

# Python **strftime** cheatsheet



Code	Example	Description
%a	Sun	Weekday as locale's abbreviated name.
%A	Sunday	Weekday as locale's full name.
%w	0	Weekday as a decimal number, where 0 is Sunday and 6 is Saturday.
%d	08	Day of the month as a zero-padded decimal number.
%-d	8	Day of the month as a decimal number. (Platform specific)
%b	Sep	Month as locale's abbreviated name.
%B	September	Month as locale's full name.
%m	09	Month as a zero-padded decimal number.
%-m	9	Month as a decimal number. (Platform specific)
%y	13	Year without century as a zero-padded decimal number.
%Y	2013	Year with century as a decimal number.
%H	07	Hour (24-hour clock) as a zero-padded decimal number.
%-H	7	Hour (24-hour clock) as a decimal number. (Platform specific)
%I	07	Hour (12-hour clock) as a zero-padded decimal number.
%-I	7	Hour (12-hour clock) as a decimal number. (Platform specific)
%p	AM	Locale's equivalent of either AM or PM.
%M	06	Minute as a zero-padded decimal number.
%-M	6	Minute as a decimal number. (Platform specific)
%S	05	Second as a zero-padded decimal number.
%-S	5	Second as a decimal number. (Platform specific)
%f	000000	Microsecond as a decimal number, zero-padded on the left.

Code	Example	Description
%z	+0000	UTC offset in the form $\pm$ HHMM[SS[.ffffff]] (empty string if the object is naive).
%Z	UTC	Time zone name (empty string if the object is naive).
%j	251	Day of the year as a zero-padded decimal number.
%-j	251	Day of the year as a decimal number. (Platform specific)
%U	36	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.
%W	35	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.
%c	Sun Sep 8 07:06:05 2013	Locale's appropriate date and time representation.
%x	09/08/13	Locale's appropriate date representation.
%X	07:06:05	Locale's appropriate time representation.
%%	%	A literal '%' character.

## 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 (-) (UNIX) or hash (#) (Windows) after the percent (%) sign.

## Source

This cheatsheet was built from the [Python standard library strftime documentation](#). See [github.com/mccutchen/strftime.org](https://github.com/mccutchen/strftime.org) for the build source code.

## See also

You might also like [PyFormat.info](#) or the interactive [strfti.me](#).

*Last updated Thu Jul 15 16:02:31 2021.*