراه باشید.

منبع:

<http://www.p30help.ir/how-make-wordpress-theme-4b-lesson-5>

چگونه برای وردپرس پوسته بسازیم؟ (بخش ۶)

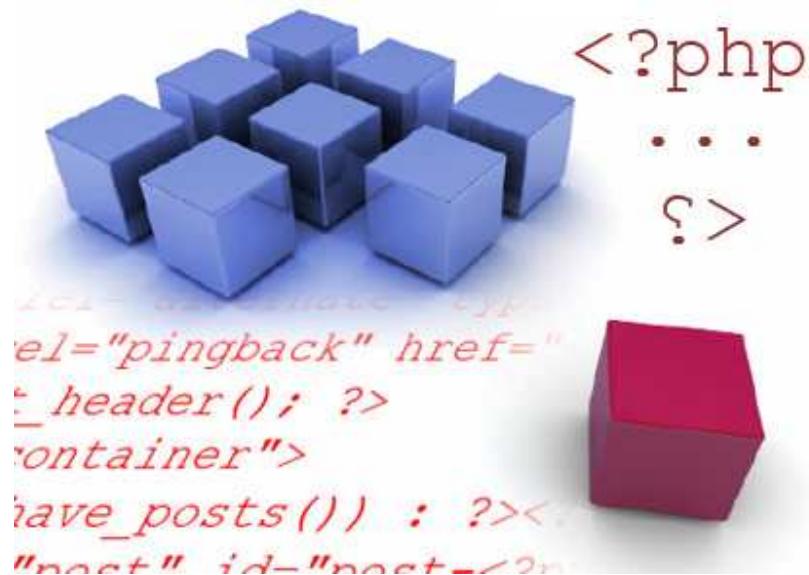
نوشته شده توسط مهندی رادی در تاریخ ۰۶ / ۱۱ / ۱۳۸۷ در دسته HTML, PHP, وبمستر, وردپرس, پوسته **حلقه ها (Loop)**

در این درس به آموزش نحوه **i** آوردن بخش ورودی های و بلاگ یا همان پست های و بلاگ تان خواهیم پرداخت. دوباره مجبورم قبل از شروع درس، این نکته را بگوییم که اگر این سری آموزشی وردپرس را از ابتدا دنبال نکرده اید، حتما از طریق [این لینک](#) این کار را انجام دهید....

امیدوارم که این کار را کرده باشید. بگذریم.....

بیشتر بدانید: اگر کمی با PHP کار کرده باشید مطمئناً با حلقة ها آشنا هستید. حلقة ها یکی از اجزای مهم در برنامه نویسی می باشند. بوسیله **i** حلقة ها قادر هستید تا یک فرمان را به اندازه ای که مایلید، تکرار کنید. روش های مختلفی برای ایجاد حلقة وجود دارد، که البته در بحث امروز ما نمی گنجد. ما در این درس به کمک حلقة ها ورودی

های سایت یا پست های سایت را وارد صفحه کنیم، اگر کمی فکر کنید، متوجه خواهید شد که حلقه ای که استفاده خواهیم کرد برای کجا و چه استفاده خواهد شد.



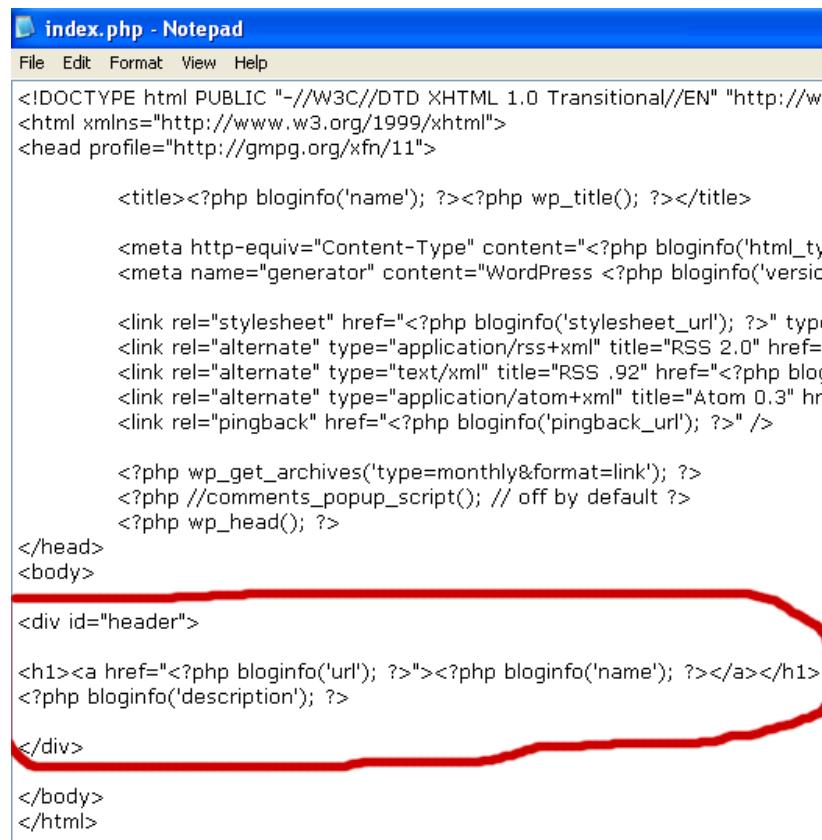
تا اینجای کار، ما به شما موارد زیر را درس داده ایم:

- قوانین ، ساختار ، و سلسله مراتب در پوسته های وردپرس.
- هر صفحه ای از چه چیز تشکیل شده است.
- چگونه پوسته ی را نصب کنیم.
- چگونه عنوان و بلاگمان را صدا بزنیم و سپس از آن یک لینک بسازیم.
- چگونه توضیحات و بلاگ را صدا بزنیم و header را از اجزای دیگر مجزا کنیم.

چون صفحه طولانی می شود برای شروع درس و آشنایی بیشتر با حلقه ها به ادامه مطلب مراجعه کنید...

همانطور که می دانید ، باید ابتدا برنامه **Xampp** را راه اندازی کنید. فolder **tutorial** را در بخش پوسته های وردپرس بگشایید، یک مرورگر وب باز کنید و آدرس **index.php** را در آن تایپ کنید، و در آخر فایل **wordpress** <http://localhost> در **notepad** هم باز کنید.

آنچه در جلسات قبل انجام داده ایم را می توانید در عکس زیر مشاهده کنید.



```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/1999/xhtml">
<html xmlns="http://www.w3.org/1999/xhtml">
<head profile="http://gmpg.org/xfn/11">
    <title><?php bloginfo('name'); ?><?php wp_title(); ?></title>
    <meta http-equiv="Content-Type" content="<?php bloginfo('html_ty
    <meta name="generator" content="WordPress <?php bloginfo('versio
    <link rel="stylesheet" href="<?php bloginfo('stylesheet_url'); ?>" type="text/css" media="all" />
    <link rel="alternate" type="application/rss+xml" title="RSS 2.0" href="<?php bloginfo('rss2_url'); ?>" />
    <link rel="alternate" type="text/xml" title="RSS .92" href="<?php bloginfo('rss_url'); ?>" />
    <link rel="alternate" type="application/atom+xml" title="Atom 0.3" href="<?php bloginfo('atom_url'); ?>" />
    <link rel="pingback" href="<?php bloginfo('pingback_url'); ?>" />
    <?php wp_get_archives('type=monthly&format=link'); ?>
    <?php //comments_popup_script(); // off by default ?>
    <?php wp_head(); ?>
</head>
<body>
    <div id="header">
        <h1><a href="<?php bloginfo('url'); ?>"><?php bloginfo('name'); ?></a></h1>
        <?php bloginfo('description'); ?>
    </div>
</body>
</html>
```

بیاد داشته باشید، هر کدی را **تایپ** کنید و هیچ چیز را کپی نکنید.

• مرحله‌ی اول:

یک تگ **div** (باکس پنهان) را زیر ناحیه **header** اضافه کنید و به آن مقدار "container" را نسبت دهید، مانند زیر:

```
<div id="container">
</div>
```

این **div** برای چه هست؟ این تگ برای این هست که محتوای اصلی و یا همان پست های سایت را از هر چیز دیگری، که بعد از آن می‌آید، جدا کند. چیزهایی مانند Sidebar و Footer

• مرحله‌ی دوم:

کد زیر را بین تگ باکس مخفی ای که مقدار id آن Container هست، اضافه کنید. کدهای زیر را بطور دقیق وارد کنید. اگر هرگونه ایرادی را مشاهده کردید، دوباره کدهایتان را چک کنید:

```
<?php if(have_posts()) : ?><?php while(have_posts()) : the_post();  
    ?>  
  
        <?php endwhile; ?>  
  
    <?php endif; ?>
```

از این مرحله، مفاهیم برایتان آساتر خواهد بود، تا متوجه شوید که کد نوشته شده چه اعمالی را انجام خواهند داد.

آنچه که شما تایپ کردید، یک حلقه (**The Loop**) در وردپرس هست. قبل از اینکه این کدها را شرح بدهم. شما باید چیزی شبیه به عکس زیر داشته باشید:

```
<body>
<div id="header">
<h1><a href=<?php bloginfo('url'); ?>"><?php bloginfo('name'); ?></a></h1>
<?php bloginfo('description'); ?>
</div>
<div id="container">
<?php if(have_posts()) : ?><?php while(have_posts()) : the_post(); ?>
<?php endwhile; ?>
<?php endif; ?>
</div>
</body>
</html>
```

توجه کنید که خطوط درون باکس مخفی یا همان **div** را تو رفته تر نسبت به خطهای دیگر تایپ کرده ام. این کار را برای سازماندهی و تشخض دادن بهتر انجام دادم.
اینکه شما فقط کدها را تایپ کنید، کافی نیست، شما باید این کدها را مدیریت کنید تا بتوانید آنها را آسانتر تشخیص بدھید. با اینکار بعدها، شما می دانید که جای هر کدی در کجاست و در کجا شروع و تمام می شود و کدهای مربوطه، در کدام قسمت قرار

دارند. من برای جلو آوردن هر کدی از دکمه **TAB** بجای **Spacebar** استفاده کرم. بعدها که صفحه‌ی شما پر از کد شد، قدر اینکار را خواهید دانست.

◆ چه چیز اتفاق افتاد؟

یک عبارت شرطی است و چک می‌کند که شما آیا پستی دارید یا خیر؟ در حقیقت این کد را می‌توان اینگونه خواند: «اگر پستی وجود داشت اجرا کن...»

این کد همان حلقه‌ی مورد نظر ماست و معنی آن می‌شود: «تا زمانی که پستی وجود داشت، **the_post()** را اجرا کن.»

پست‌ها را برای نشان دادن صدا می‌زند.

همان قانون اول است - این عبارت حلقه‌ی **while()** را می‌بندد.

شرط **(if)** را می‌بندد. (قانون اول)

توجه: لازم نیست تمام کدهایی را که بازکرده‌ام، حتماً در آخر ببندیم بعضی از کدها از دو بخش ساخته نشده اند. برخی کدها می‌توانند خودشان، خودشان را ببندند، مثل: **the_post();** یا **have_posts()**. کمی که با این مباحث بیشتر آشنا شوید، به راحتی می‌توانید بفهمید که چه کدی نیاز به بسته شدن دارد، و چه کدی نیازمند این عمل نیست.

● مرحله سوم:

در درس قبلی، شما یاد گرفتید که چطور عنوان و بلاگتان (**blog's title**) را بوسیله‌ی **bloginfo('name')** صدا بزنید. حالا، باید برایتان آسانتر باشد که چگونه عنوان پست **post titles** (The Loop) را صدا بزند. برای اینکار در بین حلقه‌ها را بعد از **<?php the_post(); ?>** و قبل از **?> <?php the_title(); ?>** تایپ کنید. مانند زیر:

```

<body>

<div id="header">

<h1><a href=<?php bloginfo('url'); ?>"><
<?php bloginfo('description'); ?>

</div>

<div id="container">

<?php if(have_posts()) : ?><?p

<?php the_title(); ?>

<?php endwhile; ?>

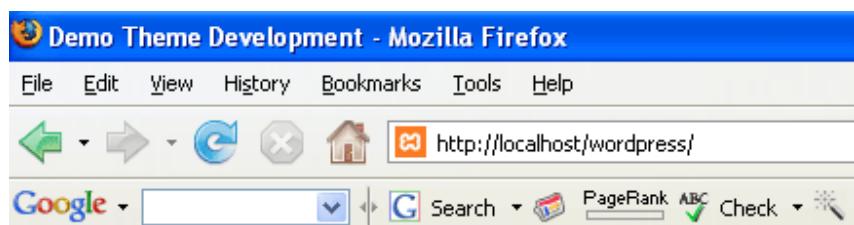
<?php endif; ?>

</div>

</body>
</html>

```

تغییرات را در فایل **index.php** ثبت کنید، مرورگر ترا **refresh** نمایید. شما باید عنوان **Hello World** را در زیر توضیحات و بلاگتان مشاهده کنید. البته در حالت پیش فرض، و بلاگ شما تنها یک پست دارد. البته شما می توانید چندین پست را خودتان به وردپرس اضافه کنید، تا چند عنوان مختلف را مشاهده کنید. برای مثال من خودم چندین پست به وردپرس اضافه کرده ام و چون عنوان پست ها را همگی **Hello world** گذاشته ام، شما یک خط طولانی شامل عبارت های **Hello world** را مشاهده خواهید کرد.



Demo Theme Development

Just another WordPress weblog

Hello World Hello World Hello World Hello World Hello
eiusmod tempor

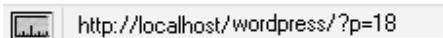
• مرحله چهارم:

الان عنوان پستهای شما قابل رویت است ولی امکان لینک ندارند. برای فعال کردن آدرس و یا لینک عنوان پست ها باید..... بیاد بیاورید که چگونه لینک را برای عنوان وبلاگ، فعال کردیم. دقیقاً مانند همان روش، تگ **< a href="#" >** و **** را بدور کد **< ?php the_title(); ?>** اضافه می کنیم

فایل index.php را save کنید و سپس صفحه را refresh نمایید. حالا عنوان پست های شما همگی حالت لینک دارند، اما لینک ها به هیچ جای خاصی نمی روند. برای اینکه برای هر پست آدرس درستی را به هر لینک بدهیم، باید عبارت **the_permalink()** را بجای # قرار دهیم. مانند عکس زیر:

```
<a href="< ?php the_permalink(); ?>">< ?php the_title(); ?></a>
```

یک تابع از php **the_permalink()** است، که آدرس یا مکان هر پست را صدا می زند. صفحه را save کنید و سپس صفحه‌ی مرورگر را refresh نمایید . نشانگر موس را بروی یکی از عنوانهای Hello World برد و به **Stutus bar** نگاه کنید. خواهید دید که در آدرس bar مطلب شما را نشان داده خواهد شد، مانند عکس زیر:



اگر شما بیشتر از یک پست داشته باشید، خواهید دید که هر لینکی به یک پست مجزا یا صفحه‌ی مجزا می رود. اما، عنوان های پست ما، در یک خط هستند، برای مجزا کردن آنها، تگهای **< h2 >** و **< /h2 >** بدور کد لینکتان اضافه کنید. یعنی:

```
< h2 >< a href="< ?php the_permalink(); ?>">< ?php the_title(); ?></a>< /h2 >
```

بیاد بیاورید **H1** را که ما برای عنوان وبلاگمان استفاده کردیم. این صفحه‌ی heading وب شما هست. حالا لینک های مطالب شما sub-headings هستند. الان باید تمام لینک پست های شما در خطوط خودشان باشند. فایل index.php را save کرده و مرورگر را refresh کنید تا تغییرات برایتان نمایان شود.

Demo Theme Development

Just another WordPress weblog

Hello World

Hello World

در پایان باید صفحه‌ی کد شما به شکل زیر باشد:

```
<body>
<div id="header">
<h1><a href="php bloginfo('url'); ?&gt;"&gt;&lt;?php bloginfo('name'); ?&gt;&lt;/a&gt;&lt;/h1&gt;
&lt;?php bloginfo('description'); ?&gt;
&lt;/div&gt;
&lt;div id="container"&gt;
&lt;?php if(have_posts()) : ?&gt;&lt;?php while(have_posts()) : the_post(); ?&gt;
    &lt;h2&gt;&lt;a href="<?php the_permalink(); ?&gt;"&gt;&lt;?php the_title(); ?&gt;&lt;/a&gt;&lt;/h2&gt;
    &lt;?php endwhile; ?&gt;
    &lt;?php endif; ?&gt;
&lt;/div&gt;
&lt;/body&gt;
&lt;/html&gt;</pre
```

خوب درس ما به پایان رسید.

توجه: اگر شما خوب منظور من را متوجه نشده‌اید. باید چندین پست مختلف به وردپرس تان اضافه کنید(مهم نیست که محتوای آن چه باشد) تا پوسته تان به درستی آزمایش شود. برای اینکار به بخش مدیریت وردپرس خود رفته و در زیر مجموعه‌ی نوشتن، چندین پست اضافه کنید. پیشنهاد می‌کنم تا عنوان هر پست را منحصر به فرد گذاشته تا کمتر دچار سردرگمی شوید.

با آموزش‌های بعدی از سری آموزشی ساخت پوسته‌ی وردپرس با ما همراه باشید. تا بعد.....

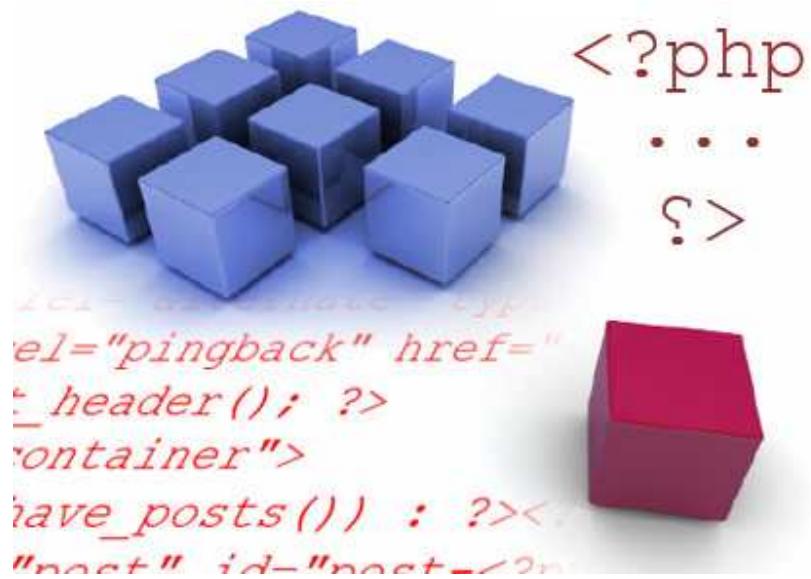
منبع:

چگونه برای وردپرس پوسته بسازیم؟ (بخش ۷)

نوشته شده توسط مهندی رادی در تاریخ ۱۶ / ۱۱ / ۱۳۸۷ در دسته وبمستر، وردپرس، پوسته

مطالب ورودی (Content)

نزدیک به یه ماه است که سری آموزشی ساخت پوسته‌ی وردپرس را آغاز کرده ایم و در این جلسه قصد داریم تا جلسه‌ی هفتم از این سری را به شما آموزش دهیم. در این جلسه علاوه بر اینکه باید جمله‌ی تکراری ام را تکرار کنم (همان جمله‌ی معروفی که از جلسه‌ی اول شروع کنید و سپس با ما همراه باشید، برای اینکار از [این لینک](#) کمک بگیرید) باید این نکته را نیز به شما گوشزد کنم که جلسه‌ی قبل یکی از جلسات بسیار مهم این سری آموزشی بود، بنابراین اگر جلسه‌ی قبل را مطالعه نکرده اید و یا آنرا خوب متوجه نشده اید، حتماً دوباره مطالعه کرده و سوالات خود را از ما دریغ نفرمایید.



در جلسه‌ی پیش‌عنوان هر یک از پست‌های و بلاگ را به داخل صفحه آورديم، در اين جلسه می خواهيم که محتوا (Content) و يا همان مطالب اصلی پست را نيز به داخل صفحه بياوريم. در اين جلسه هميچنین مقدار بيشتری تگ باكس مخفی (div) برای مجزا کردن محتواي هر پست از عنوان آن (پست)، بكار خواهيم برد. پس برای فراگيری مطالب امروز با ما همراه باشيد....

بعلت طولانی شدن صفحه، به ادامه مطلب مراجعه کنيد....

امروز هم، همان کارهای اولیه را انجام می دهیم. برنامه‌ی Xampp را اجرا کرده، فolder **tutorial** از بخش پوسته های وردپرس را گشوده، یک صفحه از مرورگر وب را باز کرده و آدرس [wordpress/http://localhost](http://localhost) را در آن نوشته و در آخر هم فایل index.php را باز می کنیم.

● مرحله‌ی اول

کد < ?php the_content(); ?> را در زیر کدهای عنوان مطالب‌itan تایپ کنید:

```
<?php if(have_posts()): ?><?php while(have_po:  
    <h2><a href="<?php the_permalink(); ?>  
        <?php the_content(); ?>  
    <?php endwhile; ?>  
    <?php endif; ?>
```

تغیيرات را save کرده و مرورگرتان را refresh کنید. شما باید الان نوشته های هر پست را در زیر عنوان همان پست مشاهده کنید.

Demo Theme Development

Just another WordPress weblog

Hello World



Lore ipsum dolor sit amet, consectetur adipisci
nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in
officia deserunt mollit anim id est laborum. 😊

❖ چه چیز اتفاق افتاد و این کار شبیه به چه بود؟

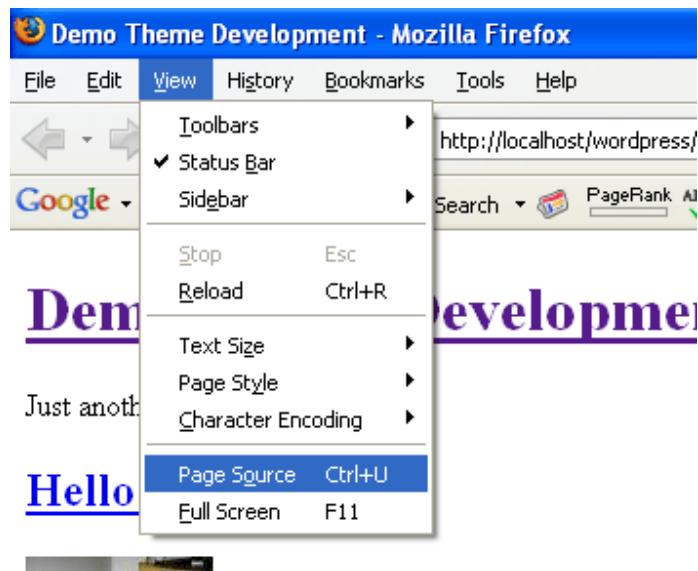
شما از تابع `the_content()` در php استفاده کردید که پست های ورودی (`post`) یا همان `content` را صدا می زند. حتماً متوجه شده اید که مطالب ورودی شما، تمام عرض پنجره‌ی مرورگر را گرفته‌اند. این بدین خاطر است که شما هنوز استایلی برای آن تعریف نکرده‌اید. بیاد داشته باشید که فایل `style.css` نیز وجود دارد که هنوز شما با آن کار نکرده‌اید، بعدها که با استایل بیشتر آشنا شدید، خواهید آموخت که چگونه به باکس‌های مخفی و سایر قسمت‌های صفحه، ویژگی بدهید.

در جلسه‌ی قبل، من از شما خواستم که یکسی‌ی مطالب ساختگی را برای آزمایش به وردپرس اضافه کنید تا چندین پست داشته باشید. اگر شما هنوز اینکار را انجام نداده‌اید، همین الان اینکار را انجام دهید. سپس بازگردید به صفحه‌ی اصلی که در اینصورت با چندین پست، صفحه‌ی اصلی شما اینچنین به نظر می‌آید.



ادامه دهید...

برگردید به مرورگرتان، بروی تب **View** کلیک کنید و گزینه **ی** **Source** را انتخاب نمایید. یک پنجره از کدها برایتان باز خواهد شد. اگر شما از IE استفاده کنید یک صفحه **ی** **Source** برایتان باز خواهد گشت.



من با افتخار از Firefox استفاده می کنم و در اینصورت، پنجره **ی** **Source** برای من اینگونه به نظر می رسد:

```
<div id="container">

    <h2><a href="http://>
        <p><img id="image13"
            alt="dcfn0032.JPG"/>Lorem ipsum do.
            ad minim veniam, quis nostrud exerci:
            velit esse cillum dolore eu fugiat :
            laborum. <img src='http://localhost,
            <p>
```

توجه کنید که بین محتوای فایل `index.php` و کدهای پنجره `Source` تفاوت وجود دارد. تمامی نوشته ها و عکس ها و غیره و به طور کلی، هرچیزی که درون این پنجره و یا همان `notepad` وجود دارد، بوسیله `the_content()` صدا زده شده است. جالب است نه؟ بدون وبلاگ و مخصوصاً قالب های وردپرس، شما تمام این عکس ها و نوشته ها را بصورت کد، مشاهده خواهید کرد.

همچنین، توجه کنید که **تگهای باز و بسته‌ی P**، بدور متن من قرار گرفته اند (تگ `P` جزو Xhtml است) در حالی که این تگها در فایل `index.php` وجود ندارند، ولی آنها در پنجره `Source` یا `notepad` شما نمایان هستند؟ برای یافتن جواب، پاراگراف بعدی را مطالعه کنید

◆ تگ های `P` ، چرا و چطور؟

«**چرا** – وقتیکه شما **ورودیهای وبلاگ** و یا همان **Content** را تایپ می کنید، هرگاه که بوسیله `Enter` به خط بعدی می روید، یک پاراگراف را تشکیل می دهید. شما بیازمند این هستید که یک راه برای نشان دادن آن پیدا کنید. هر پاراگرافی داخل **تگهای P** قرار می گیرد. در اصل تگ **P** مخفف **Paragraph** می باشد. در حقیقت بدانید که این تگ `P` جزو همان فایلهای ورودی بحساب می آید و یک قسمت مجزا

نیست که بعداً به نوشه‌ی شما اضافه شود.

«**چطور** – اما اگر برایتان این سوال پیش آمده که چطور این تگها به دور نوشه‌های من قرار گرفته‌اند، جواب خیلی ساده است، قالب‌های سیستم وردپرس بطور خودکار تگ‌های P را برای شما می‌سازند.

● مرحله‌ی دو:

یک باکس مخفی (div) بدور عبارت the_content() قرار دهید و به آن را اضافه کنید. به مانند زیر:

```
<div class="entry">  
</div>
```

شما الان باید داشته باشید.

```
<div id="container">  
    <?php if(have_posts()): ?><?php while(ha  
        <h2><a href=<?php the_permalink  
        <div class="entry">  
            <?php the_content(); ?>  
        </div>  
    <?php endwhile; ?>  
    <?php endif; ?>  
</div>
```

تغییرات را ثبت کرده و صفحه‌ی مرورگر را referesh کنید. حال اگر شما به منوی view بروید و سپس گزینه‌ی Page Source را کلیک کنید، خواهید دید، تگ div به همراه class="entry" بدور هر پست ورودی تان قرار گرفته است.

چرا؟ ♦

اولین دلیل، با اینکار شما می توانید بگویید که کجا عنوان یک مطلب تمام می شود و کجا محتوای یک پست شروع می شود. **دومین دلیل**، برای استایل بندی (یا همان صفحه آرایی) بوسیله‌ی فایل Style.css است. شما بعدها یاد خواهید گرفت.

◆ چه تفاوتی بین id و Class وجود دارد؟

تا اینجا، برای هر باکس مخفی و یا تگ div، شما از id برای نام دهی به آن استفاده می کردید. برای مثال "id="header" را ببینید! بنابراین چه تفاوتی بین این دو وجود دارد؟ تفاوت این است، id منحصر به فرد است ولی class اینگونه نیست. اگر شما نگاهی بیاندازید به کدهای پنجره‌ی Source، خواهید دید که تنها یک "id="header" وجود دارد و یا تنها یک "id="container" وجود دارد، در صورتی که چندین class="entry" را مشاهده خواهید کرد.

◆ آیا برای header و container می توان بجای استفاده از id از class استفاده کرد ؟ جواب : بله

این را در **مغزتان نگه دارید**: شما نباید هیچ id تکراری را داشته باشید. برای مثال، شما نمی توانید دو تا "id="header" بروی یک صفحه داشته باشید. بنابراین در شرایطی که شما می بایست از تگهای div چندین بار استفاده کنید، بجای استفاده از id برای نام دهی آنها، از class کمک بگیرید مانند باکس مخفی برای پست‌های ورودی که از class کمک گرفتیم.

● مرحله‌ی سوم:

تگ div را به دور عنوان پستها و پستهای ورودی (content) قرار دهید. نام آن را class="post" بگذارید.

```
<div class="post">  
</div>
```

توجه: نام‌های انتخاب شده برای class‌ها و id‌ها می تواند هر چیزی باشد. شما می توانید از این پس، نام غذاهای مورد علاقه‌تان، بازیگران مورد علاقه‌تان و ... را برای آنها قرار دهید، اما کلماتی مانند post و entry علاوه بر اینکه کوتاه و ساده هستند،

براحتی می توانند منظورمان را هم برای خود و هم برای دیگران شرح دهند. اینطور نیست؟؟

بعد از اضافه کردن کد بالا باید داشته باشید:

```
v id="container">

<?php if(have_posts()) : ?><?php while(have_posts():

    <div class="post">
        <h2><a href="<?php the_permalink() ?>">
            <div class="entry">
                <?php the_content(); ?>
            </div>
        </div>
    <?php endwhile; ?>
```

و عکس زیر ، یک نسخه از حالت سازماندهی شده کدهای بالا است:

```
<div class="post">
    <h2><a href="<?php the_permalink() ?>">
        <div class="entry">
            <?php the_content(); ?>
        </div>
    </div>
```

من برای برجسته کردن کدها از دکمه **tab** بجای دکمه **spacebar** استفاده می کنم.

چرا سازماندهی؟ برخلاف عکس بالا، کدهای شما با هایلایت قرمز و سبز مشخص نیستند. شما نیازمندید تا کدها و وظیفه‌ی آنها را بیاد بیاورید. بوسیله‌ی این نوع برجسته کردن (فاصله‌های بوجود آمده از **tab**) حالا این عمل آسان است که برای مثال بگوییم که تگ **</div>** کجا بسته شده است.

فایل index.php را save کرده و مرورگر تان را refresh کنید تا تمامی تغییرات ایجاد شده در پنجره **کدهای Source** را مشاهده کنید.

◆ چرا یک تگ div دیگر به دور عنوان پست و مطالب پست ها اضافه کردیم؟

جواب: شما عبارت "div class="entry"" را برای مجزا کردن عنوان یک پست ها از مطلب همان پست اضافه کردید. در صورتی که عبارت "div class="post"" را برای جدا کردن کل یک مطلب از دیگری اضافه کرده اید.

Hello World



Lore ipsum dolor
nisi ut aliquip ex ea commodo conse
officia deserunt mollit anim id est lab



Lore ipsum dolor
nisi ut aliquip ex ea commodo conse
officia deserunt mollit anim id est lab

Lore ipsum dolor sit amet,
nostrud exercitation ullamco
eu fugiat nulla pariatur. E

Lore ipsum dolor sit amet, consec
ea commodo consequat. [\(more...\)](#)

Post number **1**

r adipisic
e dolor ir

r adipisic
e dolor ir

lipisic
ut ali
occaeca

sed do e

Hello World

Lore ipsum dolor sit amet, consec
ea commodo consequat. Duis aute i
anim id est laborum.

Post number **2**

sed do e
nderit ir

این درس امروز ما بود؛ ممکن است کمی هضم کردن این اطلاعات برای شما سخت باشد، ولی با کمی کار کردن تمامی این مشکلات حل خواهد شد.

در جلسه‌ی بعد به شما نحوه‌ی درگیر شدن با تاریخ، دسته‌ها و بخش نظرات را به شما خواهیم آموخت.

درس‌های بعدی از این سری آموزشی را دنبال کنید....

منبع:

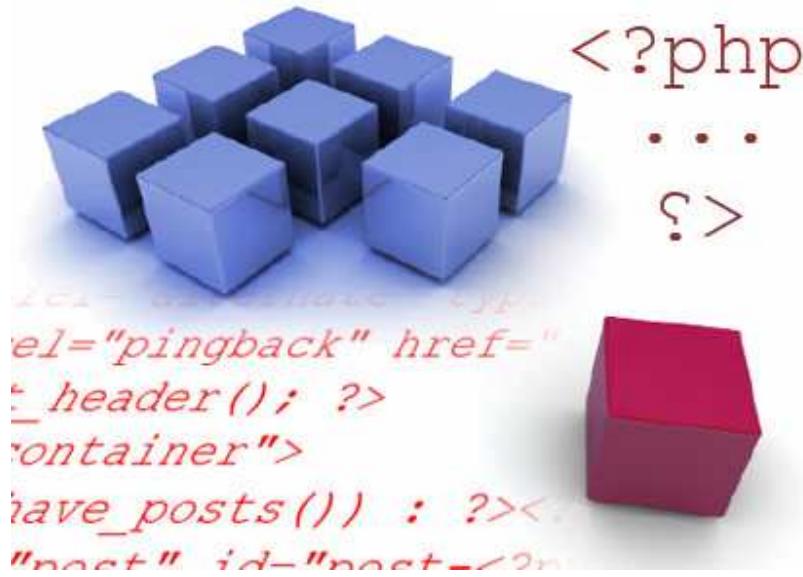
<http://www.p30help.ir/how-make-wordpress-theme-6b-lesson-7>

چگونه برای وردپرس پوسته بسازیم؟ (بخش ۸ و ۹)

نوشته شده توسط مهندی رادی در تاریخ ۱۳۸۷ / ۱۱ / ۲۳

در ابتدای عرايضم باید اين نکته را بگويم که هدف من از کند پيش رفتن و در حقیقت يواش راه رفتن در اين مسیر، اين بود که هضم مطالب برای شما بزرگواران، آساتر شود. مطمئناً يك شبه نمی توان راه صد ساله را پيمود. اما شکوه و گلایه شما عزيزان از کند راه رفتن و عقب ماندن از ساير قافله‌ها، باعث شد که بخواست شما مطالب را دو تا دوتا و يا همان جفت پا جفت پاى خودمان بنويسم. اميدوارم که سريع راه رفتن من باعث خستگی شما نشود....

ابن بار دو جلسه را بصورت يكجا به شما خواهیم گفت. اميدوارم که با اينكار بتوانم رضایت شما عزيزان را جلب كرده باشم. قبل از اينكه درس را به صورت جدي شروع کنم، باز و باز و باز هم اين مسئله‌ی تكراري را می گويم که از ابتدا شروع کنيد.....
(البته انصافاً اين بار خلاصه گفتم)



بعد از گفتن مقدمه باید این نکته را نیز عرض کنم که در ابتدای جلسه، کار با postmetadata را شروع خواهیم کرد و سپس در جلسه‌ی بعد یا همان بخش دوم که در ادامه‌ی همین درس خواهیم داشت، به شما Else, Post ID, Link Title را معرفی خواهیم کرد. درس را با نیرویی مضاعف شروع می‌کنیم....

بعلت طولانی شدن صفحه، برای مطالعه‌ی درسهای هشتم و نهم به ادامه مطلب مراجعه کنید....

نکته: برای بزرگ‌دین عکس‌ها بروی آنها کلیک کنید.

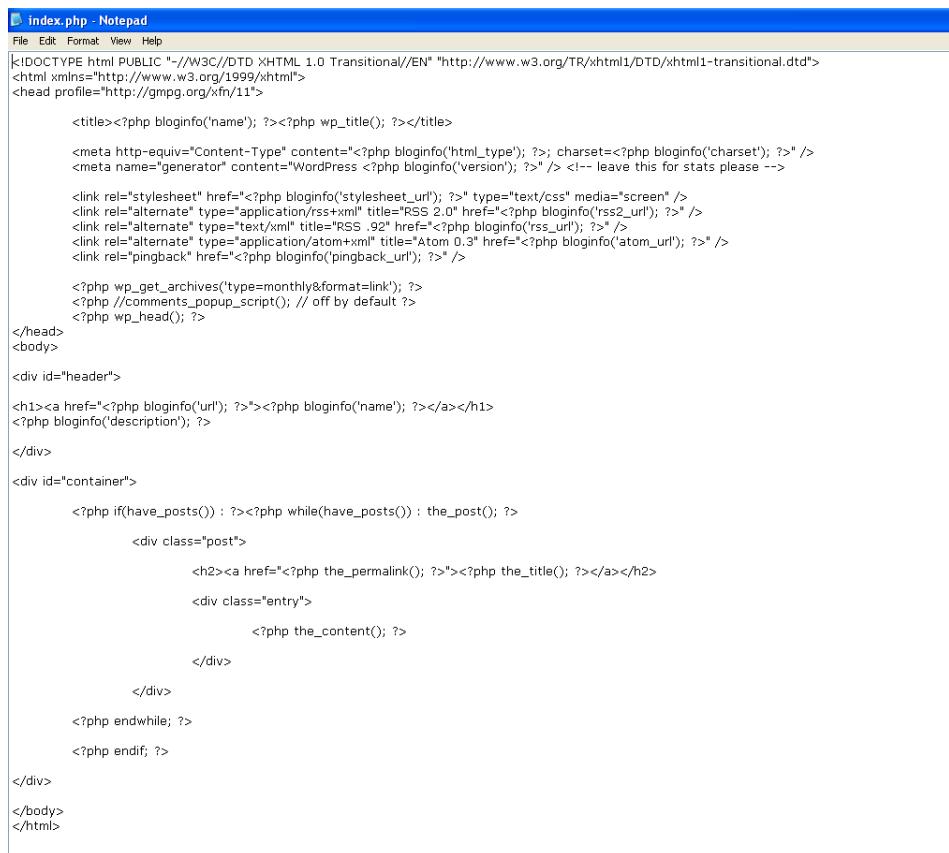
درس هشتم – Postmetadata

◆ **postmetadata** چیست؟

سوال بسیار خوبی پرسیدم، در حقیقت **postmetadata** شامل: تاریخ نوشته، دسته‌ها، نویسنده، تعداد نظرات و... می‌شود. در حقیقت هر اطلاعات جانبی که مربوط به یک پست شود را **postmetadata** گویند. برای اینکه این موضوع را بیشتر متوجه شوید به عکس زیر نگاهی بیندازید:

قبل از اینکه درس را به طور رسمی شروع کنیم، شما همان کارهای قبلی را انجام دهید، یعنی برنامه‌ی Xampp را اجرا کنید، فولدر پوسته‌های وردپرس، مرورگر و همچنین فایل index.php را نیز باز کنید.

اجازه بدهید که در ابتدا، آنچه را که در دروس قبل گفته ایم، بررسی کنیم، شما باید تا الان مانند آنچه که در عکس زیر موجود است را در فایل index.php داشته باشید:



```
index.php - Notepad
File Edit Format View Help
<!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">
<head profile="http://gmpg.org/xfn/11">
    <title><?php bloginfo('name'); ?><?php wp_title(); ?></title>
    <meta http-equiv="Content-Type" content="<?php bloginfo('html_type'); ?>; charset=<?php bloginfo('charset'); ?>" />
    <meta name="generator" content="WordPress <?php bloginfo('version'); ?>" /> <!-- leave this for stats please -->
    <link rel="stylesheet" href="<?php bloginfo('stylesheet_url'); ?>" type="text/css" media="screen" />
    <link rel="alternate" type="application/rss+xml" title="RSS 2.0" href="<?php bloginfo('rss2_url'); ?>" />
    <link rel="alternate" type="text/xml" title="RSS .92" href="<?php bloginfo('rss_url'); ?>" />
    <link rel="alternate" type="application/atom+xml" title="Atom 0.3" href="<?php bloginfo('atom_url'); ?>" />
    <link rel="pingback" href="<?php bloginfo('pingback_url'); ?>" />
    <?php wp_get_archives('type=monthly&format=link'); ?>
    <?php /*Comments_popup_script(); // off by default ?>
    <?php wp_head(); ?>
</head>
</body>
<div id="header">
<h1><a href="<?php bloginfo('url'); ?>"><?php bloginfo('name'); ?></a></h1>
<?php bloginfo('description'); ?>
</div>
<div id="container">
    <?php if(have_posts()): ?><?php while(have_posts()): the_post(); ?>
        <div class="post">
            <h2><a href="<?php the_permalink(); ?>"><?php the_title(); ?></a></h2>
            <div class="entry">
                <?php the_content(); ?>
            </div>
        </div>
    <?php endwhile; ?>
    <?php endif; ?>
</div>
</body>
</html>
```

● مرحله‌ی اول:

کدهای موجود در فایل [postmetadata.txt](#) را در زیر کد <?php the_content(); ?> کپی و پیست کنید (**توجه**: این بار شما می‌توانید از کپی و پیست استفاده کنید. شما لازم نیست که بطور کامل کدهایی که در این قسمت اضافه کردیم را متوجه بشوید. همین که بدانید هر گروه از کدها چه اعمالی را انجام می‌دهند، کافی است)

عکس زیر بخشی از کدهای اضافه شده را نشان می‌دهد. سعی کنید که کدهایتان را سازماندهی کنید (بوسیله‌ی دکمه Tab)

```
<div class="entry">  
    <?php the_content(); ?>  
    <p class="postmetadata">  
<?php _e('Filed under&#58;'); ?> <?php the_category(', ') ?> <?php  
<?php comments_popup_link('No Comments &#187;', '1 Comment &#187;', 'Comments') ?>  
    </p>  
</div>
```

حال تغییرات را ثبت کرده و صفحه را refresh کنید.

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. [\(more...\)](#)

Filed under: [Uncategorized](#), [Personal](#), [Sub Category](#) by Small Potato
[2 Comments »](#)

شما همچنین می‌توانید به کدهای پنجره‌ی source از مرورگرتان نگاهی بیاندازید تا کدهای postmetadata را مشاهده کنید و متوجه شوید که اینگونه کدها به چه شکلی ظاهر می‌شوند. بیاد بیاورد که من به شما نشان دادم که چگونه پنجره‌ی کدهای Source را بگشایید. برای اینکار به تب view رفته و سپس Page Source یا Source را انتخاب کنید.

❖ توضیحات مربوط به کدهای اضافه شده:

« **<?php the_content(); ?>** - تمامی اطلاعات postmetadata بین تگ های پاراگراف (P) با مقدار **class="postmetadata"** قرار می‌گیرند، زیرا من می‌خواهم تا postmetadata را از سایر محتوا (Content) یا همان مطالب ورودی مجزا کنم. بدون تگ پاراگراف (P)، اطلاعات postmetadata در هر کجا که محتوا (Content) یا

همان مطالب ورودی تان پایان یابد، نشان داده خواهد شد، و همچنین، هیچگونه فاصله ای بین مطالب ورودی (Content) و postmetadata وجود نخواهد داشت.

«`<?php _e('Filed under'); ?>` عبارت که هست که عبارت `: - <?php _e('Filed under'); ?>` دو نقطه (:) را صدا می زند و همچنین کد `(''); ?>` بدور `?php _e('Filed under')` قرار می گیرد، البته شما می توانید به سادگی `Filed under:` را تایپ کنید، ولی بیاد داشته باشید که می خواهیم زمانی این پوسته را به فارسی ترجمه کنیم و در صورتی که اینکار را انجام نداده باشید، با مشکل رو برو خواهیم شد. در جلسات پایانی که نحوه‌ی فارسی سازی پوسته را به شما خواهیم گفت، کاربرد کد `('e' _e` را بهتر متوجه خواهید شد.

«`<?php the_category(); ?>` یک تابع از php می باشد و دسته‌ها (category) را صدا می زند. دسته‌هایی که شما مطلب تان را در آنها قرار داده اید. حال که شما `Filed under:` را با یکدیگر قرار داده اید، خواهید داشت [. ۲ Name of category ۱, Name of category :Filed under](#)

علامت کامایی که (,) بین نقل قول (‘،’) قرار دارد، برای جدا کردن دسته‌ها بوسیله‌ی کاما(،) گذاشته شده است، که می توانید آنرا با هر چیز دیگری عوض کنید (کاما در حقیقت بین هر دسته قرار می گیرد تا دسته‌ها را از یکدیگر جدا کند. اگر متوجه نشده اید، کاما را با / عوض کنید و صفحه را Refresh نمایید تا تغییرات بوجود آمده، باعث فهم شما شود، البته این نکته را باید بگوییم که شما باید مطالبتان را در چند دسته یا Category مختلف قرار داده باشید تا بتوانید این تغییرات را مشاهده کنید. اگر باز هم متوجه نشده اید که چه می گوییم، این مبحث را بدون تغییر رها کنید و مطمئن باشید که روزی خواهید فهمید)

«`<?php _e('by'); ?>` این عبارت مانند همان کاری هست که برای `Filed under:` کردیم. اگر شما یک پوسته برای استفاده‌ی شخصی می سازید، می توانید عبارت `(e _e()` را بدور کلمه هایتان قرار ندهید. اما اگر پوسته‌ای می سازید که برای عموم است، قرار دادن این عبارت بسیار اهمیت دارد. بطور کلی بگوییم که هر پوسته‌ای را چه برای خود و چه برای هر شخص دیگری می سازید، اگر می خواهید که پوسته تان استاندارد و استفاده از آن راحت باشد، از کد `(e _e()` حتماً استفاده کنید. در جلسات پایانی اهمیت آنرا بیشتر حس خواهید کرد.

«`<?php the_author(); ?>` - نیازی به توضیح ندارد، این تابع به سادگی نام شخصی که این پست و یا مطلب را منتشر کرده است را چاپ می کند.

`Comment | <?php comments_popup_link('No Comments »', ' ' . '»', '% Comments »); ?>`

عبارت `comments_popup_link()` تعداد نظراتی که برای هر پست داده شده است را صدا زده و نمایش می دهد. عبارت `No Comments »` هنگامی نشان داده می شود که برای مطلب شما هیچ نظری وجود نداشته باشد. عبارت `Comment |` `»` هنگامی نمایش داده می شود که برای مطلب شما تنها یک نظر وجود داشته باشد، و در آخر عبارت `% Comments »` هنگامی استفاده می شود که تعداد نظرات شما بیش از یک نظر باشد. برای مثال **Comments** «`8`». علامت درصد (%) بجای تعداد نظرات و عبارت `»` نیز کد شکل «`8`» می باشد.

«`<?php edit_post_link('Edit', ' | ', '') ; ?>` - این کد تنها زمانی استفاده خواهد شد که شما بعنوان مدیر وارد وبلاگتان شده باشید. عبارت `edit_post_link()` به سادگی لینک Edit را برای هر پستی به مدیر وبلاگ نشان می دهد تا اگر مایل بود، قادر باشید که پست مورد نظرش را به آسانی تصحیح کند. عبارت `edit_post_link()` سه گروه از علامت نقل قول را دارا می باشد. اولین علامت نقل قول برای آن است که کلمه ویرایش پست با چه عنوانی نمایش داده شود. اگر شما برای مثال عبارت `Edit` را بجای کلمه `Post` در عوض کلمه `Post` نمایش داده خواهد شد. دومین گروه از علامت نقل قول ها تنها برای آن می آید که عبارتی را **قبل** از کلمه `Edit` قرار دهد. در اینجا یک خط عمود (|) که کد آن هم `|` است نمایش داده خواهد شد. سومین گروه از علامت های نقل قول برای این است که شکلی را **بعد** از کلمه `Edit` قرار دهد و چون درون آن هیچ عبارتی قرار داده نشده است، هیچ شکلی نیز مشاهده نخواهد شد. کاملا می فهمم که در این مورد کمی گیچ شده اید، می توانم بیشتر توضیح بدhem ولی اینکار را انجام نمی دهم و از خود شما می خواهم که با تغییر محتوای داخلی این سه گیومه متوجه شوید که منظور من چیست. فقط تغییرات را بعد از آزمایشات گوناگون به حالتی که الان وجود دارد، برگردانید.

توجه: بعنوان مدیر و بلاگ وارد بخش مدیریت شوید سپس به صفحه ای اصلی سایت برگردید و لینک Edit را مشاهده کنید.

❖ اما یه چیزی را فراموش کرده ایم!!

درست حدس زدید، فراموش کردیم که تاریخ نوشته شدن مطلب را اضافه کنیم. این قسمت را من جدا از بخش های دیگر گفتم زیرا بسیار تنوع در نمایش دادن تاریخ متفاوت است. برای مثال به عکس های زیر توجه کنید:



بعثت تنوع این بخش در postmetadata من قرار دادن تاریخ را به جلسات بعد موکول می کنم و چند روش را برای قرار دادن تاریخ به شما خواهم گفت.

خوب درس این جلسه‌ی ما به پایان رسید، اما فراموش نکنید که جلسه‌ی بعد را نیز در ادامه این جلسه بخارط رضایت شما عزیزان شروع می کنم. بنابراین با ما باز هم همراه باشید...

در بخش دوم از این سری، شما را با Else, Post ID, Link Title آشنا خواهیم کرد.

درس نهم (بخش دوم) – آشنایی با Else, Post ID, Link Title

این درس Else, Post ID, Link Title را پوشش می دهد، که البته تنظیمات اضافی محسوب می شوند و اضافه کردن آنها اختیاری است، اما شما این تنظیمات را در هر پوسته‌ای خواهید یافت، زیرا شما نمی دانید، اشخاص برای چه اهدافی از پوسته‌ی شما استفاده می کنند. این بار به سرعت به سراغ مراحل می رویم و در طول آموزش، توضیحات لازمه را می دهیم.

● مرحله‌ی اول:

کدهای پایین را در زیر کد <?php endwhile; ?> تایپ کنید:

```
<?php else : ?>
```

```
<div class="post">
<h2><?php _e( 'Not Found' ); ?></h2>
</div>
```

باید تصویری شبیه به زیر داشته باشید (توجه کنید برای سازماندهی کدها از tab استفاده کنید):

```
<?php endwhile; ?>
<?php else : ?>
<div class="post">
    <h2><?php _e('Not Found'); ?></h2>
</div>
<?php endif; ?>
```

تغییرات را ثبت کنید و صفحه را نیز Refresh نمایید، اما شما باید هیچ تغییری را مشاهده کنید. حال به توضیح آنچه که تایپ کردید، می پردازیم:

به کدهای زیر توجه کنید :

```
<?php if(have_posts()) : ?>
<?php while(have_posts()) : the_post(); ?>
<?php endwhile; ?>
<?php endif; ?>
```

ابتدا (`if(have_posts())`) چک می کند که آیا هیچ گونه پستی در این وبلاگ وجود دارد یا خیر. سپس حلقه‌ی `while(have_posts())` دستور `the_post()` را برای صدا زدن پست‌های شما اجرا می کند. عبارت `else` برای زمانی است که هیچ پستی وجود نداشته باشد. اگر کمی با برنامه نویسی php آشنا باشید، براحتی متوجه من را متوجه شده اید، اما اگر آگاهی چندانی از این زبان ندارید، به زبان عامیانه باید اینگونه عرض کنم که عبارت شرطی `if` که به معنای **اگر** است، ابتدا سعی می شود که اجرا گردد، اما اگر این شرط بدلیل نبود پست، اجرا نشد، عبارت `else` که به معنای **در غیر اینصورت** است، اجرا خواهد شد. یعنی `if` اجرا می شود و یا `else` و البته عبارت شرطی `if` به

ارجحیت دارد. امیدوارم که منظور من را متوجه شده باشید. توجه شود، هنگامی که شرطی را آغاز می کنید، باید پایان آن شرط را در آخر مشخص کنید، بنابراین برای پایان دادن به شرط if از دستور <?php endif; ?> کمک گرفته ایم. همین موضوع برای حلقه نیز صادق است، یعنی زمانی که حلقه ای را شروع کردیم، باید پایان آن را نیز مشخص کنیم، پس برای مشخص کردن پایان حلقه از عبارت <?php endwhile?>; ?> استفاده کردیم. اگر جلسه اول را بیاد بیاورید در مورد قانون تو در توابی توضیح دادیم، که نمود بارز آن در اینجا مشخص است. اگر دقت کنید متوجه می شوید که ابتدا **شرط آغاز شده**، سپس **حلقه آغاز شده**، بعد دستورات داخل حلقه اجرا شده، سپس **حلقه بسته شده** و در آخر **شرط نیز بسته** می شود، یعنی هیچگونه تداخلی در باز و بسته شدن اینگونه کدها نباید وجود داشته باشد.

حالا شما می دانید که else برای چیست. در زیر مجموعه‌ی `if` یک باکس مخفی ایجاد کردیم و داخل آن نوشته ایم **Not Found** یا همان **یافت نشد** خودمان. این بدین معنی است که اگر پستی وجود داشت که هیچ، ولی اگر پستی یافت نشد، آنگاه عبارت **Found** برای شخص بیننده نمایش داده خواهد شد. چه منطقی و ساده....

در مثال بالا، پیغام **Not Found** درون <?php _e('')?> قرار گرفته است. دیگر توضیح نمی دهم که اینکار برای چیست.

تمامی پیغام‌ها و کدها برای عبارت **Not Found** با تگ‌های <h2> و </h2> احاطه شده است. البته اینکار آنچنان واجب نیست. اینکار را برای این انجام داده ایم که نوشته‌ی ما بزرگتر به نظر برسد و بیننده راحت‌تر بفهمد که اشتباه کرده است و چیزی را در سایت شما پیدا نمی کند، شوخی کردم، امیدوارم که سایت شما پر از محتوا باشد.

❖ **اما تگ‌های <div class="post"> و </div>** برای چه هستند؟
خوب مشخص است، جوابش را شما بگویید. بعد از این همه جلسه دیگر نگفتنش مایه آبروریزیست.

● مرحله‌ی دو:

مقدار <div class="post"> را به id="post-<?php the_id(); ?>" اضافه کنید:

```
<div class="post" id="post-<?php the_ID(); ?>">
```

تغییرات را ثبت کرده و مرورگرتان را refresh کنید. حال به تب View رفته و Page Source یا انتخاب کنید . شما خواهید دید که هر پستی حالا یک شماره یا id دارد که به آن اضافه شده است. کد the_id() به سادگی id هر پستی را صدا می زند و آنرا در کد صفحه نشان می دهد. حتماً دیده اید که سایتها وردپرس در حالت عادی آدرس سایتشان به صورت <http://www.p30help.ir/?p=۱۲۵> است، عدد ۱۲۵ همان مقدار id می باشد.

◆ چرا اینکار را کردیم؟

اینکار برای این است که بطور اختصاصی یک پست خاص را مورد هدف قرار دهیم. برای مثال شاید شما بخواهید که ظاهر یکی از پست هایتان با سایر پست ها متمایز باشد، برای ایجاد تمایز بین پست ها، ما این عمل را انجام دادیم. بعدها، هنگامی که شما از فایل Style.css استفاده می کنید. به کمک id منحصر به فردی که هر پست دارد، قادرید تا این تمایز را بین هر پست ایجاد کنید.

◆ چطور می توان هم Class و هم id را برای یک div یا باکس مخفی قرار داد؟

یک تگ هست، اما class یک مشخصه می باشد. id نیز یک مشخصه است. هر تگی می تواند چندین مشخصه را قبول کند مثل تگ div که می تواند ۲ مشخصه ی class و id را قبول کرده است.(توجه کنید id یک مشخصه ی Xhtml و عبارت() یک تابع php است و این دو با یکدیگر متفاوتند).

● مرحله سوم :

عبارة ”[title=<?php the_title\(\); ?>"](<?php the_permalink(); ?>)“ را به لینک عنوان پست ها اضافه کنید.

```
<h2><a href="<?php the_permalink(); ?>" title="<?php the_title(); ?>">
```

عکس بالا یک قطعه از اسکرین شات است تا به شما نشان دهد که کد را کجا باید قرار دهیم. تغییرات را save کرده و مرورگرتان را refersh view source کنید. به دوباره بازگردید. وقتی که پنجره ی Source را مشاهده کردید، post title link را جستجو

کنید. اگر عنوان مطلب شما Hello World باشد، آنگاه باید **title="Hello World"** را داشته باشید.

توضیح: یکی دیگر از مشخصه های xhtml برای تگ لینک () می باشد. هرچیزی که داخل علامت نقل قول قرار گیرد، برای توصیف لینک شما بکار برده می شود. در این مورد، عنوان هر پست، توصیف همان لینک نیز می باشد، این هست دلیل اینکه چرا از تابع پی اچ پی the_title() دوباره استفاده کردیم.

اگر نمی خواهید از the_title() عنوان یک مقدار برای title="" استفاده کنید، آنگاه می توانید مقدار دیگری که title="" بگذارید، تا عنوان توصیفات برای لینک عنوان مطلب شما محسوب شود. برای مثال، اگر در عوض the_title()، شما از title="Click" استفاده کنید، توصیف برای عنوان هر پست Click me استفاده کنید.

حالا، به صفحه‌ی وب بازگردید، کرس موس را بروی عنوان هر پست قرار داهید، خواهید دید که توصیفات بصورت تول تیپ دیده خواهند شد. به عکس زیر نگاه کنید تا بهتر متوجه شوید:



❖ این عمل برای چیست؟

هرگاه که شما پستی را منتشر کردید، وردپرس Technorati.com و سایتهای دیگر را مطلع می سازد. ربات Technorati.com سپس به وبلاگ شما آمده و مطالب جدیدتان را تجسس کرده و در آخر یک خلاصه ای از پست شما را ایندکس می کند، که این خلاصه، شامل **عنوان توضیحات مطلبتان** نیز می شود. کلًّا بدانید که چیز خوبی است....

در جلسه‌ی بعد به شما یاد خواهیم داد که چگونه برای قالب‌تان امكان **صفحه‌ی بعد** و **صفحه قبل** را بگذارد. به جد، برای این دو درس سختی کشیدم، باشد که مورد پسند شما قرار گیرد.

پی نوشت: خواستم این پست شامل سه درس باشد، اما هم خودم واقعاً خسته شدم و هم اینکه یاد دهی مطالب زیاد باعث عدم هضم این مطالب می گردد.

منبع:

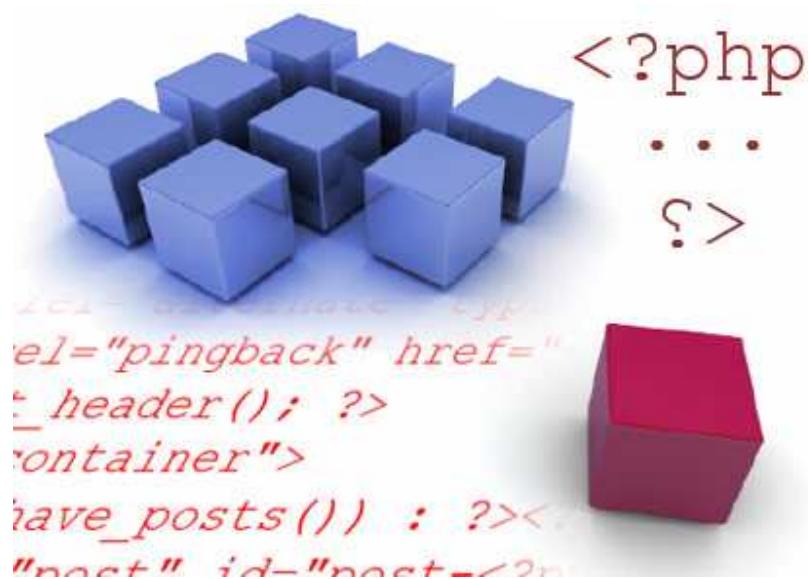
<http://www.p30help.ir/how-make-wordpress-theme-6b-and-6c-lesson-8-9>

چگونه برای وردپرس پوسته بسازیم؟ (بخش ۱۰)

نوشته شده توسط مهند رادی در تاریخ ۱۲ / ۱۰ / ۱۳۸۷ در دسته وبمستر, وردپرس, پوسته, PHP, HTML

لينک های Navigation

مدتی بود که سرگرم کاری بودم، به همین دلیل آماده کردن این آموزش کمی طول کشید، اما مهم این است که این آموزش آماده شد. بنابراین آموزش دهم از این سری آموزشی را با نیرویی مضاعف آغاز می کنیم. آموزش این هفته‌ی ما در رابطه با قراردادن لینکهای **صفحه‌ی بعد** ([Next Page](#)) و **صفحه‌ی قبل** ([Previous Page](#)) در انتهای صفحه است. این لینکها که حتماً با آنها آشنایی دارید وظیفه‌ی نشان دادن پستها و یا مطالب قبلی و بعدی را به شما دارند. راستی اگر تا حال با این سری آموزشی همراه نبوده اید، از [اینجا](#)، خواندن این سری آموزشی را پی بگیرید.



به دلیل طولانی شدن صفحه به ادامه مطلب مراجعه کنید...

آموزش این هفته‌ی ما بسیار کوتاه است و فقط شامل یک مرحله‌ی شود. در ابتدا همان کارهای همیشگی را انجام دهید: برنامه‌ی Xampp و فولدر پوسته‌های وردپرس را بگشایید. مرورگر تان را فعال کرده، و فایل `php.index` را باز کنید.

● مرحله‌ی اول:

کد زیر را بین کدهای `<?php else : ?>` و `<?php endwhile; ?>` اضافه کنید.

```
<div class="navigation">
<?php posts_nav_link( ); ?>
</div>

<?php endwhile; ?>

<div class="navigation">
<?php posts_nav_link(); ?>
</div>

<?php else : ?>
```

توجه کنید که برای سازماندهی کدها بوسیله‌ی دکمه Tab خطها را کمی جلو و عقب کرده‌اند.

❖ توضیح آنچه انجام شده است:

- `<div class="navigation">` این تگ باکس مخفی‌ای را با نام `navigation` ایجاد می‌کند. توجه کنید که ما این تگ را به دور لینک‌های `posts_nav_link()` قرار داده ام تا بتوانیم، بعدها به این باکس مخفی ویژگی بدهیم.
- `<?php` شروع `<?php` است.
- `posts_nav_link()` لینک‌های بعدی (Next) و قبلی (Previous) را صدا می‌زند.
- به صدا زدن آنها را پایان می‌دهد.
- تمام شدن `<?php` است.
- بستن باکس مخفی که نام آن `navigation` هست.

باید الان شکل پوسته‌ی شما به صورت زیر باشد:

Filed under: Uncategorized, Personal, Sub Category

[1 Comment »](#)

[— Next Page »](#)

Done



start

WPDesigner WordPre...

tutoria

تغییرات را ثبت و سپس مرورگر تان را refresh کنید حال باید لینک های Next و Previous را مشاهده کنید. بصورت پیشفرض اگر شما بیشتر از ۱۰ پست را به وردپرس تان اضافه نکرده باشید، این گزینه را نخواهید دید. اگر که نمی توانید لینک های Next و Previous را مشاهده کنید به قسمت مدیریت وردپرس رفته و در بخش نوشتن، مطالبی را اضافه کنید تا تعداد پست هایتان بیشتر از ده عدد شود و یا به بخش تنظیمات رفته، در زیر مجموعه‌ی تب خواندن، تعداد پست هایتان را از ۱۰ عدد به تعداد دلخواه در بیاورید، تا بتوانید لینک های Next و Previous را مشاهده کنید. برای مثال از ۱۰ از به ۵ پست برسانید

◆ چگونه posts_nav_link() را تطبیق دهیم:

در درس postmetadata تابع هایی را به شما معرفی کردیم که در تنظیمات آن سه علامت نقل قول را می توانستیم معین کنیم. در اینجا نیز شما قادرید تا این تنظیمات را در سه نقل قول جا دهید. این تنظیمات برای این است که شما بتوانید [بین، قبل و یا بعد](#) از لینک های Next و Previous علامت دلخواهتان را قرار دهید. دستور زیر شما را در فهم یاری می کند.

```
<?php posts_nav_link('in  
between', 'before', 'after'); ?>
```

اولین گروه از بخش علامت نقل قول که با رنگ سبز مشخص شده است، برای این منظور به کار می رود تا هر آنچه را که مایل بودید را در بین لینک های Previous و Next قرار دهید. بوسیله‌ی دومین گروه از علامت نقل قول که با رنگ آبی مشخص شده است، می توانید هر آنچه را که مایل بودید قبل این لینکها بیاورید. و در آخر سومین گروه از علامت نقل قول که با رنگ سبز مشخص شده است، وظیفه‌ی این عمل را بر عهده دارد تا هر آنچه را که مایل بودید بعد از این لینکها قرار دهید.

ممکن است که متوجه نشده باشید که منظور من چیست، پس بهتر است تا بعنوان مشق شب تان، با قرار دادن عبارت های گوناگون داخل این سه علامت نقل قول، متوجه شوید که منظور من چیست.

در اینجا یک مثال از تطبیق تابع `posts_nav_link()` داریم:



بله درس امروز به همین کوتاهی بود. امیدوارم که حداقل به اندازه‌ی کوچکی اش از آن استفاده کرده باشید.
البته روش آسانتر و بهتری برای ساخت لینک صفحات بعد و قبل هم هست که در درس‌های پایانی به شما خواهیم گفت.

منبع:

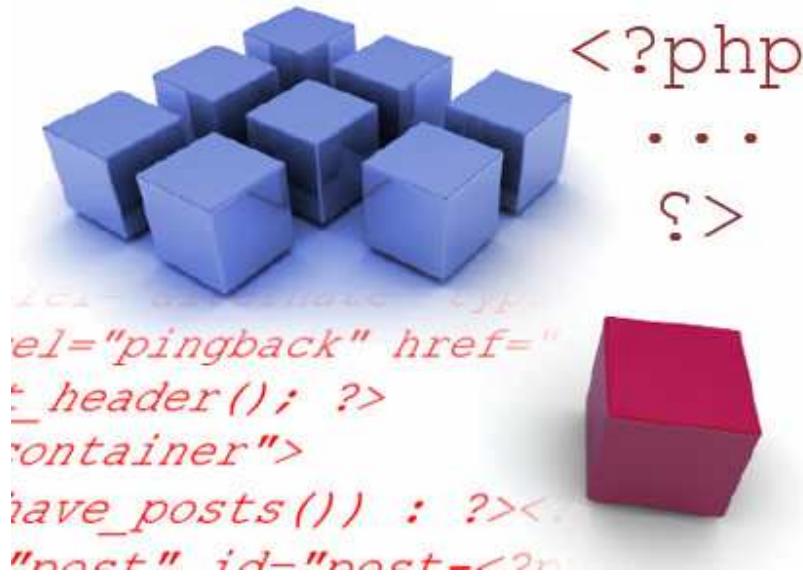
<http://www.p30help.ir/how-make-wordpress-theme-6e-lesson-10>

چگونه برای وردپرس پوسته بسازیم؟ (بخش ۱۱)

نوشته شده توسط مهندی رادی در تاریخ ۱۹ / ۱۲ / ۱۳۸۷

درس یازدهم : سایدبار (Sidebar)

مطلوب امروز خود را با یه عذرخواهی از شما به علت تأخیر در آماده شدن این مطلب آغاز می‌کنم. در این درس به شما نحوه‌ی ساخت سایدبار را خواهیم گفت. مطمئن هستم که برای شنیدن این درس بسیار مشتاق بودید، پس امیدوارم که مورد رضایت شما واقع شود.



در نگاه اول سایدبار (Sidebar) کمی مشکل به نظر می رسد ولی این اشتباهی بیش نبیست، بلکه سایدبار از یکسری ساختار تشکیل شده است که با فهم این ساختارها، کار برای شما بسیار ساده خواهد شد. البته شما در این بخش و بخش های بعدی ساختار سایدبار را تشکیل خواهید داد و در درسهای بعدی کد و استایل آن را خواهید نوشت.

خوب درس را شروع می کنیم، ادامه درس در ادامه مطلب ...

قبل از شروع مبحث، شما باید فایلی مانند [index.php](#) داشته باشید (بروی کلیک کنید)

فراموش نکنید که برنامه‌ی Xampp را فعال کنید.

● مرحله اول :

اجازه دهید که یک باکس از نوع class بنام Sidebar را ایجاد کنیم تا دور هرچیزی که جزو sidebar حساب می شود، قرار بگیرد. کد پایین را زیر باکس container و بالای تگ </body> تایپ کنید:

```
<div class="sidebar">  
</div>
```

```
</div>  
<div class="sidebar">  
</div>  
</body>  
</html>
```

● مرحله دوم :

یک لیست unordered (یا غیر سفارشی) داخل باکس جدید Sidebar ایجاد کنید.

unordered - «

unordered - »

```
<div class="sidebar">  
<ul>  
</ul>  
</div>
```

توجه: لیست غیر سفارشی یا همان تگ ul مخفف unordered list می باشد.

● مرحله ی سوم :

یک لیست آیتم ها (li) را درون unordered list که مخفف آن همان کد ul هست، ایجاد کنید و درون آن یک sub-heading قرار دهید. یعنی:

```
<li><h2><?php _e('Categories'); ?></h2>  
</li>
```

```
<ul>  
    <li><h2><?php _e('Categories'); ?></h2>  
        </li>  
</ul>
```

توجه کنید که فاصله ای بوسیله `Tab` قبل از تگ‌های `` و `` برای سازماندهی بهتر، اضافه شده است.

— لیست آیتم را باز می‌کند
— تگ `<h2>` را باز می‌کند
— کلمه‌ی `<?php _e('Categories'); ?>` را نشان می‌دهد
— تگ `</h2>` را می‌بندد
— لیست آیتم را می‌بندد

توجه: لیست آیتم یا همان تگ `li` مخفف کلمه‌ی `list item` می‌باشد.

شما نیازی ندارید که کد `<?php e('Categories'); ?>` را بدور کلمه‌ی `Categories` اضافه کنید. ولی اینکار را انجام دهید تا در فارسی سازی این پوسته با مشکل روبرو نشویم.

تغییرات را `save` کرده و مرورگرتان را `refresh` کنید. شما بایستی `Categories` را با ساختار `sub-heading` شبیه به زیر بینید:

— [Next Page »](#)



یک گوی کوچک قبل از `sub-heading` شما قابل دیدن هست، دلیل آن این است که درون یک لیست آیتم‌ها (`li`) قرار گرفته. اگر شما ۲ لیست آیتم (`li`) داخل لیست `unordered` یا همان تگ (`ul`) داشته باشید، می‌بایست ۲ گوی را مشاهده کنید، در حقیقت برای هر آیتم یک نقطه یا گوی خواهیم داشت.

● مرحله چهارم :

کد زیر از داخل لیست آیتم یا `li` خودمان اضافه کنید:

```
<ul>
<?php
```

```

wp_list_cats('sort_column=name&optioncount=1&hierarchical=0'); ?>
</ul>

<ul>
    <li><h2><?php _e('Categories'); ?></h2>
        <ul>
            <?php wp_list_cats('sort_column=name&optioncount=1&hierarchical=0'); ?>
        </ul>
    </li>
</ul>

```

❖ معنی این کدها چیست؟

- «
 یک لیست غیر سفارشی باز می کند.
- «
 کدی از php `wp_list_cats();` است که وظیفه‌ی آن صدا زدن دسته‌های وبلاگ می باشد.
- «
 لیست غیر سفارشی را می بندد.

تغییرات را save کرده و مرورگرتان را refresh کنید. حال باید دسته هایتان شبیه به عکس زیر بنظر برسد:

• Categories

- [Personal](#) (11)
- [Sub Category](#) (10)
- [Uncategorized](#) (11)



دسته‌ی پیش فرض شما **Uncategorized** هست که البته وردپرس فارسی آن را دسته‌بندی نشده معنی کرده است. اگر مطالبتان را در چندین دسته‌ی مختلف منتشر نکرده‌اید، شما در بخش دسته‌های وبلاگ باید تنها یک لینک به نام دسته‌بندی نشده داشته باشید.

❖ توضیحات بیشتر:

لیست دسته ها را بر اساس الفبا مرتب می کند. «sort_column=name» «تعداد هر پست را در زیر مجموعه هر دسته نشان می دهد. «optioncount=1» البته اگر مقدار آن را از یک به صفر تغییر دهید دیگر تعداد مطالب درون هر دسته را نخواهید دید.

«hierachial=0» دسته ها را داخل زیر مجموعه sub-list-items نمی چرخاند. «&» هرگاه بخواهید یک مشخصه ای دیگر را اضافه کنید، شما باید & را برای مجزا کردن آن از مشخصه های دیگر تایپ کنید. برای مثال شما & را آوردید ، بین optioncount و sort_column تا این دو را از یکدیگر مجزا کنید.

◆ چرا شما تگ های <?php wp_list_cats(); ?> را بدور و قرار ندادید؟

شما برای اینکه لینک دسته ها را صدا بزنید از wp_list_cats() استفاده می کنید. این کد بطور خودکار گروهی از تگ های لیست آیتم یعنی تگ های و را بدور همه ای لینکها اضافه می کند. برای اینکه بیشتر متوجه شوید به تب view رفته و سپس Source Page یا Source را انتخاب کنید. بعد از اینکه پنجره ای Pop up بازشد، اسکرول را به پایین بکشید و کدهای مربوط به لینک دسته ها را مشاهده کنید، خواهید دید که هر لینک دارای یک گروه از تگ های لیست آیتم به دور خودش می باشد.

هنگام کار با Sidebar توجه کنید که list items و unordered list بسیار مهم هستند، بنابراین بباید بیاورید قانون اول و دوم ما در کد نویسی:

قانون اول: هر چیزی را باز کرده اید را در آخر بیندید.

قانون دوم : تگها بباید بصورت تو در تو در یکدیگر ایجاد شوند.

The right way to close:

```
<ul>
  <li>
  </li>
</ul>
```

The wrong way to close:

```
<ul>
  <li>
  </ul>
</li>
```

درسهای بعدی از این سری آموزشی را دنبال کنید.....

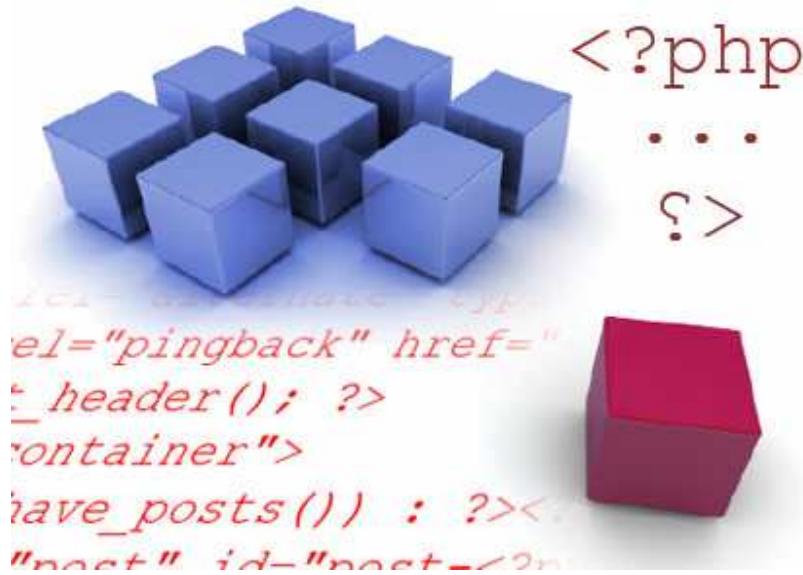
منبع:

<http://www.p30help.ir/how-make-wordpress-theme-lesson-11>

چگونه برای وردپرس پوسته بسازیم؟ (بخش ۱۲)

نوشته شده توسط مهند رادی در تاریخ ۱۰ / ۰۱ / ۱۳۸۸

این آموزش در ادامه مبحث سایدبار (Sidebar) قرار دارد. هر چند که تعویق نسبتاً طولانی بین مباحث درس قبل و درس امروز شما افتاده است، ولی با این حال حداقل باید در خاطرتان باشد که جلسه‌ی قبیل با ساختار Sidebar آشنا شدید. در این درس خواهید آموخت که چگونه سایر عناصر کاربردی را به سایدبارتان اضافه کنید. شاید این سوال برای شما پیش بیاید که کی و چگونه امکان ویجیت یا ابزارک را به پوسته اضافه می‌کنیم؟ خوب جواب این سوال بسیار ساده است، ما در ابتدا به شما یاد خواهیم داد که چگونه یک حالت پیشفرض برای پوسته بسازید و در پایان به شما خواهیم گفت که چگونه امکان ویجیت را به قالب اضافه کنید.



این درس شامل سه بخش می باشد که در هر بخش یاد خواهید گرفت تا چگونه عناصری را به سایدبار اضافه کنید. برای یادگیری این درس عجله نکنید، بلکه سعی کنید تا با آهسته پیش رفتن، تمامی مطالب را به خوبی فرا گیرید.

برای شروع درس به ادامه مطلب مراجعه کنید....

بخش اول – لیست برگه ها

در این بخش از آموزش، با لیست برگه های (Page-link) موجود در سایدبار آشنا خواهیم شد.

کد زیر را در بالای بلوک دسته ها اضافه کنید:

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

```
<ul>
    <?php wp_list_pages(); ?>
    <li><h2><?php _e('Categories'); ?></h2>
        <ul>
            <?php wp_list_cats('sc
        </ul>
    </li>
</ul>
```

تغییرات را ثبت کرده و مرورگرتان را Refresh کنید.

- Pages
 - [About](#)
 - [Parent 1](#)
 - [Child 1 Name iiii](#)
 - [Grandchild 1](#)
 - [Great Grandchild](#)
 - [Grandchild 2 iiiiiiiiiiiiiii](#)
 - [Great Grandchild 1](#)
 - [Child 2 Name](#)
 - [Child 3 Name](#)
 - [Child 4 Name](#)
 - [Child 5 Name](#)
- [Parent 2](#)

- Categories

در حالت پیشفرض، در وردپرس شما فقط یک برگه (معمولًا همان برگه‌ی درباره) وجود دارید. من خودم برگه‌های کوچک دیگری در زیر مجموعه‌ی برگه‌های دیگر اضافه کردم.

به منوی View رفته و Source یا Page Source را بگشایید تا تمام ساختار و همه کدهایی که wp_list_pages() برای شما تولید کرده است را مشاهده کنید. عکس زیر یک مثال است:

```
<li>Pages
    <ul>
        <li><a href="#">Your Link</a></li>
        <li><a href="#">Your Link 2</a></li>
    </ul>
</li>
```

❖ توضیح آنچه اتفاق افتاده است:

اول از همه، هرچیزی داخل لیست آیتم (li) قرار گرفته است. **دوم**، کلمه‌ی **Pages**، نوشته شده است، **سوم**، یک لیست غیرسفارسی یا **unordered list** و یا همان تگ **ul** داخل لیست آیتم (li) قرار گرفته است. **چهارم**، هر لینکی داخلی گروهی از تگ‌های **** و **** قرار گرفته است.

در اسکرین شات بالا، توجه کنید که سایز کلمه‌ی **برگه‌ها** (Pages) در مقایسه با عنوان **دسته‌ها** (Categories) کوچکتر است.

◆ **چطور سایز برگه‌ها و دسته‌ها را متناسب کنیم؟**
برای اینکار کد '`wp_list_pages('title_li=<h2>Pages</h2>')`' را به اضافه می‌کنیم.

```
<?php wp_list_pages('title_li=<h2>Pages</h2>'); ?>
```

تغییرات را ثبت کرده و مرورگرتان را Refresh کنید تا تغییرات را مشاهده کنید.

• Pages

- [About](#)
- [Parent 1](#)
 - [Child 1 Name iiii](#)
 - [Grandchild 1](#)
 - [Great Grandchild](#)
 - [Grandchild 2 iiiii](#)
 - [Great Grandchild 1](#)
 - [Child 2 Name](#)
 - [Child 3 Name](#)
 - [Child 4 Name](#)
 - [Child 5 Name](#)
- [Parent 2](#)

• Categories

بخش دوم – لیست آرشیو و پیوندهای و بلاگ

بخش دوم آموزش امروز مربوط به آوردن بلاگرول و آرشیو و بلاگ می باشد. امیدوارم که بحث قبلی را بخوبی متوجه شده باشید. حال بحث جدید رو آغاز می کنیم.

● مرحله اول:

در بخش اول قصد داریم تا لیست آرشیو را به سایدبار اضافه کنیم، کدهای زیر را در محل سایدبار و زیر لیست دسته ها (Categories) تایپ کنید:

```
<li><h2><?php _e('Archives'); ?></h2>
<ul>
<?php wp_get_archives('type=monthly'); ?>
</ul>
</li>
```

من برای مدیریت بهتر از دکمه TAB کمک گرفته ام. حال باید کدهای ما به شکل زیر شده باشد:

```
<li><h2><?php _e('Categories'); ?></h2>
<ul>
<?php wp_list_cats('sort_column=name&optic
</ul>
</li>

<li><h2><?php _e('Archives'); ?></h2>
<ul>
<?php wp_get_archives('type=monthly'); ?>
</ul>
</li>
```

فایل را ثبت و مرورگرتان را refresh کنید، نتیجه اینگونه خواهد بود:

• Categories

- [Personal](#) (11)
- [Sub Category](#) (10)
- [Uncategorized](#) (11)

• Archives

- [October 2006](#)

❖ چه اتفاقی افتاده است؟

شما از تابع `wp_get_archives()` همراه با مشخصه `monthly` استفاده کرده اید، تا آرشیو و بلاگtan را بصورت ماهیانه صدا بزنید.

— باز کردن لیست آیتم «
sub-heading — باز کردن <h2> «
— کلمه `Archive` را نشان خواهد داد <?php _e('Archives'); ?> «
sub-heading — بستن </h2> «
— باز کردن زیر unordered list و داخل لیست آیتم «
— لینک آرشیو و بلاگtan را <?php wp_get_archives('type=monthly'); ?> «
بصورت ماهیانه صدا می زند، هر لینکی داخل تگ و قرار می گیرد. اگر شما
صفحه تان را چک کنید، خواهید دید که برای هر لینک تابع Source Page
wp_get_archives() بطور خودکار بدور آن قرار گرفته است.
— بسیار شبیه به تابع wp_list_cats()
— بستن unordered list «
— بستن آیتم «

● مرحله ی دو:

در این مرحله قصد داریم تا لینک بلاگرول و یا همان پیوندها را اضافه کنیم:

کد زیر را در پایین لینک آرشیو تایپ کنید:

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

```

<li><h2><?php _e('Archives'); ?></h2>
    <ul>
        <?php wp_get_archives('type=monthly'); ?>
    </ul>
</li>

<?php get_links_list(); ?>

```

تغییرات را save ، مرورگر را refresh و نتیجه را مشاهده کنید:

• Archives

- [October 2006](#)

• Blogroll

- [Mike](#)
- [Dougal](#)
- [Matt](#)
- [Ryan](#)
- [Alex](#)
- [Michel](#)
- [Donncha](#)

در حالت پیش فرض ، بلاگرول من تفاوتی با مال شما ندارد، برای همین سری به کدهای زده تا اوضاع را بهتر بررسی کنیم.

```

<li id="linkcat-1"><h2>Blogroll</h2>
<ul>
<li><a href="http://zed1.com/journalized/">Mike</a></li>
<li><a href="http://dougal.gunters.org/">Dougal</a></li>
<li><a href="http://photomatt.net/">Matt</a></li>
<li><a href="http://boren.nu/">Ryan</a></li>
<li><a href="http://www.alexking.org/">Alex</a></li>
<li><a href="http://zengun.org/weblog/">Michel</a></li>
<li><a href="http://blogs.linux.ie/xeer/">Donncha</a></li>

</ul>
</li>

```

همانگونه که مشاهده می کنید تابع `get_links_list()` خود به تنها یی تمامی کارها را انجام داده و کدهای مورد نیاز را تولید کرده است، بنابراین نیاز به هیچ دستکاری دیگری نیست. در تصویر بالا تگهای `li` و `ul` را با رنگهای قرمز و سبز مشخص کرده ام تا مشاهده کنید که قانون اول نیز به خوبی رعایت شده است. (برای آگاهی از قانون اول به درس اول مراجعه کنید)

بخش دوم – تقویم ، فیلد جستجو و لیست اطلاعات

● مرحله‌ی اول:

در این مرحله یاد خواهیم گرفت که چگونه **فیلد جستجو** را به سایدبار اضافه کنیم.

یک Notepad کاملاً خالی باز کنید. این فایل را با نام **searchform.php** ثبت کنید (در همان فolderی که فایل index.php قرار دارد) اطلاعات درون فایل **searchform.txt** را در فایل searchform.php کپی کنید. حالا فایل را save کرده و از آن خارج شوید.

در فایل index.php کدهای زیر را در بالای لیستی که مربوط به Sidebar می شود، تایپ کنید.

```
<li id="search">
<?php include(THEMEPATH . '/searchform.php') ; ?>
</li>
```

```

<ul>
    <li id="search">
        <?php include(TEMPLATEPATH . '/searchform.php'); ?>
    </li>

    <?php wp_list_pages('depth=3&title_li=<h2>Pages</h2>'); ?>

    <li><h2><?php _e('Categories'); ?></h2>
        <ul>
            <?php wp_list_cats('sort_column=name&option_order=ASC'); ?>
        </ul>
    </li>

    <li><h2><?php _e('Archives'); ?></h2>
        <ul>
            <?php wp_get_archives('type=monthly'); ?>
        </ul>
    </li>

    <?php get_links_list(); ?>
</ul>

```

تغییرات را save کرده و مرورگرتان را نمایید. نتیجه باید مانند زیر باشد:

A search form consisting of a text input field and a blue 'Search' button.

- **Pages**

- [About](#)
- [Parent 1](#)
 - [Child 1 Name iiii](#)
 - [Grandchild 1](#)
 - [Grandchild 2 iiiiiiiiiiiiiiiii](#)
 - [Child 2 Name](#)
 - [Child 3 Name](#)
 - [Child 4 Name](#)
 - [Child 5 Name](#)
- [Parent 2](#)

◆ چه اتفاقی افتاده است؟

یک لیست آیتم با نام آی دی Search شروع می کند. شما به این گزینه id داده اید، بنابراین می توانید بعداً آن را استایل کنید.

بوسیله ای این کد می توانید کدهای هر فایلی را بدرون فایل مورد نظر فراخوانی کنید. در این مورد، کدهای داخل فایل searchform.php درون فایل مورد نظر (در اینجا index.php)، آورده و خوانده می شود. استفاده از این تابع، برای مدیریت آسانتر فایلها انجام می گردد.

TEMPLATEPATH مکانی که فolder پوسته‌ی شما قرار دارد را آدرس می دهد، یعنی wp-content/themes/tutorial /searchform.php ' مکان و نام فایل wp-content/themes/tutorial/searchform.php و شما بوسیله ای آوردن آنها در کنار هم، بین TEMPLATEPATH و 'searchform.php' ارتباط برقرار می کند، و در انتهای دارید: wp-content/themes/tutorial/searchform.php لیست آیتم را می بندد

● مرحله‌ی دو:

در این مرحله یاد خواهیم گرفت که چگونه تقویم را به سایدبارمان اضافه کنیم.

کدهای زیر را در پایین کدهای فیلد جستجو یا بلوک برگه‌ها تایپ کنید:

```
<li id="calendar"><h2><?php _e('Calendar'); ?></h2>
<?php get_calendar(); ?>
</li>

<li id="search">
    <?php include(THEMEPATH . '/searchform.php'); ?>
</li>

<li id="calendar"><h2><?php _e('Calendar'); ?></h2>
    <?php get_calendar(); ?>
</li>
```

تغییرات را ثبت کرده و مرورگرتان را Refresh کنید. نتیجه اینگونه خواهد بود.



• Calendar

March 2007						
M	T	W	T	F	S	S
1	2	3	4			
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	

[« Oct](#)

◆ چه اتفاقی افتاده است؟

لیست آیتم ها را با id بنام Calendar باز می کند `<li id="calendar">` « تگ sub-heading را شروع می کند `<h2>` « کلمه ی Calendar را نمایش می دهد `<?php _e('Calendar'); ?>` « تگ sub-heading را می بندد `</h2>` « تقویم را با استفاده از تابع `get_calendar()` صدا می زند `<get_calendar()>` « لیست آیتم را می بندد `` «

خوب عملیات تقویم نیز بخوبی انجام شد.

● مرحله ی سوم:

و در مرحله ی آخر این درس یاد خواهیم گرفت که چگونه بخش اطلاعات را به سایدبار اضافه کنیم

در این مرحله قصد داریم تا اطلاعات و بلاگ را که شامل عضویت در و بلاگ، ورود به و بلاگ، خروج از و بلاگ، ورود به بخش کنترل پنل و ... است را به سایدبار اضافه کنیم.

برای اینکار کدهای زیر را در پایین کد `get_links_list()` تایپ کنید:

```
<li><h2><?php _e('Meta'); ?></h2>
<ul>
<?php wp_register(); ?>
<li><?php wp_loginout(); ?></li>
<?php wp_meta(); ?>
</ul>
</li>

<?php get_links_list(); ?>

<li><h2><?php _e('Meta'); ?></h2>
<ul>
<?php wp_register(); ?>
<li><?php wp_loginout(); ?></li>
<?php wp_meta(); ?>
</ul>
</li>
```

تغییرات را ثبت کنید و نتایج را در مرورگر تان مشاهده نمایید.

(اگر بعنوان مدیر وارد وبلاگ نشده باشید نتایج اینگونه خواهد بود)

- **Meta**

- [Register](#)
- [Login](#)

(اگر وارد شده باشید.)

- **Meta**

- [Site Admin](#)
- [Logout](#)

◆ چه اتفاقی افتاده است؟

شما کار را با یک لیست آیتم (li) با یک sub-heading (ul) یا همان H2 برای Meta شروع کردید. پایین sub-heading ، شما یک ul یا همان تگ unordered list وارد کرده اید. و برای هر لینکی، شما تگ لیست آیتم (li) را بدور آنها قرار داید.

تابع wp_register() خودش به تنها یک تگهای و را تولید می کند، وقتی شما داخل وبلاگ نشده باشید، آن لینک Register را برای شما نشان می دهد، وقتی که شما داخل وبلاگ شده باشید، آن به شما لینک Site Admin را نشان خواهد داد، تابع wp_loginout() خودش تگهای لیست آیتم را تولید نمی کند، بنابراین شما با استفاده از و را بدور آنها قرار دهید. این کد هنگامیکه داخل سایت نشده باشید، به شما لینک Login را نشان می دهد، و هنگامی که داخل وبلاگ شده باشید، به شما لینک Logout را می دهد. و تابع wp_meta() کار خاصی نمی کند، این کد یک چیز مخفی در صفحه ی وب و همچنین یک چیز مخفی در Source کد می باشد. الان در مورد wp_meta() فکر نکنید، شما از این تابع به ندرت استفاده خواهید کرد.

حال تقریباً سایدبار شما ساخته شده است، در جلسه ی بعد به شما نحوه ی اضافه کردن قابلیت ابزارک را خواهیم گفت. می دانم که بی صبرانه برای آن منتظرید.

منبع :

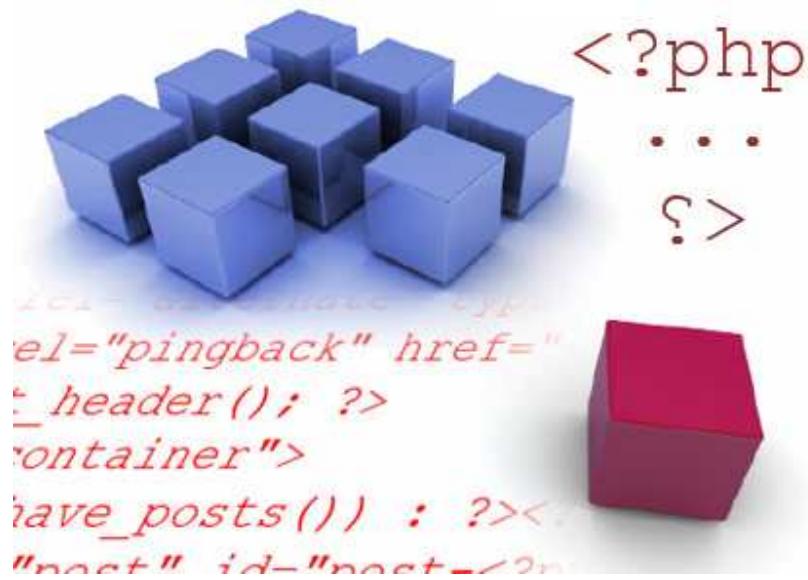
<http://www.p30help.ir/how-make-wordpress-theme-lesson-12>

چگونه برای وردپرس پوسته بسازیم؟ (بخش ۱۳)

نوشته شده توسط مهندی رادی در تاریخ ۱۳۸۸ / ۰۱ / ۲۰

همانطور که به شما قول داده بودم، در این جلسه به شما خواهیم گفت که چگونه امکان ویجیت را برای سایدبار خود فعال کنیم. اما در این جلسه علاوه بر مبحث ویجیت، بحث فوئر و اعتبار سنجی (Validate) صفحات را نیز مطرح خواهیم کرد، تا با یاری خدا در جلسات بعد به طور جدی به بحث CSS پردازیم. با توجه به اینکه مباحث بسیار زیاد است، تصمیم گرفته ام که بعد از اتمام این سری آموزشی، سوالات متداول شما

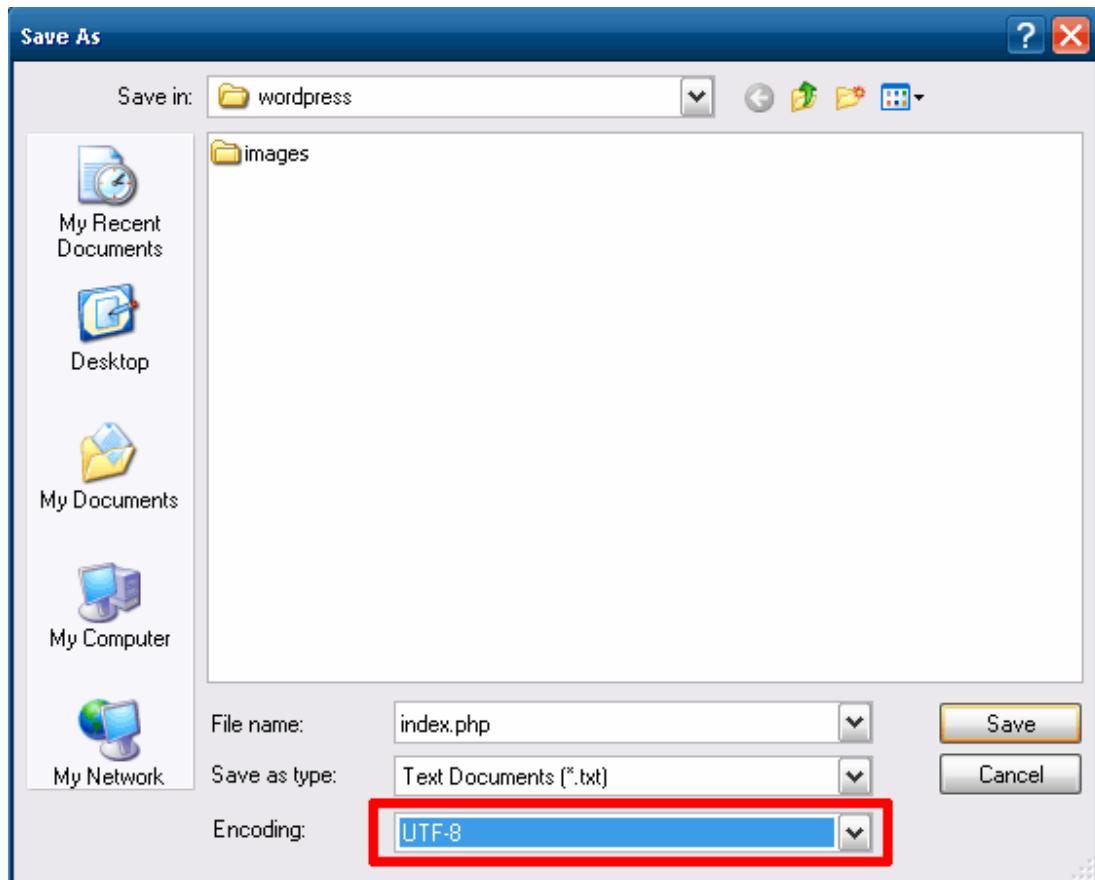
دوستان را همراه با جواب بطور پست های جداگانه مطرح کنم تا از اشکالات شما دوستان قدری بکاهم.



فعلاً باید سعی کنیم تا همین سری آموزشی به پایان برسد، برای شروع درس به ادامه مطلب مراجعه کنید...

قبل از اینکه درس را بطور جدی شروع کنم، به این نکته که یکی از دوستان به من اطلاع داد، اشاره می کنم. نکته‌ی کوچک ولی مهمی که من فراموش کرده بودم به اطلاع شما دوستان برسانم، این است که مدل **Encoding** فایل‌های ذخیره شده تان را **UTF-8** قرار دهید. از هر ادیتوری که می خواهید استفاده کنید ولی Encoding را **utf-8** قرار دهید، اگر تا حال اینکار را انجام نداده اید، هیچ مشکلی ندارد، ولی از همین الان اینکار را انجام دهید تا در بعدها با مشکل مواجه نشوید. هنگامی که مدل Encoding شما **utf-8** باشد، در هنگام فارسی نویسی با هیچ مشکلی مواجه نخواهید شد. خوب شاید به شما کمی به شما استرس وارد کرده ام. هیچ غصه‌ای نخورید چون اینکار در هر مرحله‌ای براحتی می تواند انجام شود و مشکلاتی که برای شما بوجود خواهد آمد براحتی قابل تعمیر است، البته اگر همین الان اینکار را انجام دهید، هیچ مشکلی را برای شما بوجود خواهد آمد چون هیچ متن فارسی درون فایلهایتان قرار ندارد.

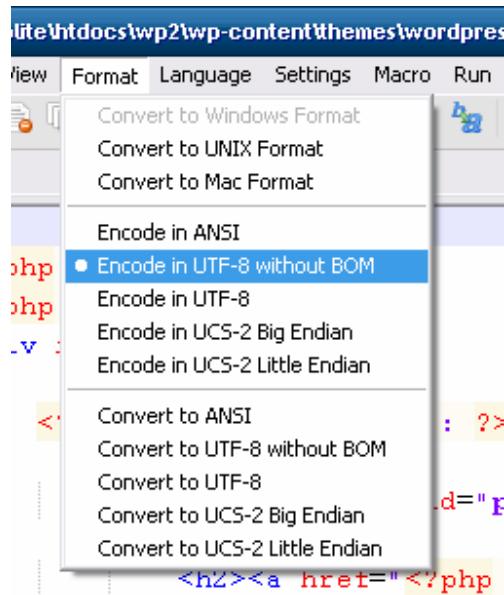
خوب برای اینکه کار را برای شما آسان کنم نحوه‌ی اینکار را به شما در Notepad نشان می‌دهم. هر یک از فایلهای مورد نظرتان را در Notepad باز کنید. به منوی فایل رفته و بروی Save As کلیک کنید. در صفحه‌ای که باز شد. در بخش Encoding گزینه‌ی UTF-8 را انتخاب کنید دقیقاً مانند عکس زیر:



فایل را دقیقاً بروی فایل قبلی ذخیره کنید. این اعمال را بروی تمامی فایلهای داخل پوشه انجام دهید. به همین سادگی....

اما اگر اینکار برایتان کمی سخت بود، یک راه بهتر و آسانتر نیز برایتان سراغ دارد. کار کردن با Notepad ساده‌ترین راهی است که می‌توان استفاده کرد، من خودم به شخصه از برنامه‌ی [Notepad++](#) استفاده می‌کنم. به شما هم پیشنهاد استفاده از این برنامه را می‌دهم. حال اگر از این برنامه استفاده می‌کنید یا خواهید کرد، برای تبدیل پاراگراف زیر را مطالعه کنید:

روش کار بسیار ساده است، فایل مورد نظر را باز کنید، به تب Format بروید و از آنجا گزینه‌ی Encode in UTF-8 without BOM را انتخاب کنید. مانند عکس زیر:



حال فایل را Save کنید. به همین راحتی به همین خوشمزگی...
تازه درسمان شروع خواهد شد، امیدوارم که خسته نباشد....

بخش اول - فعال سازی Widget برای سایدبار:

❖ اما شاید برای شما این سوال پیش بیاید که ویجیت چیست؟

در جواب این سوال باید گفت که ویجیت (Widget) همان بخشی از وردپرس است که در فارسی به نام ابزارک معنی شده است. در واقع ابزارک‌های وردپرس، لوازمی برای سایدبار می‌باشند که شما برای توانید آنها را در سایدبار سایت خود به نمایش درآورید. حقیقتاً ابزارک‌ها کار را برای شما آسان می‌کنند و به مدیران سایت اجازه می‌دهند تا بدون داشتن دانش برنامه نویسی وب، بتوانند برای سایت اجزا گذاشت و ... را درون سایدبار و بلاگشان به نمایش درآورند و یا مکان آنها را جابجا کنند. جالب است بدانید که در ورژن‌های قبلی وردپرس امکان ابزارک به این نوع وجود نداشته است و برای استفاده از قابلیت ابزارک، باید یک پلاگین خاص را نصب می‌کردیم، اما وردپرس به سرعت پی‌برد که این بخش یکی از قسمتهای بسیار لازم برای

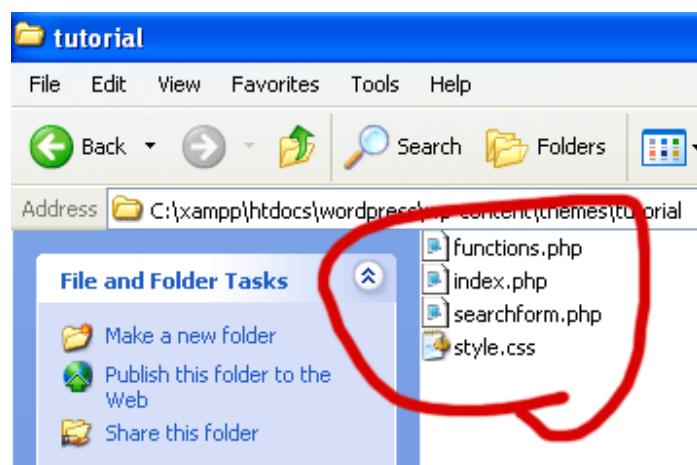
وردپرس می باشد و در بسته های بعدی این قسمت را بطور مستقیم درون وردپرس جای داد.

امیدوارم که توضیحات کوتاه بنده برای شما کافی باشد. اما برای فعال کردن قابلیت ویجیت مراحل زیر را پی بگیرید:

● مرحله‌ی اول: ساختن فایل functions.php

یک Notepad جدید باز کنید، که کاملاً خالی باشد، و آن را با نام functions.php ثبت کنید (البته توجه کنید که Encoding فایل 8 باشد). هر چیزی که داخل فایل وجود دارد را، به درون فایل **functions.php** کپی کنید و سپس **functions.txt** تغییرات مورد نظر را Save کرده و سپس صفحه‌ی Notepad را ببندید.

الان شما باید چهار فایل را درون فolder **tutorial** داشته باشید.



● مرحله‌ی دوم: ایزارک در سایدبار

کد زیر را مستقیماً بعد از اولین خط تگ در بخش سایدبار تایپ کنید:

```
<?php if ( function_exists('dynamic_sidebar') &&
dynamic_sidebar() ) : else : ?>
```

```
<ul>

<?php if ( function_exists('dynamic_sidebar') && dynamic_sidebar() ) : else : ?>

    <li id="search">
        <?php include(TEMPLATEDPATH . '/searchform.php'); ?>
    </li>

    <li id="calendar"><h2><?php _e('Calendar'); ?></h2>
        <?php get_calendar(); ?>
    </li>
```

این کد را نیز به طور مستقیم قبل از تگ **** تایپ کنید:

```
<?php endif; ?>
```

```
<?php endif; ?>

</ul>

</div>

</body>
</html>
```

حال فایل **index.php** را save کنید، شما هیچ تغییری را مشاهده نخواهید کرد تا زمانی که در قسمت ابزارک ها از یک ویجیت استفاده کنید، که در این صورت تغییرات اعمال خواهد شد.

همین تمام شد. باورتان می شد که اینکار تا این حد ساده باشد.

بخش دوم – اضافه کردن فوتر (Footer) :

این بخش از درس نیز بسیار ساده است، شما تنها باید یک **div** یا باکس مخفی زیر بخش **div** سایدبار اضافه کنید و داخل این تگ را با چیزهای مثل نوشته‌ی کپی رایت پر کنید. فکر کنم شما اینکار را بدون هیچگونه دستور العملی نیز می توانید انجام دهید، اینکار را خود به تنها یی انجام دهید و سپس برگردید و ادامه مطلب را بخوانید، تا متوجه شوید که تا چه میزان کار را درست انجام داده اید:

• مرحله‌ی اول : اضافه کردن **div**

کدهای زیر را در زیر **div** مربوط به Sidebar تایپ کنید:

```
<div id="footer">  
</div>
```

```
</div>end of Sidebar
```

```
<div id="footer">
```

```
</div>
```

```
</body>
```

```
</html>
```



● مرحله ی دوم : اضافه کنید متن فوتر

داخل تگ پاراگراف (**<p>** و **</p>**) هر نوشته ای را که مایلید، تایپ کنید. البته معمولاً این نوشته ها شامل کپی رایت می شوند:

```
<p>  
Copyright © 2008 <?php bloginfo('name'); ?>  
</p>
```

```
<div id="footer">  
<p>  
Copyright © 2008 <?php bloginfo('name'); ?>  
</p>  
</div>
```

این هست نتیجه: Refresh را کرده و مرورگرتان را save notepad کنید،

- Archives

- [October 2006](#)

Copyright © 2008 Demo Theme Development

حروف **©** کدهایی هستند که علامت کپی رایت (©) را به شما نشان می‌دهند. همچنین، بیاد بیاورید که **bloginfo** تابعی هست که از آن برای قرار دادن نام عنوان سایت در هدر استفاده کردیم که من دوباره از آنها در فوتر استفاده کردم. همچنین بدانید که کلمه **name** داخل **bloginfo** عنوان وبلاگ و همچنین **url** داخل **bloginfo** آدرس وبلاگ را صدا می‌زند.

فوتر نیز به همین سادگی ساخته شد.

بخش سوم- اعتبار سنجی (Validate) صفحات:

قبل از اینکه به شما CSS و کار با فایل Style.css آموزش دهیم، شما نیاز دارید که بدانید چگونه برای کدهایتان اعتبار نامه بگیرید.

قبل از این نحوه اعتبار سنجی را برای شما توضیح دهیم باید بگوییم که برای برنامه نویسی وب یکسری قوانین وجود دارد، حال اگر شما از قوانین تبعیت کردید، در حقیقت برای صفحات خود گواهی نامه را نیز دریافت خواهید کرد، در غیر اینصورت صفحات شما گواهی ندارد. البته اصلًا نگران نباشید. چون انصافاً رعایت تمامی این قوانین آنچنان آسان نیست و نداشتن گواهی نیز نقص بزرگی به حساب نمی‌آید. این را نیز بگوییم که ظاهر یک سایت هیچ ربطی به گواهی داشتن یا نداشتن ندارد، چه بسا سایتهای بسیار زیبایی که گواهی نداشته و سایتهایی تا زیبایی که گواهی دارند. این را نیز بدانید که بسیاری از سایتهای بزرگ فعلی نیز قادر این گواهی نامه می‌باشند.

ایراد را می‌دهد. من به شما دو لینک [CSS Validator](#) و [XHTML Validator](#) را می‌دهم، که از این دو لینک قادر به بررسی اعتبار صفحات تان می‌باشید. لینک [CSS Validator](#) برای اعتبار سنجی صفحات اصطلاحاً html می‌باشد و لینک [XHTML Validator](#) نیز برای اعتبار سنجی کدهای CSS می‌باشد که البته هنوز در این رابطه چیزی نخوانده اید.

خوب برای اعتبار سنجی روی [XHTML Validator](#) کلیک کرده تا صفحه آن باز شود. حال به صفحه ای که طراحی کرده ایم در مرورگر بروید به تب View رفته و سپس را انتخاب کنید. پنجره ای برای شما باز می‌شود. شما باید Source Page Source

همه کدها را انتخاب کنید، برای اینکار می توانید به سادگی دکمه *Ctrl+a* را فشار دهید. سپس همه کدها را انتخاب کنید، برای اینکار نیز می توانید به سادگی دکمه *Ctrl+c* را فشار دهید.

به صفحه [Validator XHTML](#) باز گردید و در داخل بخش **Validate by Direct Input** تمامی کدها را Paste کنید برای اینکار می توانید از *Ctrl+v* استفاده کنید. حال بروی دکمه **Check** کلیک کنید.

Validate by Direct Input

Input the markup you would like to validate in the text area below:

```
</div>

<div id="footer">
<p>
Copyright © 2007 <a href="http://localhost/wordpress">Development</a>
</p>

</div>

</body>
</html>
```

کدهای شما توسط **validator** چک خواهد شد، سپس نتیجه برای شما نمایان خواهد گشت. اگر پیغام سبز بود، یعنی شما هیچ ایرادی نداشتید. و اگر قرمز بود، معلوم می شود که توانسته اید اعتبار لازم را دریافت کنید. البته validator به شما می گوید که در کجا ها ایراد داشتید، که اینکار شما را در ایراد یابی کدها یاری خواهد کرد.

Result: Passed validation

File: upload://Form Submission

Encoding: utf-8

Doctype: XHTML 1.0 Transitional

Root Namespace: <http://www.w3.org/1999/xhtml>

درس امروز ما که شامل سه بخش بود به پایان رسید، در جلسات بعدی به شما خواهیم گفت که چگونه با CSS کار کنید، مطمئناً نفس در سینه تان حبس شده است، ولی نگران نباشید، خیلی ساده است....

منبع:

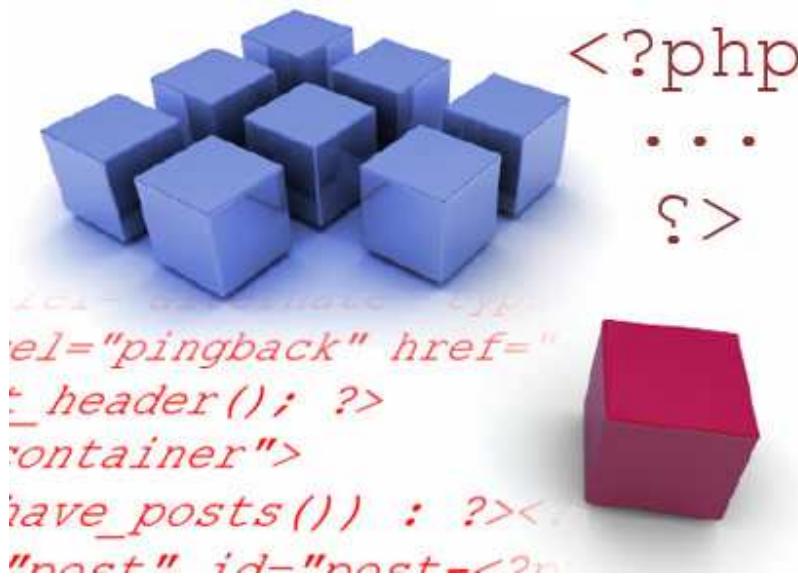
<http://www.p30help.ir/how-make-wordpress-theme-lesson-13>

چگونه برای وردپرس پوسته بسازیم؟ (بخش ۱۴)

نوشته شده توسط مهند رادی در تاریخ ۱۳۸۸/۰۱/۲۹

آشنایی با فایل Style.css و مقدمات CSS

انتظار ها به سر رسید و جلسه ای کار با Css هم از همین الان شروع شد، می دام که زمان زیادی بود که در انتظار این لحظه بوده اید. در این جلسه به شما مقدمات و Css را خواهیم گفت، کار با css بسیار ساده است و تنها باید ماهیت آن را درک کنید، از طرفی چون مبحث گسترده ای است، ما نمی توانیم همه آن را در این سری آموزشی به شما بگوییم، فقط مباحثی که به کارمان مربوط است. البته سعی داریم که ماهیت Css را برای شما تشریح کنیم که در این صورت ۹۲٪ مشکلات برای شما حل خواهد شد.



البته قبل از شروع یادگیری Css باید تراژدی عدم هماهنگی آن با مرورگرهای مختلف نیز گفته شود، مشکلی که بسیاری از اختیارات را از طراحان قالب سلب می‌کند. امیدوارم که در ابتدای درس به شما استرسی وارد نکنم. نگاه کنید اگر بخواهم قضیه را خیلی ساده برای شما شرح دهم، باید بگویم که مرورگرهای مختلف از دستورات CSS برداشتهای گوناگونی دارند(البته خیال نکنید که این برداشتها بسیار متفاوتند)، از همین جهت ممکن است قالبی در مرورگر فایرفاکس به خوبی نشان داده شود ولی در مرورگر ایترنوت اکسپلورر با ایراد و عدم هماهنگی اجزای صفحه نمایش داده شود. خود من به شخصه در بین مرورگرها می‌توانم فایرفاکس را فوق العاده نام ببرم(البته اپرا را امتحان نکرده ام)، به گونه ای که اکثر قالبها بدون مشکل و دردرس و یک سیستم منطقی نشان داده می‌شوند. اما از طرفی ایترنوت اکسپلورر و بخصوص ایترنوت اکسپلورر ۶ بسیار شما را از بابت طراحی آزار خواهد داد (انگار که در این مرورگر منطقی حکم فرما نیست). یکی از آروزهای طراحان براندازی ایترنوت اکسپلورر ۶ است که امیدوارم تحقق یابد. از درد و دل که بگذریم، بهتر است روش کار را به شما بگوییم.

شما روش‌های گوناگونی را برای کار با Css می‌توانید، استفاده کنید:
روش ۱ - چندین مرورگر مختلف را باز کرده و هر تغییری را که اعمال می‌کنید در تمامی مرورگرها بخصوص در ایترنوت اکسپلورر مشاهده کنید و در صورت مشاهده ای ایراد، مشکلات را برطرف کنید. این روش سخت ترین روش موجود است، چرا که

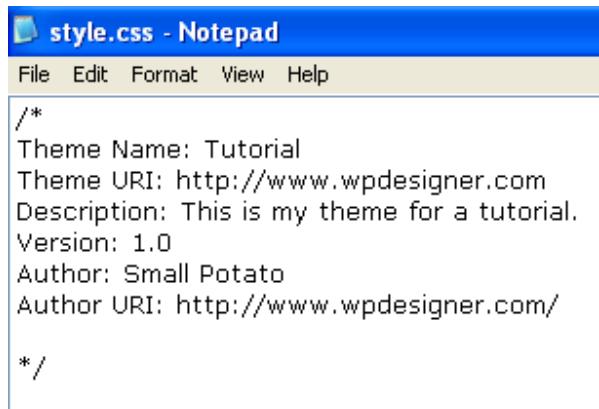
هماهنگی قالب با تمامی مرورگر ها کار آسانی نیست، البته باید در اینجا این نکته را عرض کنم که تنها اینترنت اکسپلوررها هستند که تغییرشان از قوانین CSS خیلی متفاوت است، سایر مرورگرها مانند اپرا ، فایرفاکس ، گوگل کروم و ... تغییر بسیار مشابه ای دارند و از این جهت کار شما خیلی آسان است. فکر نکنم که از این روش خوش تان آمده باشد، پس روش دوم که روش خود من است را عمل کنید:

روش ۲ – من به شخصیه ابتدا با مرورگر فایرفاکس قالبم را می سازم و در آخر هنگامی که کاملاً ساخته شد، یک CSS جدا برای مرورگر های اینترنت اکسپلورر ۶ و ۷ ساخته و قالب را با آنها سازگار می کنم. انصافاً این روش خیلی ساده تر است. این نکته را نیز همینجا بگوییم که شما می توانید یک CSS جدا ساخته که برای مثال فقط در اینترنت اکسپلورر ۷ یا فقط در اینترنت اکسپلورر ۶ قابل خواندن و استفاده باشد. چگونگی انجام اینکار را بعداً به شما خواهم گفت ولی اگر باز فراموش کردم، حتماً یادآوری کنید....

خوب به ادامه مطلب برای شروع درس مراجعه کنید....

بهترین راه برای یادگیری CSS غوطه ور شدن در آن است، برخلاف PHP و XHTML ، شما نیاز ندارید که فایلهای مرکز قالب تان را لمس کنید. همچنین در CSS بهترین روش یادگیری، روش آزمون و خطا می باشد.

اگر فایل Style.css را که درون پوشه **tutorial** وجود دارد، باز کنید. **خط ۶** را مشاهده خواهید کرد، این خط فقط برای وردپرس معنی می دهد که در ادامه معنای آنها را خواهیم گفت:



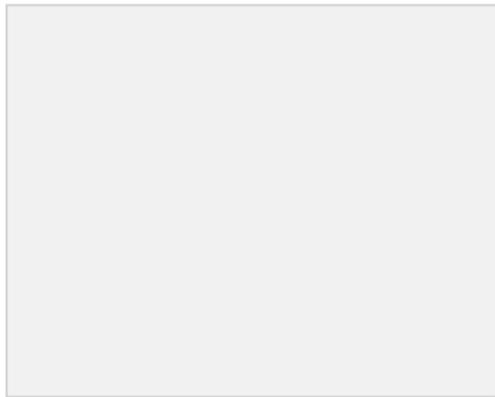
```
style.css - Notepad
File Edit Format View Help
/*
Theme Name: Tutorial
Theme URI: http://www.wpdesigner.com
Description: This is my theme for a tutorial.
Version: 1.0
Author: Small Potato
Author URI: http://www.wpdesigner.com/
*/
```

- * خط اول مربوط به نام پوسته است که الان **Tutorial** نوشته شده است، پس در وردپرس نیز قالب شما با این نام نشان داده خواهد شد. می توانید هر نامی که خواستید را برای این قسمت بگذارید.
- * خط دوم مربوط به آدرس صفحه پوسته در وب می باشد، چون بعضی از پوسته ها توضیحات اضافی نیز دارند.
- * خط سوم توضیح خود پوسته است که در وردپرس نشان داده خواهد شد.
- * خط شماره ی چهارم که دیگر تقریباً مشخص است. درباره ی ورژن پوسته ی شما می باشد، چون ممکن است برای یک پوسته چندین ورژن ارائه داده باشید، در اینجا می توانید معین کنید قالب شما الان چه ورژنی است.
- * خط پنجم و ششم، نام سازنده ی قالب و آدرس وب او می باشد، که در اینجا باید نام خود و آدرس وب سایت خود را وارد کنید.

علامت **/*** و ***/** بدور اطلاعات پوسته ی شما قرار می گیرد تا محتوای اصلی فایل Css شما از اطلاعات پوسته ی شما مجزا باقی بماند و با هم ادغام نشود. هنگام کار با Css گاهی اوقات لازم است برای بعضی از خطها توضیح بگذارید، تا هم خود شما و هم دیگرانی که از این قالب استفاده می کنند، سردرگم نشوند. برای اینکار توضیحات خود را داخل علامت **/*** و ***/** قرار دهید، تا مشکلی برای فایل Css تان پیش نیاید. اگر از استفاده کنید، خواهید دید که نوشته هایی که بین این دو علامت قرار بگیرند با رنگ **سبز** مشخص خواهند شد.

اگر به صفحه ی پوسته در بخش مدیریت وردپرس سری بزنید، مشاهده خواهید کرد که پوسته ی شما اینگونه نشان داده خواهد شد.

Tutorial 1.0



This is my theme for a tutorial.

حال شروع آموزش (انصافاً مقدمه خيلي طولاني شد)

● مرحله ی اول :

- * کنترل برنامه‌ی Xampp ، فolder پوسته‌ها، یک مرورگر فایرفاکس و فایل Style.css را باز کنید.
- * مرورگر فایرفاکس را به آدرس <http://localhost/wordpress> هدایت کنید.

نکته: اگر می‌خواهید از روش اول استفاده کنید، یک مرورگر اینترنت اکسپلورر نیز باز کنید.

اگر از روش اول استفاده می‌کنید بهتر از از چند مرورگر کمک بگیرید ولی اگر تنبل هستید بازکردن مرورگر فایرفاکس و اینترنت اکسپلورر نیز کفایت میکند، ولی در کل روش اول روش خوبی نیست.

(توجه: در این درس نیازی نیست که فایل Index.php را باز کنید)
اگر هم برنامه‌ی Notepad++ را نصب نکرده‌اید و قادر هم نیستید که فایل Style.css را در Notepad باز کنید، بروی فایل Style.css راست کلیک کرده در زیرمجموعه Open with گزینه‌ی Choose Program ... را انتخاب کنید و در پنجره‌ی باز شده، Notepad را انتخاب کرده و ok را کلیک کنید.

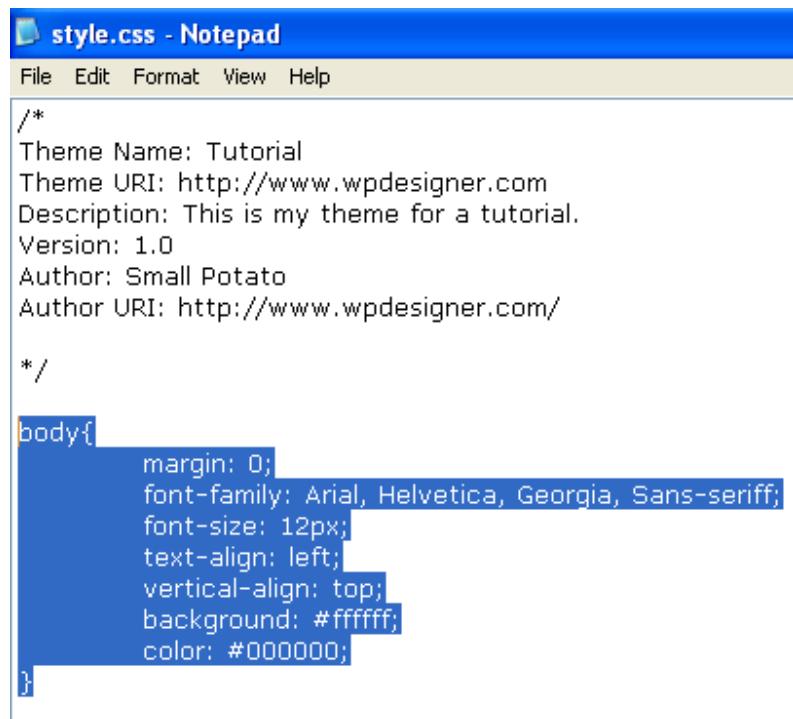
● مرحله ی دوم:

کدهای زیر را در فایل Style.css تایپ کنید :

```
body{
margin: 0;
font-family: Arial, Helvetica, Georgia, Sans-serif;
font-size: 12px;
text-align: left;
vertical-align: top;
background: #ffffff;
color: #000000;
}
```

نکته: فعلاً پوسته را چپ به راست درست می کنیم، در جلسه‌ی آخر یه بخش کوچک در مورد فارسی سازی یا راست به چپ کردن قالب به شما خواهیم گفت.

عکس زیر را بنگرید تا متوجه شوید که چه می گوییم....



The screenshot shows a Windows-style Notepad window titled "style.css - Notepad". The menu bar includes File, Edit, Format, View, and Help. The code in the editor is as follows:

```
/*
Theme Name: Tutorial
Theme URI: http://www.wpdesigner.com
Description: This is my theme for a tutorial.
Version: 1.0
Author: Small Potato
Author URI: http://www.wpdesigner.com/

body{
margin: 0;
font-family: Arial, Helvetica, Georgia, Sans-serif;
font-size: 12px;
text-align: left;
vertical-align: top;
background: #ffffff;
color: #000000;
}
```

تغییرات را در فایل Style.css ثبت کنید و صفحه را refresh کنید، تا تغییرات را مشاهده کنید.

علامت { یعنی باز و } به معنی بسته بودن است و داخل { و } علامت دو نقطه (:) یعنی شروع مقدار دهی و سیمی کولن (;) یعنی توقف کد موردنظر.

اما به چه دلیل ما کلمه‌ی { body (سلکتور CSS) را در CSS مورد استفاده قرار دادیم. دلیل این هست که هر چیزی از صفحه‌ی وب بین تگهای <body> و </body> قرار دارد، در نتیجه هنگامی که شما به این تگ استایل بدهید، یعنی به کل صفحه‌ی وب استایل داده اید، بنابراین کدهایی که در این استایل قرار دارند بروی کل صفحه‌ی وب اعمال خواهند شد. بعدها عناصر جزئی تر صفحه را نیز استایل خواهیم داد.

❖ توضیحات بیشتر:

« margin: 0 » توضیح این مورد کمی برای من سخت است، البته ترسید، کمی توضیح دادنش برای شما دشوار است، در حقیقت زمانی به درستی این موضوع را خواهید فهمید که خود شما دست به کار شده و با آزمون و خطا متوجه شوید که این گزینه‌ها برای چه هستند (این گزینه را خیلی راحت می‌توانید درک کنید، مشکل اصلی برای فهمیدن Padding هست که در جلسات بعد به شما خواهم گفت و کاری شبیه به همین margin را انجام می‌دهد. مهم نیست..... درس امروز را متوجه شوید، کافیست). در حقیقت margin یک حاشیه برای شما بوجود می‌آورد و فضای را به مقداری که شما لازم دارید تخلیه می‌کند. الان مقدار margin برابر با صفر است یعنی هیچگونه حاشیه یا فضای خالی نخواهید داشت ولی اگر می‌خواهید یک حاشیه داشته باشید، عدد صفر را به ۱۰ px یا ۲۰ px، یا غیره... تغییر دهید. برای هر عددی که می‌نویسید باید واحد آن را نیز بیان کنید. تنها عدد صفر است که اگر واحد آنرا ننویسید مشکلی پیش نمی‌آید.

نکته ۱: هر پیکسلی یک نقطه روی صفحه‌ی کامپیوتر است.

نکته ۲: خیال نکنید که این حاشیه‌ای که در مورد آن صحبت می‌کنیم قابل دیدن است، این حاشیه یه حاشیه نامرئی می‌باشد. در حقیقت **حاشیه** در اینجا معنی **فاحله** را می‌دهد.

در عکس زیر، هایلایت قرمز فضایی هست که بوسیله‌ی تگ body بصورت پیش فرض برای margin اعمال شده است. (این رنگ قرمز را خود من ایجاد کرده‌ایم)

Demo Theme Development

Just another WordPress weblog

Hello World

بعد از استایل کردن و قرار دادن مقدار صفر برای margin ، همانگونه که در عکس زیر می بینید دیگر فاصله ای وجود ندارد. یعنی نوشته کاملاً به سمت چپ چسبیده است و هیچگونه فاصله ای وجود ندارد. خود شما بهتر است حالت های گوناگون را تست کنید تا بهتر متوجه شوید.



Demo Theme Development

Just another WordPress weblog

Hello World



« **font-family: Arial, Helvetica, Georgia, Sans-serif;** – این خط، فونت هایی را برای صفحه ی وب، تعریف می کند. همانگونه که مشاهده می کنید، در این قسمت اسم چهار فونت نوشته شده است، حال فایل Style.css در ابتدا از فونت Arial برای نشان دادن متنها استفاده می کند. اگر کاربری فونت Arial را بروی کامپیوترش نداشت، فایل Style.css نوشته های صفحه ی وب را با فونت Helvetica برای کامپیوتر کاربر نشان خواهد داد، به همین ترتیب فونت Georgia و در آخر Sans-serif در صورت نداشتن سه فونت اولی نشان داده خواهد شد. شما می توانید هر فونتی را که مایل بودید در این قسمت قرار دهید.

« **font-size: 12px;** – این گزینه مشخص است، شماره را بزرگتر و یا کوچکتر کنید تا تغییرات را مشاهده کنید.

— این گزینه نوشته های صفحه را به سمت چپ، تراز می کند. این کد را به `text-align: right;` تغییر دهید، تا تغییرات را مشاهده کنید.

— این کد هر چیزی در صفحه را از بالا مقایسه می کند. اگر شما مقدار را بجای `top` به `middle` و یا `bottom` تغییر دهید، آنگاه تگ `body` از همان طرف مقایسه می شود. امتحان کردنش رایگان است....

— این کد یعنی، رنگ پس زمینه برای صفحه وب، که در اینجا ما رنگ را سفید قرار داده ایم. عبارت `#fffffff` یک کد hex برای سفید و مقدار `#000000` یک کد hex برای سیاه می باشد. در جلسات بعد خیلی خوب با این کدها آشنا می شوید، الان مغز خود را زیاد درگیر این قضیه نکنید.

— این کد مربوط به رنگ فوتها یا نوشتها می شود، که در اینجا رنگ متن ها را سیاه قرار داده ایم.

پیشنهاد می کنم که برای یادگیری بیشتر CSS حتماً به سایت w3schools.com نیز سری بزنید....

پ.ن: بعد از تغییراتی که ایجاد کردید، و آزمایشها یی که بروی کدها انجام دادید، تمامی تغییرات را به حالت اول بازگردانید.

منبع:

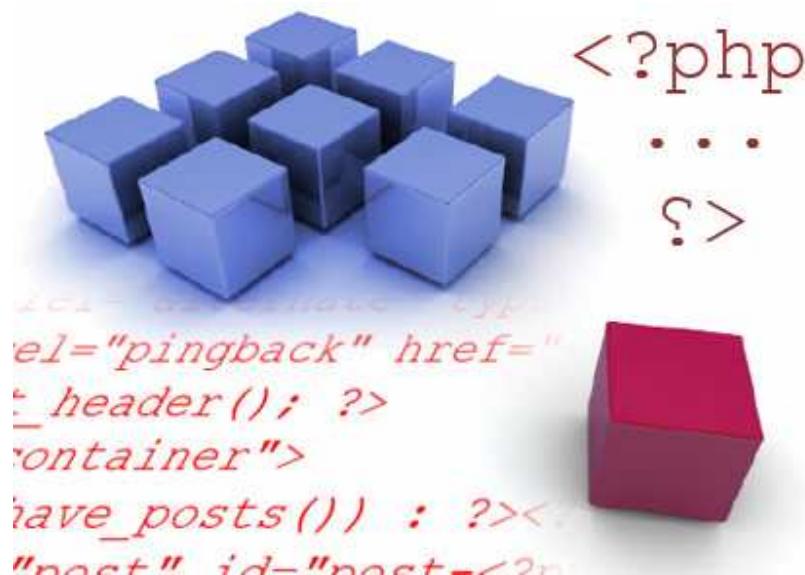
<http://www.p30help.ir/how-make-wordpress-theme-lesson-14>

چگونه برای وردپرس پوسته بسازیم؟ (بخش ۱۵)

نوشته شده توسط مهندی رادی در تاریخ ۱۲ / ۰۲ / ۱۳۸۸

درس پانزدهم : کدهای Hex و استایل دهی لینکها

با نام خدا درس جدید رو شروع می کنیم. درس امروز کوتاه است و چون این روزها کمی سرم شلوغ است، از همین جهت خواهشان زیاد اعتراض نکنید. قرار شد در این جلسه کدهای Hex را مورد بررسی قرار بدهیم. ما در این جلسه می خواهیم رنگ آمیزی بیشتری را برای اجزای صفحه انجام دهیم. پس چیزهای بیشتری را در مورد کدهای شانزده تایی (Hex Codes) خواهیم آموخت.



برای خواندن کامل متن به ادامه مطلب بروید...

این تنظیمات از یک کد شانزده تایی (hexadecimal) که مخفف آن hex است، پیروی می کند. مثلًا برای رنگ نوشه ها کد body { color: #000000;} این معنی را می دهد که تمامی نوشه ها در بدنه ای سایت، به رنگ سیاه در خواهند آمد.

همچنین شما می توانید رنگ پس زمینه ای سایت خود را به رنگ های مختلفی در بیاورید. برای مثال کد body{ background: #ffffff;} به این معنی است که پس زمینه ای صفحه ای شما سفید خواهد بود.

❖ حال توضیح و نحوه ای عملکرد کدهای شانزده تایی (Hexadecimal Codes) :

تمامی کدهای Hex شش رقمی هستند، همچنین قبل از این کدها یک علامت # می آید. رنگ این کدها از شماره #ffffffff که رنگ سفید است تا # ۰ ۰ ۰ ۰ ۰ ۰ که رنگ سیاه است، ادامه دارد.

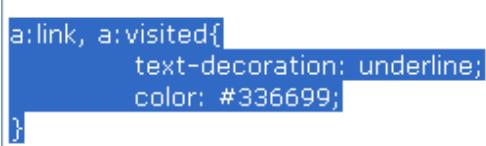
```
#ffffff, #eeeeee, #dddddd, #cccccc, #bbbbbb, #aaaaaa, #999999, #888888,  
#777777, #666666, #555555, #444444, #333333. #222222, #111111,  
#000000
```

۲ رقم اول کدهای Hex نماینده رنگ قرمز هستند، رقم های سوم و چهارم نماینده رنگ سبز و ۲ رقم آخر نیز نماینده رنگ آبی می باشند. پس با این اوصاف کد #ff0000 به شما رنگ قرمز را می دهد، همچنین کد #550000 رنگ قرمز کدر را به شما خواهد داد. کد #220000 نیز قابل پیش بینی است که رنگ قرمز کدرتری را به شما می دهد. حال کد #ff0000 رنگ سبز را به شما خواهد داد و در آخر کد #ff0000 رنگ آبی را می دهد. پس منظور این کدها را متوجه شده اید و متوجه شده اید که ترکیب این کدها رنگهای مختلفی را خواهد داد.

● مرحله‌ی اول:

کدهای زیر را در زیر سلکتور { body } تایپ کنید:

```
a:link, a:visited{  
text-decoration: underline;  
color: #336699;  
}
```



```
a:link, a:visited{  
text-decoration: underline;  
color: #336699;  
}
```

❖ این کدها برای چه هستند؟

در ابتدا شما با کد; text-decoration: underline; مطمئن می شوید که تمامی لینک های موجود در وب سایت خط زیرین(underline) خواهند داشت و همچنین چون کد color: #336699; را اضافه کرده اید، تمامی رنگ لینک هایتان به رنگ آبی در خواهد آمد. این رنگ آبی هست چون دو رقم آخر دارندۀ ی بالاترین شماره می باشد (دو رقم آخر مربوط به رنگ آبی می شود).

❖ حالا معرفی سلکتور ها:

a:link – این کد برای استایل یا صفحه آرایی لینکها بکار می رود. وقتی شما می خواهید یک نوشه را به لینک تبدیل کنید مسلماً از تگهای <a> و استفاده می کنید. بنابراین این سلکتور تمامی نوشه هایی که داخل اینگونه تگها قرار می گیرند را استایل دهی می کنند.

a:visited – این کد برای استایل یا صفحه آرایی لینکهایی که قبلاً کلیک شده است (صفحه های دیده شده)، بکار برده می شود، بنابراین **Visited** داخل a:visited نوشه شده است.

اگر دقت کرده باشید ما دو سلکتور را با هم استایل داده ام، هنگامی که شما اینکار را انجام دهید تمامی مشخصات برای هر دو سلکتور اعمال می شود. برای اینکار باید بین سلکتور ها باید علامت (،) قرار دهید. این عملی که ما انجام دادیم تنها برای کوتاهی و خلاصه کردن کار است و گرنه شما می توانید کد زیر را که دقیقاً کار کد بالا را انجام می دهد را استفاده کنید.

```
a:link{  
text-decoration: underline;  
color: #336699;  
}  
a:visited{  
text-decoration: underline;  
color: #336699;  
}
```

● مرحله دوم :

کد های زیر را زیر سکلتورهای { a:link, a:visited } تایپ کنید:

```
a:hover{  
text-decoration: none;  
}
```

◆ حال این کد برای چیست؟

سلکتور a:hover برای زمانی مورد استفاده قرار می گیرد که شما کرسر موس را بروی لینکها می بردید. و چون ما کد text-decoration: none؛ را برای این سکلتور قرار داده

ایم، کاری کردیم که هرگاه کرسر موس بروی لینکی قرار گرفت، خط زیرین آن ناپدید شود.

اگر شما می خواهید که هنگام قرار گرفتن کرسر بروی لینکها رنگ آنها نیز تغییر کند، می توانید از کد زیر استفاده کنید، کاملاً همه چیز مشخص است و فکر نکنم که نیازی به توضیح داشته باشد:

```
a:hover{  
text-decoration: none;  
color: #ff0000;  
}
```

این درس خیلی کوتاه بود، متنظر درسهای بعدی باشید..... ناراحت نباشید چون مطلب طولانی نشانه‌ی ننوشتن است..... (ضرب المثل گرفته شده از فرهنگستان و بلاگستان)

منبع:

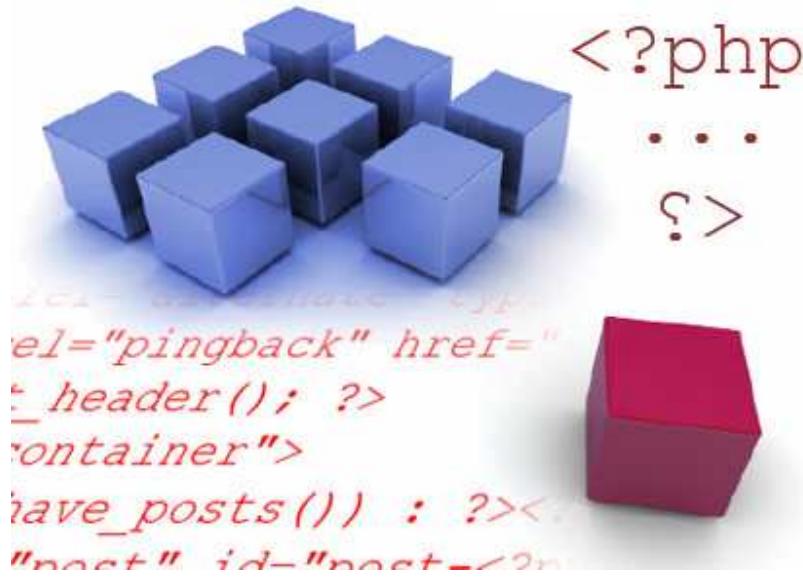
<http://www.p30help.ir/how-make-wordpress-theme-lesson-15>

چگونه برای وردپرس پوسته بسازیم؟ (بخش ۱۶)

نوشته شده توسط مهندی رادی در تاریخ ۲۵ / ۰ ۰ / ۱۳۸۸

کار با پهنا (Width) و

مدتی هست که کار با استایل یا CSS را شروع کرده ایم. در این درس کار با آن را ادامه خواهیم داد.



در این درس می خواهیم، نحوه‌ی تنظیم کردن پهنا را برای هر div یا همان باکس مخفی توضیح دهیم و اینکه چگونه آنها را مرتب کنیم. همچنین شما را از ایرادات هنگام ساخت پوسته، آگاه خواهیم کرد.

برای خواندن کامل متن به ادامه مطلب مراجعه کنید...

قبل از شروع رسمی درس اعمال زیر را انجام دهید:

Xampp Control و فolder پوستها، مرورگر فایرفاکس، فایل index.php و فایل Style.css را باز کنید.

● مرحله‌ی اول:

اولین چیزی که بایستی انجام دهید، این است که پهنا اصلی پوسته تان را مشخص کنید. ما در این آموزش می خواهیم که از پهنا ۷۵۰ پیکسل برای پوسته مان استفاده کنیم. این را نیز به خاطر داشته باشید که هر ۱۰۰ پیکسلی یک اینج است. یکی از دلایلی که ما از پهنا ۹۰۰ پیکسل استفاده نمی کنیم این است که در صورت استفاده از اندازه ۹۰۰ پیکسل، بیننده هایی که دارای رزولوشن ۸۰۰ در ۶۰۰ هستند، هنگام تماشای سایت شما حدود ۱۰۰ پیکسل را به طور مستقیم نمی توانند مشاهده کنند.

❖ حال چگونه پهنای کلی پوسته را به ۷۵۰ پیکسل محدود کنیم؟

برای اینکار باید یه باکس مخفی ایجاد کنیم که تمامی اجزای صفحه، مانند header, footer و container, sidebar زیر را انجام دهید:

- کد <div id="wrapper"> را بعد از <body> در فایل index.php وارد کنید.
- همچنین کد </div> را قبل از </body> در فایل index.php اضافه کنید.

حتماً متوجه شدید که ما یک باکس مخفی ایجاد کردیم که تمامی اجزا را درون خودش جا داده است.

حال کدهای زیر را در فایل Style.css برای استایل دهی تایپ کنید:

```
#wrapper{  
margin: 0 auto 0 auto;  
width: 750px;  
text-align: left;  
}
```

در CSS، بويژه در style.css از علامت # برای آدرس دهی به یک div که با مشخصه‌ی id نام دهی شده است، استفاده می‌کنیم. خوب شاید این سوال برایتان پیش بیابد که چگونه به div‌های دارای مشخصه‌ی class آدرس بدهیم. جواب این است که اگر شما عبارتی مانند کد <div class="wrapper"> داشته باشد باید از wrapper (نقشه). ممکن است که متوجه نشده باشید، اصلًا نگران نباشید در ادامه هنگامی که چند مثال را مشاهده کردید، کاملًا موضوع را درک خواهید کرد.

تغیرات داده شده در هر دو فایل را save کرده و مرورگر تان را refresh کنید تا تغیرات را مشاهده کنید.

❖ توضیحات بیشتر:

– معنی این کد این است که به ترتیب، صفر پیکسل حاشیه از بالا، فاصله از خودکار از راست، صفر پیکسل حاشیه از پایین و حاشیه ای خودکار از چپ بوجود آید.

– کاملاً واضح هست و نیازی به توضیح نیست.

– این خط، تراز نوشته های داخل باکس مخفی را تعیین می کند و نوشته ها را به سمت چپ متمایل می نماید. می توانید بجای کد body{ text-align: left; } استفاده کنید تا فرق این دو را متوجه شوید.

● مرحله ی دو:

باکس Header را به سمت چپ float و به آن پهنای ۷۵۰ پیکسل دهید.
مانند زیر:

```
#header{  
float: left;  
width: 750px;  
}
```

● مرحله ی سوم:

به باکس Container به سمت چپ float و پهنای ۵۰۰ پیکسل دهید.

```
#container{  
float: left;  
width: 500px;  
}
```

● مرحله ی چهار:

باکس Sidebar را به سمت چپ float و پهنای ۲۴۰ پیکسل و رنگ خاکستری دهید.
(حدود ۱۰ پیکس بی استفاده می ماند، در ادامه دلیل اینکار را توضیح خواهیم داد.)

```
.sidebar{  
float: left;  
width: 240px;  
background: #eeeeee;  
}
```

توجه: ما در اینجا از علامت # برای آدرسی دهی به باکس مخفی ای که اسمش sidebar است، استفاده نکرده‌ایم. در حالی که از یک نقطه کمک گرفته‌ایم. دلیلش این است که در باکس مخفی نام sidebar را با مشخصه‌ی class تعریف کرده‌ایم نه id. همچنین بیان بیاورید که رنگ سفید است و رنگ background: #eeeeee یک رنگ خالکستری روشن می‌باشد. در اینجا ما تنها برای اینکه بین container و sidebar تفاوت ایجاد کنیم، به پس زمینه‌ی سایدبار رنگ خالکستری روشن داده‌ایم، چون قرار است در ادامه ۱۰ پیکسل باقی مانده را بعنوان فاصله‌ی بین این دو باکس استفاده کنید.

● مرحله پنجم:

برای باکس مخفی footer مشخصه‌های زیر را بدھید:

```
#footer{  
clear: both;  
float: left;  
width: 750px;  
}
```

❖ چه تفاوتی بین Header و Footer در استایل نویسی وجود دارد؟

جواب: تنها تفاوت داشتن عبارت clear: both; برای سلکتور footer{} می‌باشد. کد مورد بحث برای این اضافه می‌شود تا مطمئن شویم که فوتر قالب به هیچ جزء دیگری از قالب اضافه نمی‌شود. برای مثلا فوتر به باکس container و یا sidebar نمی‌چسبد.

تغیرات را ثبت کرده و مرورگرتان را refresh کنید.

● مرحله‌ی ششم:

حال در این مرحله ۱۰ پیکسل باقی مانده را به سایدبار بعنوان فاصله‌ی یا حاشیه نامرئی اضافه می‌کنیم. برای اینکار از کد margin کمک می‌گیریم. حالا کد استایل سایدبار ما شبیه زیر است:

```
.sidebar{  
float: left;  
width: 240px;  
background: #eeeeee;
```

```
margin: 0 0 0 10px;  
}
```

فایل را save کنید و مرورگر را refresh نمایید تا ۱۰ پیکسل اضافه شده به چپ سایدبار را مشاهده خواهید کرد.

« margin: 0 0 0 10px; » معنی آنها به ترتیب یعنی صفر از بالا، صفر از راست، صفر از پایین، ۱۰ پیکسل از چپ. همچنین وقتی که اندازه‌ی مورد نظر ما صفر باشد لازم نیست که عبارت px را جلوی آن بنویسیم.

● مرحله‌ی هفتم (مرحله‌ی اضافی):

در این مرحله می‌خواهیم که بجای ۱۰ پیکسل فاصله‌ای که به بخش چپ سایدبار داده ایم، ۲۰ پیکسل فاصله قرار دهیم. با انجام اینکار خواهید دید که لایه‌های شما خواهد شکست و فشار آورده شده به سایدبار باعث می‌شود که باکس سایدبار در زیر باکس container قرار گیرد. دلیلش این است که ۲۰ پیکسل حاشیه باعث می‌شود که جمع پهنازی wrapper و Sidebar و Container جمعاً ۷۶ پیکسل شود در صورتی که ما باکس تنها ۷۵ پیکسل پهنا داده ایم. البته ممکن است این خطا در فایرفاکس نشان داده نشود و تنها در اینترنت اکسپلورر اینگونه نشان داده شود، چون باگ‌های فایرفاکس بسیار کمتر از اینترنت اکسپلورر هستند. در هر حال توجه کنید که حتماً اندازه‌هایی که به اجزای صفحه می‌دهید متناسب با سایر اجزا باشد.

برای برطرف کردن این باگ، عبارت sidebar را به سلکتور `display: inline;` اضافه کنید. حال کد ما به این صورت خواهد بود:

```
.sidebar{  
float: left;  
width: 240px;  
background: #eeeeee;  
margin: 0 0 0 10px;  
display: inline;  
}
```

درس امروز به پایان رسید.

و فایلهای نهایی [style](#) و [index](#) را می‌توانید دانلود کنید.

امیدوارم که از این درس هم لذت برده باشید، به زودی این دوره‌ی آموزش نیز به پایان خواهد رسید...

منبع:

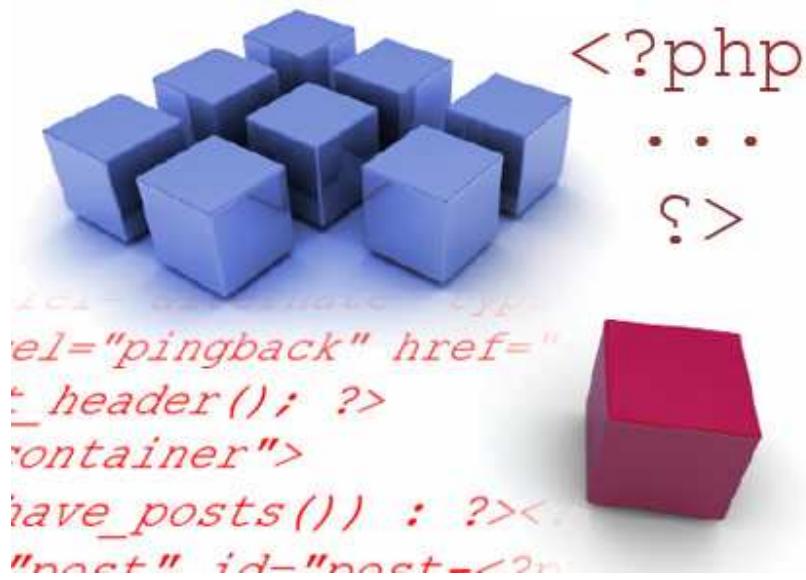
<http://www.p30help.ir/how-make-wordpress-theme-lesson-16>

چگونه برای وردپرس پوسته بسازیم؟ (بخش ۱۷)

نوشته شده توسط مهندی رادی در تاریخ ۰۳ / ۰۴ / ۱۳۸۸

قالب بندی پست‌ها و حواشی آن

نمی‌دانم که در این شرایط کشور، نوشتن یک پست آی‌تی جایز هست یا خیر ... به هر حال بعد از یک دقیقه سکوت بخاطر حوادث ناگوار اخیر درس را شروع می‌کنیم.



همانطور که از عنوان درس مشخص هست در این درس قرار است که پست‌ها یا مطالب پوسته را قالب بندی کنیم.
برای شروع رسمی درس به ادامه مطلب مراجعه کنید ...

در این درس شما به فایل **index.php** نیازی ندارید.
موارد **Xampp Control**, **فولدر پوسته**, **Firefox**, **و style.css** را باز کنید.

● مرحله‌ی اول:

فایل **style.css** را باز کنید و در بالای کد **body** کد‌های زیر را وارد کنید:

```
body, h1, h2, h3, h4, h5, h6, blockquote, p{  
margin: 0;  
padding: 0;  
}
```

- توجه کنید که عبارت **margin: 0 0 0 0** دقیقاً همان عبارت **margin: 0** است. در حقیقت وقتی که تمامی مقادیر یکسان باشند، شما می‌توانید برای خلاصه نویسی از یک شماره استفاده کنید که این یک شماره برای تمامی حالت‌ها استفاده می‌شود. همین کار را برای **padding** هم استفاده کردیم.
- فایل را **save** کرده و تغییرات را در مرورگر مورد نظر نگاه کنید.

نکته: من به شما گفتم که کد مورد نظر را بالا یا پایین کدهای دیگر قرار دهید. اما بدانید که این‌ها همگی اختیاری هستند. شما می‌توانید کدها را در هر کجا تایپ کنید و یا قرار دهید، این کار را تنها برای سازماندهی و مدیریت بهتر کدها به شما گفتم.

● مرحله‌ی دوم:

حال زمان آن رسیده که عنوان پست‌ها (**H1**) را استایل دهیم، پس کدهای زیر را در پایین **body** تاپ‌کنید:

```
h1{  
font-family: Georgia, Sans-serif;  
font-size: 24px;  
padding: 0 0 10px 0;  
}
```

Arial – این کد عنوان پست ها را از فونت Georgia, Sans-serif; «
به Sans serif تغییر می دهد. و در صورت نبود فونت Georgia، از فونت استفاده می کند.

– این گزینه کاملاً واضح است. توجه کنید، شما اندازه ی فونت را در body{} مقدار ۲۴px تنظیم کردید، در صورتیکه تگ های H1 و H2 از این قانون یا این خط کد پیروی نمی کنند، بلکه تگ های heading از قانون خودشان پیروی می کنند. پس برای کنترل آنها، شما باید آنها را بصورت جداگانه قالب بندی کنید.

– این کد به معنی این است که از تاجیه ی پایین H1 ده پیکسل فاصله بیافتد. این فضای ۱۰ پیکسلی را برای فضای بین عنوان و بلاگ ها و توضیحات و بلاگ مورد استفاده قرار خواهد گرفت.

تغییرات را save کنید و آنرا مشاهده کنید.



● مرحله ی سوم:

کد های زیر را در زیر سلکتور #container تایپ کنید. (بعد از تغییر در هر بلاک و یا تایپ هر کد تغییرات را ثبت کرده و آنها را در مرورگر خود مشاهده کنید).

```
.post{  
padding: 10px 0 10px 0;  
}
```

(شما با این کد ۱۰ پیکسل فاصله از ناحیه بالا و پایین به هر div برای هر پست اضافه کرده اید).

```
.post h2{  
font-family: Georgia, Sans-serif;  
font-size: 18px;  
}
```

(سلکتور **h2** به طور ویژه **sub-headings** های **H2** را فقط برای پست‌ها هدف گیری می‌کند، به زبان خودمان این سلکتور تنها به عنوان پستها ویژگی می‌بخشد).

```
.entry{  
line-height: 18px;  
}
```

کد **line-height** فاصله‌ی بین خطوط یک مطلب را معین می‌کند که در اینجا ما برای آن مقدار متوسط و مناسب ۱۸px را انتخاب کرده‌ایم. می‌توانید مقدار آنرا به اعداد مختلف تغییر دهید تا وظیفه‌ی این خط کد را بهتر متوجه شوید. ولی در پایان مقدار را به همان ۱۸ پیکسل بازگردانید.

● مرحله چهارم:

کد پایین را در زیر سلکتور **a:hover** تایپ کنید:

```
p{  
padding: 10px 0 0 0;  
}
```

(سلکتور **p** با پاراگراف‌های متن سروکار دارد و با توجه به مقدار **padding** داده شده، فاصله‌ی هر پاراگراف از دیگری ۱۰px پیکسل خواهد بود).

● مرحله‌ی پنجم:

کد زیر را در پایین سلکتور **.entry** تایپ کنید:

```
p.postmetadata{  
border-top: 1px solid #ccc;  
margin: 10px 0 0 0;  
}
```

اگر بیاد داشته باشید، تگ باکس مخفی **postmetadata** را درون تگ‌های پاراگراف (**p**) قرار داده اید (اگر بیاد ندارید یک نگاهی به فایل index.php بیاندازید). پس برای استایل دادن به آن می‌توانید به صورت **p.postmetadata** آدرس دهی کنید.

برای تگ باکس مخفی **postmetadata** ، شما یک حاشیه خاکستری با ضخامت یک پیکسل از سمت بالا و یک فاصله ۱۰ پیکسلی از طرف بالا را به **postmetadata** داده اید.

نکته: اگه بیاد داشته باشید بخش **postmetadata** برای این است که اطلاعات هر پست مانند دسته‌ی پست، تعداد نظرات پست و ... را درون آن جای دهیم.

معنی عبارت **border-top** حاشیه از بالا (فقط از بالا) است. عبارت **border-left** معنی حاشیه از سمت چپ (فقط از سمت چپ) و به همین ترتیب برای **border-bottom** و **border-right** نیز معنی اش واضح است. برای مثال معنی عبارت **border: 1px solid #ccc** این است که: از هر چهار طرف حاشیه خاکستری با ضخامت یک پیکسل داده شود.

● مرحله‌ی ششم:

کدهای زیر را در پایین سلکتور **{p.postmetadata}** تایپ کنید:

```
.navigation{  
padding: 10px 0 0 0;  
font-size: 14px;  
font-weight: bold;  
line-height: 18px;  
}
```

در این کد شما برای بخش **navigation** یا همان قسمت **صفحه‌ی بعدی و صفحه‌ی قبلی** قالب بندی می‌کنید. در حقیقت شما:

- ۱۰ پیکسل حاشیه از بالا اضافه کردید.
- اندازه‌ی فونت را به ۱۴px تغییر دادید.
- پنهانی فونت را به ضخیم عوض کردید
- ارتفاع خط را به ۱۸ پیکسل افزایش دادید.

خوب درس امروز ما که کار و قالب دهی به بخش پست‌ها بود به پایان رسید. دیگر چیزی به پایان کلی دروس نمانده است....

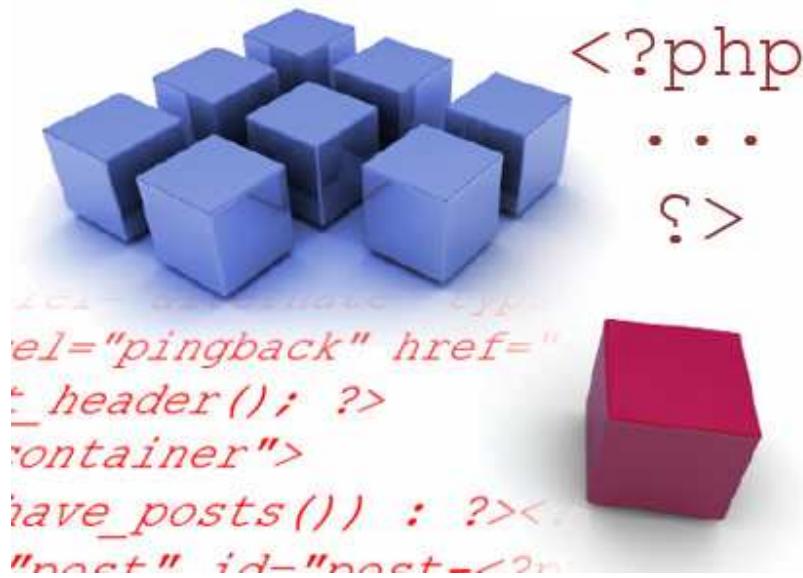
منبع:

چگونه برای وردپرس پوسته بسازیم؟ (بخش ۱۸)

نوشته شده توسط [مهندی رادی](#) در تاریخ ۰۵ / ۰۸ / ۱۳۸۸

صفحه آرایی با سایدبار

کم کم در حال رسیدن به جلسات پایانی هستیم. در این درس به شما خواهیم آموخت که چگونه به سایدبار قالب خود ویژگی ببخشید و آنرا صفحه آرایی کنید. صفحه آرایی سایدبار شامل بخش های است. از همین جهت ما در این جلسه تنها می توانیم بخش های اصلی را به شما معرفی کنیم و شما برای مطالعه ای جزئیات باید خود تحقیق و تفحص کنید...



برای شروع درس و اینکه چگونه سایدبار را صفحه آرایی کنیم به ادامه مطلب بروید...

مثل همیشه شما بایستی index.php ، فایل style.css ، مرورگر فایرفاکس، فolder پوسته‌ی مورد نظر و در آخر تنظیمات برنامه‌ی Xampp را باز کنید.

● مرحله‌ی اول :

کدهای زیر را در پایین `.sidebar` تایپ کنید

```
.sidebar ul{  
list-style-type: none;  
margin: 0;  
padding: 0 10px 0 10px;  
}
```

حال شما با این کد ها تگ های `ul` ای که درون سایدبار هستند را صفحه آرایی کرده اید و به آنها ویژگی بخشیده اید. تمامی تگ های `ul` و همچنین `ul` های تو در تو، ویژگی های مورد نظر را به ارث خواهند برداشت.

برای مثال :

```
<ul>  
  <li>  
    <ul>  
      <li></li>  
    </ul>  
  </li>  
</ul>
```

همانگونه که مشاهده می کنید، `ul` های دوم که با رنگ آبی مشخص شده اند با وجود اینکه در زیر مجموعه `ul` اصلی قرار دارند، باز هم ویژگی های `ul` اول را به ارث می برند. برای مثال اگر شما حاشیه ای را برای `ul` اول بدهید، همین حاشیه برای `ul` دوم نیز داده خواهد شد.

تغییرات را ثبت کرده و مرورگر تان را `refresh` کنید تا تغییرات را مشاهده کنید.

توجه کنید که چطور `10` پیکسل `padding` به بالا و پایین سایدبار شما اضافه شده است. (عکس زیر)

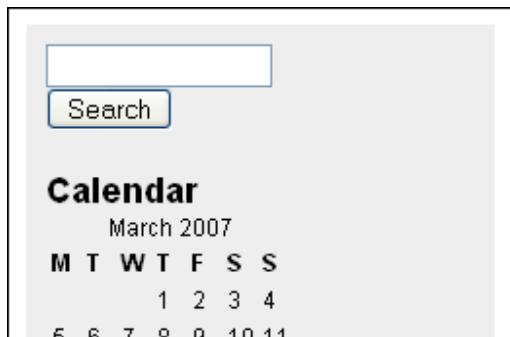


● مرحله‌ی دو:^۲

کدهای زیر را پایین کدهای `.sidebar ul{}` تایپ کنید.

```
.sidebar ul li{  
padding: 10px 0 10px 0;  
}
```

با این کد مشاهده خواهید کرد که بین هر ابزارک یا ویجیت درون سایدبار فاصله‌ای ۱۰ پیکسلی از بالا و پایین ایجاد می‌گردد. (عکس زیر را با عکس بالایی مقایسه کنید)



◆ چرا این ۱۰ پیکسل فاصله را به تگ `ul` در مکان اول اضافه نکردیم؟

دلیل اینکار این بود که ما می‌خواهیم که بین ابزارک‌های داخل سایدبار فاصله ایجاد کنیم، برای مثال می‌خواهیم بین فرم جستجو و تقویم فاصله‌ای وجود داشته باشد یا بین تقویم و برگه‌ها فاصله‌ی مشخصی وجود داشته باشد. که در این صورت بایستی به تگ `li` درون `ul` سایدبار ویژگی بخشید. حال شما با فاصله‌ای (padding) که به سایدبار داده اید بین اجزای داخل سایدبار فاصله‌ای ۱۰ پیکسلی ایجاد کرده‌اید.

● مرحله‌ی سوم:

خط های زیر را در پایین کد های `.sidebar ul li` تایپ کنید:

```
.sidebar ul li h2{  
font-family: Georgia, Sans-serif;  
font-size: 14px;  
}
```

بیاد بیاورید که چگونه **sub-heading** های زیر مجموعه‌ی `post` را صفحه آرایی کردیم. در حقیقت شما با سلکتور `ul li h2` عنوان اصلی هر ابزارک را صفحه‌بندی می‌کنید.

حالا شما می‌توانید ویژگی‌های داده شده به عنوان هر ابزارک را مشاهده کنید.



برای اینکه مدیریت بخش برگه‌ها حرفه‌ای تر باشد من خود علاوه بر برگه‌ی **درباره** یکسری برگه‌ی دیگر بصورت‌های مختلف (تصویرت تو و تو و معمولی) از طریق مدیریت بخش وردپرس ایجاد کرده‌ام. اگر دقت کرده باشید فاصله‌ی هر کدام از

برگه ها با دیگری حدود ۰ پیکسل است که کمی نازیبا بوده و بخش سایدبار ما را زشت می کند.

♦ دلیل این فاصله بین هر کدام از برگه ها چیست؟

چون شما به `.sidebar ul li{}` مقداری `padding` داده اید. بنابراین `li` های سطوح زیر دست نیز بر اساس قانون وراثت از این فاصله (`padding`) ویژگی به ارث می برند.
برای حل این مشکل مرحله ی چهارم را بخوانید.

● مرحله ی چهارم :

کد زیر را زیر کدهای بخش `.sidebar ul li h2{}` تایپ کنید:

```
.sidebar ul ul li{  
padding: 0;  
}
```

با سلکتور `.sidebar ul ul li{}` و مقدار فاصله `padding` صفر، `li` های زیر دست ویژگی فاصله ۰ پیکسلی را از دست داده و به همان حالت اول باز می گردند.

نتیجه این هست:



کد زیر به `.sidebar ul ul li{}` اضافه کنید:

```
.sidebar ul ul li{  
padding: 0;  
line-height: 24px;  
}
```

اگر با انجام اینکار شما در اینترنت اکسپلورر، فضای اضافی را در زیر فرم جستجو پیدا کرده اید، عبارت **form** را به سلکتور های زیر در style.css اضافه کنید تا این مشکل برطرف شود:

```
body, h1, h2, h3, h4, h5, h6, address, blockquote,  
dd, dl, hr, p{  
margin: 0;  
padding: 0;  
}
```

باید کد شما شبیه به این شده باشد:

```
body, h1, h2, h3, h4, h5, h6, address, blockquote,  
dd, dl, hr, p, form{  
margin: 0;  
padding: 0;  
}
```

● مرحله ی پنجم (مرحله ی اختیاری) :

این مرحله تنها برای افزایش زیبایی سایدبار و بویژه بخش تقویم می باشد. با انجام این مرحله تقویم شما کشیده شده و تمام بخش سایدبار را پوشش می دهد و از این نظر سایدبار شما را زیباتر می کند. الان شکل تقویم شما به صورت زیر است، همانطور که می بینید فضای خالی زیادی در سمت راست تقویم وجود دارد که از آن استفاده نشده است.



برای صفحه آرایی تقویم باید ابتدا تگ مربوط به آن، مکان قرار گیری آن و نام id آن را پیدا کرد. برای انجام اینکار در مرورگر فایرفاکس به تب View رفته و سپس Source را انتخاب می کنیم. کدهای مربوط به سایدبار در بخش پایین کدهای سورس هستند، بنابراین اسکرول را به پایین کشیده تا به کدهای زیر برسید:

```
<li id="calendar"><h2>Calendar</h2>
  <table id="wp-calendar">
    <caption>March 2007</caption>
    <thead>
      <tr>
```

حالا شما می دانید که تقویم در تگ Table با مشخصه id که مقدار آن wp-calendar است، قرار دارد.

❖ اما سوال این است که چطور wp-calendar را داخل فایل style.css هدف گیری کنیم؟

جواب `table#wp-calendar` هست. اما چرا؟
قبلأ به شما نحوه ی هدف گیری باکس مخفی (div) که با مشخصه id نام گذاری شده بود را آموزش داده بودیم (باید یک علامت # را در ابتدای نام آن قرار می دادیم). در این مورد شما فقط بجای div یک table دارید با مشخصه id پس بصورت آنرا `table#wp-calendar` هدفگیری می کنیم.

چون `#wp-calendar` منحصر به فرد است بنابراین این سلکتور برای سایر تگها استفاده نخواهد شد. اما اگر شما می خواهید که این مقدار را باز هم منحصر به فرد تر کنید می توانید از سلکتور `.sidebar ul li table#wp-calendar` استفاده کنید.

اما سوال اصلی اینجاست که چگونه این کشش را ایجاد کنیم تا تمامی فضای سایدبار پر شود. برای اینکار از کد `width: 100%` کمک می گیریم. به همین سادگی....

پس کدهای زیر را در پایین کدهای `.sidebar ul ul li` تایپ کنید:

```
table#wp-calendar{
  width: 100%;
}
```

تغییرات را ثبت و مشاهده کنید...

Calendar						
March 2007						
M	T	W	T	F	S	S
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	

این کد در مورد پهنانی بخش های مختلف مورد استفاده قرار گرفته می شود و در اینجا شما مقدار آن را ۱۰۰٪ گذاشته اید که در این صورت تمامی فضای سایدبار توسط این مقدار پر خواهد شد. البته بجای ۱۰۰٪ شما می توانید به آن پیکسل نیز بدھید مثلًا ۲۰۰ پیکسل.

فکر کنم که از حالت قبلی بهتر به نظر برسد. ظاهر سایدبار را کمی بهتر کرده ایم ولی هنوز کمی کار دارد که در جلسات آینده به آن خواهیم پرداخت....

منبع:

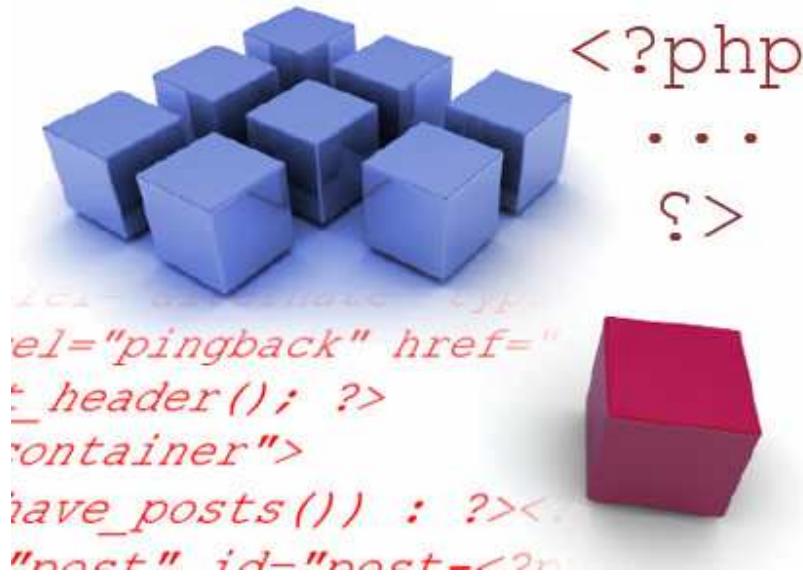
<http://www.p30help.ir/how-make-wordpress-theme-lesson-18>

چگونه برای وردپرس پوسته بسازیم؟ (بخش ۱۹)

نوشته شده توسط مهندی رادی در تاریخ ۲۰ / ۰۵ / ۱۳۸۸

فوتر و تقسیم بندی فایل index

درس امروز ما کوتاه خواهد بود و در آن به نحوه ایجاد فوتر و مهم تر از آن تقسیم بندی فایل index.php به چندین فایل کوچکتر(برای مدیریت بهتر) خواهیم پرداخت. البته این مجزا کردن باعث نمی شود که فایل index.php اهمیت خود را از دست بدهد.



برای شروع درس به ادامه مطلب مراجعه کنید....

برای شروع کار Firefox, index.php, style.css ,Xampp را باز کنید.

● مرحله اول:

۱۰ پیکسل فاصله (padding) از بالا به باکس مخفی footer بدهید. بعد از این همه درس باید اینکار را خودتان بتوانید انجام دهید. بنابراین کد را به شما نمی دهم، خود شما باید اینکار ساده را انجام دهید.

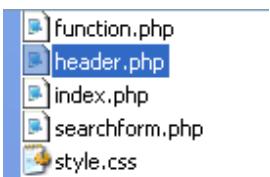
● مرحله دو:

به تمام تگهای P داخل باکس مخفی footer یک ارتفاع خط (line-height) **هجهده** پیکسلی بدهید. به شما یک کمک می کنم، سلکتور مورد نظر برای تگهای پاراگراف درون div فوتر {} #footer p می باشد. (شما امروز خودتان باید این ارتفاع خط را به این سلکتور بدهید.)

● مرحله سوم:

« یک فایل جدید بنام **header.php** بسازید. مطمئن شوید که نوع یا پسوند فایل php باشد نه یک فایل Document معمولی.

« درون فایل **index.php** ، از تگ پایانی باکس مخفی **header** شروع کنید به انتخاب کردن و تا هر آنچه که در بالای آن قرار دارد را **select** کنید. بعد از انتخاب کردن تمامی خطوط، آنها را **cut** کرده و درون فایل **header.php** بچسبانید یا همان **header.php** بچسبانید.



```
<?php wp_get_archives('type=1') ?>
<?php //comments_popup_script() ?>
<?php wp_head(); ?>
</head>
<body><div id="wrapper">

<div id="header">
<h1><a href="<?php bloginfo('url'); ?>"><?php bloginfo('description'); ?></a></h1>
</div>

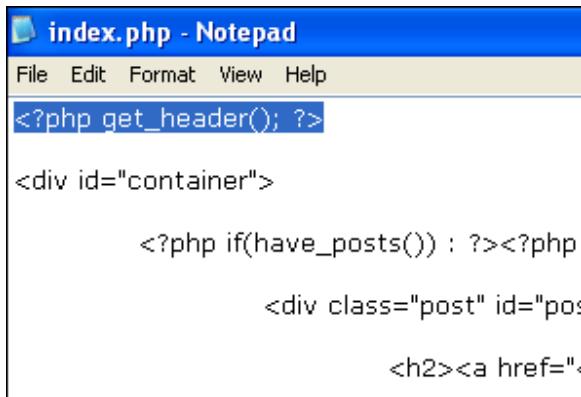
<div id="container">
```

حال می توانید فایل **header.php** من را برای فهم بیشتر دانلود کنید.

● مرحله چهارم:

شما باید به فایل **Index.php** بفهمانید که از این به بعد باید کدهای قبلی (کدهای مربوط به هدر) را از داخل فایل **header.php** بخواند. برای اینکار خط زیر را در ابتدای فایل **index.php** در عوض کدهای قبلی بنویسید:

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



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

<div id="container">

    <?php if(have_posts()) : ?><?php

        <div class="post" id="pos

            <h2><a href="#">
```

کد بالا یک تابع داخل سیستم وردپرس است که به طور ویژه کدهای داخل فایل `index.php` را درون فایل `header.php` می خواند. در حقیقت کد بالا کد طولانی `<?php include (TEMPLATEPATH . '/header.php'); ?>` را برای شما خلاصه می کند.

تغییرات را `save` کرده و مرورگر خود را `refresh` کنید. شما نباید هیچ تغییری را مشاهده کنید. اگر پوسته‌ی شما شکسته به نظر برسد یا ایرادی داشته باشد در هنگام انتقال کدها حتماً شما اشتباهی انجام داده اید.

● مرحله‌ی پنجم:

- « مرحله‌ی پنجم نیز تقریباً مانند مراحل قبل است، متنها اینبار برای فایل `sidebar.php` انجام خواهیم داد.
- « از شروع باکس مخفی `sidebar` تا پایان آن باکس، کدها را انتخاب، `cut` و درون فایل `sidebar.php` بچسبانید.
- « سپس، در داخل فایل `index.php` بجای کدهای برداشته شده کد `<?php get_sidebar(); ?>` را تایپ کنید.
- « تغییرات را `save` کرده و مرورگرتان را `refresh` کنید. دوباره شما نباید تغییری را مشاهده کنید.
- « الان فایل `sidebar.php` من را می‌توانید برای فهم بیشتر دانلود کنید.

```
<?php endif; ?>  
</div>  
<?php get_sidebar(); ?>  
<div id="footer">  
<p>  
Copyright &#169; 2007 <a href=".</a></p>  
</div>
```

```
</div></body>  
</html>
```

● مرحله ی ششم:

- « همین مراحل بالا را برای فایل **footer.php** نیز انجام دهید.
- « در داخل فایل **index.php** بجای کدهای برداشته شده کد **<?php get_footer(); ?>** را تایپ کنید.
- « فایل **footer.php** من را می توانید برای فهم بیشتر دانلود کنید.

```
<?php endif; ?>  
</div>  
<?php get_sidebar(); ?>  
<?php get_footer(); ?>
```

❖ بررسی کردن مراحل:

- « شما ۳ فایل جدید با نام های **header.php** و **sidebar.php** و **footer.php** بوجود آوردید.
- « شما با ۳ تابع جدید آشنا شدید. تابع های **get_header()** و **get_sidebar()** و **get_footer()**

« شما در پایان این درس می توانید به فایلهای من نگاهی بیاندازید: **style**, **index**, **footer**, **sidebar**, **header**.

درس امروز کوتاه بود، چیزی به پایان دروس نمانده است... صبور باشید.....

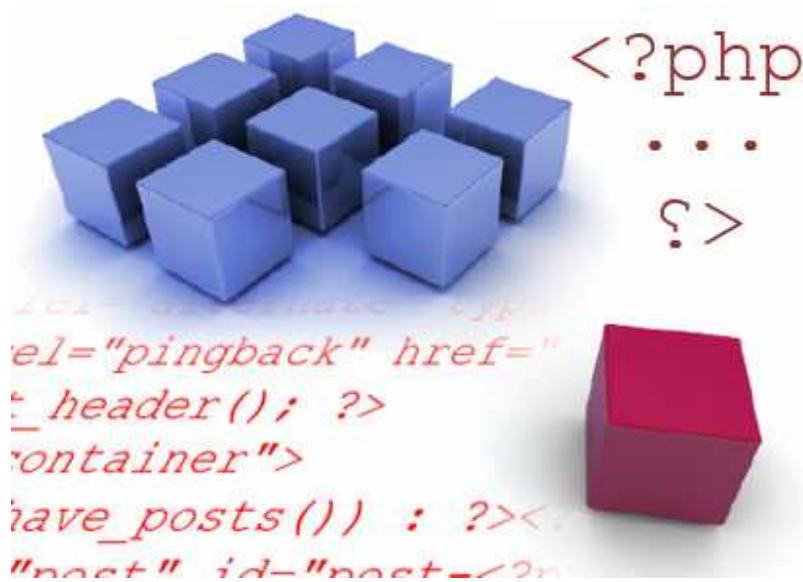
منبع:

<http://www.p30help.ir/how-make-wordpress-theme-lesson-19>

چگونه برای وردپرس پوسته بسازیم؟ (بخش ۲۰)

نوشته شده توسط مهند رادی در تاریخ ۱۳۸۸/۰۵/۳۱

در درس قبل آموختید که چگونه فایل index.php را برای مدیریت بهتر به فایلهای جداگانه ای تقسیم کنید، در این درس نیز قصد داریم تا همین عمل را برای بخش های دیگری انجام دهیم. که در اصطلاح به این فایلهای Sub Template می گویند. اگر بیاد داشته باشید در جلسه ای قبل فایلهای index.php را به سه فایل header.php و footer.php و sidebar.php مجزا کردیم تا مدیریت آنها ساده تر باشد. در این جلسه این عمل را بصورت پیشرفته تری انجام خواهیم داد.



برای ادامه درس به ادامه مطلب مراجعه کنید...

هماگونه که گفته شده در این درس نیز قصد تقسیم کردن فایل index.php را داریم ولی با پیش زمینه ای درس قبل اینکار بسیار ساده تر خواهد بود.

● مرحله‌ی اول:

در این مرحله می‌خواهیم که صفحه‌ای برای بخش آرشیو بسازیم. پس برای اینکار مراحل زیر را انجام دهید.

- « یک فایل بنام **archive.php** بسازید
- « تمامی کدهای موجود در فایل **index.php** را کپی کرده و داخل فایل **archive.php** قرار دهید.
- « در داخل فایل **archive.php** تابع **the_content** را به **the_excerpt** تغییر دهید.
- « حال دوباره فایل **archive.php** را **save** کنید.

همانطور که از نام فایل **archive.php** مشخص است این فایل برای نحوه‌ی نشان دادن صفحه‌ی آرشیو وبلاگ بکار می‌رود. و البته ما در اینجا این صفحه را دقیقاً مانند صفحه‌ی اصلی سایت طراحی کردیم بغير از یک تغییر کوچک که در آن بوجود آوردیم. حال برای فهم بیشتر بروی یکی از آرشیو‌های قدیمی وبلاگ خود کلیک کنید تا تغییرات را متوجه شوید.

بله درست است در صفحه‌ی آرشیو شما تنها خلاصه از هر مطلب را مشاهده می‌کنید نه کل مطلب را. عبارت **the_content** بجای کد **the_excerpt** باعث این عمل می‌شود.

❖ اما چرا اینکار را انجام دادیم؟

ما این عمل را برای محافظت در برابر جرمیه‌ی گوگل انجام دادیم. چون اگر مطلبی هم در آرشیو و هم در صفحه‌ی اصلی سایت باشد، گوگل این مطالب را تکراری تلقی می‌کند و ممکن است که وبلاگ شما را جرمیه‌کند و از رتبه‌ی وبلاگ شما کاسته شود.

نکته: اگر شما مایلید که صفحه‌ی دسته‌های (Category) وبلاگ تان نیز با صفحه‌ی اصلی سایت شما متفاوت باشد، یک فایل بنام **category.php** ساخته و آنرا آنطور که مایلید تنظیم کنید.

● مرحله‌ی دوم:

- « یک فایل بنام **search.php** بسازید.
- « هر آنچه که در فایل **archive.php** وجود دارد را کپی کرده و درون فایل

بچسبانید.
« تغییرات را save کنید.

با انجام این عمل تمامی **نتایج جستجوی و بلاگ** نیز بصورت گلچین نمایش داده خواهد بود.

◆ حال شاید این سوال برای شما پیش آید که اگر فایل **search.php** یا **archive.php** وجود نداشت، صفحه‌ی جستجوی و بلاگ یا بخش آرشیو و بلاگ چگونه نمایش داده می‌شد؟

باید در جواب گفت که اگر فایلهای **archive.php** یا **search.php** وجود نداشته باشند، آنگاه وردپرس بصورت هوشمندانه نحوه‌ی نمایش تمامی این صفحات را از فایل **index.php** می‌گیرد.

● مرحله‌ی سوم:

« دو فایل با نام‌های **single.php** و **page.php** بسازید.
« تمامی کدهای موجود در فایل **index.php** را درون فایلهای **single.php** و **page.php** کپی کنید. (الان باید فایلهای **single** و **Page** مانند هم باشند).
« تغییرات را در فایل **single** و **page** ثبت کرده و فایل **index.php** و فایل **single.php** را ببینید.

● مرحله‌ی چهارم:

فایل **page.php** در مورد نحوه‌ی نمایش برگه‌های و بلاگ شما وارد عمل می‌شود. حال برای بهتر شدن آن مراحل زیر را پی بگیرید.

۱ - در فایل **page.php** در زیر کد <?php the_content(); ?> کدهای زیر را تایپ کنید:

```
<?php link_pages(' <p><strong>Pages:</strong> ' ,  
'</p>' , 'number' ) ; ?><?php edit_post_link('Edit' ,  
'<p>' , '</p>' ) ; ?>
```

۲- کدهای **postmetadata** پاک کنید. چون اطلاعات **page.php** از **postmetadata** در برگه های وردپرس بی معنی هستند.

```
<div class="entry">  
  
    <?php the_content(); ?>  
    <?php link_pages('<p><strong>Pages:</strong> ',  
        <?php edit_post_link('Edit', '<p>', '</p>'); ?>  
  
</div>
```

۳- کدهای **page.php** را نیز در فایل **navigation** یا **posts_nav_link()** پاک کنید.

```
<?php endwhile; ?>  
  
    <div class="navigation">  
        <?php posts_nav_link(); ?>  
    </div>  
  
<?php else : ?>
```

◆ چه اتفاقی افتاده است؟

کد خط اول که تایپ کردیم برای نمایش تعداد صفحات یک برگه می باشد.

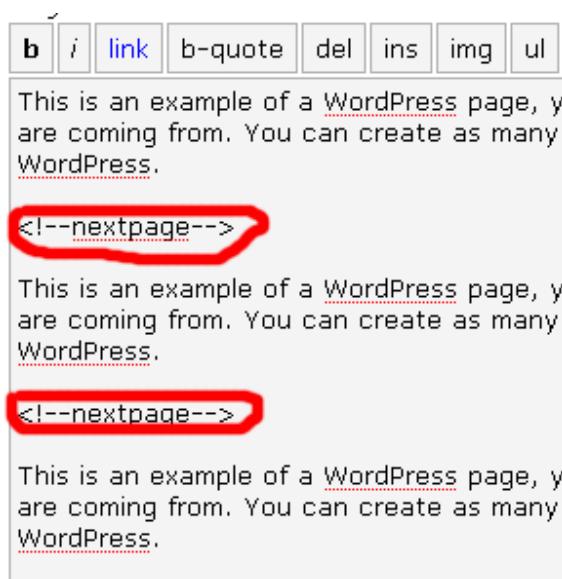
About

This is an example of a WordPress page, you could edit this yourself or your site so readers know where you are coming from. You can use the main menu to publish pages like this one or sub-pages as you like and manage them in WordPress.

Pages: [1](#) [2](#) [3](#)

[Edit](#)

که ظاهر آن برای مجزا کردن در حالت **ویرایش** وردپرس به مانند عکس زیر می باشد:



البته این امکان هنگامی استفاده می شود که شما یک برگه خیلی طولانی داشته باشید و بخواهید که مطلب مورد نظر خود را در چندین صفحه نشان دهید.

دومین خط تایپ شده به مدیر وبلاگ یک لینک بنام **Edit** می دهد تا در صورت تمایل بتوانید با کلیک بروی آن به بخش مدیریت وردپرس رفته و مستقیماً متن مورد نظر خود را ویرایش کند. در حقیقت این دکمه برای سهولت دسترسی به ویرایش برگه ها می باشد.

تغییرات را ثبت کرده و فایل **page.php** را ببینید.

● مرحله ی پنجم:

اما تنها یک فایل دیگر مانده و آن هم **single.php** است. فایل **single.php** مربوط به نحوه ی نمایش صفحه ی اصلی مطالب است یعنی زمانی که شما بروی لینک ادامه مطلب کلیک می کنید و صفحه ی اصلی مطلب گشوده می شود و در آن صفحه قادر هستید که کامنت نیز بگذارید. (شما الان که در حال خواندن این مطلب هستید دقیقاً درون صفحه ی **single.php** قرار دارید.)

۱ - درون فایل **single.php** در پایین کد زیر را تایپ کنید:

```
<?php link_pages(' <p><strong>Pages:</strong> ' ,  
' </p>' , 'number' ); ?>
```

بله ، این کد همان خط از کدهایی برای لینک های صفحات می باشد. این کد نیز دقیقاً برای صفحه بندی پست ها بکار می رود.

۲- در بخش `postmetadata`، تابع `?> <?php comments_popup_link(); ?>` و همچنین `tg
` قبل از آن را پاک کنید. ولی کل بخش `postmetadata` را پاک نکنید. و دلیل پاک شدن `tg br` نیز این است که یک فاصله ای اضافی را بوجود می آورد که اضافی است.

دلیل اینکه شما تابع تعداد نظرات را پاک می کنید این است که این تابع در بخش `single` دیگر کار نمی کند بنابراین استفاده ای نخواهد شد، پس بهتر است که در فایل `single.php` پاک شود.

۳- بجای کد `?> <? ;()php posts_nav_link?>` عبارت زیر جایگزین کنید:

```
<?php previous_post_link('&laquo; %link') ?> <?php  
next_post_link(' %link &raquo;') ?>
```

در صفحه ای اصلی، صفحه ای آرشیو، صفحه ای دسته ها و صفحه ای جستجو ، شما از تابع `posts_nav_link()` استفاده می کنید تا لینک های **Previous page** و **Next page** را صدا بزنید. اما برای صفحه ای `single` چون لینک های **Previous page** و **page Next** وجود ندارند. شما از تابع های `next_post_link()` و `previous_post_link()` استفاده کردید تا لینک پست های بعدی و قبلی را صدا بزنید.

تغییرات را در فایل `single.php` ثبت کنید و آنها را در مرورگر مشاهده کنید.

❖ خلاصه درس:

« شما ۴ فایل با نام های [single.php](#) و [page.php](#) و [search.php](#) و [archive.php](#) ساختید. « [single.php](#) و [page.php](#) دقیقاً [search.php](#) و [archive.php](#) یکسان هستند.

« بخش های اضافی را از فایل page.php حذف کردیم.
« بخش های اضافی را از فایل single.php حذف کردیم.

منبع:

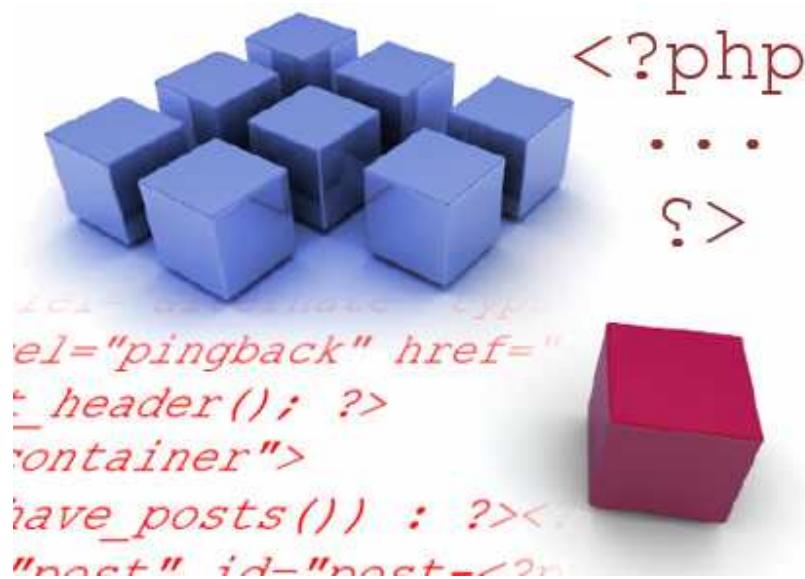
<http://www.p30help.ir/how-make-wordpress-theme-lesson-20>

چگونه برای وردپرس پوسته بسازیم؟ (بخش ۲۱ و پایانی)

نوشته شده توسط مهندی رادی در تاریخ ۰۶ / ۰۶ / ۱۳۸۸

قالب دهی نظرات

خوب باید این خبر خوش را به شما بگویم، که بعد از مدت‌ها این سری آموزشی به اتمام رسید. اولین پست این سری آموزشی در تاریخ ۱۵ / ۱۳۸۷ نوشته شد و امروز یعنی در تاریخ ۲۶ / ۱۳۸۸ این سری آموزشی به پایان می‌رسد، یعنی نزدیک به ۷ ماه این سری آموزشی که شامل ۲۱ جلسه بود، ادامه پیدا کرد. دوستان زیادی به من گفتند که این سری آموزشی را سریعتر ادامه بدهم، و باید الان به اینان گفت که این سری در نهایت به پایان رسید. با وجود تمام سختی‌ها و مشکلات، باید بگویم که خوشحالم که توانستم خدمتی هر چند کوچک به وردپرس کرده باشم...



شما تقریباً همه چیز را می‌دانید. و برای کامل کردن اطلاعات وردپرسی خود این سری آموزشی را نیز ادامه دهید. امروز شما نحوه‌ی زندگی بخشی به تمام وبلاگ‌ها را یعنی نحوه‌ی قرار دادن بخش نظرات را خواهید آموخت. در تصورات من، وبلاگی که در آن خواننده‌هایش مجاز نباشند که نظر بدهند، وبلاگ نیست. البته برای برخی، بخش نظرات آنچنان مهم نیست ولی برای من واقعاً تصور نبود نظرات هم سخت است.

برای خواندن بقیه‌ی درس ادامه مطلب را مطالعه کنید....

در ابتدا باید بگوییم که روش ساختن بخش نظرات به دو شکل ممکن است، **یک** - حالت پیش فرض وردپرس، که با تایپ فقط یک خط کد امکانش میسر می‌شود. **دو** - روشی که ما نظرات را با نظر و سلیقه‌ی خود، درون فایلی بنام `comments.php` می‌نویسیم تا سبک نمایش را خودمان تعیین کنیم. که در این درس ما روش دوم را به شما می‌آموزیم.

البته شما باید بدانید:

- « روش دوم یعنی تنظیم کردن قالب در فایل `commnets.php` راه آنچنان استانداردی نیست و بهتر است که از پیش فرض وردپرس استفاده شود.
- « بیشتر طراحان وردپرس از الگو پیش فرض نظرات وردپرس استفاده می‌کنند.
- « بعضی از طراحان از الگوی پیش فرض وردپرس صرف نظر می‌کنند چون حتی آن نیازشان را برآورده نمی‌کند.
- « شما از ورژن اصلاح شده‌ی الگو پیش فرض نظرات من استفاده می‌کنید.

● مرحله‌ی اول:

- « یک فایل جدید بنام `comments.php` بسازید.
- « کدهای داخل فایل `comments.php` را به درون فایل `comments.txt` کپی کنید.
- « کدهای داخل فایل `comments.php` را `save` کنید.

● مرحله‌ی دو:

- « کدهای درون فایل `comments-template-css` را کپی کنید.
- « این کدها را در پایین یا در زیر سلکتور `#footer` بچسبانید.

● مرحله‌ی سوم:

در فایل `single.php`, در زیر تگ باکس مخفی ای که مشخصه‌ی `id` آن `entry` است، کدهای زیر را تایپ کنید:

```
<div class="comments-template">
<?php comments_template(); ?>
</div>
```



```
<div class="entry">
    <?php the_content(); ?>
    <?php link_pages('<p><strong>Pa
        <p class="postmetadata">
            <?php the_category(', ') ?> <?php _e(
                </p>
            </div>
            <div class="comments-template">
                <?php comments_template(); ?>
            </div>
```

تابع `(comments_template()` بخش نظرات و بلاگ را صدا می‌زند. البته بخش جالب این تابع این است که اگر فایلی بنام `comment.php` وجود داشته باشد، الگوی نمایش بخش نظرات پوسته از این فایل گرفته می‌شود و اگر چنین فایلی وجود نداشته باشد، این تابع نحوه‌ی نمایش بخش نظرات را از بخش پیش فرض وردپرس می‌گیرد که البته امکانات آن ممکن است راضی کننده نباشد.

شما همچنین می‌توانید از تابع `(comments_template()` در فایل `page.php` استفاده کنید، تا به بینندگان اجازه دهید تا امکان نظر دهی برای برگه‌های قالب شما نیز میسر باشد. اگر مایلید که اینکار را انجام دهید، دقیقاً همین مراحل را برای فایل `page.php` انجام دهید.

● مرحله‌ی چهارم:

خوب مراحل اصلی را انجام دادیم. این مرحله برای اعتبار نامه‌ی قالب است. یعنی با اینکار می‌توانید متوجه شوید که کدهای شما بی‌نقص هستند یا نه. البته اگر

نتوانستید این اعتبار نامه را بگیرید خیلی دلسوز نشوید، چون گرفتن این اعتبار نامه آسان نیست. حال برای گرفتن اعتبار نامه مراحل زیر را انجام دهید:

- « به آدرس `Page Source < View` یا `Source` بروید.
- « تمامی این کدها را کپی کنید.
- « به سایت `validator` بروید.
- « تمامی این کدها را در بخش `Direct Input` بچسبانید و یا `paste` کنید.
- « بروی دکمه `check` کلیک کنید.

برای مرجع بعدی (وقتی شما پوسته‌ی خودتان و بخش نظرات را طراحی کردید) از صفحات زیر نیز اعتبار سنجی کنید:

Home page	•
Archive pages	•
Category pages	•
Search result pages	•
: About) مثل Pages (•
pageSingle post	•
بدون بخش نظراتSingle post	•
با بخش نظراتSingle post	•
با پیغام باید برای نظر دادن حتماً وارد شویدSingle post	•
بدون محدودیتی برای نظر دادنSingle post	•
comments با Password protected single post	•

❖ توضیحات بیشتر برای الگوی نظرات:

« الگوی نظرات اساساً یک لیستی با تگ `ol` می‌باشد. اگر چه این تگ با سایر تگهای این سبکی فرقی ندارد (مانند تگهای `li`) اما یک تفاوت کوچک دارد. تگ `ol` هر آیتم را با یک نقطه از دیگری تمایز می‌کنند ولی تگ `ol` هر آیتم را با شماره از دیگری تمایز می‌کند. (اگر متوجه نشده‌اید، ناراحت نباشید، این خط در حقیقت تنها برای دادن اطلاعات بیشتر بود و در امتحان پایان ترم نمی‌آید).

« در فایل `single.php`، شما تگ `div` با نام `مشخصه‌ی آی دی` `comments-template` را در حقیقت تگهای `ol` توسط تگ `div` بدور تابع `comments_template()` قرار داده‌اید. در

محصور شده اند.

« وقتی شما مطلبی را با پسورد منتشر می کنید، بخش نظرات شما نیز همچنین بوسیله
ی پسورد محافظت می شوند.

```
<h2><?php _e('Password Protected'); ?></h2>
<p><?php _e('Enter the password to view comments.');?>" ?></p>
```

این کدهای عکس زیر هستند که لیست نظرات را می سازند:

```
<ol class="commentlist">
<?php foreach ($comments as $comment) : ?>
|           <li class=<?php echo $oddcomment;
|           <div class="commentmetadata">
|             <strong><?php comment_author_link() ?></strong>
|             <comment_time() ?></a> <?php _e('Said&#58;')
|                           <?php if ($comment->comment_approved == '1') &gt;
|                             <em><?php _e('Your comment has been approved')
|                           <?php endif; ?>
|             </div>
|             <?php comment_text() ?>
|             </li>
|             <?php /* Changes every other comment to a different class */
|               if ('alt' == $oddcomment) $oddcomment = 'comment';
|               else $oddcomment = 'alt';
|             ?>
|           <?php endforeach; /* end for each comment */
|           </ol>
```

و این آنچه که واقعاً منظور ماست در عکس زیر مشهود است:

```
<ol>
    <li>
        each comment sits here
    </li>
</ol>
```

تابع `comment_text()` محتوای نظر هر شخص را صدا می زند. مثلاً اگر من نظر داده باشم، که **چطوری رفیق خیلی مطلبت باحال بود داداش...** این نوشته دقیقاً توسط تابعی که گفته شد نمایش داده می شود.

من شرح کدها درون فایل `Style` را به خود شما واگذار می کنم . بر خلاف کدهای درون فایل `comments.php` شما واقعاً می توانید با کدهای `CSS` بدون ایجاد ایرادی بازی کنید تا به آنچه که نیاز دارید برسید.

خوب دوستان عزیز این سری آموزشی که درس مقدماتی نحوه ی ساخت پوسته بود به اتمام رسید. همانطور که گفته شد، در آینده می خواهم سری آموزشی جدیدی را با نام **سوال و جواب وردپرسی** راه بیاندازم بنابراین هر شخصی که سوالی در زمینه ی وردپرس و طراحی آن دارد، می تواند سوال خود را برای من مطرح کند تا از این سری آموزشی جدید که طی پستهایی سوال آنها مطرح و جواب داده خواهد شد، هم خودتان و هم دیگران استفاده کنید.

البته یک درس متفرقه ی دیگر باقی مانده و آن نحوه ی فارسی سازی قالب های وردپرس است که به آن نیز خواهیم پرداخت...

از اینکه مدتی را در خدمت شما دوستان بودم، بسیار خوشحالم و امیدوارم که این سری آموزشی مورد توجه شما قرار گرفته باشد. موفق و سر بلند باشید....



بزرگترین مرجع تخصصی آموزش و دانلود در افغانستان

نهایت لذت دانلود را ب ما تجربه کنید!

WWW.P30AF.COM