

چگونه با استفاده از افزونه پارسی دیت استارت آپ های وردپرس فارسی را توسعه دهیم؟

HOW TO USE **WP-PARSIDATE PLUGIN** FOR PERSIAN WORDPRESS DEVELOPMENT

نویسنده : مهرشاد درزی

توسعه دهنده وردپرس [Wordpress Developer]

وب سایت : RealWp.net



اگر در حال برنامه نویسی یک استارت آپ فارسی با وردپرس باشید ، یکی از نیاز های شما مسلما بحث کار با تاریخ شمسی در وردپرس هست. در بین افزونه های موجود در زمینه فارسی سازی وردپرس دو افزونه wp-jalali (افزونه جلالی) و افزونه پارسی دیت (wp-parsidate) محبوبیت زیادی نزد کاربران دارد.

در پروژه ای که برای پرتال یکی از دانشگاه های کشور مبتنی بر هسته وردپرس ایجاد کردم با کمی تحقیق افزونه ی پارسی دیت را برای این کار انتخاب کردم . در این مقاله شما را با توابع و آموزش کار با تاریخ شمسی توسط افزونه wp-parsidate در وردپرس آشنا خواهم کرد .

چه انتظاری باید از یک افزونه فارسی ساز وردپرس داشت ؟

بطور کلی دو انتظار میتوان از یک افزونه فارسی ساز وردپرس داشت:

۱. ابتدا فارسی سازی تمامی توابع محلی یا Native در وردپرس ، توسط فیلتر ها و اکشن هایی که وردپرس در اختیار توسعه دهندگان قرار داده. به عنوان مثال زمانی که شما افزونه را نصب می کنید تابع `get_the_time` به طور کامل تاریخ را بصورت شمسی برپایمان نشان می دهد. این نیاز اصولا برای وبمستران وردپرس بیشتر مطرح هست.

۲. قاعدتا زمانی که شما یک پروژه ای بزرگ را با وردپرس استارت می زنید . حتما جدول هایی جدا از جدول های پیش فرض وردپرس در بانک اطلاعاتی دارید.

سوال اینجاست با چه ابزاری میتوان تاریخ هایی که در جداول شخصی خودمان ایجاد کردیم فارسی شود ؟ این همان نیاز دوم هست یعنی افزونه باید مستنداتی را در اختیار توسعه دهنده وردپرس قرار دهد تا بتوان وردپرس فارسی را توسعه داد.

چرا می بایست افزونه پارسی دیت را بر افزونه جلالی ترجیح دهیم ؟

با بررسی افزونه جلالی و پارسی دیت نهایت به دلایل زیر استفاده از افزونه wp-parsidate را ترجیح دادم.

۱ - بروز رسانی یک افزونه یکی از مهم ترین دلایل پیشرفت آن تلقی می شوید. نمی دانید وقتی برنامه نویس باشید و وارد مخزن گیت هاب یک پروژه شوید ، آخرین بروز رسانی آن را یک سال پیش ببینید این حس به شما تلقی می شود که این پروژه احتمالا متوقف شده و شخص

آن را با گزینه ی بهتری جایگزین کرده.

افزونه پارسی دیت بروز رسانی های منظمی در گیت هاب دارد و این یعنی خودش را با پیشرفت وردپرس هماهنگ کرده و نیاز های کاربران را ارتقا داده اما افزونه جلالی در یک نیاز ثابت مانده.

The screenshot shows the GitHub repository page for 'wordpress-parsi/wp-parsidate'. At the top, it displays '2' watches, '9' stars, and '4' forks. Below this, there are tabs for 'Code', 'Issues (3)', 'Pull requests (1)', 'Projects (1)', 'Wiki', and 'Insights'. The repository name 'WP Parsi Date' and its URL 'http://wp-parsi.com/' are shown, along with tags for 'wordpress', 'parsi', 'persian', 'plugin', 'jalali', 'shamsi', and 'calendar'. Statistics include '142 commits', '2 branches', '22 releases', '5 contributors', and 'GPL-2.0' license. A 'New pull request' button is visible. A table of files and their commit history is shown below, with the latest commit being '61dc152 on May 16' by 'man4toman'. The table lists files like 'assets', 'includes', 'languages', '.gitignore', 'LICENSE', 'README.md', 'readme.txt', 'screenshot-1.png', 'screenshot-2.png', 'screenshot-3.png', 'screenshot-4.png', and 'wp-parsidate.php' with their respective commit dates.

۲- ساختار برنامه نویسی افزونه پارسی دیت کاملا تجربی است و از هیچ فریم ورک معتبر در زمینه ی استارتر های افزونه وردپرس مانند **wppb** استفاده نشده. این کار تسلط کامل برنامه نویسی را بر وردپرس نشان می دهد. البته این را هم مد نظر قرار دهید، زمانی که محصول یا افزونه ای تولید می شود به طور حتم اگر قرار است شخصی در همان موضوع افزونه ای تولید کند نکات ضعف آن را پوشش خواهد داد.

۳- من در یک پروژه با **فریم ورک لاراول** از افزونه پارسی دیت استفاده کردم. بله درست شنیدید یکی از اصلی ترین قابلیت های افزونه parsidate مستندات آن است که به توسعه دهندگان امکان گسترش پروژه های وب فارسی از جمله وردپرس را میدهد. یک کلاس با نام bn_parsidate در این افزونه وجود دارد که این وظیفه را برعهده دارد.

آموزش نصب افزونه پارسی دیت

برای نصب افزونه فارسی ساز وردپرس کافیست در صفحه افزونه ها مدیریت وردپرس ، عبارت wp-parsidate را جستجو و افزونه را نصب و فعال کنید. بعد از نصب ، یک منو با نام تنظیمات پارسی در مدیریت ایجاد می شود. روی آن کلیک کنید و گزینه تاریخ خورشیدی را فعال و تنظیمات را ذخیره کنید با این عملکرد تمامی تابع های اصلی وردپرس که با تاریخ سرو کار دارند از فیلتر افزونه پارسی دیت عبور کرده و در نتیجه شمسی خواهند شد.

تبدیل تاریخ میلادی به شمسی در وردپرس

فرض کنید میخواهیم تاریخ ثبت نام کاربر در وردپرس را به شمسی نمایش دهیم. در جدول wp_users فیلد user_registered تاریخ ثبت نام کاربر را با به میلادی با فرمت datetime یعنی تاریخ به همراه ساعت ثبت نام ، قرار داده است. ابتدا این مقدار را دریافت میکنیم سپس توسط تابع parsidate آن را به شمسی نمایش می دهیم:

```
1. $user_id = 1;
2. $user_data = get_userdata($user_id);
3. $user_date_register = $user_data->user_registered;
4. echo parsidate("Y-m-d", $register_date, "per");
```

در خط دوم توسط تابع `get_userdata` اطلاعات پایه کاربر با شناسه ۱ را از دیتابیس دریافت و در خط سوم تاریخ ثبت نام فرد را به میلادی در متغیر قرار دادیم. در نهایت توسط تابع `parsidate` آن را به شمسی تبدیل کردیم. خروجی این تابع مثلا ۱۳۹۸-۰۲-۱۰ می باشد.

همانطوری که مشاهده می کنید تابع `parsidate` دارای سه پارامتر اصلی می باشد:

۱. پارامتر اول نوع نمایش خروجی تاریخ هست که بر اساس پارامتر های پیش فرض تعریف شده در تابع `date` زبان برنامه نویسی `php` هست. لیست کامل این پارامتر ها در پایان مقاله ضمیمه شده است.

۲. پارامتر دوم باید تاریخ میلادی را به تابع بدهیم. که در مثال بالا متغیر تاریخ ثبت نام فرد به تابع داده شده است. دقت کنید در این تابع دیگر نیازی به `strtotime` کردن تاریخ نیست و این مبدل در خوده هسته ی این تابع وجود دارد.

۳. پارامتر سوم مشخص میکنیم اعدادی که قرار است به عنوان خروجی نمایش داده شود به صورت فارسی یا `per` باشد و یا به صورت انگلیسی یا `eng`

ایجاد شورت کد نمایش تاریخ امروز در وردپرس

در مثال زیر یک شورت کد به نام `nowdate` ایجاد می کنیم. این شورت کد تاریخ امروز را در هر کجای سایت که `shortcode` را قرار دهید نمایش می دهد:

```

1. function wp_nowdate() {
2. return parsidate("l j F Y", 'now', "per");
3. }
4. add_shortcode( 'nowdate ', 'wp_nowdate' );

```

اگر در ویرایشگر وردپرس شورت کد [nowdate] را منتشر کنیم خروجی آن تاریخ همان روز به ما می باشد مثلا "دوشنبه ۱۰ مرداد ۱۳۹۸". دقت کنید که در این مثال ما بجای تاریخ میلادی تنها عبارت now را نوشتیم.

شمسی کردن تاریخ های میلادی برای کوئری های دلخواه

اگر یک جدول دلخواه در پروژه وردپرس خود ایجاد کرده اید میتوانید از کلاس `wpdb` ابتدا مقدار را دریافت کنید و سپس آن را توسط تابع `parsidate` به شمسی تبدیل کنید. فرض کنید در یک پروژه دانشگاهی لیست تمامی دانشجویان را در یک جدول به نام `person` قرار دادیم و یک فیلد به نام `date` وجود دارد که تاریخ ارائه پایان نامه آن دانشجو می باشد. برای نمایش تاریخ آن به شمسی از کد زیر استفاده میکنیم:

```

1. global $wpdb;
2. $result = $wpdb->get_row( "SELECT * FROM `".$wpdb->prefix."person`
WHERE `code` = 498" );
3. $person_date = $result->date;
4. echo parsidate("Y-m-d", $person_date, "per");

```

در مثال بالا ابتدا به یک درخواست از جدول `wp_person` ، اطلاعات دانشجو با کد ۴۹۸ را دریافت کردیم. سپس فیلد `date` آن را برای شمسی سازی به تاریخ `parsidate` واگذار کردیم.

توجه : اگر میخواهید روی خروجی تابع `parsidate` عملیاتی انجام دهید نایستی خروجی آن را بصورت فارسی یا "per" منتشر کنید. چون کاراکتر های خروجی در این حالت بصورت رشته برای `php`

معرفی می شود ن عدد.

فرض کنید میخواهید سال تولد یکی از اعضا رو برگردانید و به آن ۴ سال اضافه کنید و نمایش دهید در این حالت باید حتما خروجی "eng" باشد تا php بتواند عملیات ریاضی را انجام دهد:

```
1. $date = "2018-02-10";  
2. echo parsidate("Y", $date, "eng") + 4;
```

در مثال بالا من خروجی تاریخ سال شمسی را به صورت eng در آوردم تا بتوانم به آن ۴ عدد اضافه کنم.

اگر کاربر افزونه پارسی دیت را نصب نکرده باشد چه می شود ؟

ما افزونه ای می سازیم و در آن از تابع parsidate استفاده میکنیم. اما زمانی که کاربری میخواهد آن افزونه را در وردپرس خود نصب کند ، افزونه ی پارسی دیت را نصب نکرده باشد و یا به هر دلیلی از افزونه جلالی در حال استفاده کردن باشد. اتفاقی که می افتد با خطای php در وردپرس مواجه می شود مبنی بر این که تابع parsidate یافت نشد. این یعنی افزونه ما بدرستی کار نخواهد کرد. برای رفع این مشکل دو راه حل پیشنهاد میدهم:

۱- میتوانید از کتابخانه **TGM Plugin Activation** استفاده کنید. (که برای این مورد من زیاد پیشنهاد نمیکنم.)

حتما تا بحال قالبی را نصب کرده اید که بعد از نصب آن ، به طور اتوماتیک افزونه هایی مثل Contact Form 7 نصب شود. کتابخانه TGM دقیقا همین کار را برایتان انجام میدهد. شما میتوانید توسط ضمیمه کردن این کتابخانه به افزونه خودتان ، به وردپرس دستور بدهید بعد از نصب افزونه شما حتما افزونه wp-parsidate یا هر افزونه ای که بخواهید را نصب کند.

۲- راه حل بهتر این است که کلاس پایه افزونه parsidate که با نام bn_parsidate و در مسیر زیر در پوشه افزونه قرار دارد را ضمیمه پروژه خود کنیم.

همان طور که گفتیم این کلاس bn_parsidate را می‌توانید حتی در پروژه‌های غیر وردپرس هم استفاده کنید.

1. wp-parsidate/includes/parsidate.php

خوب من ابتدا این فایل را داخل پروژه خود قرار میدهم اما قبلش باید چک کنم که کاربر این افزونه را نصب کرده هست یا خیر تا در عملکرد php اختلال ایجاد نشود. برای این کار می‌توانید از دو کد زیر استفاده کنید:

```
1. /* Check Exist function is_active_plugin */
2. if ( ! function_exists( 'is_plugin_active' ) ) {
3. include_once( ABSPATH . 'wp-admin/includes/plugin.php' );
4. }
5. /*Check Wp-parsidate Plugin is Active*/
6. if ( is_plugin_active( 'wp-parsidate/wp-parsidate.php' ) ===false ) {
7. include_once( 'parsidate.php' );
8. }
```

در وردپرس اگر میخواهید چک کنید که یک افزونه نصب هست یا خیر می‌توانید از تابع is_active_plugin استفاده کنید. در بخش اول ابتدا ما چک کردیم در صورتی که این تابع فراخوانی نشده باشد، از هسته ی وردپرس آن را فراخوانی کند و در بخش بعدی چک میکنیم، اگر افزونه wp-parsidate نصب نشده است فایل کلاس bn_parsidate را ضمیمه وب سایت کند.

روش دیگری هم وجود دارد که من همیشه از آن استفاده میکنم، آن هم چک کردن موجودیت کلاس هست در مقابل افزونه. چون در برخی پروژه‌ها دیدم که عینا برنامه نویسان برای شخصی سازی نام پوشه wp-parsidate را تغییر میدهند و در این حالت ما نمیتوانیم چه پارامتری به تابع is_active_plugin دهیم که بفهمیم افزونه نصب هست یا خیر.

مورد بعدی اینکه اگر بخواهید در یک فریم ورک مثل **codeigniter** از کلاس استفاده کنید دیگر تابع `is_active_plugin` در آن معنی ندارد و می بایست کلاس را چک کنید. کد دوم به صورت زیر هست:

```
1. if (class_exists('bn_parsidate') ===false) {  
2. include_once( 'parsidate.php' );  
3. }
```

اگر یک افزونه دو زبانه فارسی و انگلیسی نوشتیم چه کنیم ؟

این مشکل را من زیاد داشتم اکثر افزونه هایی که تولید میکنم برای هر دو زبان انگلیسی و فارسی هست. خوب مشکلی که ما داریم این است که افراد غیر ایرانی نیازی به افزونه `parsidate` ندارند و به تاریخ میلادی نیاز دارند. از طرفی ما در افزونه خود از تابع شمسی استفاده کردیم. راه حل چیست ؟ برای جلوگیری از این کار میتوانیم ابتدا چک کنیم که وردپرس کاربر در چه زبانی هست بعد از آن مطابق با فارسی یا انگلیسی بودن آن نوع تاریخ را نمایش دهیم:

```
1. /*Check Farsi Locale WordPress*/  
2. function wp_is_fa() {  
3. if ( get_locale() == 'fa_IR' ) { return true; } else { return false;  
   }  
4. }  
5. $date = "2019-05-10";  
6. if (wp_is_fa() ) {  
7. $date = parsidate("Y-m-d", $date, "eng");  
8. }  
9. echo $date;
```

در مثال بالا ابتدا یک تابع ساده نوشتیم تا چک کند در حال حاضر وردپرس کاربر فارسی هست یا خیر. سپس در هر کجای برنامه نویسی افزونه چک میکنم که اگر وردپرس کاربر فارسی بود برای ما به شمسی نمایش دهد.

البته اگر بخواهیم استاندارد تر کدنویسی کنیم میتوانیم از Filter در وردپرس استفاده کنیم مانند:

```
1. $date = "2019-05-10";
2. $date = apply_filters( 'wp_show_datetime_my_plugin', $date );
3. echo $date;
4. /* Adding filter */
5. if ( get_locale() == 'fa_IR' ) {
6. add_filter( 'wp_show_datetime_my_plugin', function($date){
7. $date = parsidate("Y-m-d", $date, "eng");
8. return $date;
9. });
10. }
```

آموزش استفاده از افزونه پارسی دیت در یک پروژه غیر از وردپرس

همانند روش بالا میتوانید فایل کلاس bn_parsidate را در پروژه ی خودتان قرار دهید و آن را ضمیمه کنید. سپس کلاس parsidate را به شکل زیر فراخوانی کرده و از آن استفاده کنید:

```
1. $parsdate_class = bn_parsidate::getInstance();
2. $nowdate = $parsdate_class->persian_date("Y-m-d", "2018-10-05", "eng");
```

تبدیل تاریخ شمسی به میلادی در وردپرس

اگر تمامی تاریخ هایی که در بانک اطلاعاتی هست میلادی باشد. یکی از نیاز های ما همواره این است ، که تاریخ شمسی از کاربر دریافت کنیم مانند صفحات گزارش و آن را برای جستجو در بانک اطلاعاتی به تاریخ میلادی تبدیل کنیم.

فرض کنید میخواهیم لیست تمامی کاربرانی که از تاریخ ۱۳۸۹-۰۲-۱۰ تا تاریخ ۱۳۹۶-۱۰-۰۵ ثبت نام کرده اند را در وردپرس لیست کنیم. برای این کار ابتدا توسط تابع **gregdate** تاریخ شمسی را به میلادی تبدیل کرده ، سپس پرس و جو در بانک اطلاعاتی را انجام می دهیم:

```
1. $after = gregdate("Y-m-d", "1389-02-10");
2. $before = gregdate("Y-m-d", "1396-10-05");
3. global $wpdb;
4. $result = $wpdb->get_results( "SELECT * FROM `".$wpdb->users."`
    WHERE (user_registered BETWEEN '".$after.'" 00:00:00' AND
    '".$before.'" 10:15:55')" );
5. foreach ($result as $user) {
6. echo $user->display_name;
7. echo "<br>";
8. }
```

در مثال بالا ابتدا دو تاریخ را تبدیل به میلادی کردیم. سپس از آنها در رشته ی SQL برای ایجاد یک درخواست توسط کلاس wpdb استفاده کردیم.

نمایش آرشیو مطالب در وردپرس با کلاس wp_Query و افزونه پارس دیت

یکی از روش های نمایش آرشیو محتوا در وردپرس استفاده از کلاس قدرتمند WP_Query می باشد. در مثال زیر ما تمامی پست هایی که از تاریخ ۱۳۸۹-۰۲-۱۰ تا تاریخ ۱۳۹۶-۱۰-۰۵ منتشر شد را در سایت میتوانیم نمایش دهیم:

```

1. /* After Date */
2. $after = gregdate("Y-m-d", "1389-02-10");
3. $array_after = explode("-", $after);
4. /* Before Date */
5. $before = gregdate("Y-m-d", "1396-10-05");
6. $array_before = explode("-", $before);
7. /* Set Query */
8. $args = array(
9. 'date_query' => array(
10. array(
11. 'after'      => array(
12. 'year'      => $array_after[0],
13. 'month'     => $array_after[1],
14. 'day'       => $array_after[2],
15. ),
16. 'before'    => array(
17. 'year'      => $array_before[0],
18. 'month'     => $array_before[1],
19. 'day'       => $array_before[2],
20. ),
21. 'inclusive' => true,
22. ),
23. ),
24. 'posts_per_page' => -1,
25. );
26. $query = new WP_Query( $args );
27. while ($query->have_posts()):
28. $query->the_post();
29. echo get_the_title();
30. echo "<br>";
31. endwhile;
32. wp_reset_postdata();

```

اگر دقت کنید کلاس Wp_Query مقادیر سال ، ماه و روز را به صورت جداگانه از ما میخواهد. به همین دلیل من مقدار خروجی تابع میلادی رو با استفاده از تابع explode جداسازی کردم و هر کدام از ماه و سال و روز رو براحتی برگرداندم.

فارسی سازی اعداد در وردپرس

برای فارسی سازی اعداد در وردپرس می توانید از تابع per_number براحتی استفاده کنیم:

```
1. $number = 1486;  
2. echo per_number($number);
```

تبدیل اعداد فارسی به انگلیسی

گاهی هم اتفاق می افتد که میخواهید اعداد فارسی را به انگلیسی تبدیل کنید یعنی دقیقا برعکس حالت بالا.تابع eng_number برای همین امر مورد استفاده قرار بگیرد.

```
1. echo eng_number(۱۴۸۶);
```

ضمیمه

پارامترهای تاریخ برای استفاده در تابع Parsidate

| مثال | دامنه | توضیحات | کاراکتر |
|---------------------------------|----------------|-------------------------------------|-------------|
| - | - | کاراکترها با حروف کوچک | کوچک |
| ق.ظ | ق.ظ ب.ظ | اوقات روز - به صورت خلاصه | a |
| ۱ | ۱ - ۴ | شماره ی فصل (ربع) از سال | b |
| ۱۳:۰۸:۲۶، ۴/۴/۱۳۹۰ +۰۴:۳۰ | طبق نمونه : | قالب مرگب ۱ (ltr): Y/n/j ,H:i:s P | c |
| ۹ | ۰۱ - ۳۱ | شماره ی روز از ماه - ۲ رقمی | d |
| Greenwich | مناطق زمانی | منطقه ی زمانی تنظیم شده | e |
| تابستان | بهار تا زمستان | نام فصل با حروف فارسی | f |
| ۷ | ۱ - ۱۲ | ساعت در روز - ۱۲ساعته - ۲ رقمی | g |
| ۰۷ | ۰۱ - ۱۲ | ساعت در روز - ۱۲ ساعته - ۲ رقمی | h |
| ۱۷ | ۰۰ - ۵۹ | دقیقه در ساعت - ۲ رقمی | i |
| ۲ | ۱ - ۳۱ | شماره ی روز از ماه - ۲ رقمی | j |
| ۷۶/۲ | ۰ - ۱۰۰ | در صد باقیمانده از سال (**.?) | k |
| شنبه | شنبه تا جمعه | نام روز در هفته - کامل | l |
| ۰۳ | ۰۱ - ۱۲ | شماره ی ماه از سال - ۲ رقمی | m |
| ۳ | ۱ - ۱۲ | شماره ی ماه از سال - ۲ رقمی | n |
| ۱۳۹۱ | شماره ی سال | سال هفته ای (به عدد) چهار رقمی | o |
| جوزا | حمل تا حوت | نام باستانی برج ها | p |
| گاو | موش تا خوک | نام حیوانی سال ها | q |
| شنبه، ۴ تیر ۱۳۹۰ +۰۴۳۰ ۱۳:۰۸:۲۶ | طبق نمونه : | قالب مرگب ۲ (ltr): H:i:s O Y F z ,ا | r |
| ۰۹ | ۰۰ - ۵۹ | شماره ی ثانیه در دقیقه - ۲ رقمی | s |
| ۳۱ | ۳۱ ۳۰ ۲۹ | تعداد روزهای (همان) ماه | t |
| ۰۰۰۰۰۰ | طبق نمونه : | میکرو ثانیه - ۶ رقمی | u |
| نود و یک | از تا نود و نه | سال به حروف - خلاصه ی دو رقمی | v |
| ۰ | ۰ - ۶ | شنبه=۰، عدد روز در هفته | w |

| | | | |
|------------------------|------------------|---------------------------------------|-------------|
| ۹۱ | ۰۰ - ۹۹ | سال (به عدد) دو رقمی | y |
| ۴۱ | ۰ - (۳۶۴ ۳۶۵) | تعداد روز (کامل) گذشته از سال | z |
| - | - | کاراکترها با حروف بزرگ | بزرگ |
| قبل از ظهر | طبق نمونه : | اوقات روز - کامل | A |
| ۰۹۸ | ۰۰۰ - ۹۹۹ | زمان اینترنتی، ۱/۴۴ دقا از ۰:۰۰ GMT۲۳ | B |
| ۱۴ | ... ۱۵ ۱۴ ... | شماره ی قرن هجری شمسی | C |
| ش | ش تا ج | حرف اول نام روز هفته، ش=شنبه | D |
| خرداد | فروردین تا اسفند | نام ماه از سال - کامل | F |
| ۷ | ۰ - ۲۳ | ساعت در روز - ۲۴ساعته - یا ۲رقم | G |
| ۰۷ | ۰۰ - ۲۳ | ساعت در روز - ۲۴ساعته - ۲رقمی | H |
| ۰ | ۰ ۱ | نشانگر طولانی بودن روز = ۱ | I |
| دو | یک تا سی و یک | شماره ی روز از ماه به حروف | J |
| ۱۴ | ۱۰۰ - ۰ | در صد گذشته از سال (?.*) | K |
| ۱ | ۰ ۱ | سال : کیبسه=۱ و غیر کیبسه=۰ | L |
| خر | فر تا اس | نام ماه از سال - خلاصه | M |
| ۷ | ۱ - ۷ | یک شنبه=۱، عدد روز در هفته | N |
| +۰۴۳۰ | - ، ۱۲۰۰+۱۴۰۰ | اختلاف ساعت جهانی - بدون ":" | O |
| +۰۴:۳۰ | - ، ۱۲:۰۰+۱۴:۰۰ | اختلاف ساعت جهانی - با ":" | P |
| ۳۲۴ | ۰ - (۳۶۵ ۳۶۴) | تعداد روز (کامل) باقی مانده از سال | Q |
| منتظر باشید | به زودی | " در دست ساخت " | R |
| ام | ام | واژه ی "ام" | S |
| IRDT | مخفف مناطق | مخفف منطقه ی زمانی تنظیم شده | T |
| ۱۲۴۳۹۵۵۳۴۶ | طبق نمونه : | برچسب زمانی (Unix) timestamp | U |
| هزار و سیصد و نود و یک | طبق نمونه : | سال به حروف - کامل | V |
| ۱۱ | ۰۰ - (۵۲ ۵۳) | شماره ی این هفته در سال - ۲رقم | W |
| منتظر باشید | به زودی | " در دست ساخت " | X |
| ۱۳۹۱ | طبق نمونه : | سال (به عدد) چهار رقمی | Y |
| ۱۶۲۰۰ | -۵۰۴۰۰ ، ۴۳۲۰۰ | اختلاف ساعت جهانی به ثانیه | Z |

برای مشاهده مقالات در زمینه آموزش وردپرس از وب سایت "وردپرس واقعی" دیدن فرمائید.

WWW.REALWP.NET