Python's `strftime` directives

Note: Examples are based on `datetime.datetime(2013, 9, 30, 7, 6, 5)`

Code	Meaning	Example
`%a`	Weekday as locale's abbreviated name.	`Mon`
`%A`	Weekday as locale's full name.	`Monday`
`%w`	Weekday as a decimal number, where 0 is Sunday and 6 is Saturday.	`1`
`%d`	Day of the month as a zero-padded decimal number.	`30`
`%-d`	Day of the month as a decimal number. (Platform specific)	`30`
`%b`	Month as locale's abbreviated name.	`Sep`
`%B`	Month as locale's full name.	`September`
`%m`	Month as a zero-padded decimal number.	`09`
`%-m`	Month as a decimal number. (Platform specific)	`9`
`%y`	Year without century as a zero-padded decimal number.	`13`
`%Y`	Year with century as a decimal number.	`2013`
`%H`	Hour (24-hour clock) as a zero-padded decimal number.	`07`
`%-H`	Hour (24-hour clock) as a decimal number. (Platform specific)	`7`
`%I`	Hour (12-hour clock) as a zero-padded decimal number.	`07`
`%-I`	Hour (12-hour clock) as a decimal number. (Platform specific)	`7`
`%p`	Locale's equivalent of either AM or PM.	`AM`
`%M`	Minute as a zero-padded decimal number.	`06`
`%-M`	Minute as a decimal number. (Platform specific)	`6`
`%S`	Second as a zero-padded decimal number.	`05`
`%-S`	Second as a decimal number. (Platform specific)	`5`
`%f`	Microsecond as a decimal number, zero-padded on the left.	`000000`
`%z`	UTC offset in the form +HHMM or -HHMM (empty string if the the object is naive).	
`%Z`	Time zone name (empty string if the object is naive).	
`%j`	Day of the year as a zero-padded decimal number.	`273`
`%-j`	Day of the year as a decimal number. (Platform specific)	`273`

Code	Meaning	Example
%U	Week number of the year (Sunday as the first day of the week) as a zero padded decimal number. All days in a new year preceding the first Sunday are considered to be in week 0.	39
%W	Week number of the year (Monday as the first day of the week) as a decimal number. All days in a new year preceding the first Monday are considered to be in week 0.	39
%c	Locale's appropriate date and time representation.	Mon Sep 30 07:06:05 2013
%x	Locale's appropriate date representation.	09/30/13
%X	Locale's appropriate time representation.	07:06:05
%%	A literal '%' character.	%

Source: Python's `strftime` documentation.

Platform specific directives: The full set of format codes supported varies across platforms, because Python calls the platform C library's strftime() function, and platform variations are common. To see the full set of format codes supported on your platform, consult the `strftime(3)` documentation.

The Python docs contain all the format codes that the C standard (1989 version) requires, and these work on all platforms with a standard C implementation. Note that the 1999 version of the C standard added additional format codes. These include codes for non-zero-padded numbers, that can be obtained by appending a dash (-) after the percent (%) sign.

Why? I need to use Python's `strftime` rarely enough that I can't remember it off the top of my head and never bookmark it but often enough to be annoyed with having to Google "python strftime" and then find the table above in the Python documentation. So I decided to put this reference page up.

Who? This was slapped together on a whim by Will McCutchen, who is just slouching on the shoulders of the awesome Python documentation team. The source code is on Github and pull requests are welcome!

When? Last updated 2015-11-24.

What else? You might also like PyFormat by Ulrich Petri and Horst Gutmann.