yt-dlp / yt-dlp

A youtube-dl fork with additional features and fixes

discord.gg/h5mncfw63r

Unlicense License

376 stars 44 forks

Unstar Unwatch

Code Issues 18 Pull requests 4 Discussions Actions Projects 1

master

pukkandan [version] update ... 3 days ago 18,724

View code

README.md

YT-DLP

Release v2021.03.24.1 License Unlicense Core Tests passing discord 60 online

commits 106/month last commit last wednesday downloads 13k PyPi 1.2k/month docs passing

A command-line program to download videos from youtube.com and many other video platforms

This is a youtube-dl fork based on the now inactive youtube-dlc. The main focus of this project is adding new features and patches while also keeping up to date with the original project

NEW FEATURES

The major new features from the latest release of blackjack4494/yt-dlc are:

SponSkrub Integration: You can use SponSkrub to mark/remove sponsor sections in youtube videos by utilizing the SponsorBlock API

Format Sorting: The default format sorting options have been changed so that higher resolution and better codecs will be now preferred instead of simply using larger bitrate. Furthermore, you can now specify the sort order using `-S`. This allows for much easier format selection that what is possible by simply using `--format` (examples)

Merged with youtube-dl v2021.03.25: You get all the latest features and patches of youtube-dl in addition to all the features of youtube-dlc

Merged with animelover1984/youtube-dl: You get most of the features and improvements from animelover1984/youtube-dl including `--get-comments`, `BiliBiliSearch`, `BilibiliChannel`, Embedding thumbnail in mp4/ogg/opus, Playlist infojson etc. Note that the NicoNico improvements are not available. See #31 for details.

Youtube improvements:

- All Youtube Feeds (`:ytfav`, `:ytwatchlater`, `:ytsubs`, `:ythistory`, `:ytrec`) works correctly and supports downloading multiple pages of content
- Youtube search (`ytsearch:`, `ytsearchdate:`) along with Search URLs works correctly
- Redirect channel's home URL automatically to `/video` to preserve the old behaviour

Split video by chapters: Videos can be split into multiple files based on chapters using `--split-chapters`

Multithreaded fragment downloads: Fragment downloads can be natively multi-threaded. Use `--concurrent-fragments` (`-N`) option to set the number of threads used

Aria2c with HLS/DASH: You can use aria2c as the external downloader for DASH(mpd) and HLS(m3u8) formats. No more slow ffmpeg/native downloads

New extractors: AnimeLab, Philo MSO, Rcs, Gedi, bitwave.tv, mildom, audius, zee5, mtv.it, wimtv, pluto.tv

Fixed extractors: archive.org, roosterteeth.com, skyit, instagram, itv, SouthparkDe, spreaker, Vlive, tiktok, akamai, ina, rumble, tennistv, amcnetworks

Plugin extractors: Extractors can be loaded from an external file. See plugins for details

Multiple paths and output templates: You can give different output templates and download paths for different types of files. You can also set a temporary path where intermediary files are downloaded to using `--paths (-P)`

Portable Configuration: Configuration files are automatically loaded from the home and root directories. See configuration for details

Other new options: `--parse-metadata`, `--list-formats-as-table`, `--write-link`, `--force-download-archive`, `--force-overwrites`, `--break-on-reject` etc

Improvements: Multiple `--postprocessor-args` and `--external-downloader-args`, Date/time formatting in `-o`, faster archive checking, more format selection options etc

Self-updater: The releases can be updated using `yt-dlp -U`

See changelog or commits for the full list of changes

PS: Some of these changes are already in youtube-dlc, but are still unreleased. See this for details

If you are coming from youtube-dl, the amount of changes are very large. Compare options and supported sites with youtube-dl's to get an idea of the massive number of features/patches youtube-dlc has accumulated.

INSTALLATION

You can install yt-dlp using one of the following methods:

Download the binary from the latest release (recommended method)

Use PyPI package: `python -m pip install --upgrade yt-dlp`

Use pip+git: `python -m pip install --upgrade git+https://github.com/yt-dlp/yt-dlp.git@release`

Install master branch: `python -m pip install --upgrade git+https://github.com/yt-dlp/yt-dlp`

UNIX users (Linux, macOS, BSD) can also install the latest release one of the following ways:

```
sudo curl -L https://github.com/yt-dlp/yt-dlp/releases/latest/download/yt-dlp -o
/usr/local/bin/yt-dlp
sudo chmod a+rx /usr/local/bin/yt-dlp
```

```
sudo wget https://github.com/yt-dlp/yt-dlp/releases/latest/download/yt-dlp -O
/usr/local/bin/yt-dlp
sudo chmod a+rx /usr/local/bin/yt-dlp
```

```
sudo aria2c https://github.com/yt-dlp/yt-dlp/releases/latest/download/yt-dlp -o
/usr/local/bin/yt-dlp
sudo chmod a+rx /usr/local/bin/yt-dlp
```

UPDATE

Starting from version `2021.02.09`, you can use `yt-dlp -U` to update if you are using the provided release. If you are using `pip`, simply re-run the same command that was used to install the program.

COMPILE

For Windows: To build the Windows executable, you must have pyinstaller (and optionally mutagen and pycryptodome)

```
python -m pip install --upgrade pyinstaller mutagen pycryptodome
```

Once you have all the necessary dependancies installed, just run `py pyinst.py`. The executable will be built for the same architecture (32/64 bit) as the python used to build it. It is strongly reccomended to use python3 although python2.6+ is supported.

You can also build the executable without any version info or metadata by using:

```
pyinstaller.exe yt_dlp\__main__.py --onefile --name yt-dlp
```

For Unix: You will need the required build tools: `python`, `make` (GNU), `pandoc`, `zip`, `nosetests`
Then simply run `make`. You can also run `make yt-dlp` instead to compile only the binary without updating any of the additional files

Note: In either platform, `devscripts\update-version.py` can be used to automatically update the version number

DESCRIPTION

yt-dlp is a command-line program to download videos from youtube.com many other video platforms. It requires the Python interpreter, version 2.6, 2.7, or 3.2+, and it is not platform specific. It should work on your Unix box, on Windows or on macOS. It is released to the public domain, which means you can modify it, redistribute it or use it however you like.

```
yt-dlp [OPTIONS] [--] URL [URL...]
```

OPTIONS

`Ctrl+F` is your friend :D

General Options:

```
-h, --help                      Print this help text and exit
--version                       Print program version and exit
-U, --update                    Update this program to latest version. Make
                                sure that you have sufficient permissions
                                (run with sudo if needed)
-i, --ignore-errors             Continue on download errors, for example to
                                skip unavailable videos in a playlist
                                (default) (Alias: --no-abort-on-error)
--abort-on-error                Abort downloading of further videos if an
                                error occurs (Alias: --no-ignore-errors)
--dump-user-agent               Display the current browser identification
--list-extractors               List all supported extractors
--extractor-descriptions        Output descriptions of all supported
                                extractors
--force-generic-extractor       Force extraction to use the generic
                                extractor
--default-search PREFIX         Use this prefix for unqualified URLs. For
                                example "gvsearch2:" downloads two videos
                                from google videos for youtube-dl "large
                                apple". Use the value "auto" to let
                                youtube-dl guess ("auto_warning" to emit a
                                warning when guessing). "error" just throws
                                an error. The default value "fixup_error"
```

```
                                  repairs broken URLs, but emits an error if
                                  this is not possible instead of searching
--ignore-config, --no-config      Disable loading any configuration files
                                  except the one provided by --config-location.
                                  When given inside a configuration
                                  file, no further configuration files are
                                  loaded. Additionally, (for backward
                                  compatibility) if this option is found
                                  inside the system configuration file, the
                                  user configuration is not loaded
--config-location PATH            Location of the main configuration file;
                                  either the path to the config or its
                                  containing directory
--flat-playlist                   Do not extract the videos of a playlist,
                                  only list them
--no-flat-playlist                Extract the videos of a playlist
--mark-watched                    Mark videos watched (YouTube only)
--no-mark-watched                 Do not mark videos watched (default)
--no-colors                       Do not emit color codes in output
```

Network Options:

```
--proxy URL                       Use the specified HTTP/HTTPS/SOCKS proxy.
                                  To enable SOCKS proxy, specify a proper
                                  scheme. For example
                                  socks5://127.0.0.1:1080/. Pass in an empty
                                  string (--proxy "") for direct connection
--socket-timeout SECONDS          Time to wait before giving up, in seconds
--source-address IP               Client-side IP address to bind to
-4, --force-ipv4                  Make all connections via IPv4
-6, --force-ipv6                  Make all connections via IPv6
```

Geo Restriction:

```
--geo-verification-proxy URL      Use this proxy to verify the IP address for
                                  some geo-restricted sites. The default
                                  proxy specified by --proxy (or none, if the
                                  option is not present) is used for the
                                  actual downloading
--geo-bypass                      Bypass geographic restriction via faking
                                  X-Forwarded-For HTTP header
--no-geo-bypass                   Do not bypass geographic restriction via
                                  faking X-Forwarded-For HTTP header
--geo-bypass-country CODE         Force bypass geographic restriction with
                                  explicitly provided two-letter ISO 3166-2
```

```
                                  country code
--geo-bypass-ip-block IP_BLOCK    Force bypass geographic restriction with
                                  explicitly provided IP block in CIDR
                                  notation
```

Video Selection:

```
--playlist-start NUMBER           Playlist video to start at (default is 1)
--playlist-end NUMBER             Playlist video to end at (default is last)
--playlist-items ITEM_SPEC        Playlist video items to download. Specify
                                  indices of the videos in the playlist
                                  separated by commas like: "--playlist-items
                                  1,2,5,8" if you want to download videos
                                  indexed 1, 2, 5, 8 in the playlist. You can
                                  specify range: "--playlist-items
                                  1-3,7,10-13", it will download the videos
                                  at index 1, 2, 3, 7, 10, 11, 12 and 13
--match-title REGEX               Download only matching titles (regex or
                                  caseless sub-string)
--reject-title REGEX              Skip download for matching titles (regex or
                                  caseless sub-string)
--max-downloads NUMBER            Abort after downloading NUMBER files
--min-filesize SIZE               Do not download any videos smaller than
                                  SIZE (e.g. 50k or 44.6m)
--max-filesize SIZE               Do not download any videos larger than SIZE
                                  (e.g. 50k or 44.6m)
--date DATE                       Download only videos uploaded in this date.
                                  The date can be "YYYYMMDD" or in the format
                                  "(now|today)[+-][0-9](day|week|month|year)(s)?"
--datebefore DATE                 Download only videos uploaded on or before
                                  this date. The date formats accepted is the
                                  same as --date
--dateafter DATE                  Download only videos uploaded on or after
                                  this date. The date formats accepted is the
                                  same as --date
--min-views COUNT                 Do not download any videos with less than
                                  COUNT views
--max-views COUNT                 Do not download any videos with more than
                                  COUNT views
--match-filter FILTER             Generic video filter. Specify any key (see
                                  "OUTPUT TEMPLATE" for a list of available
                                  keys) to match if the key is present, !key
                                  to check if the key is not present,
                                  key>NUMBER (like "view_count > 12", also
                                  works with >=, <, <=, !=, =) to compare
                                  against a number, key = 'LITERAL' (like
                                  "uploader = 'Mike Smith'", also works with
                                  !=) to match against a string literal and &
```

```
                                  to require multiple matches. Values which
                                  are not known are excluded unless you put a
                                  question mark (?) after the operator. For
                                  example, to only match videos that have
                                  been liked more than 100 times and disliked
                                  less than 50 times (or the dislike
                                  functionality is not available at the given
                                  service), but who also have a description,
                                  use --match-filter "like_count > 100 &
                                  dislike_count <? 50 & description"
--no-match-filter                 Do not use generic video filter (default)
--no-playlist                     Download only the video, if the URL refers
                                  to a video and a playlist
--yes-playlist                    Download the playlist, if the URL refers to
                                  a video and a playlist
--age-limit YEARS                 Download only videos suitable for the given
                                  age
--download-archive FILE           Download only videos not listed in the
                                  archive file. Record the IDs of all
                                  downloaded videos in it
--break-on-existing               Stop the download process when encountering
                                  a file that is in the archive
--break-on-reject                 Stop the download process when encountering
                                  a file that has been filtered out
--no-download-archive             Do not use archive file (default)
--include-ads                     Download advertisements as well
                                  (experimental)
--no-include-ads                  Do not download advertisements (default)
```

Download Options:

```
-N, --concurrent-fragments N      Number of fragments to download
                                  concurrently (default is 1)
-r, --limit-rate RATE             Maximum download rate in bytes per second
                                  (e.g. 50K or 4.2M)
-R, --retries RETRIES             Number of retries (default is 10), or
                                  "infinite"
--fragment-retries RETRIES        Number of retries for a fragment (default
                                  is 10), or "infinite" (DASH, hlsnative and
                                  ISM)
--skip-unavailable-fragments      Skip unavailable fragments for DASH,
                                  hlsnative and ISM (default)
                                  (Alias: --no-abort-on-unavailable-fragment)
--abort-on-unavailable-fragment   Abort downloading if a fragment is unavailable
                                  (Alias: --no-skip-unavailable-fragments)
--keep-fragments                  Keep downloaded fragments on disk after
                                  downloading is finished
--no-keep-fragments               Delete downloaded fragments after
```

```
                                  downloading is finished (default)
    --buffer-size SIZE            Size of download buffer (e.g. 1024 or 16K)
                                  (default is 1024)
    --resize-buffer               The buffer size is automatically resized
                                  from an initial value of --buffer-size
                                  (default)
    --no-resize-buffer            Do not automatically adjust the buffer size
    --http-chunk-size SIZE        Size of a chunk for chunk-based HTTP
                                  downloading (e.g. 10485760 or 10M) (default
                                  is disabled). May be useful for bypassing
                                  bandwidth throttling imposed by a webserver
                                  (experimental)
    --playlist-reverse            Download playlist videos in reverse order
    --no-playlist-reverse         Download playlist videos in default order
                                  (default)
    --playlist-random             Download playlist videos in random order
    --xattr-set-filesize          Set file xattribute ytdl.filesize with
                                  expected file size
    --hls-prefer-native           Use the native HLS downloader instead of
                                  ffmpeg
    --hls-prefer-ffmpeg           Use ffmpeg instead of the native HLS
                                  downloader
    --hls-use-mpegts              Use the mpegts container for HLS videos;
                                  allowing some players to play the video
                                  while downloading, and reducing the chance
                                  of file corruption if download is
                                  interrupted. This is enabled by default for
                                  live streams
    --no-hls-use-mpegts           Do not use the mpegts container for HLS
                                  videos. This is default when not
                                  downloading live streams
    --external-downloader NAME    Name or path of the external downloader to
                                  use. Currently supports aria2c, avconv,
                                  axel, curl, ffmpeg, httpie, wget
                                  (Recommended: aria2c)
    --downloader-args NAME:ARGS   Give these arguments to the external
                                  downloader. Specify the downloader name and
                                  the arguments separated by a colon ":". You
                                  can use this option multiple times
                                  (Alias: --external-downloader-args)
```

Filesystem Options:

```
    -a, --batch-file FILE         File containing URLs to download ('-' for
                                  stdin), one URL per line. Lines starting
                                  with '#', ';' or ']' are considered as
                                  comments and ignored
    -P, --paths TYPE:PATH         The paths where the files should be
```

```
                                  downloaded. Specify the type of file and
                                  the path separated by a colon ":". All the
                                  same types as --output are supported.
                                  Additionally, you can also provide "home"
                                  and "temp" paths. All intermediary files
                                  are first downloaded to the temp path and
                                  then the final files are moved over to the
                                  home path after download is finished. This
                                  option is ignored if --output is an
                                  absolute path
-o, --output [TYPE:]TEMPLATE     Output filename template, see "OUTPUT
                                  TEMPLATE" for details
--output-na-placeholder TEXT     Placeholder value for unavailable meta
                                  fields in output filename template
                                  (default: "NA")
--autonumber-start NUMBER        Specify the start value for %(autonumber)s
                                  (default is 1)
--restrict-filenames             Restrict filenames to only ASCII
                                  characters, and avoid "&" and spaces in
                                  filenames
--no-restrict-filenames          Allow Unicode characters, "&" and spaces in
                                  filenames (default)
--windows-filenames              Force filenames to be windows compatible
--no-windows-filenames           Make filenames windows compatible only if
                                  using windows (default)
--trim-filenames LENGTH          Limit the filename length (excluding
                                  extension) to the specified number of
                                  characters
-w, --no-overwrites              Do not overwrite any files
--force-overwrites               Overwrite all video and metadata files.
                                  This option includes --no-continue
--no-force-overwrites            Do not overwrite the video, but overwrite
                                  related files (default)
-c, --continue                   Resume partially downloaded files/fragments
                                  (default)
--no-continue                    Do not resume partially downloaded
                                  fragments. If the file is unfragmented,
                                  restart download of the entire file
--part                           Use .part files instead of writing directly
                                  into output file (default)
--no-part                        Do not use .part files - write directly
                                  into output file
--mtime                          Use the Last-modified header to set the
                                  file modification time (default)
--no-mtime                       Do not use the Last-modified header to set
                                  the file modification time
--write-description              Write video description to a .description
                                  file
--no-write-description           Do not write video description (default)
--write-info-json                Write video metadata to a .info.json file
                                  (this may contain personal information)
```

```
--no-write-info-json            Do not write video metadata (default)
--write-annotations             Write video annotations to a
                                .annotations.xml file
--no-write-annotations          Do not write video annotations (default)
--write-playlist-metafiles      Write playlist metadata in addition to the
                                video metadata when using --write-info-json,
                                --write-description etc. (default)
--no-write-playlist-metafiles   Do not write playlist metadata when using
                                --write-info-json, --write-description etc.
--clean-infojson                Remove some private fields such as
                                filenames from the infojson. Note that it
                                could still contain some personal
                                information (default)
--no-clean-infojson             Write all fields to the infojson
--get-comments                  Retrieve video comments to be placed in the
                                .info.json file. The comments are fetched
                                even without this option if the extraction
                                is known to be quick
--load-info-json FILE           JSON file containing the video information
                                (created with the "--write-info-json"
                                option)
--cookies FILE                  File to read cookies from and dump cookie
                                jar in
--no-cookies                    Do not read/dump cookies (default)
--cache-dir DIR                 Location in the filesystem where youtube-dl
                                can store some downloaded information
                                permanently. By default
                                $XDG_CACHE_HOME/youtube-dl or
                                ~/.cache/youtube-dl . At the moment, only
                                YouTube player files (for videos with
                                obfuscated signatures) are cached, but that
                                may change
--no-cache-dir                  Disable filesystem caching
--rm-cache-dir                  Delete all filesystem cache files
```

Thumbnail Images:

```
--write-thumbnail               Write thumbnail image to disk
--no-write-thumbnail            Do not write thumbnail image to disk
                                (default)
--write-all-thumbnails          Write all thumbnail image formats to disk
--list-thumbnails               Simulate and list all available thumbnail
                                formats
```

Internet Shortcut Options:

```
--write-link                    Write an internet shortcut file, depending
                                on the current platform (.url, .webloc or
                                .desktop). The URL may be cached by the OS
--write-url-link                Write a .url Windows internet shortcut. The
                                OS caches the URL based on the file path
--write-webloc-link             Write a .webloc macOS internet shortcut
--write-desktop-link            Write a .desktop Linux internet shortcut
```

Verbosity and Simulation Options:

```
-q, --quiet                     Activate quiet mode
--no-warnings                   Ignore warnings
-s, --simulate                  Do not download the video and do not write
                                anything to disk
--skip-download                 Do not download the video but write all
                                related files (Alias: --no-download)
-g, --get-url                   Simulate, quiet but print URL
-e, --get-title                 Simulate, quiet but print title
--get-id                        Simulate, quiet but print id
--get-thumbnail                 Simulate, quiet but print thumbnail URL
--get-description               Simulate, quiet but print video description
--get-duration                  Simulate, quiet but print video length
--get-filename                  Simulate, quiet but print output filename
--get-format                    Simulate, quiet but print output format
-j, --dump-json                 Simulate, quiet but print JSON information.
                                See "OUTPUT TEMPLATE" for a description of
                                available keys
-J, --dump-single-json          Simulate, quiet but print JSON information
                                for each command-line argument. If the URL
                                refers to a playlist, dump the whole
                                playlist information in a single line
--print-json                    Be quiet and print the video information as
                                JSON (video is still being downloaded)
--force-write-archive           Force download archive entries to be
                                written as far as no errors occur,even if
                                -s or another simulation switch is used
                                (Alias: --force-download-archive)
--newline                       Output progress bar as new lines
--no-progress                   Do not print progress bar
--console-title                 Display progress in console titlebar
-v, --verbose                   Print various debugging information
--dump-pages                    Print downloaded pages encoded using base64
                                to debug problems (very verbose)
--write-pages                   Write downloaded intermediary pages to
                                files in the current directory to debug
                                problems
--print-traffic                 Display sent and read HTTP traffic
```

Workarounds:

```
--encoding ENCODING             Force the specified encoding (experimental)
--no-check-certificate          Suppress HTTPS certificate validation
--prefer-insecure               Use an unencrypted connection to retrieve
                                information about the video (Currently
                                supported only for YouTube)
--user-agent UA                 Specify a custom user agent
--referer URL                   Specify a custom referer, use if the video
                                access is restricted to one domain
--add-header FIELD:VALUE        Specify a custom HTTP header and its value,
                                separated by a colon ":". You can use this
                                option multiple times
--bidi-workaround               Work around terminals that lack
                                bidirectional text support. Requires bidiv
                                or fribidi executable in PATH
--sleep-requests SECONDS        Number of seconds to sleep between requests
                                during data extraction
--sleep-interval SECONDS        Number of seconds to sleep before each
                                download. This is the minimum time to sleep
                                when used along with --max-sleep-interval
                                (Alias: --min-sleep-interval)
--max-sleep-interval SECONDS    Maximum number of seconds to sleep. Can
                                only be used along with --min-sleep-interval
--sleep-subtitles SECONDS       Number of seconds to sleep before each
                                subtitle download
```

Video Format Options:

```
-f, --format FORMAT             Video format code, see "FORMAT SELECTION"
                                for more details
-S, --format-sort SORTORDER     Sort the formats by the fields given, see
                                "Sorting Formats" for more details
--S-force, --format-sort-force  Force user specified sort order to have
                                precedence over all fields, see "Sorting
                                Formats" for more details
--no-format-sort-force          Some fields have precedence over the user
                                specified sort order (default), see
                                "Sorting Formats" for more details
--video-multistreams            Allow multiple video streams to be merged
                                into a single file
--no-video-multistreams         Only one video stream is downloaded for
                                each output file (default)
--audio-multistreams            Allow multiple audio streams to be merged
                                into a single file
--no-audio-multistreams         Only one audio stream is downloaded for
                                each output file (default)
```

```
--all-formats                   Download all available video formats
--prefer-free-formats           Prefer video formats with free containers
                                over non-free ones of same quality. Use
                                with "-S ext" to strictly prefer free
                                containers irrespective of quality
--no-prefer-free-formats        Don't give any special preference to free
                                containers (default)
-F, --list-formats              List all available formats of requested
                                videos
--list-formats-as-table         Present the output of -F in tabular form
                                (default)
--list-formats-old              Present the output of -F in the old form
                                (Alias: --no-list-formats-as-table)
--merge-output-format FORMAT    If a merge is required (e.g.
                                bestvideo+bestaudio), output to given
                                container format. One of mkv, mp4, ogg,
                                webm, flv. Ignored if no merge is required
--allow-unplayable-formats      Allow unplayable formats to be listed and
                                downloaded. All video postprocessing will
                                also be turned off
--no-allow-unplayable-formats   Do not allow unplayable formats to be
                                listed or downloaded (default)
```

Subtitle Options:

```
--write-subs                    Write subtitle file
--no-write-subs                 Do not write subtitle file (default)
--write-auto-subs               Write automatically generated subtitle file
                                (Alias: --write-automatic-subs)
--no-write-auto-subs            Do not write auto-generated subtitles
                                (default) (Alias: --no-write-automatic-subs)
--all-subs                      Download all the available subtitles of the
                                video
--list-subs                     List all available subtitles for the video
--sub-format FORMAT             Subtitle format, accepts formats
                                preference, for example: "srt" or
                                "ass/srt/best"
--sub-langs LANGS               Languages of the subtitles to download
                                (optional) separated by commas, use --list-
                                subs for available language tags
```

Authentication Options:

```
-u, --username USERNAME         Login with this account ID
-p, --password PASSWORD         Account password. If this option is left
```

	out, yt-dlp will ask interactively
-2, --twofactor TWOFACTOR	Two-factor authentication code
-n, --netrc	Use .netrc authentication data
--video-password PASSWORD	Video password (vimeo, youku)

Adobe Pass Options:

--ap-mso MSO	Adobe Pass multiple-system operator (TV provider) identifier, use --ap-list-mso for a list of available MSOs
--ap-username USERNAME	Multiple-system operator account login
--ap-password PASSWORD	Multiple-system operator account password. If this option is left out, yt-dlp will ask interactively
--ap-list-mso	List all supported multiple-system operators

Post-Processing Options:

-x, --extract-audio	Convert video files to audio-only files (requires ffmpeg and ffprobe)
--audio-format FORMAT	Specify audio format: "best", "aac", "flac", "mp3", "m4a", "opus", "vorbis", or "wav"; "best" by default; No effect without -x
--audio-quality QUALITY	Specify ffmpeg audio quality, insert a value between 0 (better) and 9 (worse) for VBR or a specific bitrate like 128K (default 5)
--remux-video FORMAT	Remux the video into another container if necessary (currently supported: mp4|mkv|flv|webm|mov|avi|mp3|mka|m4a|ogg|opus). If target container does not support the video/audio codec, remuxing will fail. You can specify multiple rules; eg. "aac>m4a/mov>mp4/mkv" will remux aac to m4a, mov to mp4 and anything else to mkv.
--recode-video FORMAT	Re-encode the video into another format if re-encoding is necessary. The supported formats are the same as --remux-video
--postprocessor-args NAME:ARGS	Give these arguments to the postprocessors. Specify the postprocessor/executable name and the arguments separated by a colon ":" to give the argument to the specified postprocessor/executable. Supported

```
                                    postprocessors are: SponSkrub,
                                    ExtractAudio, VideoRemuxer, VideoConvertor,
                                    EmbedSubtitle, Metadata, Merger,
                                    FixupStretched, FixupM4a, FixupM3u8,
                                    SubtitlesConvertor, EmbedThumbnail and
                                    SplitChapters. The supported executables
                                    are: SponSkrub, FFmpeg, FFprobe, and
                                    AtomicParsley. You can also specify
                                    "PP+EXE:ARGS" to give the arguments to the
                                    specified executable only when being used
                                    by the specified postprocessor.
                                    Additionally, for ffmpeg/ffprobe, "_i"/"_o"
                                    can be appended to the prefix optionally
                                    followed by a number to pass the argument
                                    before the specified input/output file. Eg:
                                    --ppa "Merger+ffmpeg_i1:-v quiet". You can
                                    use this option multiple times to give
                                    different arguments to different
                                    postprocessors. (Alias: --ppa)
    -k, --keep-video                Keep the intermediate video file on disk
                                    after post-processing
    --no-keep-video                 Delete the intermediate video file after
                                    post-processing (default)
    --post-overwrites               Overwrite post-processed files (default)
    --no-post-overwrites            Do not overwrite post-processed files
    --embed-subs                    Embed subtitles in the video (only for mp4,
                                    webm and mkv videos)
    --no-embed-subs                 Do not embed subtitles (default)
    --embed-thumbnail               Embed thumbnail in the audio as cover art
    --no-embed-thumbnail            Do not embed thumbnail (default)
    --add-metadata                  Write metadata to the video file
    --no-add-metadata               Do not write metadata (default)
    --parse-metadata FIELD:FORMAT   Parse additional metadata like title/artist
                                    from other fields. Give a template or field
                                    name to extract data from and the format to
                                    interpret it as, seperated by a ":". Either
                                    regular expression with named capture
                                    groups or a similar syntax to the output
                                    template can be used for the FORMAT.
                                    Similarly, the syntax for output template
                                    can be used for FIELD to parse the data
                                    from multiple fields. The parsed parameters
                                    replace any existing values and can be used
                                    in output templates. This option can be
                                    used multiple times. Example: --parse-
                                    metadata "title:%(artist)s - %(title)s"
                                    matches a title like "Coldplay - Paradise".
                                    Example: --parse-metadata "%(series)s
                                    %(episode_number)s:%(title)s" sets the
                                    title using series and episode number.
                                    Example (regex): --parse-metadata
```

```
                                    "description:Artist - (?P<artist>.+?)"
    --xattrs                        Write metadata to the video file's xattrs
                                    (using dublin core and xdg standards)
    --fixup POLICY                  Automatically correct known faults of the
                                    file. One of never (do nothing), warn (only
                                    emit a warning), detect_or_warn (the
                                    default; fix file if we can, warn
                                    otherwise)
    --ffmpeg-location PATH          Location of the ffmpeg binary; either the
                                    path to the binary or its containing
                                    directory
    --exec CMD                      Execute a command on the file after
                                    downloading and post-processing, similar to
                                    find's -exec syntax. Example: --exec 'adb
                                    push {} /sdcard/Music/ && rm {}'
    --convert-subs FORMAT           Convert the subtitles to another format
                                    (currently supported: srt|ass|vtt|lrc)
                                    (Alias: --convert-subtitles)
    --split-chapters                Split video into multiple files based on
                                    internal chapters. The "chapter:" prefix
                                    can be used with "--paths" and "--output"
                                    to set the output filename for the split
                                    files. See "OUTPUT TEMPLATE" for details
    --no-split-chapters             Do not split video based on chapters
                                    (default)
```

SponSkrub (SponsorBlock) Options:

SponSkrub is a utility to mark/remove sponsor segments from downloaded YouTube videos using SponsorBlock API

```
    --sponskrub                     Use sponskrub to mark sponsored sections.
                                    This is enabled by default if the sponskrub
                                    binary exists (Youtube only)
    --no-sponskrub                  Do not use sponskrub
    --sponskrub-cut                 Cut out the sponsor sections instead of
                                    simply marking them
    --no-sponskrub-cut              Simply mark the sponsor sections, not cut
                                    them out (default)
    --sponskrub-force               Run sponskrub even if the video was already
                                    downloaded
    --no-sponskrub-force            Do not cut out the sponsor sections if the
                                    video was already downloaded (default)
    --sponskrub-location PATH       Location of the sponskrub binary; either
                                    the path to the binary or its containing
                                    directory
```

Extractor Options:

```
--extractor-retries RETRIES      Number of retries for known extractor
                                 errors (default is 3), or "infinite"
--allow-dynamic-mpd              Process dynamic DASH manifests (default)
                                 (Alias: --no-ignore-dynamic-mpd)
--ignore-dynamic-mpd             Do not process dynamic DASH manifests
                                 (Alias: --no-allow-dynamic-mpd)
--hls-split-discontinuity        Split HLS playlists to different formats at
                                 discontinuities such as ad breaks
--no-hls-split-discontinuity     Do not split HLS playlists to different
                                 formats at discontinuities such as ad
                                 breaks (default)
--youtube-include-dash-manifest  Download the DASH manifests and related
                                 data on YouTube videos (default)
                                 (Alias: --no-youtube-skip-dash-manifest)
--youtube-skip-dash-manifest     Do not download the DASH manifests and
                                 related data on YouTube videos
                                 (Alias: --no-youtube-include-dash-manifest)
--youtube-include-hls-manifest   Download the HLS manifests and related data
                                 on YouTube videos (default)
                                 (Alias: --no-youtube-skip-hls-manifest)
--youtube-skip-hls-manifest      Do not download the HLS manifests and
                                 related data on YouTube videos
                                 (Alias: --no-youtube-include-hls-manifest)
```

CONFIGURATION

You can configure yt-dlp by placing any supported command line option to a configuration file. The configuration is loaded from the following locations:

1. **Main Configuration**: The file given by `--config-location`
2. **Portable Configuration**: `yt-dlp.conf` in the same directory as the bundled binary. If you are running from source-code (`<root dir>/yt_dlp/__main__.py`), the root directory is used instead.
3. **Home Configuration**: `yt-dlp.conf` in the home path given by `-P "home:<path>"` , or in the current directory if no such path is given
4. **User Configuration**:

    `%XDG_CONFIG_HOME%/yt-dlp/config` (recommended on Linux/macOS)

    `%XDG_CONFIG_HOME%/yt-dlp.conf`

`%APPDATA%/yt-dlp/config` (recommended on Windows)

`%APPDATA%/yt-dlp/config.txt`

`~/yt-dlp.conf`

`~/yt-dlp.conf.txt`

Note that `~` points to `C:\Users\<user name>` on windows. Also, `%XDG_CONFIG_HOME%` defaults to `~/.config` if undefined

5. **System Configuration**: `/etc/yt-dlp.conf` or `/etc/yt-dlp.conf`

For example, with the following configuration file yt-dlp will always extract the audio, not copy the mtime, use a proxy and save all videos under `YouTube` directory in your home directory:

```
# Lines starting with # are comments

# Always extract audio
-x

# Do not copy the mtime
--no-mtime

# Use this proxy
--proxy 127.0.0.1:3128

# Save all videos under YouTube directory in your home directory
-o ~/YouTube/%(title)s.%(ext)s
```

Note that options in configuration file are just the same options aka switches used in regular command line calls; thus there **must be no whitespace** after `-` or `--`, e.g. `-o` or `--proxy` but not `- o` or `-- proxy`.

You can use `--ignore-config` if you want to disable all configuration files for a particular yt-dlp run. If `--ignore-config` is found inside any configuration file, no further configuration will be loaded. For example, having the option in the portable configuration file prevents loading of user and system configurations. Additionally, (for backward compatibility) if `--ignore-config` is found inside the system configuration file, the user configuration is not loaded.

Authentication with `.netrc` file

You may also want to configure automatic credentials storage for extractors that support authentication (by providing login and password with `--username` and `--password`) in order not to pass credentials as command line arguments on every yt-dlp execution and prevent tracking plain text passwords in the shell command history. You can achieve this using a `.netrc` file on a per extractor basis. For that you will need to create a `.netrc` file in your `$HOME` and restrict permissions to read/write by only you:

```
touch $HOME/.netrc
chmod a-rwx,u+rw $HOME/.netrc
```

After that you can add credentials for an extractor in the following format, where *extractor* is the name of the extractor in lowercase:

```
machine <extractor> login <login> password <password>
```

For example:

```
machine youtube login myaccount@gmail.com password my_youtube_password
machine twitch login my_twitch_account_name password my_twitch_password
```

To activate authentication with the `.netrc` file you should pass `--netrc` to yt-dlp or place it in the configuration file.

On Windows you may also need to setup the `%HOME%` environment variable manually. For example:

```
set HOME=%USERPROFILE%
```

OUTPUT TEMPLATE

The `-o` option is used to indicate a template for the output file names while `-P` option is used to specify the path each type of file should be saved to.

tl;dr: navigate me to examples.

The basic usage of `-o` is not to set any template arguments when downloading a single file, like in `yt-dlp -o funny_video.flv "https://some/video"` (hard-coding file extension like this is not recommended). However, it may contain special sequences that will be replaced when downloading each video. The special sequences may be formatted according to python string formatting operations. For example, `%(NAME)s` or `%(NAME)05d`. To clarify, that is a percent symbol followed by a name in parentheses, followed by formatting operations. Date/time fields can also be formatted according to strftime formatting by specifying it inside the parantheses seperated from the field name using a `>`. For example, `%(duration>%H-%M-%S)s`.

Additionally, you can set different output templates for the various metadata files seperately from the general output template by specifying the type of file followed by the template seperated by a colon ":". The different filetypes supported are `subtitle`, `thumbnail`, `description`, `annotation`, `infojson`, `pl_description`, `pl_infojson`, `chapter`. For example, `-o '%(title)s.%(ext)s' -o 'thumbnail:%(title)s\%(title)s.%(ext)s'` will put the thumbnails in a folder with the same name as the video.

The available fields are:

- `id` (string): Video identifier
- `title` (string): Video title
- `url` (string): Video URL
- `ext` (string): Video filename extension
- `alt_title` (string): A secondary title of the video
- `description` (string): The description of the video
- `display_id` (string): An alternative identifier for the video
- `uploader` (string): Full name of the video uploader
- `license` (string): License name the video is licensed under
- `creator` (string): The creator of the video
- `release_date` (string): The date (YYYYMMDD) when the video was released
- `timestamp` (numeric): UNIX timestamp of the moment the video became available
- `upload_date` (string): Video upload date (YYYYMMDD)
- `uploader_id` (string): Nickname or id of the video uploader
- `channel` (string): Full name of the channel the video is uploaded on
- `channel_id` (string): Id of the channel
- `location` (string): Physical location where the video was filmed
- `duration` (numeric): Length of the video in seconds
- `duration_string` (string): Length of the video (HH-mm-ss)

`view_count` (numeric): How many users have watched the video on the platform

`like_count` (numeric): Number of positive ratings of the video

`dislike_count` (numeric): Number of negative ratings of the video

`repost_count` (numeric): Number of reposts of the video

`average_rating` (numeric): Average rating give by users, the scale used depends on the webpage

`comment_count` (numeric): Number of comments on the video (For some extractors, comments are only downloaded at the end, and so this field cannot be used)

`age_limit` (numeric): Age restriction for the video (years)

`is_live` (boolean): Whether this video is a live stream or a fixed-length video

`was_live` (boolean): Whether this video was originally a live stream

`playable_in_embed` (string): Whether this video is allowed to play in embedded players on other sites

`availability` (string): Whether the video is 'private', 'premium_only', 'subscriber_only', 'needs_auth', 'unlisted' or 'public'

`start_time` (numeric): Time in seconds where the reproduction should start, as specified in the URL

`end_time` (numeric): Time in seconds where the reproduction should end, as specified in the URL

`format` (string): A human-readable description of the format

`format_id` (string): Format code specified by `--format`

`format_note` (string): Additional info about the format

`width` (numeric): Width of the video

`height` (numeric): Height of the video

`resolution` (string): Textual description of width and height

`tbr` (numeric): Average bitrate of audio and video in KBit/s

`abr` (numeric): Average audio bitrate in KBit/s

`acodec` (string): Name of the audio codec in use

`asr` (numeric): Audio sampling rate in Hertz

`vbr` (numeric): Average video bitrate in KBit/s

`fps` (numeric): Frame rate

`vcodec` (string): Name of the video codec in use

`container` (string): Name of the container format

`filesize` (numeric): The number of bytes, if known in advance

`filesize_approx` (numeric): An estimate for the number of bytes

`protocol` (string): The protocol that will be used for the actual download

`extractor` (string): Name of the extractor

`extractor_key` (string): Key name of the extractor

`epoch` (numeric): Unix epoch when creating the file

`autonumber` (numeric): Number that will be increased with each download, starting at `--autonumber-start`

`playlist` (string): Name or id of the playlist that contains the video

`playlist_index` (numeric): Index of the video in the playlist padded with leading zeros according to the total length of the playlist

`playlist_id` (string): Playlist identifier

`playlist_title` (string): Playlist title

`playlist_uploader` (string): Full name of the playlist uploader

`playlist_uploader_id` (string): Nickname or id of the playlist uploader

Available for the video that belongs to some logical chapter or section:

`chapter` (string): Name or title of the chapter the video belongs to

`chapter_number` (numeric): Number of the chapter the video belongs to

`chapter_id` (string): Id of the chapter the video belongs to

Available for the video that is an episode of some series or programme:

`series` (string): Title of the series or programme the video episode belongs to

`season` (string): Title of the season the video episode belongs to

`season_number` (numeric): Number of the season the video episode belongs to

`season_id` (string): Id of the season the video episode belongs to

`episode` (string): Title of the video episode

`episode_number` (numeric): Number of the video episode within a season

`episode_id` (string): Id of the video episode

Available for the media that is a track or a part of a music album:

`track` (string): Title of the track

`track_number` (numeric): Number of the track within an album or a disc

`track_id` (string): Id of the track

`artist` (string): Artist(s) of the track

`genre` (string): Genre(s) of the track

`album` (string): Title of the album the track belongs to

`album_type` (string): Type of the album

`album_artist` (string): List of all artists appeared on the album

`disc_number` (numeric): Number of the disc or other physical medium the track belongs to

`release_year` (numeric): Year (YYYY) when the album was released

Available when using `--split-chapters` for videos with internal chapters:

`section_title` (string): Title of the chapter

`section_number` (numeric): Number of the chapter within the file

`section_start` (numeric): Start time of the chapter in seconds

`section_end` (numeric): End time of the chapter in seconds

Each aforementioned sequence when referenced in an output template will be replaced by the actual value corresponding to the sequence name. Note that some of the sequences are not guaranteed to be present since they depend on the metadata obtained by a particular extractor. Such sequences will be replaced with placeholder value provided with `--output-na-placeholder` (`NA` by default).

For example for `-o %(title)s-%(id)s.%(ext)s` and an mp4 video with title `yt-dlp test video` and id `BaW_jenozKcj`, this will result in a `yt-dlp test video-BaW_jenozKcj.mp4` file created in the current directory.

For numeric sequences you can use numeric related formatting, for example, `%(view_count)05d` will result in a string with view count padded with zeros up to 5 characters, like in `00042`.

Output templates can also contain arbitrary hierarchical path, e.g. `-o '%(playlist)s/%(playlist_index)s - %(title)s.%(ext)s'` which will result in downloading each video in a directory corresponding to this path template. Any missing directory will be automatically created for you.

To use percent literals in an output template use `%%`. To output to stdout use `-o -`.

The current default template is `%(title)s [%(id)s].%(ext)s`.

In some cases, you don't want special characters such as 中, spaces, or &, such as when transferring the downloaded filename to a Windows system or the filename through an 8bit-unsafe channel. In these cases, add the `--restrict-filenames` flag to get a shorter title:

Output template and Windows batch files

If you are using an output template inside a Windows batch file then you must escape plain percent characters (`%`) by doubling, so that `-o "%(title)s-%(id)s.%(ext)s"` should become `-o "%%(title)s-%%(id)s.%%(ext)s"` . However you should not touch `%` 's that are not plain characters, e.g. environment variables for expansion should stay intact: `-o "C:\%HOMEPATH%\Desktop\%%(title)s.%%(ext)s"` .

Output template examples

Note that on Windows you need to use double quotes instead of single.

```
$ yt-dlp --get-filename -o '%(title)s.%(ext)s' BaW_jenozKc
youtube-dl test video ''_ä↭𝕐.mp4    # All kinds of weird characters

$ yt-dlp --get-filename -o '%(title)s.%(ext)s' BaW_jenozKc --restrict-filenames
youtube-dl_test_video_.mp4          # A simple file name

# Download YouTube playlist videos in separate directory indexed by video order in a
$ yt-dlp -o '%(playlist)s/%(playlist_index)s - %(title)s.%(ext)s' https://www.youtub

# Download YouTube playlist videos in seperate directories according to their upload
$ yt-dlp -o '%(upload_date>%Y)s/%(title)s.%(ext)s' https://www.youtube.com/playlist?

# Download all playlists of YouTube channel/user keeping each playlist in separate d
$ yt-dlp -o '%(uploader)s/%(playlist)s/%(playlist_index)s - %(title)s.%(ext)s' https

# Download Udemy course keeping each chapter in separate directory under MyVideos di
$ yt-dlp -u user -p password -P '~/MyVideos' -o '%(playlist)s/%(chapter_number)s - %

# Download entire series season keeping each series and each season in separate dire
$ yt-dlp -P "C:/MyVideos" -o "%(series)s/%(season_number)s - %(season)s/%(episode_nu

# Stream the video being downloaded to stdout
$ yt-dlp -o - BaW_jenozKc
```

FORMAT SELECTION

By default, yt-dlp tries to download the best available quality if you **don't** pass any options. This is generally equivalent to using `-f bestvideo*+bestaudio/best` . However, if multiple audiostreams is enabled (`--audio-multistreams`), the default format changes to `-f bestvideo+bestaudio/best` . Similarly, if ffmpeg is unavailable, or if you use yt-dlp to stream to `stdout` (`-o -`), the default becomes `-f best/bestvideo+bestaudio` .

The general syntax for format selection is `--f FORMAT` (or `--format FORMAT`) where `FORMAT` is a *selector expression*, i.e. an expression that describes format or formats you would like to download.

tl;dr: navigate me to examples.

The simplest case is requesting a specific format, for example with `-f 22` you can download the format with format code equal to 22. You can get the list of available format codes for particular video using `--list-formats` or `-F`. Note that these format codes are extractor specific.

You can also use a file extension (currently `3gp`, `aac`, `flv`, `m4a`, `mp3`, `mp4`, `ogg`, `wav`, `webm` are supported) to download the best quality format of a particular file extension served as a single file, e.g. `-f webm` will download the best quality format with the `webm` extension served as a single file.

You can also use special names to select particular edge case formats:

- `all`: Select all formats
- `b*`, `best*`: Select the best quality format irrespective of whether it contains video or audio.
- `w*`, `worst*`: Select the worst quality format irrespective of whether it contains video or audio.
- `b`, `best`: Select the best quality format that contains both video and audio. Equivalent to `best*[vcodec!=none][acodec!=none]`
- `w`, `worst`: Select the worst quality format that contains both video and audio. Equivalent to `worst*[vcodec!=none][acodec!=none]`
- `bv`, `bestvideo`: Select the best quality video-only format. Equivalent to `best*[acodec=none]`
- `wv`, `worstvideo`: Select the worst quality video-only format. Equivalent to `worst*[acodec=none]`
- `bv*`, `bestvideo*`: Select the best quality format that contains video. It may also contain audio. Equivalent to `best*[vcodec!=none]`
- `wv*`, `worstvideo*`: Select the worst quality format that contains video. It may also contain audio. Equivalent to `worst*[vcodec!=none]`
- `ba`, `bestaudio`: Select the best quality audio-only format. Equivalent to `best*[vcodec=none]`
- `wa`, `worstaudio`: Select the worst quality audio-only format. Equivalent to `worst*[vcodec=none]`

`ba*`, `bestaudio*`: Select the best quality format that contains audio. It may also contain video. Equivalent to `best*[acodec!=none]`

`wa*`, `worstaudio*`: Select the worst quality format that contains audio. It may also contain video. Equivalent to `worst*[acodec!=none]`

For example, to download the worst quality video-only format you can use `-f worstvideo`. It is however recomended to never actually use `worst` and related options. When your format selector is `worst`, the format which is worst in all respects is selected. Most of the time, what you actually want is the video with the smallest filesize instead. So it is generally better to use `-f best -S +size,+br,+res,+fps` instead of `-f worst`. See sorting formats for more details.

If you want to download multiple videos and they don't have the same formats available, you can specify the order of preference using slashes. Note that formats on the left hand side are preferred, for example `-f 22/17/18` will download format 22 if it's available, otherwise it will download format 17 if it's available, otherwise it will download format 18 if it's available, otherwise it will complain that no suitable formats are available for download.

If you want to download several formats of the same video use a comma as a separator, e.g. `-f 22,17,18` will download all these three formats, of course if they are available. Or a more sophisticated example combined with the precedence feature: `-f 136/137/mp4/bestvideo,140/m4a/bestaudio`.

You can merge the video and audio of multiple formats into a single file using `-f <format1>+<format2>+...` (requires ffmpeg installed), for example `-f bestvideo+bestaudio` will download the best video-only format, the best audio-only format and mux them together with ffmpeg. If `--no-video-multistreams` is used, all formats with a video stream except the first one are ignored. Similarly, if `--no-audio-multistreams` is used, all formats with an audio stream except the first one are ignored. For example, `-f bestvideo+best+bestaudio` will download and merge all 3 given formats. The resulting file will have 2 video streams and 2 audio streams. But `-f bestvideo+best+bestaudio --no-video-multistreams` will download and merge only `bestvideo` and `bestaudio`. `best` is ignored since another format containing a video stream (`bestvideo`) has already been selected. The order of the formats is therefore important. `-f best+bestaudio --no-audio-multistreams` will download and merge both formats while `-f bestaudio+best --no-audio-multistreams` will ignore `best` and download only `bestaudio`.

Filtering Formats

You can also filter the video formats by putting a condition in brackets, as in `-f "best[height=720]"` (or `-f "[filesize>10M]"`).

The following numeric meta fields can be used with comparisons `<`, `<=`, `>`, `>=`, `=` (equals), `!=` (not equals):

`filesize` : The number of bytes, if known in advance

`width` : Width of the video, if known

`height` : Height of the video, if known

`tbr` : Average bitrate of audio and video in KBit/s

`abr` : Average audio bitrate in KBit/s

`vbr` : Average video bitrate in KBit/s

`asr` : Audio sampling rate in Hertz

`fps` : Frame rate

Also filtering work for comparisons `=` (equals), `^=` (starts with), `$=` (ends with), `*=` (contains) and following string meta fields:

`ext` : File extension

`acodec` : Name of the audio codec in use

`vcodec` : Name of the video codec in use

`container` : Name of the container format

`protocol` : The protocol that will be used for the actual download, lower-case (`http`, `https`, `rtsp`, `rtmp`, `rtmpe`, `mms`, `f4m`, `ism`, `http_dash_segments`, `m3u8`, or `m3u8_native`)

`format_id` : A short description of the format

`language` : Language code

Any string comparison may be prefixed with negation `!` in order to produce an opposite comparison, e.g. `!*=` (does not contain).

Note that none of the aforementioned meta fields are guaranteed to be present since this solely depends on the metadata obtained by particular extractor, i.e. the metadata offered by the video hoster. Any other field made available by the extractor can also be used for filtering.

Formats for which the value is not known are excluded unless you put a question mark (`?`) after the operator. You can combine format filters, so `-f "[height<=?720][tbr>500]"` selects up to 720p videos (or videos where the height is not known) with a bitrate of at least 500 KBit/s. You can also use the filters with `all` to download all formats that satisfy the filter. For example, `-f "all[vcodec=none]"` selects all audio-only formats.

Format selectors can also be grouped using parentheses, for example if you want to download the best mp4 and webm formats with a height lower than 480 you can use `-f '(mp4,webm)[height<480]'`.

Sorting Formats

You can change the criteria for being considered the `best` by using `-S` (`--format-sort`). The general format for this is `--format-sort field1,field2...`. The available fields are:

- `hasvid`: Gives priority to formats that has a video stream
- `hasaud`: Gives priority to formats that has a audio stream
- `ie_pref`: The format preference as given by the extractor
- `lang`: Language preference as given by the extractor
- `quality`: The quality of the format as given by the extractor
- `source`: Preference of the source as given by the extractor
- `proto`: Protocol used for download (`https` / `ftps` > `http` / `ftp` > `m3u8-native` > `m3u8` > `http-dash-segments` > other > `mms` / `rtsp` > unknown > `f4f` / `f4m`)
- `vcodec`: Video Codec (`av01` > `vp9.2` > `vp9` > `h265` > `h264` > `vp8` > `h263` > `theora` > other > unknown)
- `acodec`: Audio Codec (`opus` > `vorbis` > `aac` > `mp4a` > `mp3` > `ac3` > `dts` > other > unknown)
- `codec`: Equivalent to `vcodec,acodec`
- `vext`: Video Extension (`mp4` > `webm` > `flv` > other > unknown). If `--prefer-free-formats` is used, `webm` is prefered.
- `aext`: Audio Extension (`m4a` > `aac` > `mp3` > `ogg` > `opus` > `webm` > other > unknown). If `--prefer-free-formats` is used, the order changes to `opus` > `ogg` > `webm` > `m4a` > `mp3` > `aac`.
- `ext`: Equivalent to `vext,aext`
- `filesize`: Exact filesize, if know in advance. This will be unavailable for mu38 and DASH formats.
- `fs_approx`: Approximate filesize calculated from the manifests
- `size`: Exact filesize if available, otherwise approximate filesize
- `height`: Height of video
- `width`: Width of video
- `res`: Video resolution, calculated as the smallest dimension.
- `fps`: Framerate of video
- `tbr`: Total average bitrate in KBit/s

`vbr` : Average video bitrate in KBit/s

`abr` : Average audio bitrate in KBit/s

`br` : Equivalent to using `tbr,vbr,abr`

`asr` : Audio sample rate in Hz

Note that any other **numerical** field made available by the extractor can also be used. All fields, unless specified otherwise, are sorted in decending order. To reverse this, prefix the field with a `+` . Eg: `+res` prefers format with the smallest resolution. Additionally, you can suffix a prefered value for the fields, seperated by a `:` . Eg: `res:720` prefers larger videos, but no larger than 720p and the smallest video if there are no videos less than 720p. For `codec` and `ext` , you can provide two prefered values, the first for video and the second for audio. Eg: `+codec:avc:m4a` (equivalent to `+vcodec:avc,+acodec:m4a`) sets the video codec preference to `h264 > h265 > vp9 > vp9.2 > av01 > vp8 > h263 > theora` and audio codec preference to `mp4a > aac > vorbis > opus > mp3 > ac3 > dts` . You can also make the sorting prefer the nearest values to the provided by using `~` as the delimiter. Eg: `filesize~1G` prefers the format with filesize closest to 1 GiB.

The fields `hasvid` , `ie_pref` , `lang` are always given highest priority in sorting, irrespective of the user-defined order. This behaviour can be changed by using `--force-format-sort` . Apart from these, the default order used is: `quality,res,fps,codec:vp9.2,size,br,asr,proto,ext,hasaud,source,id` . Note that the extractors may override this default order, but they cannot override the user-provided order.

If your format selector is `worst` , the last item is selected after sorting. This means it will select the format that is worst in all repects. Most of the time, what you actually want is the video with the smallest filesize instead. So it is generally better to use `-f best -S +size,+br,+res,+fps` .

Tip: You can use the `-v -F` to see how the formats have been sorted (worst to best).

Format Selection examples

Note that on Windows you may need to use double quotes instead of single.

```
# Download and merge the best best video-only format and the best audio-only format,
# or download the best combined format if video-only format is not available
$ yt-dlp -f 'bv+ba/b'

# Download best format that contains video,
# and if it doesn't already have an audio stream, merge it with best audio-only form
$ yt-dlp -f 'bv*+ba/b'
```

```
# Same as above
$ yt-dlp

# Download the best video-only format and the best audio-only format without merging
# For this case, an output template should be used since
# by default, bestvideo and bestaudio will have the same file name.
$ yt-dlp -f 'bv,ba' -o '%(title)s.f%(format_id)s.%(ext)s'



# The following examples show the old method (without -S) of format selection
# and how to use -S to achieve a similar but better result

# Download the worst video available (old method)
$ yt-dlp -f 'wv*+wa/w'

# Download the best video available but with the smallest resolution
$ yt-dlp -S '+res'

# Download the smallest video available
$ yt-dlp -S '+size,+br'



# Download the best mp4 video available, or the best video if no mp4 available
$ yt-dlp -f 'bv*[ext=mp4]+ba[ext=m4a]/b[ext=mp4] / bv*+ba/b'

# Download the best video with the best extension
# (For video, mp4 > webm > flv. For audio, m4a > aac > mp3 ...)
$ yt-dlp -S 'ext'



# Download the best video available but no better than 480p,
# or the worst video if there is no video under 480p
$ yt-dlp -f 'bv*[height<=480]+ba/b[height<=480] / wv*+ba/w'

# Download the best video available with the largest height but no better than 480p,
# or the best video with the smallest resolution if there is no video under 480p
$ yt-dlp -S 'height:480'

# Download the best video available with the largest resolution but no better than 4
# or the best video with the smallest resolution if there is no video under 480p
# Resolution is determined by using the smallest dimension.
# So this works correctly for vertical videos as well
$ yt-dlp -S 'res:480'



# Download the best video (that also has audio) but no bigger than 50 MB,
```

```
# or the worst video (that also has audio) if there is no video under 50 MB
$ yt-dlp -f 'b[filesize<50M] / w'

# Download largest video (that also has audio) but no bigger than 50 MB,
# or the smallest video (that also has audio) if there is no video under 50 MB
$ yt-dlp -f 'b' -S 'filesize:50M'

# Download best video (that also has audio) that is closest in size to 50 MB
$ yt-dlp -f 'b' -S 'filesize~50M'



# Download best video available via direct link over HTTP/HTTPS protocol,
# or the best video available via any protocol if there is no such video
$ yt-dlp -f '(bv*+ba/b)[protocol^=http][protocol!*=dash] / (bv*+ba/b)'

# Download best video available via the best protocol
# (https/ftps > http/ftp > m3u8_native > m3u8 > http_dash_segments ...)
$ yt-dlp -S 'proto'



# Download the best video with h264 codec, or the best video if there is no such vid
$ yt-dlp -f '(bv*+ba/b)[vcodec^=avc1] / (bv*+ba/b)'

# Download the best video with best codec no better than h264,
# or the best video with worst codec if there is no such video
$ yt-dlp -S 'codec:h264'

# Download the best video with worst codec no worse than h264,
# or the best video with best codec if there is no such video
$ yt-dlp -S '+codec:h264'



# More complex examples

# Download the best video no better than 720p prefering framerate greater than 30,
# or the worst video (still prefering framerate greater than 30) if there is no such
$ yt-dlp -f '((bv*[fps>30]/bv*)[height<=720]/(wv*[fps>30]/wv*)) + ba / (b[fps>30]/b)

# Download the video with the largest resolution no better than 720p,
# or the video with the smallest resolution available  if there is no such video,
# prefering larger framerate for formats with the same resolution
$ yt-dlp -S 'res:720,fps'



# Download the video with smallest resolution no worse than 480p,
# or the video with the largest resolution available if there is no such video,
```

```
# prefering better codec and then larger total bitrate for the same resolution
$ yt-dlp -S '+res:480,codec,br'
```

PLUGINS

Plugins are loaded from `<root-dir>/ytdlp_plugins/<type>/__init__.py`. Currently only `extractor` plugins are supported. Support for `downloader` and `postprocessor` plugins may be added in the future. See ytdlp_plugins for example.

Note: `<root-dir>` is the directory of the binary (`<root-dir>/yt-dlp`), or the root directory of the module if you are running directly from source-code (`<root dir>/yt_dlp/__main__.py`)

DEPRECATED OPTIONS

These are all the deprecated options and the current alternative to achieve the same effect

```
--cn-verification-proxy URL      --geo-verification-proxy URL
--id                             -o "%(id)s.%(ext)s"
-A, --auto-number                -o "%(autonumber)s-%(id)s.%(ext)s"
-t, --title                      -o "%(title)s-%(id)s.%(ext)s"
-l, --literal                    -o accepts literal names
--autonumber-size NUMBER         Use string formatting. Eg: %(autonumber)03d
--metadata-from-title FORMAT     --parse-metadata "title:FORMAT"
--prefer-avconv                  avconv is no longer officially supported (Alias:
--no-prefer-ffmpeg)
--prefer-ffmpeg                  Default (Alias: --no-prefer-avconv)
--avconv-location                avconv is no longer officially supported
-C, --call-home                  Not implemented
--no-call-home                   Default
--write-srt                      --write-subs
--no-write-srt                   --no-write-subs
--srt-lang LANGS                 --sub-langs LANGS
--prefer-unsecure                --prefer-insecure
--rate-limit RATE                --limit-rate RATE
--force-write-download-archive   --force-write-archive
--dump-intermediate-pages        --dump-pages
--dump-headers                   --print-traffic
--youtube-print-sig-code         No longer supported
--trim-file-names LENGTH         --trim-filenames LENGTH
--yes-overwrites                 --force-overwrites
--load-info                      --load-info-json
--split-tracks                   --split-chapters
```

```
--no-split-tracks               --no-split-chapters
--sponskrub-args ARGS           --ppa "sponskrub:ARGS"
--test                          Only used for testing extractors
```

MORE

For FAQ, Developer Instructions etc., see the original README

Releases 25

yt-dlp 2021.03.24.1 Latest
3 days ago

+ 24 releases

Used by 4

Contributors 795

+ 784 contributors

Languages

- Python 99.7%
- Other 0.3%