

به نام خدا

چندی قبل در تاریخ سوم دی سال ۱۳۸۷ فرصتی شد تا توفیق این را بیابم که نحوه ی ساخت پوسته های وردپرس را طی یک سری آموزشی برای شما آماده کنم. هرچند که در ابتدا پیش بینی می شد 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 اندازه ی ممکن بتوانیم وبلاگ نویسان عزیز را با ساختار وردپرس آشنا کنیم. البته هر که مایل بود، همین سری آموزشی می تواند برایش پله ای برای حرکت به سوی وردپرس پیشرفته باشد. با توجه به اینکه وردپرس حقی بزرگ به گردن ما دارد، شاید این آموزش کمی از محبت های او را نسبت ما جبران کند....

سخن را دراز نمی کنم....

امیدوارم که این سری آموزشی مورد توجه شما دوستان وردپرسی قرار گیرد.

پیروز و سربلند باشید.....

مهدی رادی

۱۳۸۸/۷/۱

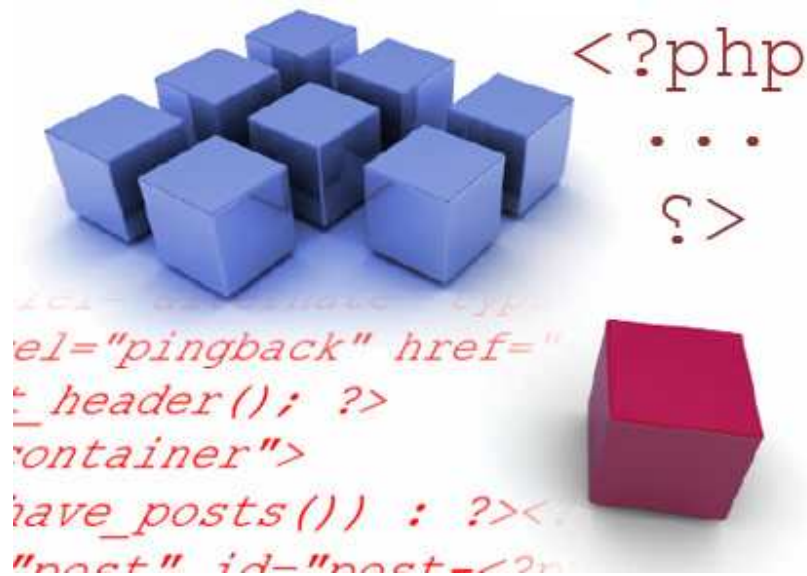


این سری آموزش همانگونه که بروی سایت [کمک رایانه \(p30help\)](#) قرار گرفته، با اندکی تغییر در اینجا نیز نمایش داده شده است. در انتهای هر پست، آدرس منبع مطلب نیز گذاشته شده است تا در صورتی که با مشکلی برخوردید یا سوالی داشتید، به کمک آن آدرس، براحتی بتوانید به صفحه ی اصلی مطلب دسترسی پیدا کرده و مشکل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

چگونه برای وردپرس پوسته بسازیم؟

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

مدتی است که قصد داشتم آموزش نحوه ی ساخت پوسته برای وردپرس را بدهم ، امروز تصمیم خود را عملی کرده و اولین درس از این سری آموزشی را به استحضار بینندگان عزیز می رسانم. این سری آموزشی احتمالاً شامل ۱۲ الی ۱۴ درس می باشد. مطمئناً با خواندن این سری آموزشی قرار نیست که یک شبه قالب های آنچنانی بسازید، اما به شما قول می دهم که اگر این سری را به صورت کامل پی بگیرید و مطالب گفته شده را آنگونه که من می خواهم دریافت کنید ، علاوه بر آشنایی با بخش های اصلی وردپرس (البته در بخش پوسته ها) قادر به رفع بسیاری از مسائل موجود در پوسته تان نیز خواهید بود. البته به غیر از اینکه این سری آموزشی شاید پله ای باشد برای حرفه ای شدنتان . به هر حال ، به امید خدا ، این سری را شروع می کنیم ، 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 مقبولى را بسازيد ، اما بايد كمى هم كار كنيد و مسلماً از دانش بكار رفته در ساير پوسته ها نيز استفاده كنيد . شما قادريد تا در پايان اين سري آموزشي قالبى همچون [قالب راديان](#) بسازيد (و البته خيلى بهتر از اين) . من براى ساخت اين قالب از معلوماتى در حد همين سري آموزشي استفاده كرده ام.

در اولين درس از اين سري آموزش به آموزش مفاهيم اوليه و قوانين اصلى در كد نويسى را خواهيم پرداخت. مطمئناً اگر كمى php و xhtml كار كرده باشيد فهم بسيارى از مطالب برايتان راحت تر خواهد بود ، ولى اگر هيچ گونه آشنائى با اين دو زبان نداريد ، حتماً به قوانين گفته شده در اين مباحث توجه كنيد.

قبل از اينكه درس را به صورت جدى شروع كنيم ، دو كار را بايد انجام داده باشيد.

۱ – نصب يك سرور خانگى بروى رایانه تان با استفاده از برنامه هاىي مانند WinLAMP يا Xampp

۲ – نصب وردپرس بروى سرور خانگى

توجه ۱ : پيشنهاد مى كنم كه از برنامه ي Xampp استفاده كنيد . براى اينكه بتوانيد هر دو مورد بالا را پوشش بدهيد حتماً [اين مطلب](#) را مشاهده كنيد.

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 صفحه ی زیر را مشاهده کنید:

[WordPress.org's documentations, Design and Layout](https://WordPress.org's_documentations,_Design_and_Layout)

در غیر اینصورت با ما همراه باشید. فقط ۲ نکته ی آخر را نیز بگویم 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 می برم ، زیرا بسیاری از مفاهیم ، با معنی کردن ، هویت خود را از دست می دهند و آنچه را که من می خواهم را به خواننده القا نمی کنند. البته سعی کرده ام که تا حدودی فارسی را نیز پاس بدارم.

از آنجایی که هر درس شامل عکس و متن می باشد ، 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

« سلسله مراتبی بودن

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

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

```

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

```

نکته : تگ چیست ؟ هر تگ بوسیله ی دو کاراکتر < و > ساخته می شود. 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 اینگونه ” باز شده است و اینگونه ” نیز بسته شده است . اگر دقت کنید در بین تگ باز و بسته یکسری کدهای دیگر آمده است. این کدها چون بین تگ قرار دارند بنابراین خاصیت تگ را نیز به خود می گیرند. در ادامه با نحوه ی 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>
```

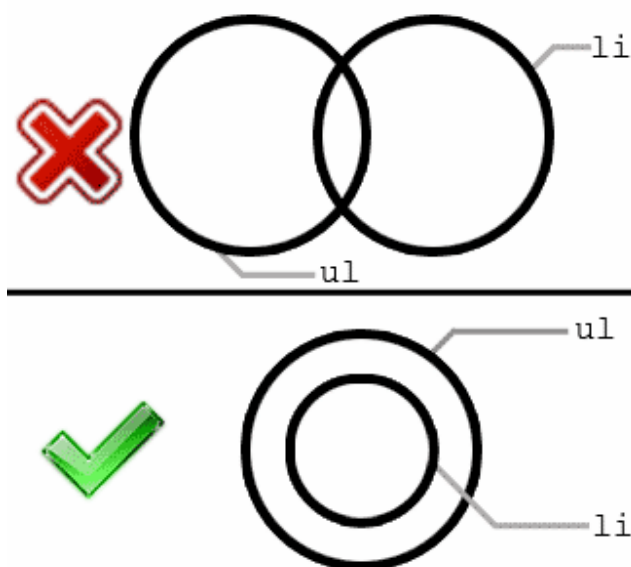


The wrong way to close:

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



اگر یک تگ باز و بسته را دایره ای در نظر بگیریم ، هیچ دایره ای نباید همدیگر را قطع کند ، بلکه باید دایره 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 مسلماً شما پوسته هایی را که تا بحال مشاهده کرده اید ، شامل فایل های 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 تترسید ، در درس‌های بعدی خواهید فهمید که هر فایل برای چیست و اینکه ترستان بی‌مورد بوده است.

« زبان ویژه / اصطلاحات مخصوص (Lingo):

مسلماً در هر مبحثی اصطلاحات و کلمات ویژه ای 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 چندباره ی آنها در مکان‌های مختلف استفاده کنید.
- ۲ – **Template file** : یک فایل که شامل یک یا چند گروه از کدها (Template) می‌شود. هر صفحه ی وبی شامل چندین Template file می‌شود ، مانند فایل‌های index.php ، style.css ، sidebar.php و...
- 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 وبلاگ شما و یا هر وبلاگ دیگری هستند . برای مثال یک صفحه از دفترچه ی خاطرات شما هم یک Post محسوب می شود.

۵- Page : یک نوع خاص از Post که در زیر مجموعه ی دسته ها (Category) نیز قرار نمی گیرد. Page ها از گروه Post سایت شما مجزا هستند. در زبان فارسی (بهتر است بگوییم در وردپرس فارسی) Page را همان برگه معنی کرده اند.
توجه : در وردپرس Page و page دو چیز مختلف هستند. یک P و p بین آنها تفاوت بزرگی را ایجاد می کند ، که در آینده بیشتر با این تفاوت آشنا می شویم.

« سلسله مراتبی بودن:

شکل زیر به راحتی به شما نشان می دهد که ، سیستم وردپرس به جستجوی چه فایل هایی می گردد. البته در عکس زیر بعضی از فایلها آورده نشده اند . من تنها ۶ فایل از ۱۳ فایل موجود را برایتان لیست کرده ام. با این فایلها در درس های آینده بیشتر آشنا خواهیم شد.



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

اگر از مجموعه ی 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 می باشد و در صورت نبود هر یک از فایل های موجود ، فایل index.php مسئولیت آنرا قبول می کند تا محتویات آن صفحه را نشان دهد.

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

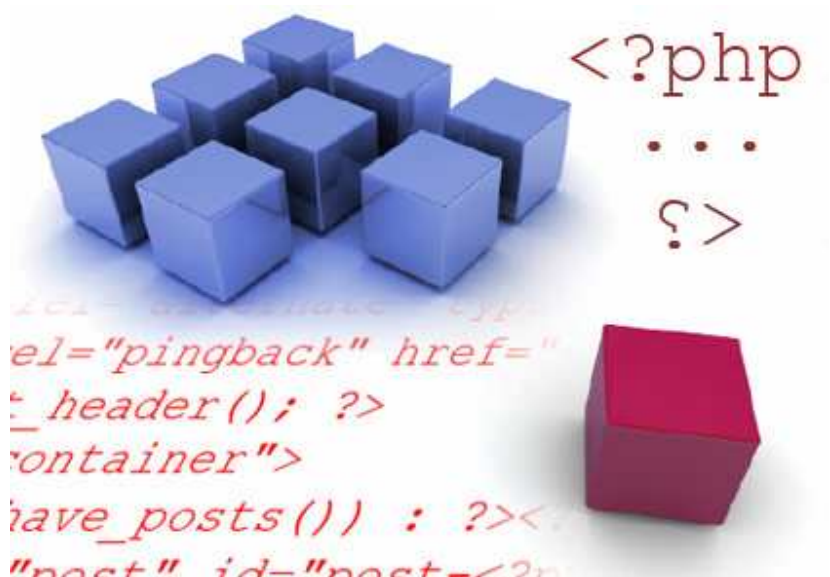
منبع :

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

چگونه برای وردپرس پوسته بسازیم؟ (بخش ۲)

نوشته شده توسط [مهدی رادی](#) در تاریخ ۱۳۸۷ / ۱۰ / ۰۶

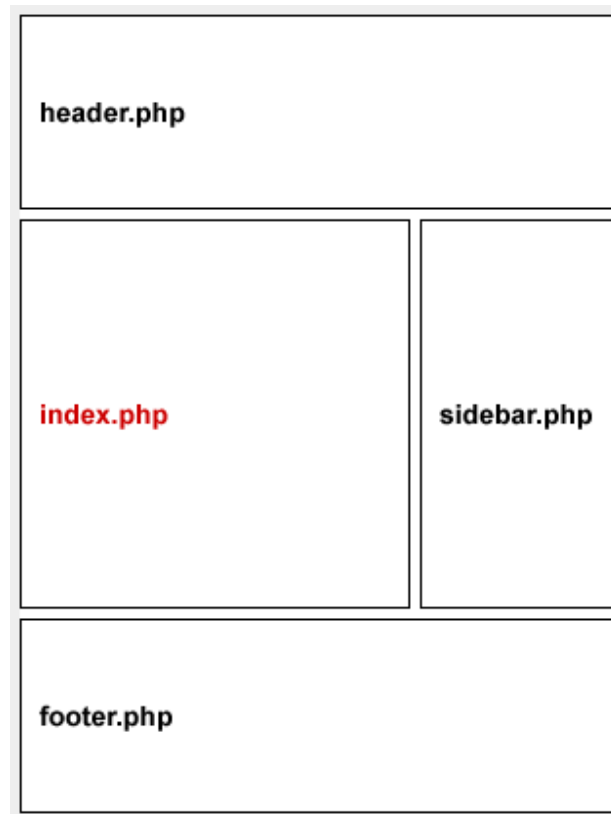
در دومین درس از سری آموزشی ساخت پوسته ی وردپرس بازهم به توضیح و معرفی برخی دیگر از مفاهیم می پردازیم . و از جلسه ی بعد به طور رسمی کار با فایل index.php را شروع خواهیم کرد. اگر شما [درس اول](#) از این سری را نخوانده اید ، این بخش را ادامه ندهید و در ابتدا درس اول را مطالعه کنید ، در غیر این صورت برخی از کلمات و توضیحات گفته شده در این بخش برای شما نا آشنا خواهند بود.



در این درس قصد داریم تا با ساختار **files template** و **templates** 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 ، صفحه ی مورد نظر از چهار template files تشکیل شده است که عبارتند از:

header ، index ، sidebar و footer

« و اما **Header Template File** چیست؟

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

<p>Blog title</p> <p>Description of your blog.</p>

« حال **Index Template File** چیست؟

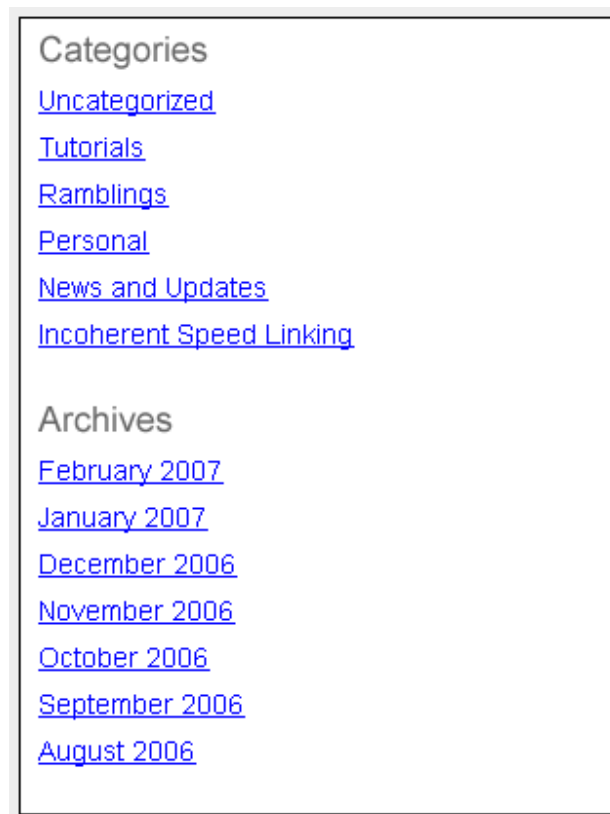
این فایل یکی از مهمترین بخش ها محسوب می گردد و شامل عنوان مطالبان (*post title*) ، محتوای مطالبان (*content*) (مانند نوشته و عکس هر *post*) و اطلاعات هر پست (*post meta data*) (مانند اسم نویسنده ی مطالب ، تاریخ انتشار ، زیر دسته ، تعداد نظرات و ..) می شود.



« Sidebar Template File چیست؟

توسط این بخش ، برگه ها (*Page links listing*) ، دسته ها (*Category*) ، آرشیو (*archive links listing*) ، پیوندها (*blogroll listing*) و وبلاگتان کنترل می شود . (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) شما نیز توسط Sidebar Template File کنترل می شوند.



« و در آخر Footer Template File چیست؟

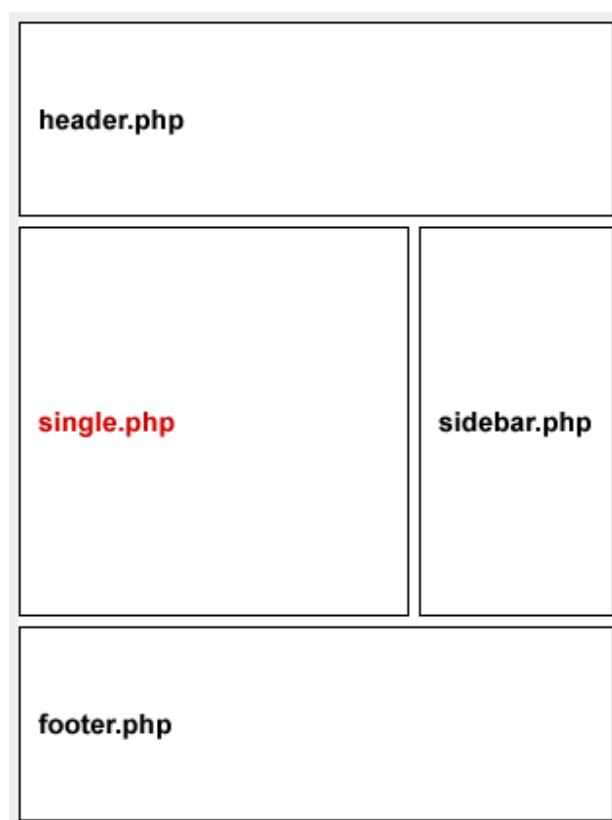
این template file شبیه بخش header.php و یا همان Header Template File می باشد. **فوتر** (Footer) معمولاً از صفحه ای به صفحه ی دیگر ثابت می ماند . شما هر چیزی را می توانید در این فایل قرار دهید ، اما معمولاً داخل این فایل اطلاعات **کپی** رایت نوشته می شود.

Copyright 2007 YourBlog.com

حالا به من اجازه دهید تا توضیح بدهم 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 Footer ، Header و sidebar که با رنگ سیاه مشخص شده اند معمولاً قسمتهای **ثابتی** هستند.

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

اما اگر نمی دانید که صفحات تنها (single) چه هستند باید بگویم ، همان صفحاتی که ادامه ی مطلب در آن نوشته می شود، برای مثال شما که همین الان در حال خواندن این مطلب هستید در یک صفحه ی تنها (single) قرار دارید . در صفحات تنها امکان نظر گذاشتن نیز وجود دارد.



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

منبع :

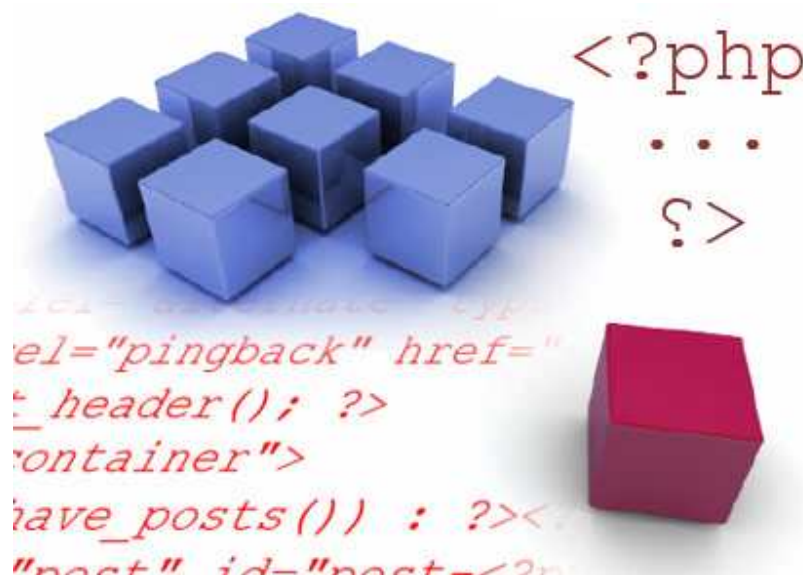
<http://www.p30help.ir/how-make-wordpress-theme-lesson-2>

چگونه برای وردپرس پوسته بسازیم؟ (بخش ۳)

نوشته شده توسط [مهدی رادی](#) در تاریخ ۱۱ / ۱۰ / ۱۳۸۷ در دسته ی [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 پوسته ی وردپرس نیز خواهد شد. اگر شما درسهای اول و دوم را نخوانده اید و تازه کار نیز هستید، برای اینکه سرنخی دست تان بیاید، همین الان از [مطلب اول](#) شروع کرده و به جلو حرکت کنید.



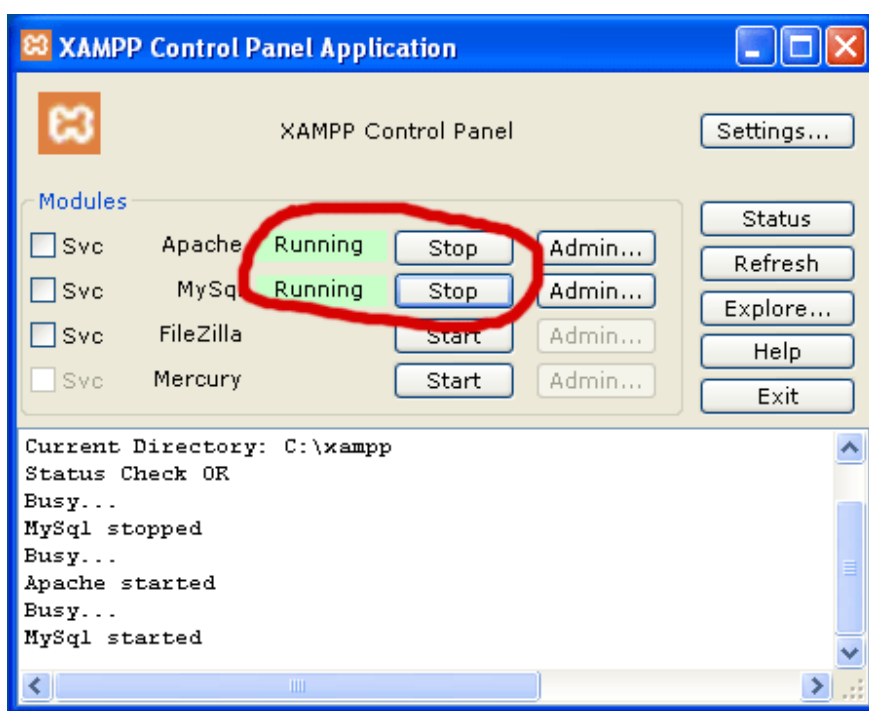
حال وقت آن رسیده است که خواندن محض را رها کرده و بصورت عملی دست به ساخت پوسته ی وردپرس بزنید. در این درس، شما دست هایتان به کدهای وردپرس کمی آشنا خواهد شد. قبل از شروع درس باید حتماً یک وبلاگ وردپرس، بروی رایانه تان نصب کنید (نه بصورت آنلاین، بلکه آفلاین زیرا بسیار آسانتر است بویژه برای ما ایرانیان)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

بعلت طولانی شدن صفحه به ادامه مطلب مراجعه کنید.

● مرحله ی اول: باز و فعال کردن برنامه ی Xampp

به فولدر اصلی این برنامه بروید. *xampp < My Computer* و یا <C:\xampp> (یا هر جای دیگری که این برنامه را نصب کرده اید).

بروی **xampp-control.exe** کلیک کنید . پنجره ای 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.

● مرحله ی دوم : ساخت یک فولدر برای پوسته

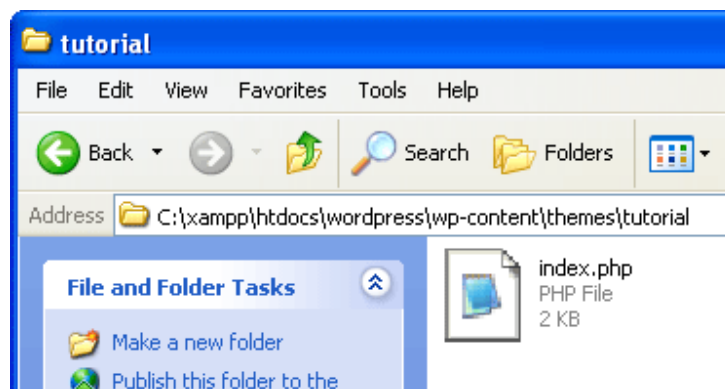
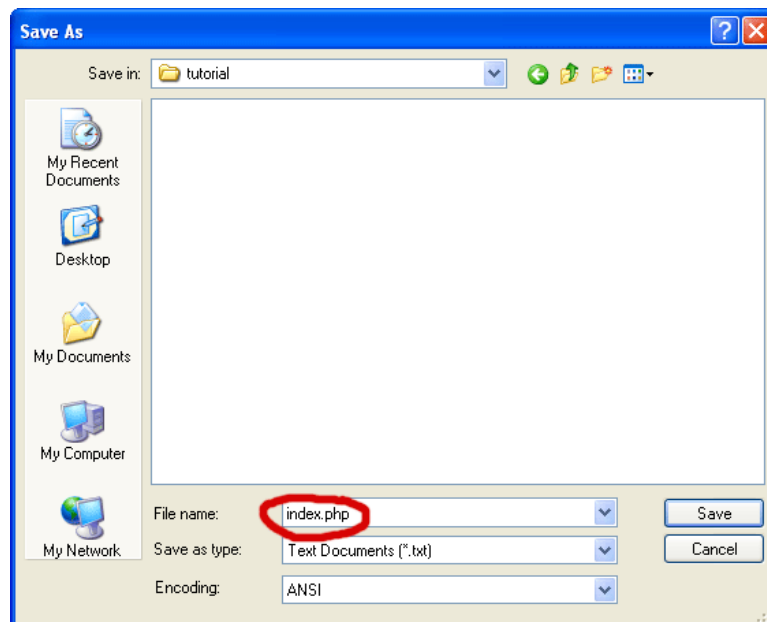
به بخش پوسته های وردپرس بروید ، یعنی آدرس زیر:
xampp/htdocs/wordpress/wp-content/themes
 در این مکان فولدری بنام **tutorial** بسازید. به همین سادگی...

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

برنامه ی notepad یا هر ویرایشگر متن مورد علاقه ی خود را باز کنید.

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

بعد از اینکار فایل notepad را درون پوشه ی **tutorial** بنام **index.php** ذخیره کنید.



دوباره ی یک notepad جدید باز کنید (باید کاملاً خالی باشد) . در همان حالت کاملاً خالی فایل notepad را با نام **style.css** درون فولدر **tutorial** ذخیره کنید. حال پنجره های notepad را ببندید.

بعد از این اعمال شما درون این فولدر باید فایل‌هایی با نام های **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 ناحیه هر کد را شرح دهم. (برای بزرگ دیدن عکس بروی آن کلیک کنید.)

```
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 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>
<?php
</body>
</html>
```

« **Doctype** - این عبارت نشان می دهد که شما از چه نوع کدی در پوسته تان استفاده کرده اید. فهم Doctype زیاد اهمیت ندارد . من خودم ویژگیهای Doctype را تعیین کردم ، بنابراین شما نیازی نیست که در این مورد زیاد فکر کنید.

« **<html>** - این تگ نشان می دهد که صفحه ی وب شما از کجا شروع می شود.

« **<head>** - این تگ به ما نشان می دهد که سر (**head**) صفحه ی وب ما از کجا شروع می شود. هر صفحه ی وبی 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 فولدر که در اینجا فولدر tutorial می شود) بنابراین پوسته ی ما می تواند به آن و یا هر استایل دیگری برای صفحه ی وب مان لینک بدهد . هرگاه کدی قرار بگیرد بین عبارت های **<?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 دهنده ی پایان کدهای پی اچ پی می باشد.

بنابراین :

« **<?php** - یعنی شروع کدهای php
 « **bloginfo('stylesheet_url')** - صدا می زند برای مکانی که style.css 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 **>?>** بیاید ، 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 بدنه ی اصلی صفحه ی وب ما از کجا شروع می شود. بدنه شامل هر چیزی می شود که ما بروی صفحه ی وب می بینم و می خوانم . شمایی که در حال خواندن این آموزش هستید یعنی در حال نگاه کردن به بدنه ی

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

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

• **مرحله ی چهارم :** هر چیزی که داخل فایل [style.txt](#) هست را به درون فایل style.css که در فولدر tutorial قرار دارد ، کپی و پیست کنید و تغییرات را save نموده و صفحه را ببینید.

• **مرحله ی پنجم :** پوسته تان را منتشر کنید
برای اینکار یک مرورگر را باز کنید

درون آدرس بار مرورگرتان تایپ کنید عبارت :

<http://localhost/wp-admin/wordpress>

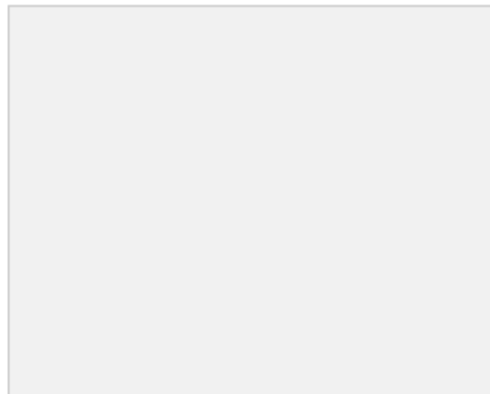
و سپس به داخل بخش مدیریت وردپرس بروید.

(**wordpress** که با رنگ قرمز مشخص شده است نشان دهنده ی این است که این عبارت فقط یک مثال است و بستگی دارد به نامی که شما برای فولدر وردپرس گذاشته اید.)

هنگامی که داخل بخش وردپرس شدید ، به بخش مدیریت پوسته ها بروید و پوسته ی **Tutorial** را فعال نمایید.

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

Tutorial 1.0

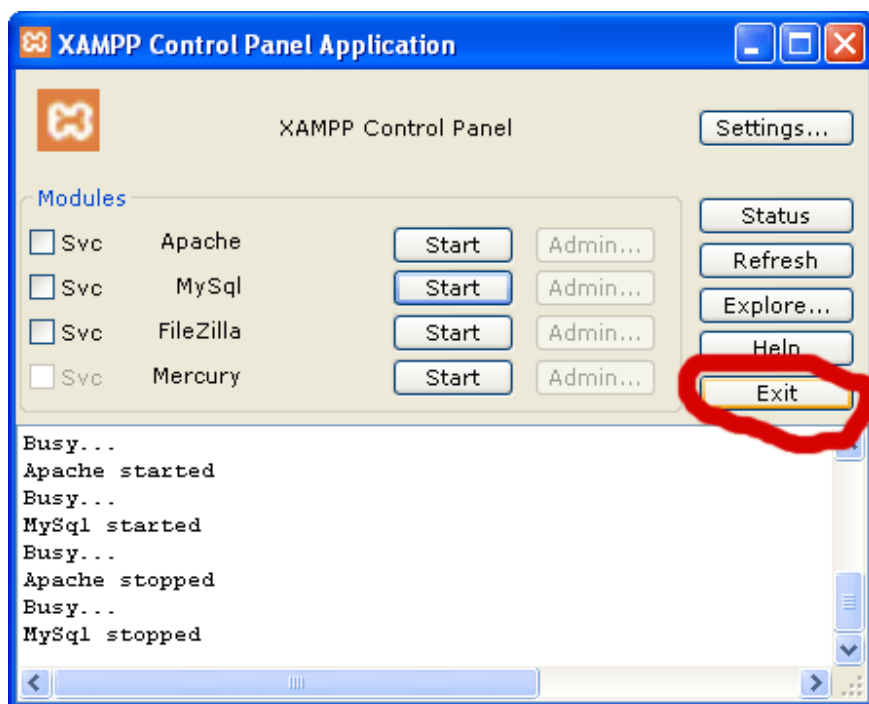


This is my theme for a tutorial.

حال یک صفحه ی جدید در مرورگر باز کنید و بروید به آدرس <http://localhost/wordpress> . شما یک صفحه ی خالی را (page blank) خواهید دید . منظورم یک صفحه ی کاملاً خالی است. اگر صفحه ی باز شده کاملاً خالی نیست ، شما صفحه ی اشتباهی را گشوده اید و یا اشتباهی در کارتان بوده است. توجه : **wordpress** که با رنگ قرمز مشخص شده است نشان دهنده ی این است که این عبارت فقط یک مثال است و بستگی دارد به نامی که شما برای فولدر وردپرس انتخاب کرده اید.

حال پوسته ی شما منتشر شده است. فکر کنم که برای این جلسه کافیه . در جلسه ی 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>

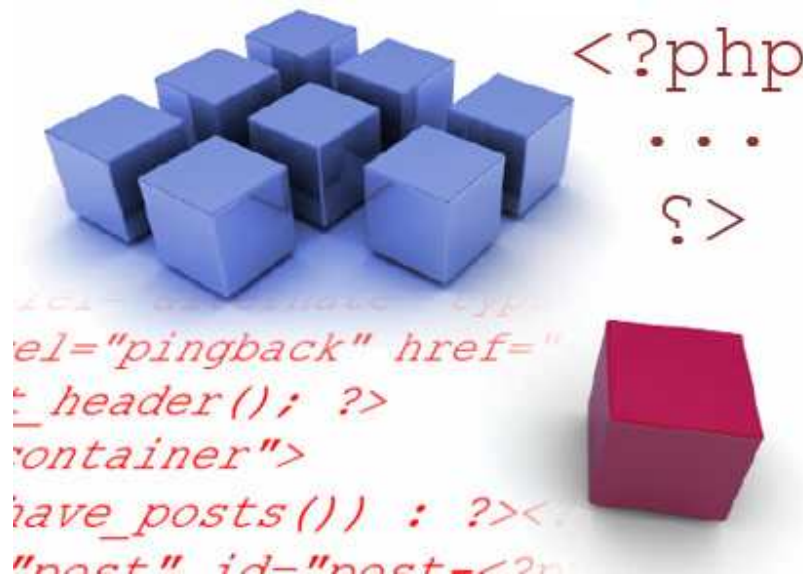
چگونه برای وردپرس پوسته بسازیم؟ (بخش ۴)

نوشته شده توسط مهدی 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 پوسته ی 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، همین جا دست نگه داشته و از

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



در جلسه ی قبل، من به شما نشان دادم که چگونه در بخش کنترل برنامه Xampp ، آپاچی و MySQL را فعال کنید. پوسته تان 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 فراخوانی کنید.

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

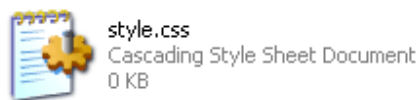
نکته مهم: بیاد داشته باشید ، هر کدی را **تایپ کنید**، و از انجام کپی و پیست پرهیز نمایید. این مسئله باعث می شود تا هر آنچه را که فرا گرفته اید، در خاطرتان بماند.

خوب درس را شروع می کنیم:

● مرحله اول:

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

دارند را باز کنید. شما هم اکنون باید ۲ فایل که جلسه ی قبل ساخته اید را مشاهده کنید. فایل های **index.php** و **Style.css**

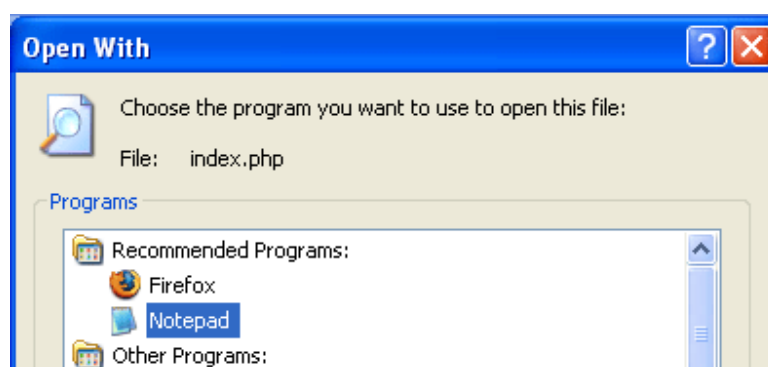
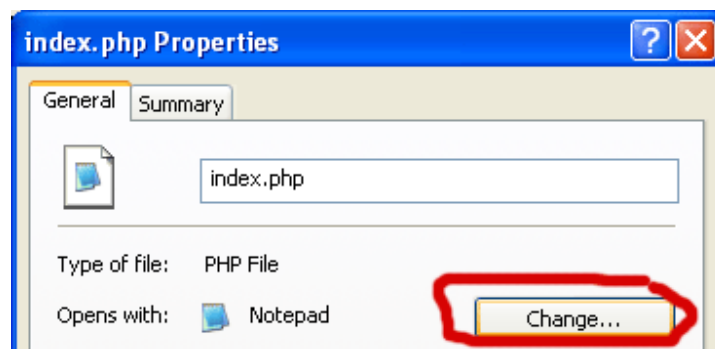
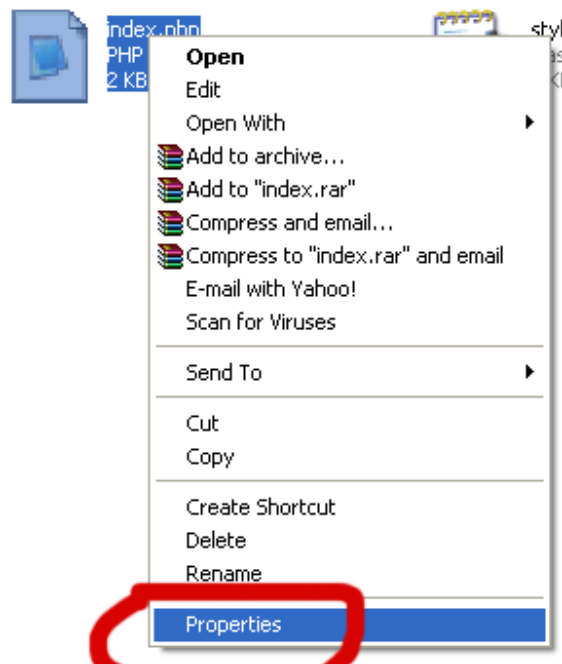


فایل های **index.php** و **style.css** باید با فایل های **index.txt** و **style.txt** پر شده باشند (یعنی باید محتویات داخل آنها یکی باشد).

● مرحله دوم:

– یک پنجره از مرورگر وب باز کنید. سپس به آدرس <http://localhost/wordpress> بروید. شما باید یک صفحه ی خالی مشاهده کنید ، چون شما یک پوسته ی خالی را در جلسه ی قبل نصب کردید. (**wordpress** که با رنگ قرمز مشخص شده است نشان دهنده ی این است که این عبارت فقط یک مثال است و بستگی دارد به نامی که شما برای فولدر وردپرس انتخاب کرده اید).

– به فولدر پوسته هایتان باز گردید و فایل index.php در notepad باز کنید. برای اینکار بروی Index.php راست کلیک کرده ، properties 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، فولدر پوسته هایتان، یک مرورگر وب و فایل index.php را (با notepad) باز کرده اید.



● مرحله ی سوم:

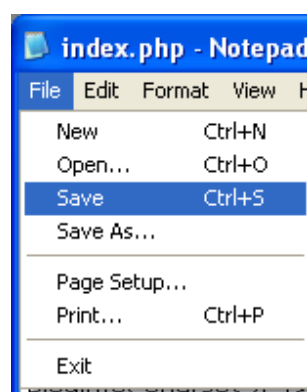
شما باید به محتویات داخل فایل index.php دقت کنید. حال برای شروع:

عبارت `<?php bloginfo('name'); ?>` را در بین تگ های `<body>` و `</body>` تایپ کنید و تغییرات را save نمایید.

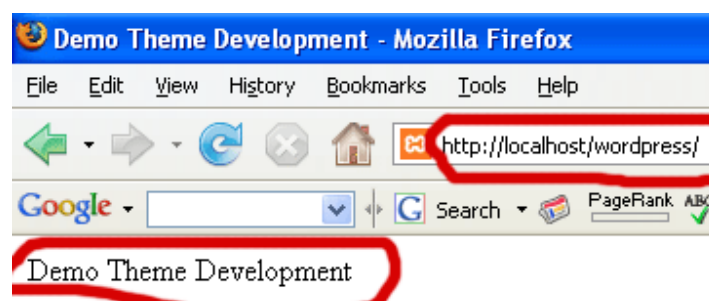
```
<body>

<?php bloginfo('name'); ?>

</body>
```



سپس بازگردید به صفحه ی مرورگر وب تان و آن صفحه را Refresh کنید(بوسیله ی دکمه ی F5 هم می توانید 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 پراتنز 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.
>? - پایان کدهای php

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:

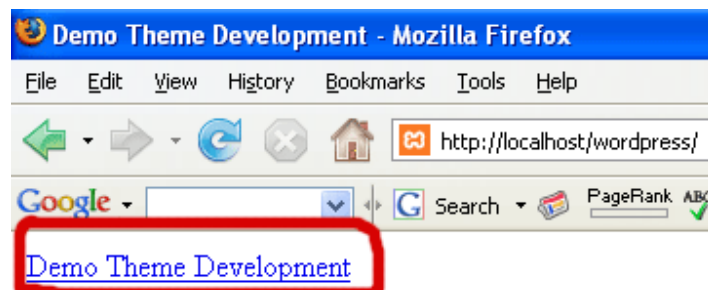
اگر دقت کرده باشید عنوان و بلاگتان (***title blog's***) قابل دیدن است، ولی قدرت لینک ندارد، یعنی قادر نیستم که بروی عنوان و بلاگ کلیک کنیم تا به صفحه ی اول سایت برویم.

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 مروگرتان بازگردید و صفحه را refresh کنید. هم اکنون باید امکان لینک را برای عنوان وبلاگتان 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 وبلاگتان، یک لینک بسازید، این روش را بکار بگیرید. یعنی:

عبارت `<?php bloginfo('url'); ?>` در بین نقل قول (' ') دستور **href=** 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، Status bar باید به شما آدرس <http://localhost/wordpress/> را به شما نشان می دهد(دلیل اینکه Wordpress با رنگ قرمز مشخص شده است را دیگر نمی گویم).



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

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

♦ حال توضیح و آنچه اتفاق افتاده است؟

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

« **bloginfo('url')** - اطلاعات وبلاگ شما را فراخوانی می کند. مخصوصاً در اینجا که آدرس یا URL صفحه ی اصلی تان را فراخوانی می کند.

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

« **** - تگی است که امکان لینک دار شدن یک نوشته را می بندد . در صورت عدم استفاده از این تگ ، صفحه ی وب شما نمی داند که نوشته ای که لینکدار کرده اید کجا پایان می یابد و در این صورت یک لینک بزرگ خواهیم داشت، یعنی تمام نوشته های وبلاگ به یک آدرس لینک دار خواهند شد. **همچنین قانون اول** را بیاد بیاورد، هر چیزی را که باز کرده اید، باید ببندید.

« **href=""** - این عبارت مخفف کلمه ی **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 <http://localhost/wordpress> را بعنوان مقدار **href=""** قرار ندادیم.

دلیلش کاملاً واضح است، بدین خاطر که در صورتی که شما <http://localhost/wordpress> را بعنوان مقدار قرار دهید ، حتی اگر این پوسته بروی یک وب سایت قرار بگیرد. هنگامی که بروی عنوان وبلاگ آن سایت برویم ، بجای آوردن آدرس صفحه ی اصلی آن سایت، آدرس <http://localhost/wordpress> را به ما خواهد 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 صفحه ی اصلی وبلاگم قرار دادم. برای اینکار من از تابع PHP یعنی **bloginfo('url')** استفاده کردم، تا آدرس و یا URL وبلاگم را صدا بزنم. در آخر هم اسم لینک را با آوردن **<?php bloginfo('name'); ?>** همان عنوان وبلاگم 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 نگرانید که چرا اینقدر کند پیش می رویم باید بگویم 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:

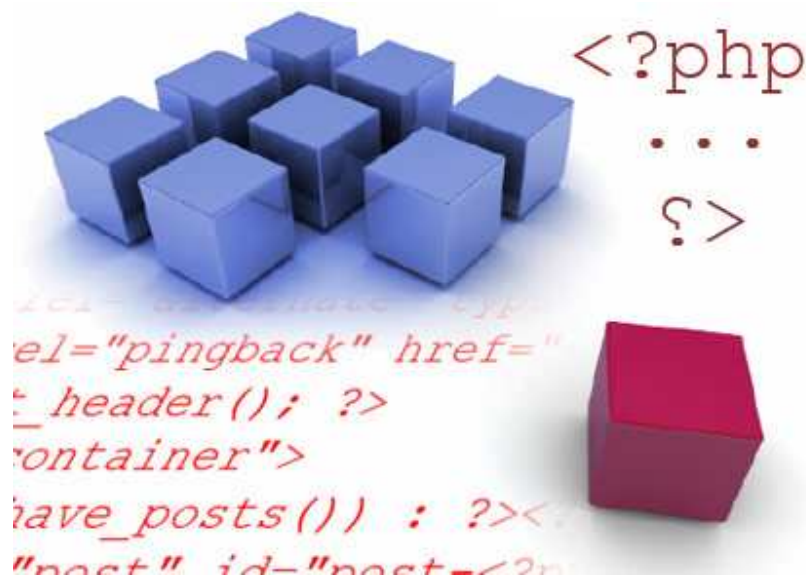
<http://www.p30help.ir/how-make-wordpress-theme-lesson-4>

چگونه برای وردپرس پوسته بسازیم؟ (بخش ۵)

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

کار با Header Template (بخش ۲)

در این پست بخش پنجم از این سری آموزشی را پی خواهیم گرفت . این درس ادامه ی درس قبلی است. باز هم می گویم که اگر این سری را از ابتدا دنبال نکرده اید، 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 فهمیده اید(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** **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.

● مرحله ی اول:

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

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

● مرحله دوم :

درون فایل **index.php** باید عبارت زیر را که از جلسه ی قبل کار کرده ایم، وجود داشته باشد:

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

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

نکته: اگر بخواهیم به زبان ساده تگ **H1** را توصیف کنم، باید بگویم که در واقع تگ **H1**، یک سایز پیش فرض از فونت می باشد. یعنی اگر این تگ بدور یک نوشته ی ما قرار گیرد، همانگونه که در بالا گفته شد، نوشته ی ما بزرگترین سایز پیش فرض را می گیرد و اگر تگ **H6** قرار بگیرد، نوشته ی ما کوچکترین سایز پیش فرض را به خود می گیرد. دلیل اینکه می گوئیم سایز پیش فرض این است که می توانیم در فایل **Style.css** این تنظیمات را تغییر داده و اندازه را با توجه به خواسته ی خودمان تغییر دهیم. که البته در جلسات بعد با اینگونه اعمال بیشتر آشنا می شوید.

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 وبلاگتان در زیر لینک عنوان وبلاگتان (*blog's title*) اضافه شده است. این نکته را نیز بگوییم که شما می توانید توضیحات وبلاگتان (*description*) را در بخش مدیریت وردپرس تنظیم کنید.

♦ توضیح آنچه اتفاق افتاده است:

«<?php - شروع کدهای php
bloginfo('description') - اطلاعات وبلاگتان را (بویژه در اینجا توضیحات وبلاگتان (*description*) را) صدا می زند. (به دلیل اینکه مقدار bloginfo("") را description قرار داده ایم، فقط توضیحات وبلاگتان (*description*) فراخوانی می شود).

« - صدا زدن توضیحات وبلاگ (*description*) را متوقف می کند.
«>? - پایان کدهای php

● مرحله ی چهار:

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

بدور کدهایی که تاکنون نوشته اید تگ های **<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 سازنده ی 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 وبلاگ (*blog's title*) و توضیحات وبلاگ (*description*) را از هر چیز دیگر موجود در صفحه جدا کنیم، حال اگر شما از این تگ استفاده نکنید. هیچ محتوایی از پوسته ی 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، می توانید بعد ها در فایل **style.css** برای باکس های مخفی (تگ **div**) ویژگی تعریف کنید و در حقیقت سایت تان را بسازید. کارهایی نظیر حاشیه دادن ، لایه گذاری ، لبه گذاشتن ، عکس پس زمینه ، رنگ پس زمینه و غیره.. را می توانید بروی این تگها اعمال کنید.

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

● مرحله ی پنجم:

مقدار **id="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 تغییری را مشاهده نخواهید کرد. ما یک **id** را به تگ **div** اضافه کردیم. دلیل اینکه ما به این تگ **div** یک **id** 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 ی 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 ی 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 مهدی رادی در تاریخ ۰۶ / ۱۱ / ۱۳۸۷ در دسته ی PHP, HTML, وبمستر, وردپرس, پوسته

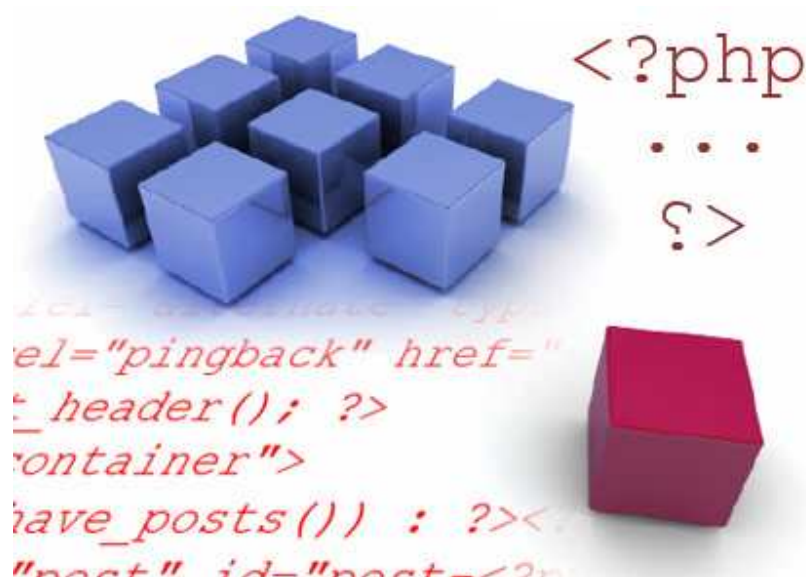
حلقه ها (Loop)

در این درس به آموزش نحوه ی آوردن بخش ورودی های وبلاگ یا همان پست های وبلاگ تان خواهیم پرداخت. دوباره مجبورم قبل از شروع درس، این نکته را بگویم که اگر این سری آموزشی وردپرس را از ابتدا دنبال نکرده اید، حتماً 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 حلقه ها آشنا هستید. حلقه ها یکی از اجزای مهم در برنامه نویسی می باشند. بوسیله ی حلقه ها قادر هستید تا یک فرمان را به اندازه ای که مایلید، تکرار کنید. روشهای مختلفی برای ایجاد حلقه وجود دارد، که البته در بحث امروز ما نمی گنجد. ما در این درس به کمک حلقه 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 وبلاگمان را صدا بزنیم و سپس از آن یک لینک بسازیم.
- چگونه توضیحات وبلاگ 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 ی Xampp را راه اندازی کنید. فولدر tutorial را در بخش پوسته های وردپرس بگشایید، یک مرورگر وب باز کنید و آدرس <http://localhost/wordpress> را در آن تایپ کنید، و در آخر فایل **index.php** را در notepad هم باز کنید.

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

```

index.php - Notepad
File Edit Format View Help

<!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_type'); ?>" />
    <meta name="generator" content="WordPress <?php bloginfo('version'); ?>" />

    <link rel="stylesheet" href="<?php bloginfo('stylesheet_url'); ?>" type="text/css" />
    <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.

● مرحله ی اول:

یک تگ **div** (باکس پنهان) را زیر ناحیه **header** اضافه کنید و به آن مقدار **id** 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 می آید، جدا کند. چیزهایی مانند Sidebar و Footer

● مرحله ی دوم:

کد زیر را بین تگ باکس مخفی ای که مقدار id آن container هست، اضافه کنید. کد های 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 آسانتر خواهد بود، تا متوجه شوید که کد نوشته شده چه عملی 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 تشخیص 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 بدهید. 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 کردم. بعدها که صفحه ی شما پر از کد شد، قدر اینکار را خواهید دانست.

♦ چه چیز اتفاق افتاد؟

« **if(have_posts())** - یک عبارت شرطی است و چک می کند که شما آیا پستی دارید یا خیر؟ در حقیقت این کد را می توان اینگونه خواند: «اگر پستی وجود داشت اجرا کن...»

« **while(have_posts())** - این کد همان حلقه ی مورد نظر ماست و معنی آن می شود: «تا زمانی که پستی وجود داشت، **the_post()** را اجرا کن.»

« **the_post()** - پست ها را برای نشان دادن صدا می زند.


« **endwhile;** - همان قانون اول است - این عبارت حلقه ی **while()** را می بندد.

« **endif;** - شرط **if()** را می بندد. (قانون اول)







توجه: لازم نیست تمام کدهایی را که باز کرده ام، حتماً در آخر ببندیم بعضی از کدها از دو بخش ساخته نشده اند. برخی کدها می توانند خودشان، خودشان را ببندند، مثل: **have_posts()** یا **the_post();** . کمی که با این مباحث بیشتر آشنا شوید، به راحتی می توانید 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 وبلاگتان (**blog's title**) را بوسیله ی **bloginfo('name')** صدا بزنید. حالا، باید برایتان آسانتر باشد که چگونه عنوان پست ها (**post titles**) را صدا بزنند. برای اینکار در بین حلقه ها (The Loop) عبارت **<?php the_title(); ?>** را بعد از **<?php the_post(); ?>** و قبل از **<?php endwhile;** تایپ کنید. مانند زیر:

 Demo Theme Development - Mozilla Firefox

File Edit View History Bookmarks Tools Help

      http://localhost/wordpress/

Google  Search  PageRank  ABC  Check 

Just another WordPress weblog
Hello World Hello World Hello World Hello World Hello World Hello
eiusmod tempor

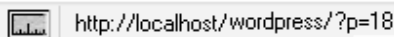
● مرحله چهارم:

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 `` 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>
```

`the_permalink()` یک تابع از `php` است، که آدرس یا مکان هر پست را صدا می زند. صفحه را `save` کنید و سپس صفحه ی مرورگر را `refresh` نمایید . نشانگر موس را بروی یکی از عنوانهای `Hello World` برده و به `Status bar` نگاه کنید. خواهید دید که در `Status bar` آدرس 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 صفحه ی 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 heading صفحه ی 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 صفحه ی کد شما به شکل زیر باشد:

```
<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(); ?>
        <h2><a href="<?php the_permalink(); ?>"><?php the_title(); ?></a></h2>
    <?php endwhile; ?>
<?php endif; ?>
</div>
</body>
</html>
```

خوب درس ما به پایان رسید.

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

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

منبع:

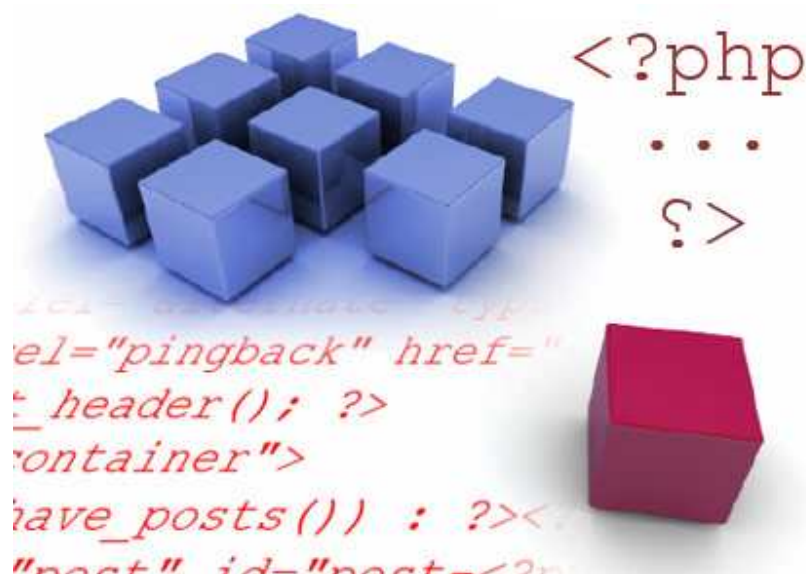
<http://www.p30help.ir/how-make-wordpress-theme-lesson-6>

چگونه برای وردپرس پوسته بسازیم؟ (بخش ۷)

نوشته شده توسط [مهدی رادی](#) در تاریخ ۱۶ / ۱۱ / ۱۳۸۷ در دسته ی [HTML](#), [PHP](#), [وبمستر](#), [وردپرس](#), [پوسته](#)

مطالب ورودی (Content)

نزدیک به یه ماه است که سری آموزشی ساخت پوسته ی وردپرس را آغاز کرده ایم و در این جلسه قصد داریم تا جلسه ی هفتم از این سری را به شما آموزش دهیم. در این جلسه علاوه بر اینکه باید جمله ی تکراری ام را تکرار کنم (همان جمله ی معروفی که از جلسه ی 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 جلسه ی قبل یکی از جلسات بسیار مهم این سری آموزشی بود، بنابراین اگر جلسه ی 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 جلسه ی پیش عنوان هر یک از پست های وبلاگ را به داخل صفحه آوردیم، در این جلسه می خواهیم که محتوا (Content) و یا همان مطالب اصلی پست را نیز به داخل صفحه بیاوریم. در این جلسه همچنین مقدار بیشتری تگ باکس مخفی (div) برای مجزا کردن محتوای هر پست از عنوان آن (پست)، بکار خواهیم برد. پس برای فراگیری مطالب امروز با ما همراه باشید...

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

امروز هم، همان کارهای اولیه را انجام می دهیم. برنامه ی Xampp را اجرا کرده، فولدر **tutorial** از بخش پوسته های وردپرس را گشوده، یک صفحه از مرورگر وب را باز کرده و آدرس <http://localhost/wordpress/> را در آن نوشته و در آخر هم فایل index.php را باز می کنیم.

● مرحله ی اول

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

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

تغییرات 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



Lorem ipsum dolor sit amet, consectetur adipiscing
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 entries`) یا همان `centent` را صدا می زند. حتماً متوجه شده اید که مطالب ورودی شما، تمام عرض پنجره ی مرورگر را گرفته اند. این بدین خاطر است که شما هنوز استایلی برای آن تعریف نکرده اید. بیاد داشته باشید که فایل `style.css` نیز وجود دارد که هنوز شما با آن کار نکرده اید، بعدها که با استایل بیشتر آشنا شدید، خواهید آموخت که چگونه به باکسهای مخفی و سایر قسمت های صفحه، ویژگی بدهید.

در جلسه ی قبل ، من از شما خواستم که یکسری مطالب ساختگی را برای آزمایش به وردپرس اضافه کنید تا چندین پست داشته باشید. اگر شما هنوز اینکار را انجام نداده اید. همین الان اینکار را انجام دهید. سپس بازگردید به صفحه ی اصلی که در اینصورت با چندین پست، **صفحه ی اصلی** شما اینچنین به نظر می آید.

Hello World



Lorem ipsum dolor sit amet, consectetur adipiscing 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. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.



Lorem ipsum dolor sit amet, consectetur adipiscing 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. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Locum ipsum dolor sit amet, consectetur adipiscing 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. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Lorem ipsum dolor sit amet, consectetur adipiscing 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. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

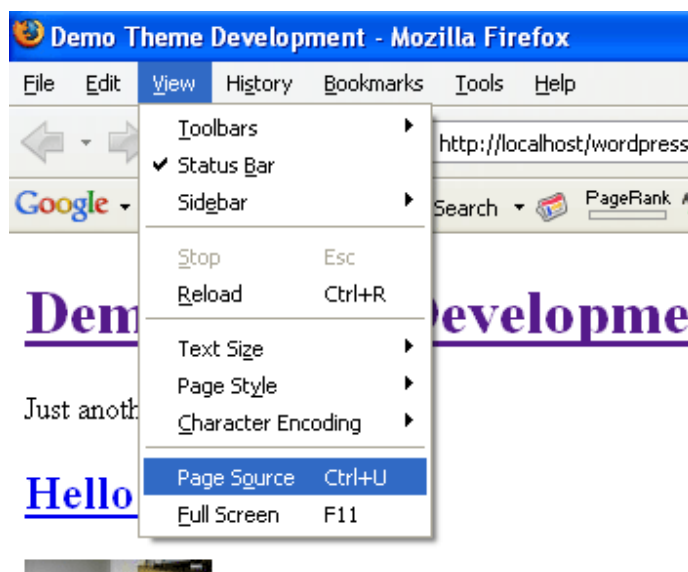
Hello World

Lorem ipsum dolor sit amet, consectetur adipiscing 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. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Lorem ipsum dolor sit amet, consectetur adipiscing 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. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

ادامه دهید...

برگردید به مرورگرتان، بروی تب **View** کلیک کنید و گزینه ی **Source Page** یا **Source** را انتخاب نمایید. یک پنجره از کدها برایتان باز خواهد شد. اگر شما از IE استفاده کنید یک صفحه ی Notepad 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 exerc:
    velit esse cillum dolore eu fugiat i
    laborum. <img src='http://localhost,
    <p>

```

توجه کنید که بین محتوای فایل `index.php` و کدهای پنجره ی `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 ی تابع `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 بسته ی 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 ی کدهای `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 ی دکمه ی `Enter` به خط بعدی می روید، یک پاراگراف را تشکیل می دهید. شما نیازمند این هستید که یک راه برای نشان دادن آن پیدا کنید. هر پاراگرافی داخل **تگ های 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 نوشته ی شما اضافه شود.

«**چطور** – اما اگر برایتان این سوال پیش آمده که چطور این تگها به دور نوشته های من قرار گرفته اند، جواب خیلی ساده است، قالب های سیستم وردپرس بطور خودکار تگ های P را برای شما می سازند.

● مرحله ی دوم:

یک باکس مخفی (div) بدور عبارت the_content() قرار دهید و به آن class="entry" 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 صفحه ی مرورگر را refresh کنید. حال اگر شما به منوی view بروید و سپس گزینه ی 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 آرای) بوسیله ی 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 بیاد بیاورید! **بنابراین چه تفاوتی بین این دو وجود دارد؟** تفاوت این است، **id منحصر به فرد** است ولی `class` اینگونه نیست. اگر شما نگاهی ببندازید به کدهای پنجره ی 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 پست های ورودی که از class کمک گرفتیم.

● مرحله ی 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: نام های انتخاب شده برای class ها و id ها می تواند هر چیزی باشد. شما می توانید از این پس، نام غذاهای مورد علاقه تان، بازیگران مورد علاقه تان 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_
        <div class="post">
            <h2><a href="<?php the_permalink(
                <div class="entry">
                    <?php the_content(); ?>
                </div>
            </div>
        </div>
    <?php endwhile; ?>
```

و عکس زیر ، یک نسخه از حالت سازماندهی شده ی 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 ی **tab** بجای دکمه ی **spacebar** استفاده می کنم.

چرا سازماندهی؟ بر خلاف عکس بالا، کدهای شما با هایلایت قرمز و سبز مشخص نیستند. شما نیازمندید تا کدها و وظیفه ی آنها را بیاد بیاورید. بوسیله ی این نوع برجسته کردن (فاصله های 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 مرورگرتان را refresh کنید تا تمامی تغییرات ایجاد شده در پنجره ی 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



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



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

Lorem ipsum dolor sit amet,
 nostrud exercitation ullamc
 eu fugiat nulla pariatur. E

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

Post
 number
 1

r adipisic
 e dolor ir

Hello World

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

Post
 number
 2

sed do
 enderit 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 جلسه ی بعد به شما نحوه ی درگیر شدن با تاریخ ، دسته ها و بخش نظرات را به شما خواهیم آموخت.

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

منبع:

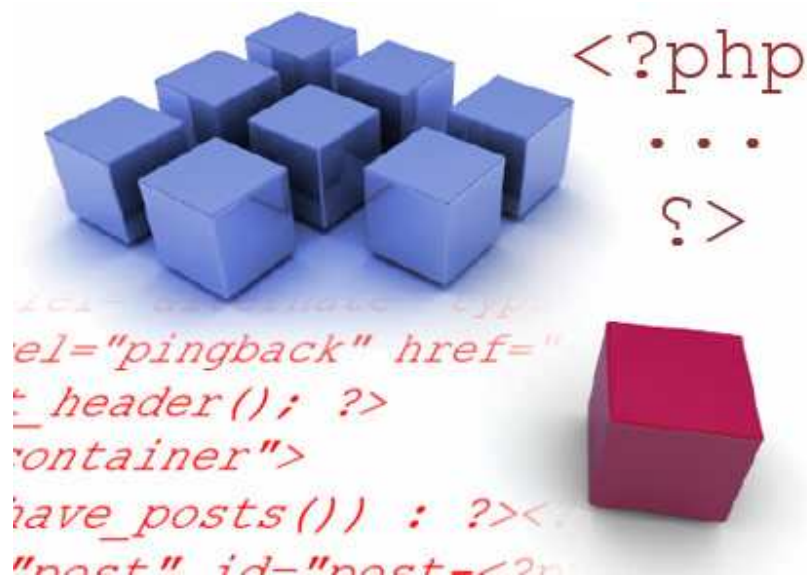
<http://www.p30help.ir/how-make-wordpress-theme-6b-lesson-7>

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

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

در ابتدای عرایض باید این نکته را بگویم که هدف من از کند پیش رفتن و در حقیقت یواش راه رفتن در این مسیر، این بود که هضم مطالب برای شما بزرگواران، آسانتر شود. مطمئناً یک شبه نمی توان راه صد ساله را پیمود. اما شکوه و گلایه شما عزیزان از کند راه رفتن و عقب ماندن از سایر قافله ها، باعث شد که بخواست شما مطالب را دو تا دوتا و یا همان جفت پا جفت پای خودمان بنویسم. امیدوارم که سریع راه رفتن من باعث خستگی شما نشود....

این بار دو جلسه را بصورت یکجا به شما خواهیم گفت. امیدوارم که با اینکار بتوانم رضایت شما عزیزان را جلب کرده باشم. قبل از اینکه درس را به صورت جدی شروع کنم، باز و باز و باز هم این مسئله ی تکراری را می گویم که از ابتدا شروع کنید.....
(البته انصافاً این 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 جلسه ی بعد یا همان بخش دوم که در ادامه ی همین درس خواهیم داشت، به شما Else, Post ID, Link Title را معرفی خواهیم کرد. درس را با نیرویی مضاعف شروع می کنیم....

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

نکته: برای بزرگ دین عکس ها بروی آنها کلیک کنید.

درس هشتم – Postmetadata:

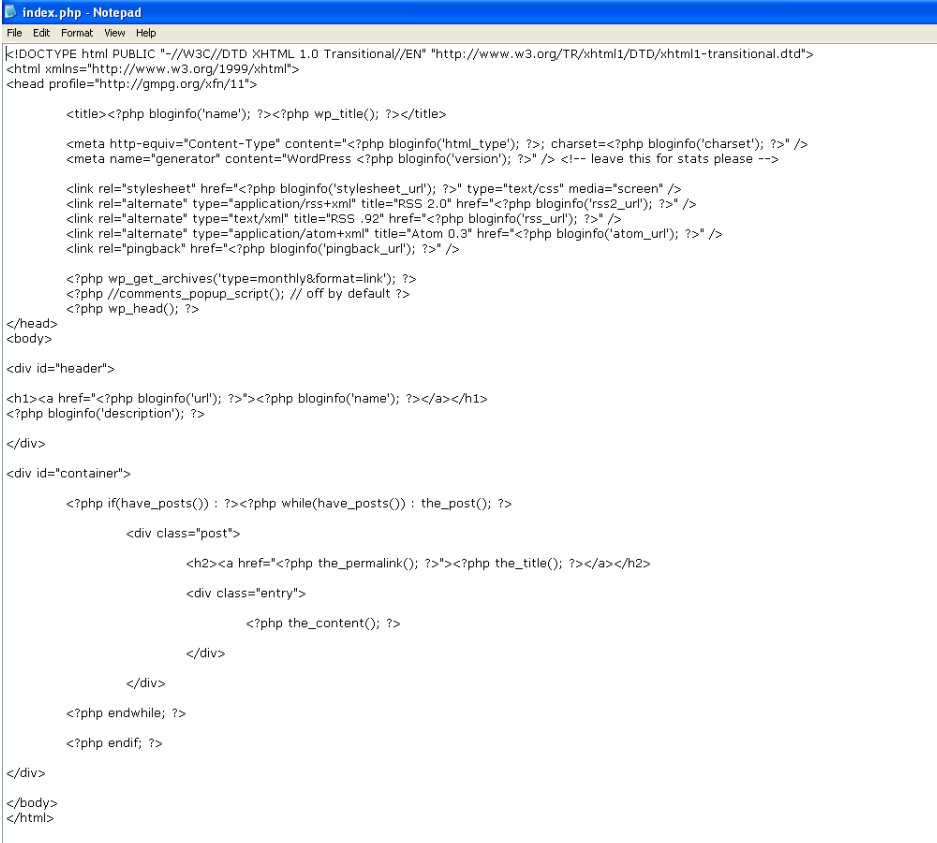
♦ postmetadata چیست؟

سوال بسیار خوبی پرسیدم، در حقیقت postmetadata شامل: تاریخ نوشته، دسته ها، نویسنده، تعداد نظرات و... می شود. 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 برنامه ی Xampp را اجرا کنید، فولدر پوسته های 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>

```

● مرحله ی اول:

کدهای موجود در فایل [postmetadata.txt](#) را در زیر کد `<?php the_content(); ?>` کپی و پیست کنید (توجه: این بار شما می توانید 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 می دهند، کافی است)

عکس زیر بخشی از کدهای اضافه شده را نشان می دهد. سعی کنید که کدهایتان را سازماندهی کنید (بوسیله ی 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 &#
    </p>

</div>
```

حال تغییرات را ثبت کرده و صفحه را refresh کنید.

Lorem ipsum dolor sit amet, consectetur adipisicing elit, s
nostrud exercitation ullamco laboris nisi ut aliquip ex ea
eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod temp
ea commodo consequat. [\(more...\)](#)

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

شما همچنین می توانید به کدهای پنجره ی source از مرورگرتان نگاهی بیاندازید تا کدهای postmetadata را مشاهده کنید و متوجه شوید که اینگونه کدها به چه شکلی ظاهر می شوند. بیاد بیاورد که من به شما نشان دادم که چگونه پنجره ی 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:

«> <p class="postmetadata"> و </p> - تمامی اطلاعات postmetadata بین تگ های پاراگراف (P) با مقدار class="postmetadata" قرار می گیرند، زیرا من می خواهم 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 : کدی هست که عبارت دو نقطه (:) را صدا می زند و همچنین کد <?php _e(' ');> در **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 روبرو خواهیم شد. در جلسات پایانی که نحوه ی فارسی سازی پوسته را به شما خواهیم گفت، کاربرد کد **e(' ')** را بهتر متوجه خواهید شد.

«<?php the_category(' ');> - عبارت **the_category()** یک تابع از php می باشد و دسته ها (category) را صدا می زند. دسته هایی که شما مطلب تان را در آنها قرار داده اید. حال که شما **Filed under:** و **the_category()** را با یکدیگر قرار داده اید، خواهید داشت **Filed under: Name of category 1, Name of category 2** . علامت کامایی که (،) بین نقل قول **the_category(' ');** قرار دارد، برای جدا کردن دسته ها بوسیله ی کاما(،) گذاشته شده است، که می توانید آنرا با هر چیز دیگری عوض کنید (کاما در حقیقت بین هر دسته قرار می گیرد تا دسته ها 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 بگویم 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 گویم، 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 استفاه ی شخصی می سازید، می توانید عبارت **e()** را بدور کلمه هایتان قرار ندهید. اما اگر پوسته ای می سازید که برای عموم است، قرار دادن این عبارت بسیار اهمیت دارد. بطور کلی بگویم که هر پوسته ای 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()** 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.

«>?php comments_popup_link('No Comments »', '% Comments »');<» Comment ۱ -

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 ۱ »` 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 >۸** . علامت درصد (%) بجای تعداد نظرات و عبارت `»` نیز کد شکل « 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 Post را بجای کلمه ی Edit بنویسید آنگاه به شما کلمه Edit Post در عوض کلمه ی Edit نمایش داده خواهد شد. دومین گروه از علامت نقل قول ها تنها برای آن می آید که عبارتی را **قبل** از کلمه ی Edit قرار دهد. در اینجا یک خط عمود (|) که کد آن هم **&124** است نمایش داده خواهد شد. سومین گروه از علامت های نقل قول برای این است که شکلی را **بعد** از کلمه ی Edit قرار دهد و چون درون آن هیچ عبارتی قرار داده نشده است، هیچ شکلی نیز مشاهده نخواهد شد. کاملاً می فهمم که در این مورد کمی گیج شده اید، می توانم بیشتر توضیح بدهم 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 وبلاگ وارد بخش مدیریت شوید سپس به صفحه ی 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 عکسهای زیر توجه کنید:



بعلت تنوع این بخش در postmetadata من قرار دادن تاریخ را به جلسات بعد موكول می كنم و چند روش را برای قرار دادن تاریخ به شما خواهم گفت.

خوب درس این جلسه ی ما به پایان رسید، اما فراموش نکنید که جلسه ی بعد را نیز در ادامه این جلسه بخاطر 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 پوسته ای خواهید یافت، زیرا شما نمی دانید، اشخاص برای چه اهدافی از پوسته ی شما استفاده می کنند. این بار به سرعت به سراغ مراحل می رویم و در طول آموزش، توضیحات لازم را می دهیم.

● مرحله ی 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 نباید 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 حلقه ی **while(have_posts())** دستور **the_post()** را برای صدا زدن پست های شما اجرا می کند. عبارت **else** برای زمانی است که هیچ پستی وجود نداشته باشد. اگر کمی با برنامه نویسی php آشنا باشید، براحتی منظور 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 یا **if** اجرا می شود و یا **else** و البته عبارت شرطی **if** به **else**

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 ی اول](#) را بیاد بیاورید در مورد قانون تو در تویی 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 مجموعه ی else 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 **Not 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 پیغام ها و کدها برای عبارت **Not Found** با تگ های `<h2>` و `</h2>` احاطه شده است. البته اینکار آنچنان واجب نیست. اینکار را برای این انجام داده ایم که نوشته ی ما بزرگتر به نظر برسد و بیننده راحت تر 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 تگ های `<div class="post">` و `</div>` برای چه هستند؟

خوب مشخص است، جوابش را شما بگویید. بعد از این همه جلسه دیگر نگفتنش مایه آبروریزیست.

● مرحله ی دوم:

مقدار `"id="post-<?php the_id(); ?>"` را به `<div class="post">` 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 Source را 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 سایتهای 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.

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

div 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 `the_ID()` یک تابع php است و این دو با یکدیگر متفاوتند).

● مرحله سوم :

عبارت `<?php the_title(); ?>` را به لینک عنوان پست ها اضافه کنید.

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

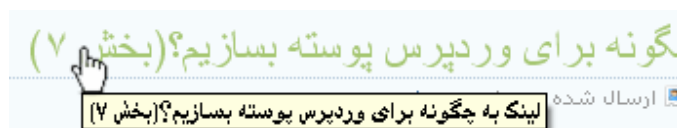
عکس بالا یک قطعه از اسکرین شات است تا به شما نشان دهد که کد را کجا باید قرار دهید. تغییرات را save کرده و مرورگرتان را refresh کنید. به view source دوباره بازگردید. وقتی که پنجره ی Source را مشاهده کردید، `post title link` را جستجو

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

توضیح: `title=""` یکی دیگر از مشخصه های xhtml برای تگ لینک (`<a>`) می باشد. هر چیزی 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()` بعنوان یک مقدار برای `title=""` استفاده کنید، آنگاه می توانید مقدار دیگری که `title=""` بگذارید، تا بعنوان توصیفات برای لینک عنوان مطلب شما محسوب شود. برای مثال، اگر در عوض `the_title()` ، شما از `title="Click me"` استفاده کنید، توصیف برای عنوان هر پست Click me خواهد بود.

حالا ، به صفحه ی وب بازگردید، کرسر موس را بروی عنوان هر پست قرار دهید، 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 رامطلع 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. کلاً بدانید که چیز خوبی است....

در جلسه ی بعد به شما یاد خواهیم داد که چگونه برای قالبتان امکان **صفحه ی 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:

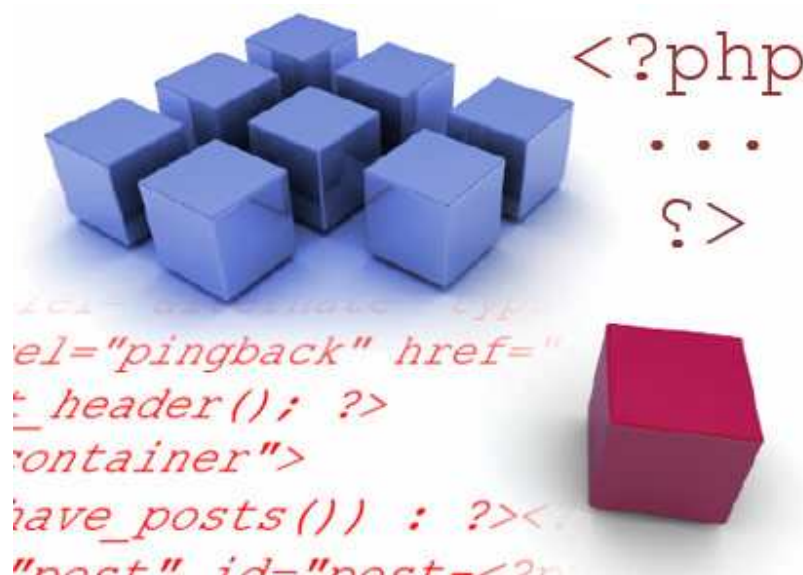
<http://www.p30help.ir/how-make-wordpress-theme-6b-and-6c-lesson-8-9>

چگونه برای وردپرس پوسته بسازیم؟ (بخش ۱۰)

نوشته شده توسط [مهدی رادی](#) در تاریخ ۱۳۸۷ / ۱۲ / ۰۲ در دسته ی [HTML](#), [PHP](#), [وبمستر](#), [وردپرس](#), [پوسته](#)

لینک 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 هفته ی ما در رابطه با قرار دادن لینکهای **صفحه ی بعد (Next Page)** و **صفحه ی قبل (Previous Page)** در انتهای صفحه است. این لینکها که حتماً با آنها آشنایی دارید وظیفه ی نشان دادن پستها و یا مطالب قبلی و بعدی را به شما دارند. راستی اگر تا بحال با این سری آموزشی همراه نبوده اید، از [اینجا](#)، خواندن این سری آموزشی را پی بگیرید.



آموزش این هفته ی ما بسیار کوتاه است و فقط شامل یک مرحله می شود.
در ابتدا همان کارهای همیشگی را انجام دهید: برنامه ی Xampp و فولدر پوسته های وردپرس را بگشاییید. مرورگرتان را فعال کرده، و فایل **php.index** را باز کنید.

● مرحله اول:

کد زیر را بین کدهای `<?php endwhile; ?>` و `<?php else : ?>` 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 بوسیله ی دکمه Tab خطها را کمی جلو و عقب کرده ام.

♦ توضیح آنچه انجام شده است:

« `<div class="navigation">` - این تگ باکس مخفی ای را با نام navigation ایجاد می کند. توجه کنید که ما این تگ را به دور لینک های `posts_nav_link()` قرار داده ام تا بتوانیم، بعدها به این باکس مخفی ویژگی بدهیم.

« `<?php` - شروع php

« `posts_nav_link()` - لینک های بعدی (Next) و قبلی (Previous) را صدا می زند.

« ; - به صدا زدن آنها را پایان می دهد.

« `>?>` - تمام شدن php

« `</div>` - بستن باکس مخفی که نام آن navigation هست.

باید الان شکل پوسته ی شما به صورت زیر باشد:

Filed under: [Uncategorized](#), [Personal](#), [Sub Category](#)

[1 Comment »](#)

[Next Page »](#)



تغییرات را ثبت و سپس مرورگرتان را refresh کنید. حال باید لینک های Next یا 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 لینکهای 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 مجموعه ی تب **خواندن**، تعداد پست هایتان را از ۱۰ عدد به تعداد دلخواه در بیاورید، تا بتوانید لینک های Next یا Previous را مشاهده کنید. برای مثال از ۱۰ 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 لینکهای Next و Previous قرار دهید. بوسیله ی 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، وظیفه ی 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:



◀◀ Previous Page

بله درس امروز به همین کوتاهی بود. امیدوارم که حداقل به اندازه ی 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:

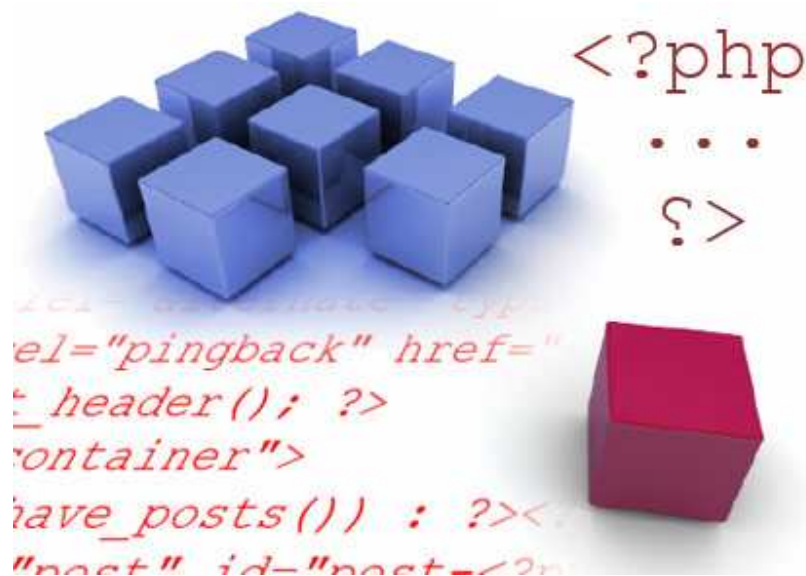
<http://www.p30help.ir/how-make-wordpress-theme-6e-lesson-10>

چگونه برای وردپرس پوسته بسازیم؟ (بخش ۱۱)

نوشته شده توسط [مهدی رادی](#) در تاریخ ۱۳۸۷ / ۱۲ / ۱۹

درس یازدهم : سایدبار (Sidebar)

مطلب امروز خود را با یه عذرخواهی از شما به علت تاخیر در آماده شدن این مطلب آغاز می کنم. در این درس به شما نحوه ی 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 می‌رسد ولی این اشتباهی بیش نیست، 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 بخش‌های بعدی ساختار سایدبار را تشکیل خواهید داد و در درسهای بعدی کد و استایل آن را خواهید نوشت.

خوب درس را شروع می‌کنیم، ادامه درس در ادامه مطلب...

قبل از شروع مبحث، شما باید فایلی مانند [index.php](#) داشته باشید (بروی index.php 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 بدور هرچیزی که جزو sidebar حساب می‌شود، 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.

« – کد باز کننده ی لیست unordered

« – کد بستن لیست 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 ی Tab قبل از تگهای و برای سازماندهی بهتر، اضافه شده است.

« – لیست آیتم را باز می کند

« <h2> – تگ sub-heading را باز می کند

« <?php _e('Categories'); ?> – کلمه ی Categories را نشان می دهد

« </h2> – تگ sub-heading را می بندد

« – لیست آیتم را می بندد

توجه: لیست آیتم یا همان تگ li مخفف کلمه ی list item می باشد.

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

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

— [Next Page »](#)



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

● مرحله چهارم :

کد زیر را 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.

« **<?php wp_list_cats(); ?>** – کدی از php است که وظیفه ی آن صدا زدن دسته های 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)



دسته ی پیش فرض شما **Uncategorized** هست که البته وردپرس فارسی آن را **دسته بندی نشده** معنی کرده است. اگر مطالبتان را در چندین دسته ی مختلف منتشر نکرده اید، شما در بخش دسته های وبلاگ باید تنها یک لینک به نام **دسته بندی نشده** داشته باشید.

♦ توضیحات بیشتر:

« **sort_column=name** – لیست دسته ها را بر اساس الفبا مرتب می کند.

« **optioncount=1** – تعداد هر پست را در زیر مجموعه ی هر دسته نشان می دهد.

البته اگر مقدار آن را از یک به صفر تغییر دهید دیگر تعداد مطالب درون هر دسته را نخواهید دید.

« **hierarchial=0** – دسته ها را داخل زیر مجموعه ی sub-list-items نمی چرخاند.

« **&** – هرگاه بخواهید یک مشخصه ی دیگر را اضافه کنید، شما باید & را برای مجزا کردن آن از مشخصه های دیگر تایپ کنید. برای مثال شما & را آوردید ، بین sort_column و optioncount تا این دو را از یکدیگر مجزا کنید.

♦ چرا شما تگ های و را بدور <?php wp_list_cats();> 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 ی لینکها اضافه می کند. برای اینکه بیشتر متوجه شوید به تب view رفته و سپس Page Source یا Source را انتخاب کنید. بعد از اینکه پنجره ی Pop up باز شد، 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 unordered list و list items بسیار مهم هستند، بنابراین بیاد بیاورید قانون اول و دوم ما در کد نویسی:

قانون اول: هرچیزی را باز کرده اید را در آخر ببندید.

قانون دوم : تگها نباید 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.....

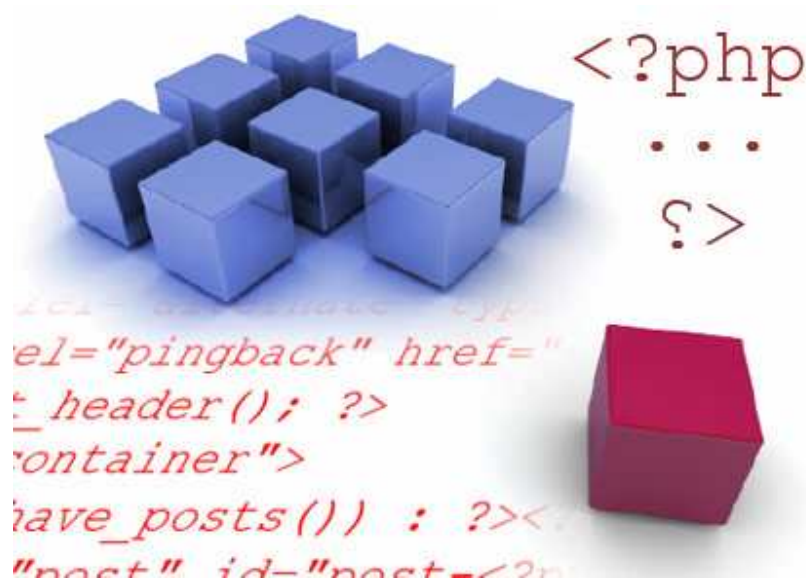
منبع:

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

چگونه برای وردپرس پوسته بسازیم؟ (بخش ۱۲)

نوشته شده توسط مهدی 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 جلسه ی قبل 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 می کنیم؟ 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>
```

..

- ◊ About

اول از همه، هرچیزی داخل لیست آیتم (li) قرار گرفته است. **دوم**، کلمه ی 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 تگ های و قرار گرفته است.

در اسکرین شات بالا، توجه کنید که سایز کلمه ی برگه ها (Pages) در مقایسه با عنوان دسته ها (Categories) کوچکتر است.

♦ چطور سایز برگه ها و دسته ها را متناسب کنیم؟
برای اینکار کد `'title_li=<h2>Pages</h2>` را به `wp_list_pages()` اضافه می کنیم.

```
<?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.....iiiii](#)
 - [Grandchild 1](#)
 - [Great Grandchild](#)
 - [Grandchild 2 iiii.....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 وبلاگ

بخش دوم آموزش امروز مربوط به آوردن بلاگرویل و آرشیو وبلاگ 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&option=sort_order=asc'); ?>
</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 پی اچ پی `wp_get_archives()` همراه با مشخصه ی `monthly` استفاده کرده اید، تا آرشیو وبلاگتان را بصورت ماهیانه صدا بزنید.

« `` – باز کردن لیست آیتم

« `<h2>` – باز کردن sub-heading

« `<?php _e('Archives'); ?>` – کلمه ی Archive را نشان خواهد داد

« `</h2>` – بستن sub-heading

« `` – بازکردن unordered list زیر sub-heading و داخل لیست آیتم ها

« `<?php wp_get_archives('type=monthly'); ?>` – لینک آرشیو وبلاگتان را

بصورت ماهیانه صدا می زند، هر لینکی داخل تگ `` و `` قرار می گیرد. اگر شما Source Page یا Source صفحه تان را چک کنید، خواهید دید که برای هر لینک تابع `wp_get_archives()` ، تگ لیست آیتم (`li`) بطور خودکار بدور آن قرار گرفته است.

بسیار شبیه به تابع `wp_list_cats()`

« `` – بستن unordered list که در زیر sub-heading قرار گرفته است

« `` – بستن لیست آیتم ها

● مرحله ی دوم:

در این مرحله قصد داریم تا لینک **بلاگ رول** 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 ، بلاگroll من تفاوتی با مال شما ندارد، برای همین سری به کدهای Source 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 تنهایی 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

● مرحله ی 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 فولدري 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(TEMPLATEPATH . '/searchform.php'); ?>
</li>
```


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

• Pages

- ## ❖ چه اتفاقی افتاده است؟

«<li id="search"> – 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.

«include() – بوسیله ی 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 فولدر پوسته ی شما قرار دارد را آدرس می دهد، یعنی wp-content/themes/tutorial

«'/searchform.php' – مکان و نام فایل /searchform.php

شما بوسیله ی آوردن آنها در کنار هم، بین TEMPLATEPATH و 'wp-content/themes/tutorial/searchform.php' ارتباط برقرار می کند، و در انتها دارید:

« – لیست آیتم را می بندد

● مرحله ی دوم:

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

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

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

```
<li id="search">
    <?php include(TEMPLATEPATH . '/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؟

« `<li id="calendar">` – لیست آیتم ها را با id بنام Calendar باز می کند
« `<h2>` – تگ sub-heading را شروع می کند
« `<?php _e('Calendar'); ?>` – کلمه ی Calendar را نمایش می دهد
« `</h2>` – تگ sub-heading را می بندد
« `get_calendar()` – تقویم را با استفاده از تابع `get_calendar()` صدا می زند
« `` – 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 وبلاگ را که شامل عضویت در وبلاگ، ورود به وبلاگ، خروج از وبلاگ، 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 مرورگرتان 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 یا همان H2 برای Meta شروع کردید. پایین sub-heading ، شما یک unordered list یا همان تگ ul وارد کرده اید. و برای هر لینکی، شما تگ لیست آیتم (li) را بدور آنها قرار دادید.

تابع `wp_register()` خودش به تنهایی 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 بایستی تگهای `` 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()` هیچ 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 :

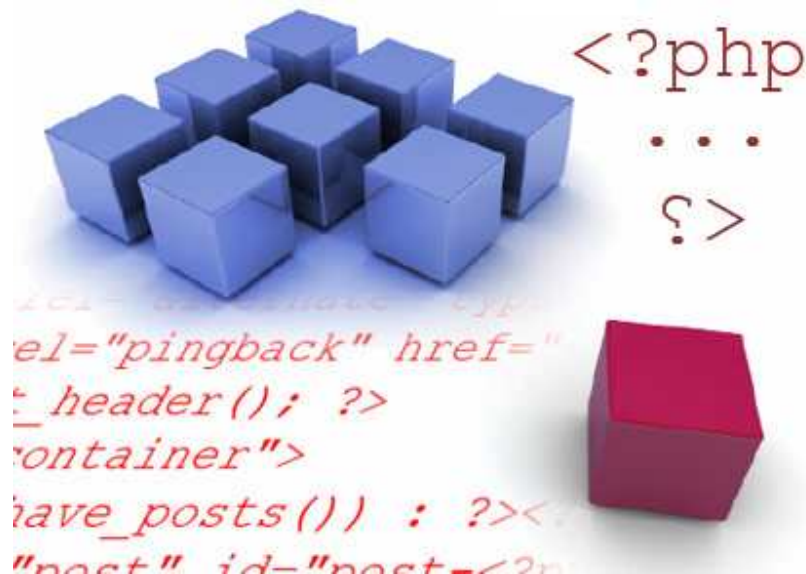
<http://www.p30help.ir/how-make-wordpress-theme-lesson-12>

چگونه برای وردپرس پوسته بسازیم؟ (بخش ۱۳)

نوشته شده توسط [مهدی رادی](#) در تاریخ ۲۰ / ۰۱ / ۱۳۸۸

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

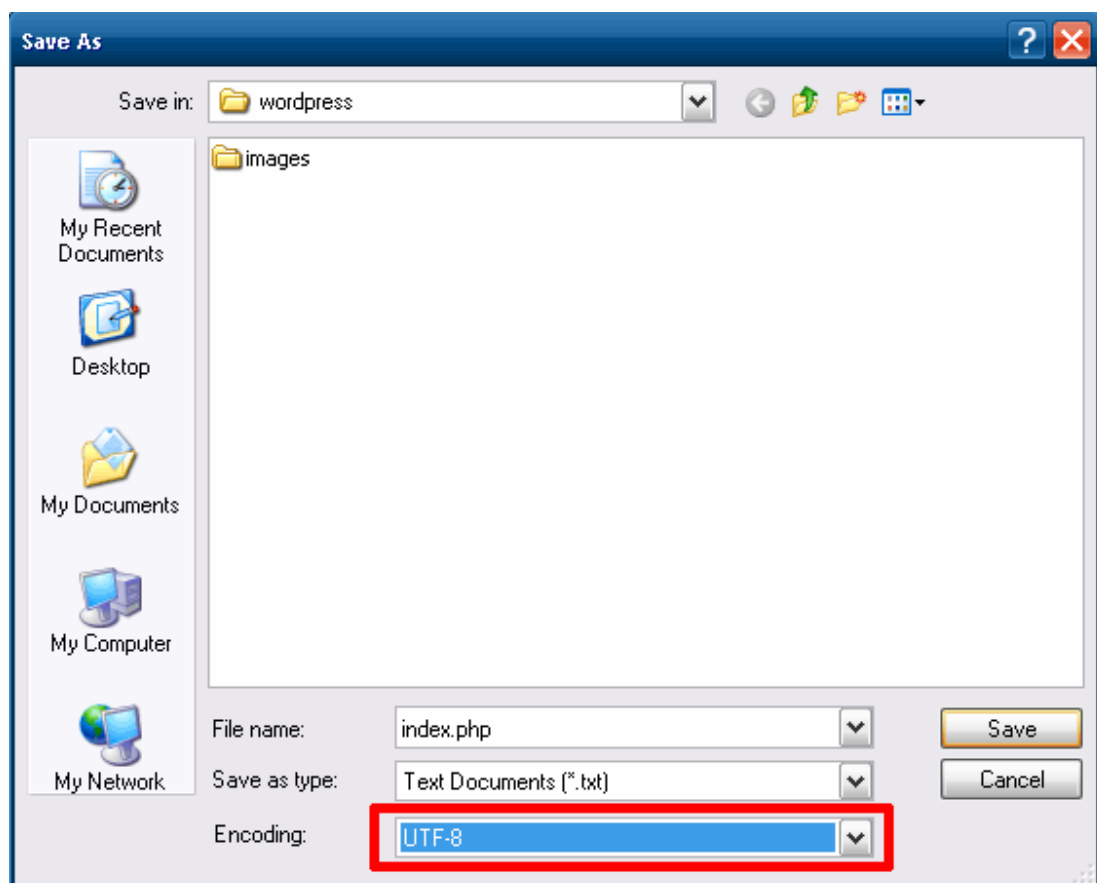
دوستان را همراه با جواب بطور پست های جداگانه مطرح کنم تا از اشکالات شما دوستان قدری بکاهم.



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. نکته ی کوچک ولی مهمی که من فراموش کرده بودم به اطلاع شما دوستان برسانم، این است که مدل **Encoding** فایل های 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 بحال 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 غصه ای نخورید چون اینکار در هر مرحله ای راحتی می تواند انجام شود و مشکلاتی که برای شما بوجود خواهد آمد راحتی قابل تعمیر است، البته اگر همین الان اینکار را انجام دهید، هیچ مشکلی را برای شما بوجود نخواهد آمد چون هیچ متن فارسی درون فایلهایتان قرار ندارد.

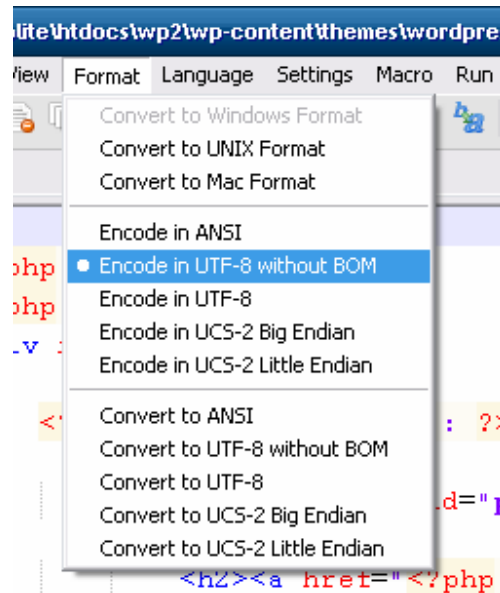
خوب برای اینکه کار را برای شما آسان کنم نحوه ی اینکار را به شما در Notepad نشان می دهم. هر یک از فایل های مورد نظرتان را در Notepad باز کنید. به منوی فایل رفته و بروی Save As کلیک کنید. در صفحه ای که باز شد. در بخش Encoding گزینه ی UTF-8 را انتخاب کنید دقیقاً مانند عکس زیر:



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

اما اگر اینکار برایتان کمی سخت بود، یک راه بهتر و آسانتر نیز برایتان سراغ دارم. کار کردن با Notepad ساده ترین راهی است که می توان استفاده کرد، من خودم به شخصه از برنامه ی [Notepad++](#) استفاده می کنم. به شما هم پیشنهاد استفاده از این برنامه را می دهم. حال اگر از این برنامه استفاده می کنید یا خواهید کرد، برای تبدیل Encoding پاراگراف زیر را مطالعه کنید:

روش کار بسیار ساده است، فایل مورد نظر را باز کنید، به تب **Format** بروید و از آنجا گزینه ی **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 نباشید....

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 می باشند که شما براحتی می توانید آنها را در سایدبار سایت خود به نمایش در آورید. حقیقتاً ابزارک ها کار را برای شما آسان می کنند و به مدیران سایت اجازه می دهند تا بدون داشتن دانش برنامه نویسی وب، بتوانند براحتی عکس ، مطلب ، گجت و ... را درون سایدبار وبلاگشان به نمایش در آورند و یا مکان آنها را جابجا کنند. جالب است بدانید که در ورژن های قبلی وردپرس امکان ابزارک به این نوع وجود نداشته است و برای استفاده از قابلیت ابزارک، باید یک پلاگین خاص را نصب می کردیم، اما وردپرس به سرعت پی برد که این بخش یکی از قسمتهای بسیار لازم برای

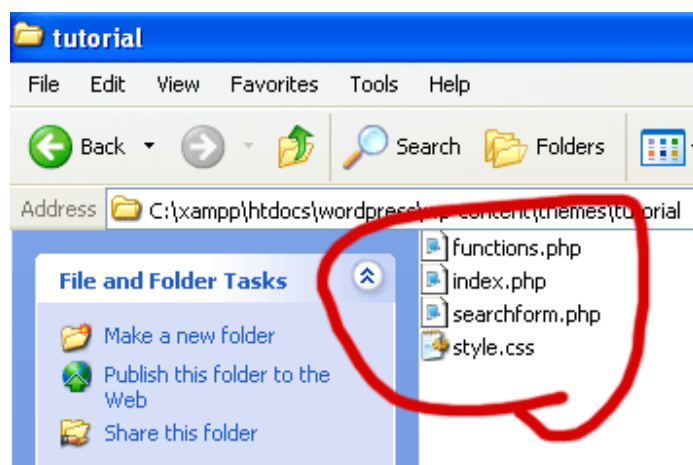
وردپرس می باشد و در بسته های بعدی این قسمت را بطور مستقیم درون وردپرس جای داد.

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

● مرحله ی 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 UTF-8 باشد). هر چیزی که داخل فایل [functions.txt](#) وجود دارد را، به درون فایل functions.php کپی کنید و سپس تغییرات مورد نظر را Save کرده و سپس صفحه ی Notepad را ببندید.

الان شما باید چهار فایل را درون فولدر tutorial داشته باشید.



● مرحله ی دوم: ابزارک 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(TEMPLATEPATH . '/searchform.php'); ?>
    </li>

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

```

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) :

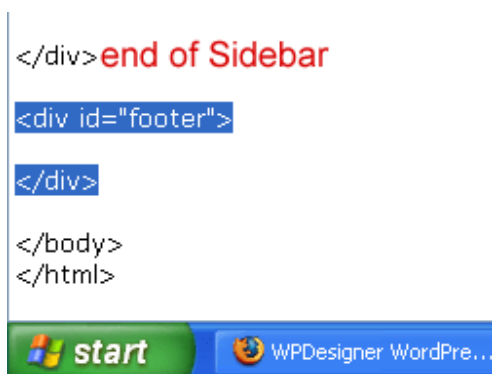
این بخش از درس نیز بسیار ساده است، شما تنها باید یک div یا باکس مخفی زیر بخش div سایدبار اضافه کنید و داخل این تگ را با چیزهای مثل نوشته ی کپی رایت پر کنید. فکر کنم شما اینکار را بدون هیچگونه دستورالعملی نیز می توانید انجام دهید، اینکار را خود به تنهایی انجام دهید و سپس برگردید و ادامه مطلب را بخوانید،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

● مرحله ی اول : اضافه کردن div

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

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

```
</div>
```



• 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 &#169; 2008 <?php bloginfo('name'); ?>
```

```
</p>
```

```
<div id="footer">  
<p>  
Copyright &#169; 2008 <?php bloginfo('name'); ?>  
</p>  
</div>
```

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

• Archives

◊ [October 2006](#)

Copyright © 2008 Demo Theme Development

حروف **©** کدهایی هستند که علامت کپی رایت (©) را به شما نشان می دهند. 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 می زند.

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 ی 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 اصلاً نگران نباشید. چون انصافاً رعایت تمامی این قوانین آنچنان آسان نیست و نداشتن گواهی نیز نقص بزرگی به حساب نمی آید. 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 نا زیبایی که گواهی دارند. این را نیز بدانید که بسیاری از سایتهای بزرگ فعلی نیز فاقد این گواهی نامه می باشند.

Validate یا **Validating** یا **Validation** به سادگی معنی مطمئن شدن درباره نداشتن ایراد را می دهد. من به شما دو لینک [XHTML Validator](#) و [CSS Validator](#) را می دهم، که از این دو لینک قادر به بررسی اعتبار صفحات تان می باشید. لینک [XHTML Validator](#) برای اعتبار سنجی صفحات اصطلاحاً **html** می باشد و لینک [CSS Validator](#) نیز برای اعتبار سنجی کدهای **CSS** می باشد 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 **Page Source** یا **Source** را انتخاب کنید. پنجره ای برای شما باز می شود. شما باید

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

به صفحه ی [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 ی **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 &#169; 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 نتوانسته 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.

Help & FAQ

Feedback

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:

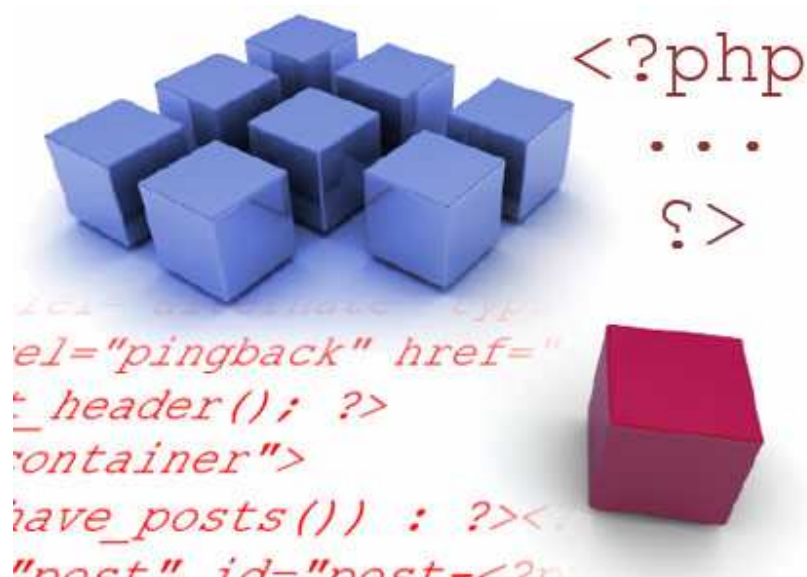
<http://www.p30help.ir/how-make-wordpress-theme-lesson-13>

چگونه برای وردپرس پوسته بسازیم؟ (بخش ۱۴)

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

آشنایی با فایل Style.css و مقدمات CSS

انتظار ها به سر رسید و جلسه ی کار با CSS هم از همین الان شروع شد، می دانم که زمان زیادی بود که در انتظار این لحظه بوده اید. در این جلسه به شما مقدمات CSS و Style.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.



هماهنگی قالب با تمامی مروگر ها کار آسانی نیست، البته باید در اینجا این نکته را عرض کنم که تنها اینترنت اکسپلوررها هستند که تعبیرشان از قوانین CSS خیلی متفاوت است، سایر مروگرها مانند اپرا ، فایرفاکس ، گوگل کروم و... تعبیر بسیار متشابه 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 شخصه ابتدا با مروگر فایرفاکس قالبم را می سازم و در آخر هنگامی که کاملاً ساخته شد، یک CSS جدا برای مروگر 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 همین جا بگویم 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 فایل های 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 ۶ خط فقط برای وردپرس معنی می دهند 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/

*/

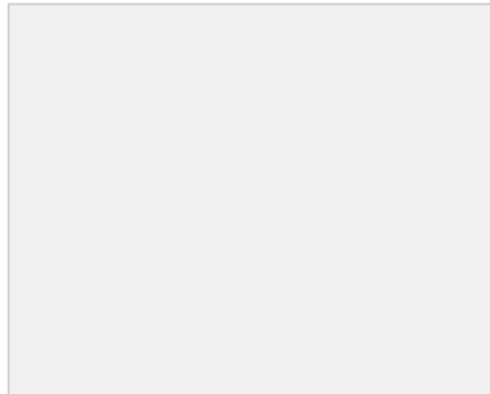
```

- * 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 [Notepad++](#) 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 خیلی طولانی شد).....

● مرحله ی اول :

* کنترل برنامه ی Xampp ، فولدر پوسته ها، یک مرورگر فایرفاکس و فایل Style.css را باز کنید.

* مرورگر فایرفاکس را به آدرس <http://localhost/wordpress> هدایت کنید.

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

اگر از روش اول استفاده می کنید 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 برنامه ی Notepad++ را نصب نکرده اید و قادر هم نیستید که فایل Style.css را در Notepad باز کنید، بروی فایل Styl.css راست کلیک کرده در زیرمجموعه Open with گزینه ی Choose Program ... را انتخاب کنید و در پنجره ی باز شده، Notepad را انتخاب کرده و ok را کلیک کنید.

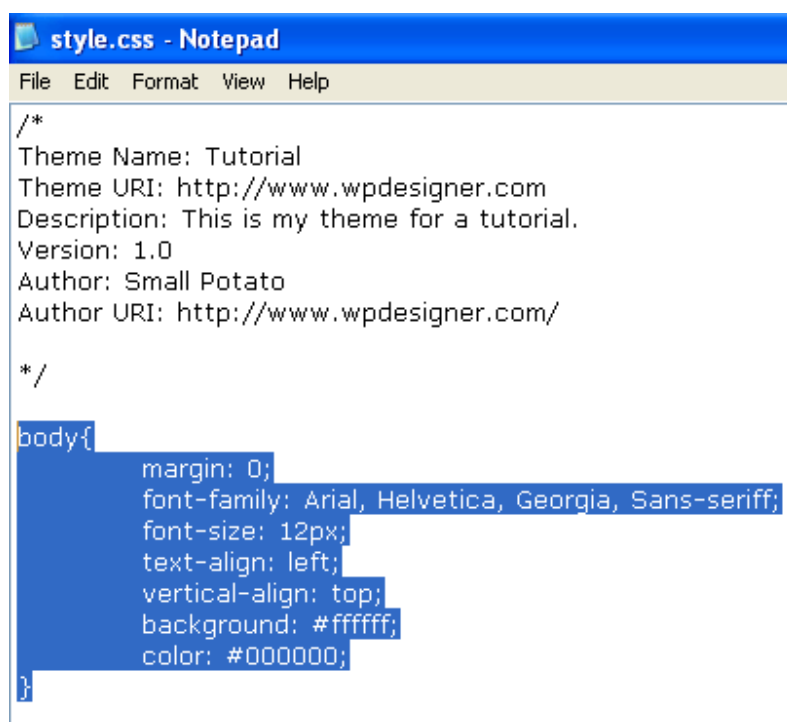
● مرحله ی دوم:

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 جلسه ی 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 گویم....



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 کلمه ی `body{}` (سلکتور CSS) را در CSS مورد استفاده قرار دادیم. دلیل این هست که هر چیزی از صفحه ی وب بین تگهای `<body>` و `</body>` قرار دارد، در نتیجه هنگامی که شما به این تگ استایل بدهید، یعنی به کل صفحه وب استایل داده اید، بنابراین کدهایی که در این استایل قرار دارند بروی کل صفحه ی 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 تترسید، 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 گزینه ها برای چه هستند (این گزینه را خیلی راحت می توانید درک کنید، مشکل اصلی برای فهمیدن **Padding** هست که در جلسات بعد به شما خواهم گفت و کاری شبیه به همین **margin** را انجام می دهد. مهم نیست.... درس امروز را متوجه شوید، کافیت). در حقیقت **margin** یک حاشیه برای شما بوجود می آورد و فضا را به مقداری که شما لازم دارید تخلیه می کند. الان مقدار **margin** برابر با صفر است یعنی هیچگونه حاشیه یا فضای خالی نخواهید داشت ولی اگر می خواهید یک حاشیه داشته باشید، عدد صفر را به `px ۱۰` یا `px ۲۰`، یا غیره... تغییر دهید. برای هر عددی که می نویسید باید واحد آن را نیز بیان کنید. تنها عدد صفر است که اگر واحد آنرا ننویسید مشکلی پیش نمی آید.

نکته ۱: هر پیکسلی یک نقطه روی صفحه ی کامپیوتر است.

نکته ۲: خیال نکنید که این حاشیه ای که در مورد آن صحبت می کنیم قابل دیدن است، این حاشیه یه حاشیه نامرئی می باشد. در حقیقت **حاشیه** در اینجا معنی **فاصله** را می دهد.

در عکس زیر، هایلایت قرمز فضایی هست که بوسیله ی تگ `body` بصورت پیش فرض برای **margin** اعمال شده است. (این رنگ قرمز را خود من ایجاد کرده ایم)

Demo Theme Development

Just another WordPress weblog

[Hello World](#)

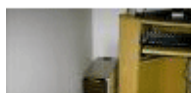
بعد از استایل کردن و قرار دادن مقدار صفر برای **margin** ، همانگونه که در عکس زیر می بینید دیگر فاصله ای وجود ندارد. یعنی نوشته کاملاً به سمت چپ چسبیده است و هیچگونه فاصله ای وجود ندارد. خود شما بهتر است حالت های گوناگون را تست کنید تا بهتر متوجه شوید.



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 فوتتی 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.

« **text-align: left;** - 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.

« **vertical-align: top;** - 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....

« **background: #ffffff;** - این کد یعنی، رنگ پس زمینه برای صفحه وب، که در اینجا ما رنگ را سفید قرار داده ایم. عبارت **#ffffff** 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.

« **color: #000000;** - این کد مربوط به رنگ فونتها یا نوشته ها 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 آزمایشهایی که بروی کدها انجام دادید، تمامی تغییرات را به حالت اول بازگردانید.

منبع:

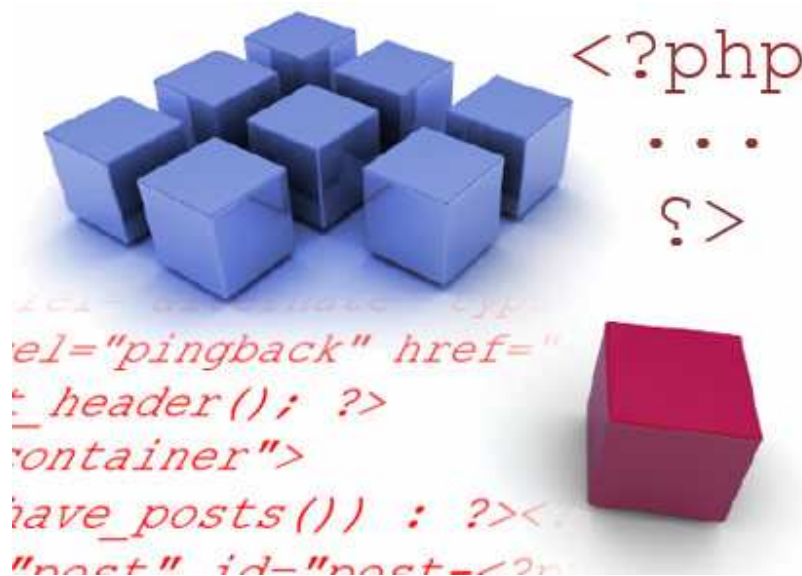
<http://www.p30help.ir/how-make-wordpress-theme-lesson-14>

چگونه برای وردپرس پوسته بسازیم؟ (بخش ۱۵)

نوشته شده توسط [مهدی 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 خواهشاً 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) خواهید 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. مثلاً برای رنگ نوشته ها کد `body { color: #000000;}` این معنی را می دهد که تمامی نوشته ها در بدنه ی سایت، به رنگ سیاه در خواهند آمد.

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

♦ حال توضیح و نحوه ی عملکرد کدهای شانزده تایی (Hexadecimal Codes):

تمامی کدهای Hex شش رقمی هستند، همچنین قبل از این کدها یک علامت # می آید. رنج این کدها از شماره #ffffff که رنگ سفید است تا #000000 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 ی **رنگ آبی** می باشند. پس با این اوصاف کد #ff0000 به شما **رنگ قرمز** را می دهد، همچنین کد #۵۵۰۰۰۰ **رنگ قرمز کدر** را به شما خواهد داد. کد #۲۲۰۰۰۰ نیز قابل پیش بینی است که **رنگ قرمز کدرتری** را به شما می دهد. حال کد #ff00۰۰ **رنگ سبز** را به شما خواهد داد و در آخر کد #ff۰۰۰۰ **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.

● مرحله ی 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 دارنده 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 نوشته را به لینک تبدیل کنید مسلماً از تگهای `<a>` و `` استفاده می کنید. بنابراین این سلکتور تمامی نوشته 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** نوشته 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 وگرنه 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 سلکتورهای { **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 کursor موس را بروی لینکها می برید. و چون ما کد **text-decoration: none;** را برای این سلکتور 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، منتظر درسهای بعدی باشید..... ناراحت نباشید چون مطلب طولانی نشانه ی نوشتن است..... (ضرب المثل گرفته شده از فرهنگستان وبلاگستان)

منبع:

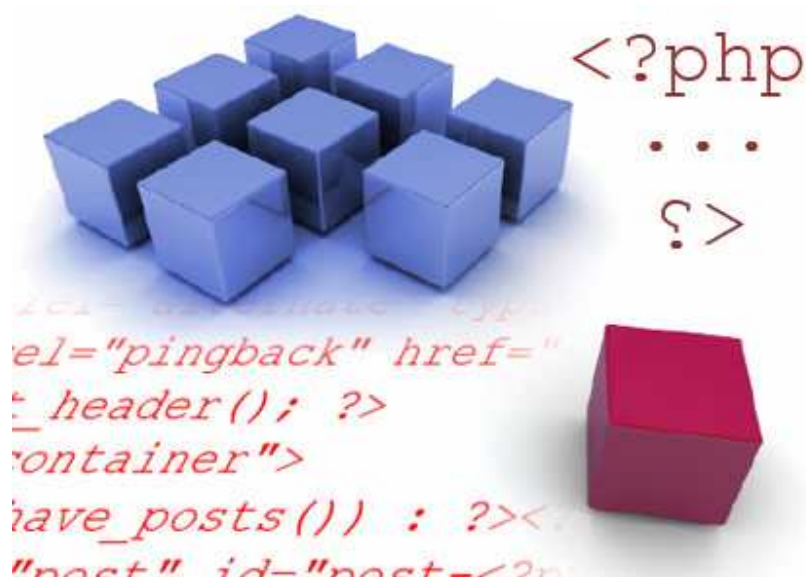
<http://www.p30help.ir/how-make-wordpress-theme-lesson-15>

چگونه برای وردپرس پوسته بسازیم؟ (بخش ۱۶)

نوشته شده توسط مهدی رادی در تاریخ ۲۵ / ۰۲ / ۱۳۸۸

کار با پهنا (Width) و Float

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، نحوه ی 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 فولدر پوسته، مرورگر فایرفاکس، فایل index.php و فایل Style.css را باز کنید.

● مرحله ی اول:

اولین چیزی که بایستی انجام دهید، این است که پهنای اصلی پوسته تان را مشخص کنید. ما در این آموزش می خواهیم که از پهنای ۷۵۰ پیکسل برای پوسته مان استفاده کنیم. این را نیز به خاطر داشته باشید که هر ۱۰۰ پیکسلی یک اینچ است. یکی از دلایلی که ما از پهنای ۹۰۰ پیکسل استفاده نمی کنیم این است که در صورت استفاده از اندازه ی 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, container, sidebar و footer را درون خود قرار دهد. خوب پس برای اینکار مراحل 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، بویژه در style.css از علامت # برای آدرس دهی به یک div که با مشخصه ی id نام دهی شده است، استفاده می کنیم. خوب شاید این سوال برایتان پیش بیاید که چگونه به div های دارای مشخصه ی class آدرس بدهیم. جواب این است که اگر شما عبارتی مانند کد `<div class="wrapper">` داشته باشد باید از **wrapper**. (نقطه) بجای **#wrapper** (مربع wrapper) استفاده کنید. ممکن است که متوجه نشده باشید، اصلاً نگران نباشید در ادامه هنگامی که چند مثال را مشاهده کردید، کاملاً موضوع را درک خواهید کرد.

تغییرات داده شده در هر دو فایل را save کرده و مرورگرتان را refresh کنید تا تغییرات را مشاهده کنید.

♦ توضیحات بیشتر:

« **margin: 0 auto 0 auto;** - معنی این کد این است که به ترتیب، صفر پیکسل حاشیه از بالا، فاصله ی خودکار از راست، صفر پیکسل حاشیه از پایین و حاشیه ی خودکار از چپ بوجود آید.

« **width: 750px;** - کاملاً واضح هست و نیازی به توضیح نیست.

« **text-align: left;** - 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;}** از کد **text-align: center;** استفاده کنید تا فرق این دو را متوجه شوید.

● مرحله ی دوم:

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 چهارم:

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 **نکرده ایم**. در حالی که از یک **نقطه** کمک گرفته ایم. دلیلش این است که در باکس مخفی نام sidebar را با مشخصه ی class تعریف کرده ایم نه id. همچنین بیاد بیاورید که #ffffff رنگ سفید است و رنگ #eeeeee یک رنگ خیلی خاکستری روشن می باشد. در اینجا ما تنها برای اینکه بین sidebar و container تفاوت ایجاد کنیم، به پس زمینه ی سایدبار رنگ خاکستری روشن داده ایم، چون قرار است در ادامه ۱۰ پیکسل باقی مانده را بعنوان فاصله ی بین این دو باکس استفاده کنید.

● مرحله پنجم:

برای باکس مخفی footer مشخصه های زیر را بدهید:

```
#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{}` می باشد. کد مورد بحث برای این اضافه می شود تا مطمئن شویم که فوتر قالب به هیچ جزء دیگری از قالب اضافه نمی شود. برای مثلا فوتر به باکس container و یا sidebar نمی چسبد.

تغییرات را ثبت کرده و مرورگرتان را refresh کنید.

● مرحله ی ششم:

حال در این مرحله ۱۰ پیکسل باقی مانده را به سایدبار بعنوان فاصله یا حاشیه نامرئی اضافه می کنیم. برای اینکار از کد **margin** کمک می گیریم. 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 اندازه ی مورد نظر ما صفر باشد لازم نیست که عبارت px را جلوی آن بنویسیم.

● مرحله ی هفتم (مرحله ی اضافی):

در این مرحله می خواهیم که بجای ۱۰ پیکسل فاصله ای که به بخش چپ سایدبار داده ایم، ۲۰ پیکسل فاصله قرار دهیم. با انجام اینکار خواهید دید که لایه های شما خواهد شکست و فشار آورده شده به سایدبار باعث می شود که باکس سایدبار در زیر باکس container قرار گیرد. دلیلش این است که ۲۰ پیکسل حاشیه باعث می شود که جمع پهنای Container و Sidebar جمعاً ۷۶۰ پیکسل شود در صورتی که ما باکس wrapper تنها ۷۵۰ پیکسل پهنا داده ایم. البته ممکن است این خطا در فایرفاکس نشان داده نشود و تنها در اینترنت اکسپلورر اینگونه نشان داده شود، چون باگ های فایرفاکس بسیار کمتر از اینترنت اکسپلورر هستند. در هر حال توجه کنید که حتماً اندازه هایی که به اجزای صفحه می دهید متناسب با سایر اجزا باشد.

برای برطرف کردن این باگ، عبارت **display: inline;** را به سلکتور sidebar 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 فایل های نهایی [index](#) و [style](#) را می توانید دانلود کنید.

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

منبع:

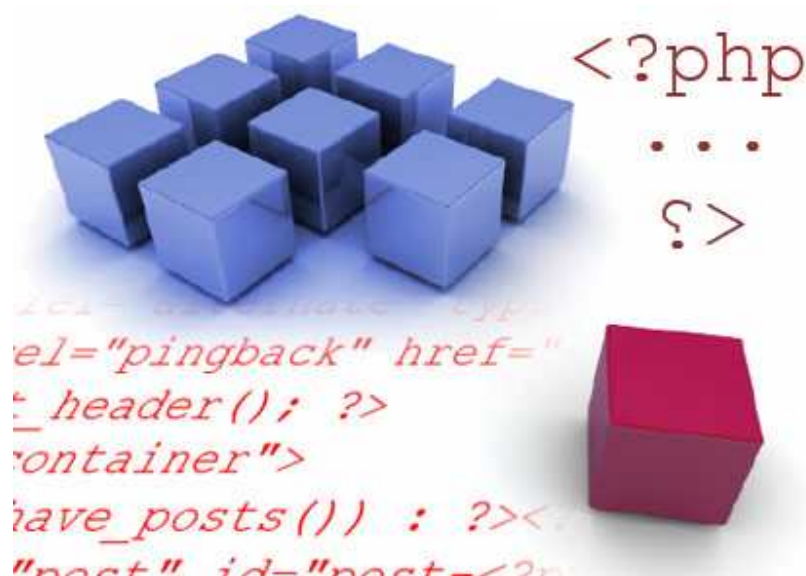
<http://www.p30help.ir/how-make-wordpress-theme-lesson-16>

چگونه برای وردپرس پوسته بسازیم؟ (بخش ۱۷)

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

قالب بندی پست ها و حواشی آن

نمی دانم که در این شرایط کشور، نوشتن یک پست آی تی جایز هست یا خیر به هر حال بعد از یک دقیقه سکوت بخاطر حوادث ناگوار اخیر درس را شروع می کنیم.



همانطور که از عنوان درس مشخص هست در این درس قرار است که پست ها 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.

● مرحله ی 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;** دقیقاً همان عبارت **margin: 0 0 0 0;** است. در حقیقت وقتی که تمامی مقادیر یکسان باشند، شما می توانید برای خلاصه نویسی از یک شماره استفاده کنید که این یک شماره برای تمامی حالت ها استفاده می شود. 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 این ها همگی اختیاری هستند. شما می توانید کدها را در هر کجا تایپ کنید و یا قرار دهید، این کار را تنها برای سازماندهی و مدیریت بهتر کدها به شما گفتم.

● مرحله ی دوم :

حال زمان آن رسیده که **عنوان پست ها (H1)** را استایل دهیم، پس کدهای زیر را در پایین **body{}** تایپ کنید:

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

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

« **font-size: 24px;** این گزینه کاملاً واضح است. توجه کنید، شما اندازه ی فونت را در **body{}** مقدار ۲۴ px 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.

« **padding: 0 0 10px 0;** این کد به معنی این است که از ناحیه ی پایین H1 ده پیکسل فاصله بیافتد. این فضای ۱۰ پیکسلی را برای فضای بین عنوان وبلاگ ها و توضیحات وبلاگ 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 post** به طور ویژه sub-headings های H2 را فقط برای پست ها هدف گیری می کند، به زبان خودمان این سلکتور تنها به عنوان پستها ویژگی می بخشد).

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

(کد **line-height** فاصله ی بین خطوط یک مطلب را معین می کند که در اینجا ما برای آن مقدار متوسط و مناسب ۱۸px را انتخاب کرده ایم. می توانید مقدار آنرا به اعداد مختلف تغییر دهید تا وظیفه ی 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 پاراگراف های متن سروکار دارد و با توجه به مقدار **padding** داده شده، فاصله ی هر پاراگراف از دیگری ۱۰px پیکسل خواهد بود).

● مرحله ی 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 تگ های پاراگراف (**p**) قرار داده اید (اگر بیاد ندارید یک نگاهی به فایل **index.php** بباندازید). پس برای استایل دادن به آن می توانید 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 ی ۱۰ پیکسلی از طرف بالا را به `postmetadata` داده اید.

نکته: اگر بیا داشتید بخش `postmetadata` برای این است که اطلاعات هر پست مانند دسته ی 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.

● مرحله ی 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 **صفحه ی بعدی و صفحه ی قبلی** قالب بندی می کنید. در حقیقت شما:

- ۱۰ پیکسل حاشیه از بالا اضافه کردید.
- اندازه ی فونت را به ۱۴px تغییر دادید.
- پهنای 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 پست ها بود به پایان رسید. دیگر چیزی به پایان کلی دروس نمانده است...

منبع:

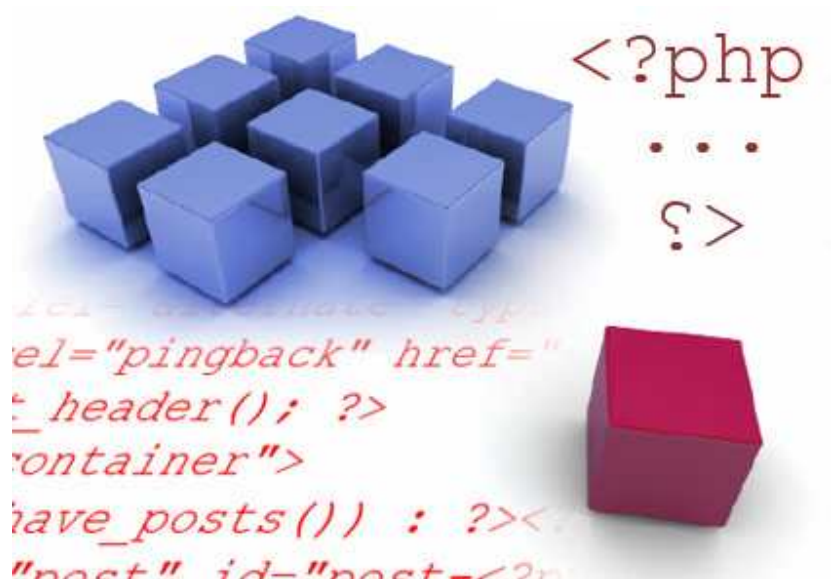
<http://www.p30help.ir/how-make-wordpress-theme-lesson-17>

چگونه برای وردپرس پوسته بسازیم؟ (بخش ۱۸)

نوشته شده توسط [مهدی رادی](#) در تاریخ ۰۸ / ۰۵ / 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 ی جزئیات باید خود تحقق 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، فولدر پوسته ی مورد نظر و در آخر تنظیمات برنامه ی Xampp را باز کنید.

● مرحله ی 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 کدها 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 برد.

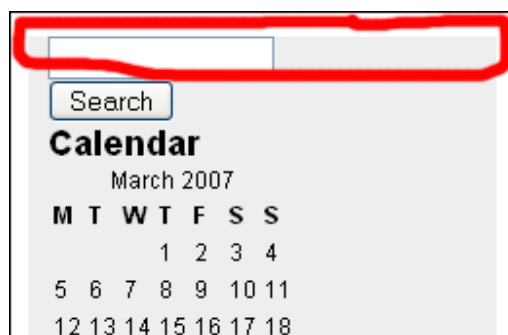
برای مثال :

```
<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 ی 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 مرورگرتان را refresh کنید تا تغییرات را مشاهده کنید.

توجه کنید که چطور ۱۰ پیکسل **padding** به بالا و پایین سایدبار شما اضافه شده است. (عکس زیر)

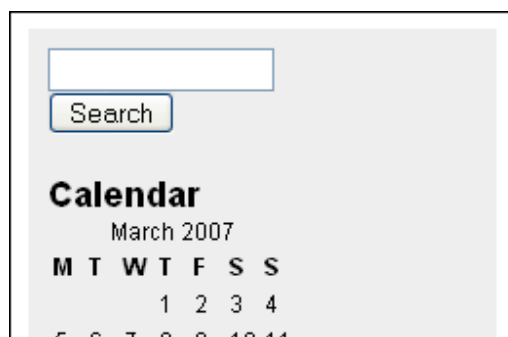


● مرحله ی دوم :

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

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

با این کد مشاهده خواهید کرد که بین هر ابزارک یا ویجیت درون سایدبار فاصله ای ۱۰ پیکسلی از بالا و پایین ایجاد می گردد. (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 می خواهیم که بین اجزا یا ابزارک های داخل سایدبار فاصله ایجاد کنیم، برای مثال می خواهیم بین فرم جستجو و تقویم فاصله ای وجود داشته باشد یا بین تقویم و برگه ها فاصله ی مشخصی وجود داشته باشد. که در این صورت بایستی به تگ `li` درون `ul` سایدبار ویژگی بخشید. حال شما با فاصله ای (`padding`) که به سایدبار داده اید بین اجزای داخل سایدبار فاصله ای ۱۰ پیکسلی ایجاد کرده اید.

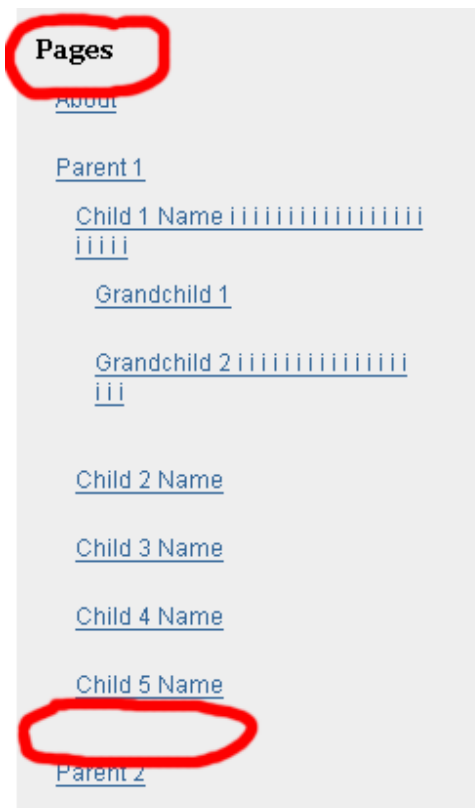
● مرحله ی سوم :

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 مجموعه ی `post` را صفحه آرایی کردیم. در حقیقت شما با سلکتور `ul li h2` عنوان اصلی هر ابزارک را صفحه بندی می کنید.

حالا شما می توانید ویژگی های داده شده به عنوان هر ابزارک را مشاهده کنید.



برای اینکه مدیریت بخش برگه ها حرفه ای تر باشد من خود علاوه بر برگه ی **درباره** یکسری برگه ی دیگر بصورت های مختلف (بصورت تو در تو و معمولی) از طریق مدیریت بخش وردپرس ایجاد کرده ام. اگر دقت کرده باشید فاصله ی هر کدام از

برگه ها با دیگری حدود ۱۰ پیکسل است که کمی نازیبی بوده و بخش سایدبار ما را زشت می کند.

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

چون شما به `li{}` sidebar مقدار `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.

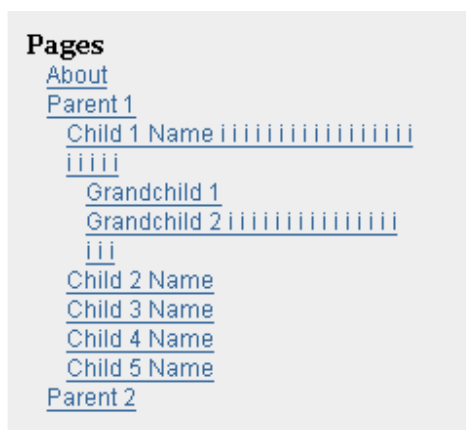
● مرحله ی چهارم :

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

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

با سلکتور `li{}` sidebar و مقدار فاصله ی (`padding`) **صفر**، `li` های زیر دست ویژگی فاصله ی ۱۰ پیکسلی را از دست داده و به همان حالت اول باز می گردند.

نتیجه این هست:



کد `line-height: 24px` را نیز به `li{}` sidebar اضافه کنید:

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


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 Page 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 ی 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؟

قبلاً به شما نحوه ی هدف گیری باکس مخفی (div) که با مشخصه ی 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 ی id پس بصورت `table#wp-calendar` آنرا 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 `table#wp-calendar` sidebar ul li 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	
« Oct						

«; width: 100%»- این کد در مورد پهنای 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 بدهید مثلاً 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:

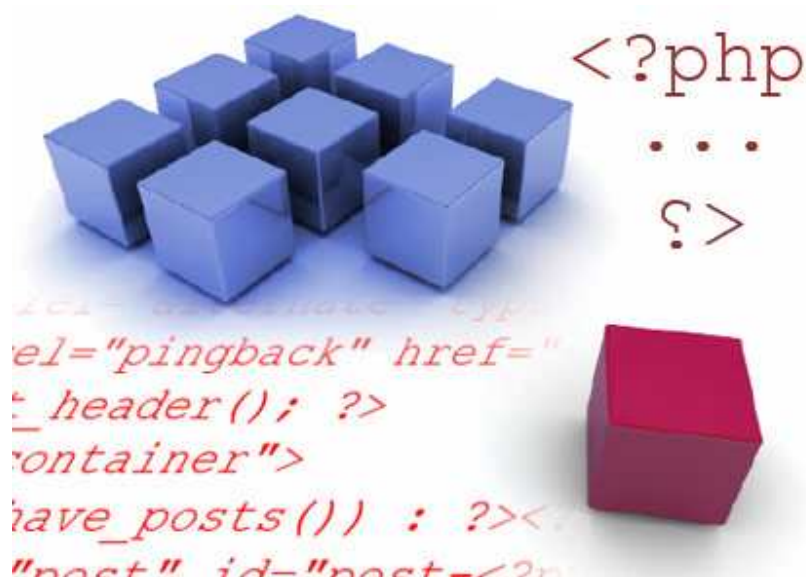
<http://www.p30help.ir/how-make-wordpress-theme-lesson-18>

چگونه برای وردپرس پوسته بسازیم؟ (بخش ۱۹)

نوشته شده توسط [مهدی رادی](#) در تاریخ ۲۰ / ۰۵ / ۱۳۸۸

فوتر و تقسیم بندی فایل index

درس امروز ما کوتاه خواهد بود و در آن به نحوه ی 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 Xampp, style.css, index.php, Firefox 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 دهیم، خود شما باید اینکار ساده را انجام دهید.

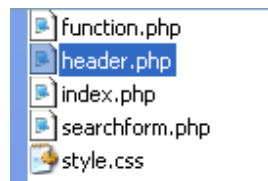
● مرحله دوم:

به تمام تگهای P داخل باکس مخفی footer یک ارتفاع خط (line-height) **هجده** پیکسلی بدهید. به شما یک کمک می کنم، سلکتور مورد نظر برای تگهای پاراگراف درون div فوتر {p} footer # 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 paste کنید.



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

<div id="header">

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

</div>

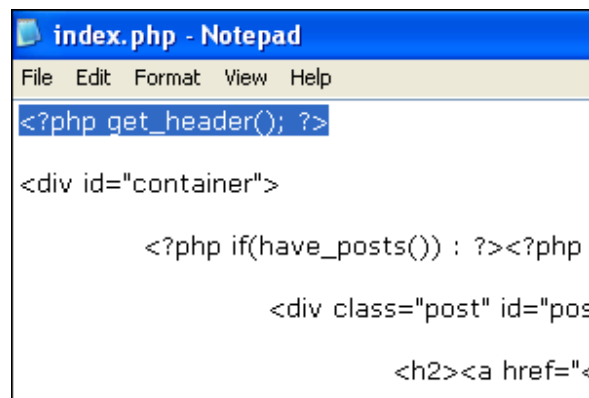
<div id="container">
```

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( ) ; ?>
```



کد بالا یک تابع داخل سیستم وردپرس است که به طور ویژه کدهای داخل فایل **header.php** را درون فایل **index.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 پوسته ی شما شکسته به نظر برسد یا ایرادی داشته باشد در هنگام انتقال کدها حتماً شما اشتباهی انجام داده اید.

● مرحله ی پنجم:

- « مرحله ی 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 می توانید برای فهم بیشتر دانلود کنید.

```

        <?php endif; ?>
    </div>
    <?php get_sidebar(); ?>
    <div id="footer">
    <p>
    Copyright &#169; 2007 <a href="
    </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 فایل های من نگاهی بیاندازید: [style](#), [index](#), [header](#), [sidebar](#), [footer](#). تا اگر به ایرادی برخوردید از آنها کمک بگیرید.
- درس امروز کوتاه بود، چیزی به پایان دروس نمانده است... صبور باشید.....

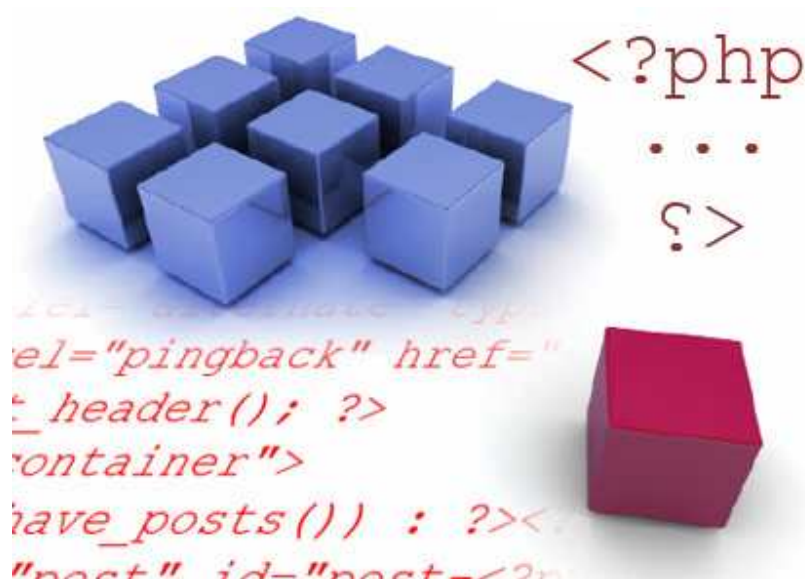
منبع:

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

چگونه برای وردپرس پوسته بسازیم؟ (بخش ۲۰)

نوشته شده توسط [مهدی رادی](#) در تاریخ ۱۳۸۸ / ۰۵ / ۳۱

در درس قبل آموختید که چگونه فایل **index.php** را برای مدیریت بهتر به فایل‌های جداگانه‌ای تقسیم کنید، در این درس نیز قصد داریم تا همین عمل را برای بخش‌های دیگری انجام دهیم. که در اصطلاح به این فایل‌ها Sub Template می‌گویند. اگر بپاد داشته باشید در جلسه‌ی قبل فایل‌های **index.php** را به سه فایل **header.php** و **sidebar.php** و **footer.php** مجزا کردیم تا مدیریت آنها ساده‌تر باشد. در این جلسه این عمل را بصورت پیشرفته‌تری انجام خواهیم داد.



برای ادامه درس به ادامه مطلب مراجعه کنید...

همانگونه که گفته شده در این درس نیز قصد تقسیم کردن فایل **index.php** را داریم ولی با پیش زمینه‌ی درس قبل اینکار بسیار ساده‌تر خواهد بود.

● مرحله ی اول:

در این مرحله می خواهیم که صفحه ای 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** بچسبانید.
- « 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 نحوه ی نشان دادن صفحه ی آرشیو وبلاگ بکار می رود. و البته ما در اینجا این صفحه را دقیقاً مانند صفحه ی اصلی سایت طراحی کردیم بغیر از یک تغییر کوچک که در آن بوجود آوردیم. حال برای فهم بیشتر بروی یکی از آرشیو های قدیمی وبلاگ خود کلیک کنید تا تغییرات را متوجه شوید.

بله درست است در صفحه ی آرشیو شما تنها خلاصه از هر مطلب را مشاهده می کنید نه کل مطلب را. عبارت **the_excerpt** بجای کد **the_content** باعث این عمل می شود.

♦ اما چرا اینکار را انجام دادیم؟

ما این عمل را برای محافظت در برابر جریمه ی گوگل انجام دادیم. چون اگر مطلبی هم در آرشیو و هم در صفحه ی اصلی سایت باشد، گوگل این مطالب را تکراری تلقی می کند و ممکن است که وبلاگ شما را جریمه کند و از رتبه ی وبلاگ شما کاسته شود.

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

● مرحله ی دوم:

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

search.php بچسبانید.

« تغییرات را save کنید.

با انجام این عمل تمامی نتایج جستجوی وبلاگ نیز بصورت گلچین نمایش داده خواهید بود.

♦ حال شاید این سوال برای شما پیش آید که اگر فایل search.php یا archive.php وجود نداشت، صفحه ی جستجوی وبلاگ یا بخش آرشیو وبلاگ چگونه نمایش داده می شد؟

باید در جواب گفت که اگر فایل های search.php یا archive.php وجود نداشته باشند، آنگاه وردپرس بصورت هوشمندانه نحوه ی نمایش تمامی این صفحات را از فایل index.php می گیرد.

● مرحله ی سوم:

« دو فایل با نام های **page.php** و **single.php** بسازید.

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

● مرحله ی چهارم:

فایل **page.php** در مورد نحوه ی نمایش برگه های وبلاگ شما وارد عمل می شود. 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 از page.php پاک کنید. چون اطلاعات 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 posts_nav_link() یا navigation را نیز در فایل page.php پاک کنید.

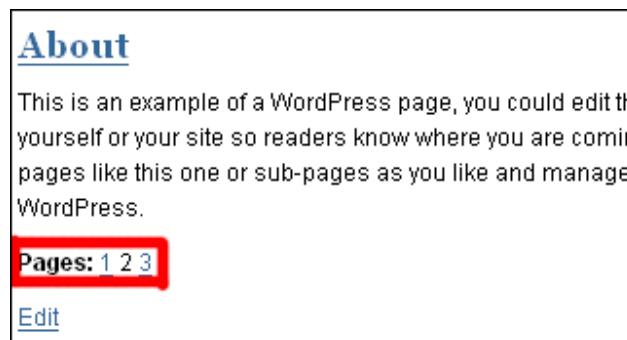
```
<?php endwhile; ?>

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

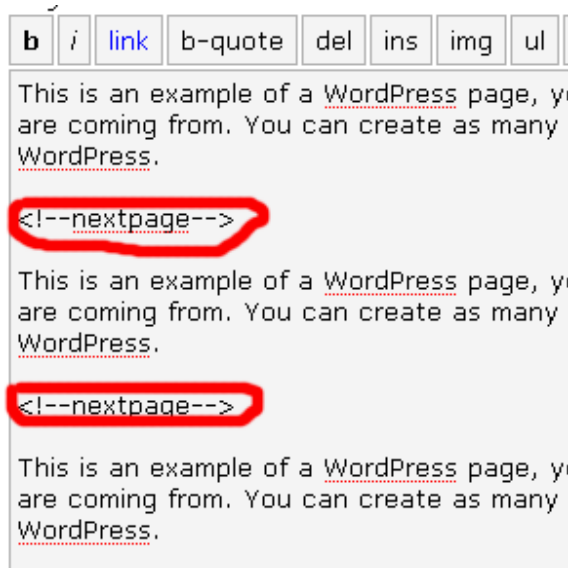
<?php else : ?>
```

♦ چه اتفاقی افتاده است؟

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



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 کنید. 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 `<?php the_content() ?>` 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 تگ `
` قبل از آن را پاک کنید. ولی کل بخش postmetadata را پاک نکنید. و دلیل پاک شدن تگ `br` نیز این است که یک فاصله ی 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 ی اصلی، صفحه ی آرشیو، صفحه ی دسته ها و صفحه ی جستجو ، شما از تابع `posts_nav_link()` استفاده می کنید تا لینک های **Next page** و **Previous page** را صدا بزنید. اما برای صفحه ی **single** چون لینک های **Next page** و **Previous page** وجود ندارند. شما از تابع های `previous_post_link()` و `next_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 مروگر مشاهده کنید.

♦ خلاصه درس:

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

« فایل های `archive.php` و `search.php` دقیقاً یکسان هستند.

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

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

منبع:

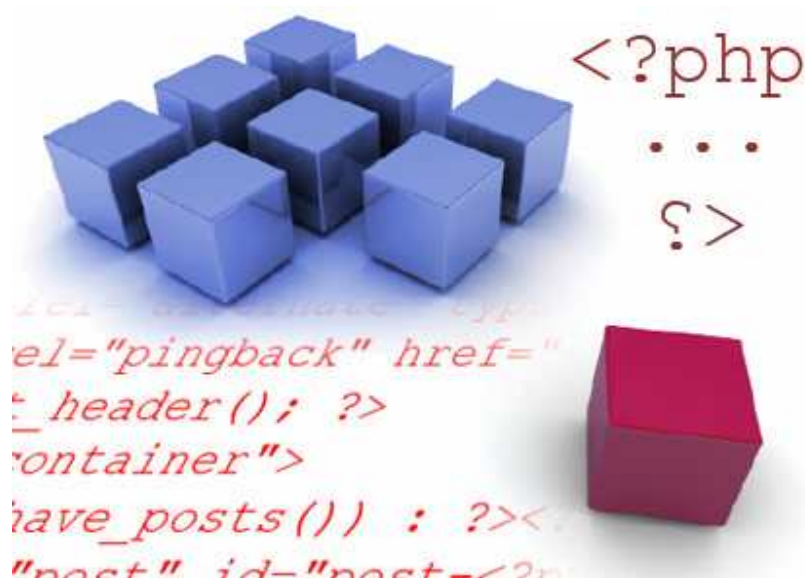
<http://www.p30help.ir/how-make-wordpress-theme-lesson-20>

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

نوشته شده توسط مهدی رادی در تاریخ ۲۶ / 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 ۱۳۸۷/۱۰/۰۳ نوشته شد و امروز یعنی در تاریخ ۱۳۸۸/۰۶/۲۶ این سری آموزشی به پایان می رسد، 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 سختی ها و مشکلات، باید بگویم که خوشحالم که توانستم خدمتی هر چند کوچک به وردپرس کرده باشم...



شما تقریباً همه چیز را می دانید. و برای کامل کردن اطلاعات وردپرسی خود این سری آموزشی را نیز ادامه دهید. امروز شما نحوه ی زندگی بخشی به تمام وبلاگ ها را یعنی نحوه ی قرار دادن بخش نظرات را خواهید آموخت. در تصورات من، وبلاگی که در آن خواننده هایش 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 بقیه ی درس ادامه مطلب را مطالعه کنید....

در ابتدا باید بگویم که روش ساختن بخش نظرات به دوشکل ممکن است، **یک** - حالت پیش فرض وردپرس، که با تایپ فقط یک خط کد امکانش میسر می شود. **دو** - روشی که ما نظرات را با نظر و سلیقه ی خود، درون فایلی بنام `comments.php` می نویسیم تا سبک نمایش را خودمان تعیین کنیم. که در این درس ما روش دوم را به شما می آموزیم.

البته شما باید بدانید:

- « روش دوم یعنی تنظیم کردن قالب در فایل `commnets.php` راه آنچنان استاندارد نیست و بهتر است که از پیش فرض وردپرس استفاده شود.
- « بیشتر طراحان وردپرس از الگو پیش فرض نظرات وردپرس استفاده می کنند.
- « بعضی از طراحان از الگوی پیش فرض وردپرس صرف نظر می کنند چون حتماً آن نیازشان را برآورده نمی کند.
- « شما از ورژن اصلاح شده ی الگو پیش فرض نظرات من استفاده می کنید.

● مرحله ی اول:

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

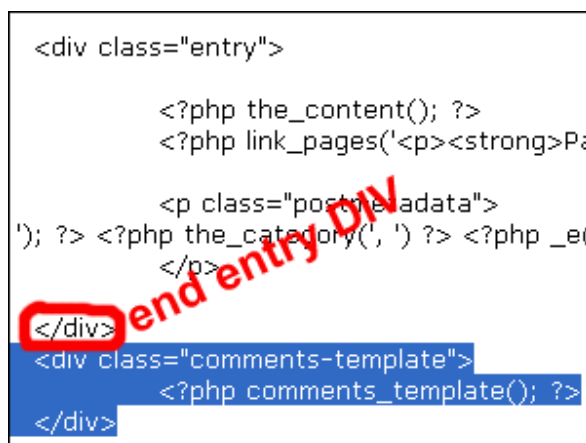
● مرحله ی دوم:

- « کدهای درون فایل [comments-template-css](#) را کپی کنید.
- « این کدها را در پایین یا در زیر سلکتور **#footer** بچسبانید.

● مرحله ی سوم:

در فایل **single.php**، در زیر تگ باکس مخفی ای که مشخصه ی **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 وبلاگ را صدا می زند. 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 می شود و اگر چنین فایلی وجود نداشته باشد، این تابع نحوه ی نمایش بخش نظرات را از بخش پیش فرض وردپرس می گیرد که البته امکانات آن ممکن است راضی کننده نباشد.

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

● مرحله ی چهارم:

خوب مراحل اصلی را انجام دادیم. این مرحله برای اعتبار نامه ی قالب است. یعنی با اینکار می توانید متوجه شوید که کدهای شما بی نقص هستند یا نه. البته اگر

توانستید این اعتبار نامه را بگیرید خیلی دلسرد 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 ی **check** کلیک کنید.

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

- Home page
- Archive pages
- Category pages
- Search result pages
- Pages (مثل) About :
- Single post با page
- Single post بدون بخش نظرات
- Single post با بخش نظرات
- Single post با پیغام باید برای نظر دادن حتماً وارد شوید
- Single post بدون محدودیتی برای نظر دادن
- Password protected single post با comments

♦ توضیحات بیشتر برای الگوی نظرات:

« الگوی نظرات اساساً یک لیستی با تگ **ol** می باشد. 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 متمایز می کنند ولی تگ **ol** هر آیتم را با شماره از دیگری متمایز می کند. (اگر متوجه نشده اید، ناراحت نباشید، این خط در حقیقت تنها برای دادن اطلاعات بیشتر بود و در امتحان پایان ترم نمی آید.)

« در فایل **single.php**، شما تگ **div** با نام مشخصه ی آی دی **comments-template** بدور تابع **comments_template()** قرار داده اید. در حقیقت تگهای **ol** توسط تگ **div**

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 $oddc comment;

<div class="commentmetadata">
<strong><?php comment_author_link() ?></strong>
comment_time() ?></a> <?php _e('Said&#58;')
    <?php if ($comment->comm
    <em><?php _e('Your comme
    <?php endif; ?>

</div>

<?php comment_text() ?>
    </li>

<?php /* Changes every other comment to a di
    if ('alt' == $oddc comment) $oddc comme
    else $oddc comment = 'alt';
?>

<?php endforeach; /* end for each comment */
    </ol>
```

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



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 بیندازم 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 پستیهای 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 سربلند باشید....

به زودی نسخه ی PDF این سری آموزشی را که شامل تمامی درسها است نیز منتشر می کنم تا راحت تر بتوانید استفاده کنید.

منبع:

<http://www.p30help.ir/how-make-wordpress-theme-lesson-21>