

date Cheatsheet

By Dejan Panovski • Updated on Jun 23, 2026 • [Download PDF](#)

Quick reference for the Linux date command covering display formats, format specifiers, relative dates, timezones, Unix timestamps, and setting the system clock.

date is the Linux command for displaying and setting the system date and time. This cheatsheet covers default and custom output formats, the most common format specifiers, relative date strings, timezone overrides, Unix timestamp conversion, and manual clock changes.

Display Date and Time

Print the current date and time in common forms.

date	Current date and time in the default format
date +%F	Date in YYYY-MM-DD format
date +%T	Time in HH:MM:SS format
date +"%F %T"	Date and time combined
date -u	Current time in UTC
date -R	Date in RFC 5322 email format
date --iso-8601=seconds	ISO 8601 with seconds

Common Format Specifiers

Build custom output with `%` controls after a `+`.

%Y %m %d	Year, month, day of month
%H %M %S	Hour (24h), minute, second
%I %p	Hour (12h), AM/PM
%A %a	Full / abbreviated weekday name
%B %b	Full / abbreviated month name
%j %u	Day of year, day of week (Mon=1)
%F %T %D	YYYY-MM-DD, HH:MM:SS, mm/dd/yy
%s %n	Unix timestamp, newline

Format Examples

Combine specifiers into ready-to-use strings.

<code>date +"%Y-%m-%d"</code>	2026-06-23
<code>date +"%H:%M:%S"</code>	14:31:01
<code>LC_TIME=C date +"%A, %d %B %Y"</code>	Tuesday, 23 June 2026
<code>date +"%Y%m%d"</code>	Compact stamp for filenames
<code>date +"%Y-%m-%d_%H-%M-%S"</code>	Safe timestamp for logs
<code>date +"%Z (%Z)"</code>	Timezone name and offset

Relative Dates

Use `-d` (or `--date`) with a date string.

<code>date -d "tomorrow"</code>	Tomorrow's date
<code>date -d "yesterday"</code>	Yesterday's date
<code>date -d "monday"</code>	Upcoming Monday (today if it is Monday)
<code>date -d "last week"</code>	One week ago
<code>date -d "30 days ago"</code>	30 days in the past
<code>date -d "2 hours"</code>	Two hours from now
<code>date -d "16 Dec 1974" +%A</code>	Weekday of a fixed date

Timezones

Override the timezone for one invocation.

<code>TZ='Asia/Tokyo' date</code>	Show time in another timezone
<code>TZ='UTC' date</code>	Show current UTC time
<code>date -d 'TZ="Australia/Sydney" 09:00 next Mon'</code>	Convert a time across zones
<code>date +"%Z"</code>	Show the current timezone abbreviation
<code>date +"%z"</code>	Show the numeric UTC offset

Unix Timestamps

Convert between epoch seconds and dates.

<code>date +%s</code>	Current Unix timestamp
<code>date -d @1234567890</code>	Convert a timestamp to a date
<code>date -d @1234567890 +"%F %T"</code>	Convert and format a timestamp
<code>date -u -d @0</code>	Print the Unix epoch start
<code>date -d "2026-06-23 12:00" +%s</code>	Convert a date to a timestamp

Files and Scripts

Read file times and capture the date in scripts.

<code>date -r /etc/hosts</code>	Last modification time of a file
<code>backup-\$(date +%F).sql</code>	Date-stamped filename
<code>now=\$(date +%s)</code>	Store the timestamp in a variable
<code>date +"%F %T" >> log.txt</code>	Append a timestamp to a log
<code>date --debug -d "next fri"</code>	Show how a date string is parsed

Set the System Clock

Disable NTP before a manual change, then re-enable it afterward.

<code>sudo timedatectl set-ntp false</code>	Disable automatic time synchronization
<code>sudo date --set="2026-06-23 17:30"</code>	Set the date and time
<code>sudo date --set="17:30"</code>	Set the time only
<code>sudo date MMDDhhmmCCYY</code>	Set the clock with the legacy positional format
<code>sudo timedatectl set-time '2026-06-23 17:30:00'</code>	Preferred way on systemd systems
<code>sudo timedatectl set-ntp true</code>	Re-enable automatic time synchronization

Related Guides

Use these references for deeper date and time tasks.

date Command in Linux	Full walkthrough of formatting, timezones, and epoch conversion
timedatectl Cheatsheet	Manage system time, timezone, and NTP sync
Set or Change the Time Zone in Linux	Change the system timezone
crontab Cheatsheet	Schedule time-based jobs