tqdm.contrib

Thin wrappers around common functions.

Subpackages contain potentially unstable extensions.

tenumerate

[view source]

```
def tenumerate(iterable,
          start=0,
          total=None,
          tqdm_class=tqdm_auto,
          **tqdm_kwargs)
```

Equivalent of `numpy.ndenumerate` or builtin `enumerate` .

Parameters

- **tqdm_class***: [default: tqdm.auto.tqdm].*

tzip

[view source]

```
def tzip(iter1, *iter2plus, **tqdm_kwargs)
```

Equivalent of builtin `zip` .

Parameters

- **tqdm_class***: [default: tqdm.auto.tqdm].*

tmap

[view source]

```
def tmap(function, *sequences, **tqdm_kwargs)
```

Equivalent of builtin `map` .

Parameters

- **tqdm_class***: [default: tqdm.auto.tqdm].*

tqdm.contrib.concurrent

Thin wrappers around `concurrent.futures`.

ensure_lock

[view source]

```
@contextmanager
def ensure_lock(tqdm_class, lock_name="")
```

get (create if necessary) and then restore `tqdm_class`'s lock

thread_map

[view source]

```
def thread_map(fn, *iterables, **tqdm_kwargs)
```

Equivalent of `list(map(fn, *iterables))` driven by `concurrent.futures.ThreadPoolExecutor`.

Parameters

- **tqdm_class***: optional*
  `tqdm` class to use for bars [default: tqdm.auto.tqdm].
- **max_workers***: int, optional*
  Maximum number of workers to spawn; passed to `concurrent.futures.ThreadPoolExecutor.__init__`. [default: max(32, cpu_count() + 4)].

process_map

[view source]

```
def process_map(fn, *iterables, **tqdm_kwargs)
```

Equivalent of `list(map(fn, *iterables))` driven by `concurrent.futures.ProcessPoolExecutor`.

Parameters

- **tqdm_class***: optional*
  `tqdm` class to use for bars [default: tqdm.auto.tqdm].
- **max_workers***: int, optional*
  Maximum number of workers to spawn; passed to `concurrent.futures.ProcessPoolExecutor.__init__`. [default: min(32, cpu_count() + 4)].
- **chunksize***: int, optional*
  Size of chunks sent to worker processes; passed to `concurrent.futures.ProcessPoolExecutor.map`. [default: 1].
- **lock_name***: str, optional*
  Member of `tqdm_class.get_lock()` to use [default: mp_lock].

tqdm/tqdm Releases

v4.66.1 (2023-08-10)

whl|asc

- □ fix `utils.envwrap` types (#1493 <- #1491, #1320 <- #966, #1319)
  - e.g. cloudwatch & kubernetes workaround: `export TQDM_POSITION=-1`
- ⚠ drop mentions of unsupported Python versions

v4.66.0 (2023-08-09)

whl|asc

- □ environment variables to override defaults (`TQDM_*`) (#1491 <- #1061, #950 <- #614, #1318, #619, #612, #370)
  - e.g. in CI jobs, `export TQDM_MININTERVAL=5` to avoid log spam
  - add tests & docs for `tqdm.utils.envwrap`
- □ fix & update CLI completion
- □ fix & update API docs
- □ minor code tidy: replace `os.path` => `pathlib.Path`
- □ fix docs image hosting
- □ release with CI bot account again (https://github.com/cli/cli/issues/6680)

v4.65.2 (2023-08-08)

whl|asc

- □ exclude `examples` from distributed wheel (#1492)

v4.65.1 (2023-08-08)

whl|asc

- □ migrate `setup.{cfg,py}` => `pyproject.toml` (#1490)
  - fix `asv` benchmarks
  - update docs
- □ fix snap build (#1490)
- □ fix & update tests (#1490)
  - fix flaky notebook tests

  - bump `pre-commit`
  - bump workflow actions

v4.65.0 (2023-03-03)

whl|asc

- ⚠ add Python 3.11 and drop Python 3.6 support (#1439, #1419, #502 <- #720, #620)
- □ misc code & docs tidy
- □ fix & update CI workflows & tests

v4.64.1 (2022-09-03)

whl|asc

- □ support `ipywidgets>=8` (#1366, #1361 <- #1310, #1359, #1360, #1364)
  - fix jupyter lab display
  - update notebook tests

v4.64.0 (2022-04-04)

whl|asc

- □ add `contrib.slack` (#1313)

v4.63.2 (2022-04-04)

whl|asc

- □ `rich`: expose `options` kwargs (#1282)
- □ `autonotebook`: re-enable VSCode (#1309)
- □ misc docs typos (#1301, #1299)
- □ update dev dependencies (#1311)

v4.63.1 (2022-03-23)

whl|asc

- □ fix stderr/stdout missing `flush()` (#1248 <- #1177)
- ⚡ misc speed improvements/optimisations

v4.63.0 (2022-02-28)

whl|asc

- □ add `__reversed__()`
- □ add efficient `__contains__()`

- ⚡ improve CLI startup time (replace `pkg_resources` => `importlib`)
- ⚠ `tqdm.autonotebook` warning & `std` fallback on missing `ipywidgets` (#1218 <- #1082, #1217)
- ⚠ warn on positional CLI arguments
- □ misc build/test framework updates
  - enable `py3.10` tests
  - add `conda` dependencies
  - update pre-commit hooks
  - fix `pytest` config (`nbval`, `asyncio`)
  - fix dependencies & tests
  - fix site deployment

v4.62.3 (2021-09-20)

whl|asc

- □ fix minor typo (#1246)
- □ minor example fix (#1246)
- ⚒ misc tidying & refactoring
- □ misc build/dev framework updates
  - update dependencies
  - update linters
  - update docs deployment branches
- □ misc test/ci updates
  - test forks
  - tidy OS & Python version tests
  - bump primary python version 3.7 => 3.8
  - beta py3.10 testing
  - fix py2.7 tests
  - better timeout handling

v4.62.2 (2021-08-23)

whl|asc

- □ fix notebook memory leak (#1216)
- □ fix `contrib.concurrent` with generators (#1233 <- #1231)

v4.62.1 (2021-08-14)

whl|asc

- □ `contrib.logging`: inherit existing handler output stream (#1191)
- □ fix `PermissionError` by using `weakref` in `DisableOnWriteError` (#1207)
- □ fix `contrib.telegram` creation rate limit handling (#1223, #1221 <- #1220, #1076)

- □ tests: fix py27 `keras` dependencies (#1222)
- □ misc tidy: use relative imports (#1222)
- □ minor documentation updates (#1222)

v4.62.0 (2021-07-30)

whl|asc

- □ `asyncio.gather` API consistency with stdlib (#1212)
- □ fix shutdown exception (#1209 <- #1198)
- □ misc build framework updates (#1209)
- ⚡ add GH Sponsors & merch links

v4.61.2 (2021-07-16)

whl|asc

- □ install `colorama` on Windows (#1139, #454)
- □ add telegram support for `leave=False` (#1189)
- □ support `pandas==1.3.0` (#1199)
- □ fix `keras` potential `AttributeError` (#1184 <- #1183)
- □ fix py3.10 `asyncio` tests (#1176)
- □ flush `stdout` / `err` before first render (#1177)
- □ misc minor build & test framework updates (#1180)

v4.61.1 (2021-06-12)

whl|asc

- □ fix `utils._screen_shape_linux()` sometimes raising `ValueError` (#1174)
- □ minor build/CI framework updates (#1175)
- □ minor documentation updates
  - fix typo (#1178)
  - link to merch! :billed_cap: :shirt:

v4.61.0 (2021-05-24)

whl|asc

- □ `keras` support for `initial_epoch` (#1150 <- #1138)
- □ misc documentation updates
  - update & shorten URLs (#1163)
  - fix typos (#1162)
- □ fix & update tests (#1163)
- □ minor framework updates (#1163)

v4.60.0 (2021-04-06)

whl|asc

- □ add `contrib.logging` helpers for redirecting to `tqdm.write()` (#1155 <- #786)
- □ support `delay` in `notebook` (#1142)
- □ fix `contrib.tmap`, `tzip` not using `tqdm_class` (#1148)
- □ add `notebook` tests (#1143)
- □ updates & misc minor fixes for documentation

v4.59.0 (2021-03-05)

whl|asc

- □ add `tqdm.dask.TqdmCallback` (#1079, #279 <- #278)
- □ add `asyncio.gather()` (#1136)
- □ add basic support for `length_hint` (#1068)
- □ add & update tests
- □ misc documentation updates (#1132)
  - update contributing guide
  - update URLs
  - bash completion: add missing `--delay`
- □ misc code tidy
  - add `[notebook]` extra (#1135)

v4.58.0 (2021-02-25)

whl|asc

- □ add start `delay` in seconds (#836 <- #1069, #704)
- □ add tests
- □ misc code tidy (#1130)
- □ misc documentation updates

v4.57.0 (2021-02-18)

whl|asc

- □ add line buffering for `DummyTqdmFile` (#960)
- □ fix & update demo notebook (#1127)
- □ fix py3 urllib examples (#1127)
- ⚠ suppress deprecated `pandas` warnings (#824, #1094)
- □ misc framework updates
- □ misc tests updates

- □ misc code tidy

v4.56.2 (2021-02-10)

whl|asc

- □ fix attribute errors when disabled (#1126)
  - `reset()` (#1125)
  - `unpause()`
- □ add tests

v4.56.1 (2021-02-09)

whl|asc

- □ fix `repr()` & `format_dict` when disabled (#1113 <- #624)
- □ rename `__repr__()` => `__str__()`
- □ minor documentation updates (#1113)
  - fix Binder demo notebook (#1119)
  - remove explicit Dockerfile
  - move some images to external repo
- □ add & update tests

v4.56.0 (2021-01-11)

whl|asc

- □ add `tqdm.tk` (#1006)
- □ add `tqdm.rich`
- □ minor formatting improvements for `tqdm.gui`
- □ fix `display()` inheritance/override
- □ add tests
- □ add documentation

v4.55.2 (2021-01-09)

whl|asc

- □ update tests (#1108)
  - make pre-commit `pytest` quicker
  - switch pre-commit from `make` to `python`
  - add and update (auto) formatters & CI (#1108, #1093)
- □ update contributing guidelines (#1108)
  - fix formatting

  - test dependencies (#1109)
- □ update `.gitignore`
- □ fix (auto & manual) formatting
- □ fix minor detected bugs
- □ misc build/CI framework upgrades

v4.55.1 (2021-01-02)

whl|asc

- □ fix `(Rolling|Expanding).progress_apply()` on `pandas==1.2.0` (#1106)
- □ minor documentation updates

v4.55.0 (2020-12-25)

whl|asc

- □ fix ASCII notebook export (#937, #1035, #1098)
- □ fix notebook gui-mode extra spaces (#433, #479, #550, #935)
- □ better ETA for early iterations (#1101)
- □ better ETA for wildly varying iteration rates (#1102)
- □ update submodule inheritance
  - `tqdm.gui`
  - `tqdm.notebook`
  - `tqdm.contrib.telegram`
  - `tqdm.contrib.discord`
- □ documentation updates
- ⚡ misc code optimisations
- □ add tests
- □ framework updates
  - build
  - CI & test
- □ misc code linting/formatting

v4.54.1 (2020-12-05)

whl|asc

- ⚠ drop `py3.4` (no longer tested) (#1091)
- □ misc CI updates (#1091)
  - update `snap` build & deploy method
  - bot releases

v4.54.0 (2020-11-27)

whl|asc

- □ get rid of `get_new` (#1085 <- #1084, #509)
- ⚡ minor CI framework optimisations

v4.53.0 (2020-11-21)

whl|asc

- □ provide `get_new()` helper for mixed subclasses in nested mode (#509)
- □ fix nested `asyncio` (#1074)
  - document async `break` hazard
- □ add tests
- ⚠ drop py2.6/3.2/3.3 and distutils (no longer tested)
  - drop py2.6 (#502 <- #620, #127)
  - drop `distutils` in favour of `setuptools` / `setup.cfg` (#723, #721)
- □ CI framework overhaul
  - drop appveyor (Windows already tested by GHA)
  - skip `devel` PRs
  - automate linting comments on failure
- □ use `setuptools_scm` (#722)
  - fix & update tests
  - fix & upgrade snap build
  - update CONTRIBUTING docs

v4.52.0 (2020-11-16)

whl|asc

- □ allow delaying `display()` to a different notebook cell (#1059 <- #909, #954)
  - add `notebook` argument `display=True` (use `display=False` with `display(tqdm_object.container)`)
  - add `keras.TqdmCallback` support for initialiser arguments (use `display=False` with `tqdm_callback_object.display()`) (#1059 <- #1065)
  - add documentation
- □ add CI on windows (#507)
- □ enable CI on OSX
- □ migrate CI Travis => GHA
  - add tests for MacOS & Windows
  - add tests for py3.9 (#1073)
  - update documentation
- □ minify docker build
- □ update tests

- □ misc tidy

v4.51.0 (2020-10-25)

whl|asc

- □ add `{eta}` datetime `bar_format` argument (#1055 <- #1051)
  - e.g. `bar_format='{l_bar}{bar}| {n_fmt}/{total_fmt} [{rate_fmt} ETA:{eta:%y-%m-%d %H:%M}{postfix}]'`
- □ fix py3 CLI `--update` & `--update_to`
- □ replace `nosetests` with `pytest` (#1052, #1045)
- □ add & update tests

v4.50.2 (2020-10-08)

whl|asc

- □ fixed `platform.system()` causing `fork()` warnings (#691)
- □ fixed `contrib.concurrent.process_map` pickling error with `threading.RLock` (#920)
- □ updated documentation & examples
- □ updated CI framework
- □ updated tests
- □ misc code tidy

v4.50.1 (2020-10-07)

whl|asc

- □ fix multiprocessing lock creation leak (#982, #936, #759)
  - fixes #617 which introduced this bug (v4.29.0, released 2019-01-06, undiagnosed until now) where multiple threads could concurrently create and append process locks to a global list, then try to release them without first acquiring :imp:
- ⚡ major test overhaul: fix, update, and speed up
- □ misc CI framework updates
- □ code linting
- □ minor documentation tidy

v4.50.0 (2020-09-28)

whl|asc

- □ add bar `colour` option (#1040 <- #450)
  - for `notebook` as well as `std`/CLI
  - plain text as well as hex (e.g. 'green' or '#00ff00')
- □ `notebook` improvements (#1032 <- #1031)
  - split bar description into separate widget

    - improve `reset()` (handle width and colour)
- □ fix exceptions when `file` is closed (#1041, #1036 <- #1033)
- □ add & update tests
- □ minor documentation updates
- □ move and configure issue/pr templates

v4.49.0 (2020-09-12)

whl|asc

- □ CLI: add `--tee` (#1014 <- #1013)
- □ CLI: add `--update` and `--update_to` (#996 <- #975)
- □ CLI: add `--null` (#996)
- □ CLI: accept `-` instead of `_` in options (e.g. `--update-to`)
- □ make `update()` return `True` on `display` to ease efficient use of custom callbacks (#845)
- □ fix `py>=3` CLI `--delim` encoding error
- □ fix `py>=3.5` version detection in `tqdm.auto` (#1029 <- #1028)
- □ fix final ETA when using `initial` (#1021 <- #689)
- □ update documentation
    - add & update custom callback examples
    - improve help formatting of boolean CLI options
- □ add & update tests

v4.48.2 (2020-08-03)

whl|asc

- □ fix `notebook` formatting

v4.48.1 (2020-08-03)

whl|asc

- □ fix ANSI escape codes breaking `tqdm.write` (#692, #777)
- □ fix long-lived strongref (#1007, https://bugs.python.org/issue39093)
- □ fix cli `--version` crash on missing `git/refs/heads` (#635)
- □ fix `contrib.tenumerate` ignoring `total` (#1017)
- □ fix potential deep => shallow `kwargs` copy issues
- □ improve `notebook` display for unknown `total` (#1015)
- □ make `asyncio` inherit from `std` rather than `auto`
- □ make `auto` multi-inherit from `autonotebook`, `asyncio` on Python3.5+
- □ misc documentation & examples updates
    - mention `docker-compose` requirements (#771)
- □ misc linting & tidy

- □ misc minor testing framework updates

v4.48.0 (2020-07-16)

whl|asc

- □ add `tqdm.asyncio` (#1004, #772, #91 <- #65)
  - supports async iterables (`async for row in tqdm(iterable)`)
  - supports iterables
  - supports `coroutine.send` (`tqdm(coroutine).send`)
  - add `as_completed` wrapper (`for f in tqdm.as_completed(fs)`)
- □ reduce `contrib.discord` logging verbosity
- □ fix potential `pandas()` modifying `**kwargs` bug
- □ update documentation
  - add missing inline docstrings
  - update contributing notes
  - update readme
  - add to `examples/` folder
    - async_coroutines
    - coroutine_pipe
  - standardise variable naming
- □ add tests
  - add benchmark tests against popular alternative libraries
  - test new functionality
  - more thorough `miniters` test (#1003)
- □ misc framework updates
  - tidy changelog helper
  - add helper bots
    - comment-bot `/tag`
    - post-release-bot
      - wiki updater
      - website updater
    - benchmark regressions
      - full/thorough test on schedule & release
      - quick test on PR

v4.47.0 (2020-06-28)

whl|asc

- □ add `contrib.discord` (similar to `contrib.telegram`) (#976)
- □ add `contrib.bells` to auto-enable all extras

- □ add `contrib.utils_worker` for common slow tasks (e.g. web I/O)
  - fix lazy large memory usage & discard unsent messages (unprocessed tasks)
- □ fix slow notebook imports (#955 <- #709)
- □ fix `gui` `TypeError` on unknown `len()` (#971)
- □ misc documentation/error message updates
  - more succinct ImportError on missing `ipywidgets` (#872)
  - fix broken/deprecated link (#981)
  - add inline usage for `contrib.discord` and `contrib.telegram`
- □ misc framework updates
  - add official `py3.8` support (#986)
  - fix `snap` builds

v4.46.1 (2020-06-03)

whl|asc

- □ fix missing `sys.setcheckinterval` in py3.9 (#978)
- □ fix `keras.TqdmCallback` compatibility with `tensorflow==2.2.0` (#979)
- □ update documentation
  - correct `contrib.concurrent` correct `max_workers` (#977)
  - drop prominent mention of `xrange` (#965)
- □ minor linting

v4.46.0 (2020-05-03)

whl|asc

- □ add `contrib.telegram` #949 <- #948
- □ add `bash` tab completion and `--comppath` #946, #947 <- #858
- □ fix exception safety in `external_write_mode` #940
- □ add `requests` examples (#242)
- □ update documentation

v4.45.0 (2020-04-02)

whl|asc

- □ propagate and `close()` on exceptions (#926 <- #548)
- □ fix nested `clear()`
- ⚒ rework nested `close()` (#930 <- #924 <- #918, #677)
  - reduces screen flicker/blank space at the cost of ordering

- □ update all documentation & demos
- □ add and update tests
- □ dev framework: add pre-commit helper

v4.44.1 (2020-03-29)

whl|asc

- □ fix `_utils` `ImportError` => `DeprecationWarning` (#928 <- #927)

v4.44.0 (2020-03-28)

whl|asc

- □ add automatic `nrows` and expose for manual override (#918 -> #924)
- ⚠ expose and warn about small `chunksize` in `tqdm.contrib.concurrent.process_map` (#912)
- □ fix py2 file stream comparison (#727 -> #730)
- ⚠ deprecate `utils._environ_cols_wrapper`
- □ add and update tests
- □ add documentation

v4.43.0 (2020-02-19)

whl

- □ fix `notebook` exceptions (#669 <- #650)
  - `set_description()` when `disable=True`
- □ `contrib.concurrent`: avoid unnecessary `len(iterable)` (#893)
- □ update documentation
  - clarify etymology (#895)
  - fix minor typo (#890)
  - update contributors and years
- □ update tests
- □ fix CI tests for PRs

v4.42.1 (2020-02-03)

whl|asc

- □ support `pandas==1.0.0` (#780, #555)
- □ minor metadata updates
- □ minor CI test update

v4.42.0 (2020-01-25)

whl|asc

- □ `notebook`: update `disable=None` behaviour (#880)
- □ `tqdm.keras`: support `tensorflow.keras` as well as `keras` (#885)
- □ add `contrib` (#882)
  - `tenumerate` (#840 <- #480, #402)
  - `tzip` <= `zip`
  - `tmap` <= `map`
  - add `concurrent` module
    - `thread_map` <= `concurrent.futures.ThreadPoolExecutor.map`
    - `process_map` <= `concurrent.futures.ProcessPoolExecutor.map`
  - add `itertools` module (#225) stub
    - `product`
- □ add & update tests
- □ add & update documentation
  - README
  - examples/
- □ update CI framework

v4.41.1 (2020-01-05)

whl|asc

- □ `keras` module tidy and fixes
  - fix `batch_size=None` (#869)
  - add missing `pop('size')`

v4.41.0 (2019-12-20)

whl|asc

- □ trim on `ncols` overflow with ANSI handling (#850, #716 <- #690)
- □ add `notebook.reset()` (#864)
- □ add `keras.TqdmCallback` (#867 <- #835)
- □ documentation updates
  - document newly added features (above)
  - notebook `ncols` percentage/pixels (#276)
- □ test updates
  - test newly added features (above)
  - add CI for `win` and `osx` (#841)
  - `py2` threading

v4.40.2 (2019-12-11)

whl|asc

- □ fix `pandas` unhashable func (#862 -> #863)
  - add tests

v4.40.1 (2019-12-11)

whl|asc

- □ fix floating point imprecision raising errors (#859)
- □ fix auto gh releases (travis-ci/dpl#914)
- □ update CONTRIBUTING (#84 (comment))

v4.40.0 (2019-12-01)

whl|asc

- □ officially support `float` for `n` and `total` (#802)
  - `notebook`: use `FloatProgress` <= `IntProgress` (#471, #456)
  - allow imprecision (`n <= total + epsilon`) (#849)
- □ fix unicode bar format arguments (#803 -> #851)
- □ add `contrib` submodule (#815)
- □ add `wrapattr`, `utils.CallbackIOWrapper`, `contrib.DummyTqdmFile` (#84 -> #844)
- □ update tests
- □ update documentation
- □ tidy automatic `snap` deployments
- □ minor doc update (#854)

v4.39.0 (2019-11-22)

whl|asc

- □ add `pandas` builtin operations check (#843 <- #697)
- □ avoid unnecessary `dedent` (#837)
- □ remove unneeded bar logic
- □ misc code tidy
- □ update documentation
  - document default argument overriding (#370)
  - add missing `isatty()` (#713)
  - update badges
  - add code of conduct
- □ update framework
  - clean CI stages
  - update CI default `py3.6` => `py3.7`

  - automate snap releases (#686 <- #685)

v4.38.0 (2019-11-09)

whl|asc

- □ support `lock_args` for e.g. non-blocking intermediate writes (#838 -> #839)
  - use `self.refresh()` in more places
  - add and update performance tests
  - support args for `TqdmDefaultWriteLock.acquire()`
- □ fix colorama on win (#678, #764)
- □ framework updates
  - CI cleanup
    - move to stages (check/test/deploy)
    - auto deploy notes
    - deploy only on parent repo
    - separate docker deploy
  - update documentation sync/generation
- □ update examples
- □ update documentation
- □ add and update tests
- □ misc tidy and linting

v4.37.0 (2019-10-31)

whl|asc

- □ potential future pandas fix (#824)
- □ better unicode widechar support (#803 -> #410, #805)
- □ fix example/tqdm_wget unknown total (#826)
- □ add `stacklevel=2` to warnings (#821)
- □ misc framework updates
  - snapcraft version fix
  - update issue/pr templates (#830)
  - update unit tests
- □ misc documentation updates
  - update parallel (multiprocessing, ThreadPoolExecutor) example (#407)
  - add slides, video images
  - add CII Best Practices badge
  - add repology badge
  - update badge icons

v4.36.1 (2019-09-20)

whl|asc

- □ fix CLI entrypoint

v4.36.0 (2019-09-17)

whl|asc

- □ expose more custom format params
- □ potential thread safety fix (#548)
- ⚒ update submodule architecture (#198 -> #800)
  - backward-compatibility
  - expose `utils`
  - rename `main` -> `cli`
  - add/fix tests
  - fix minor cached var optimisation
  - `gui` partial upgrade (#790)
  - `notebook` upgrade (#790)
    - support `{bar}` in `bar_format` (#594)
    - inherit methods including `set_*(refresh)` (#740 -> #741)
  - ready for `contrib` releases (#252)
- □ support custom `bar_format` with unknown `total`
- □ fix `tqdm_notebook` red block on explicit `file=None` (#791)
- □ update notebook to fully support custom `bar_format` (#594, #649)
- □ misc tidy
  - warnings
  - linting
  - update submodule contribution guide (#252)
  - parallel `(py)make test`
  - fix Zenodo metadata
- □ update documentation
  - `pandas`, `notebook`, `gui` (#245)
  - indentation
  - update multiprocessing and multithreading examples (#796)

v4.35.0 (2019-08-24)

whl|asc

- □ add `{bar}` format specifier (#623 -> #799)
  - `[width][type]`
- □ add tests and documentation

- ⚡ update performance tests

v4.34.0 (2019-08-18)

whl|asc

- □ add `leave=None` convenience option for `leave = position == 0`
- □ ensure nested completed bars respect `leave=True` (#230)
- □ ensure nested bars are cleared before being moved up (#795)
- □ support both and `=` syntax for CLI options (#761 -> #774)
- □ misc documentation updates (#778, #782)
- □ fix/update tests
- □ update GitHub docker package deployment

v4.33.0 (2019-08-08)

whl|asc

- □ fix `pandas==0.25` API change (#780)
- □ add contributor badges (`sourcerer`)
- □ fix py26 travis CI
- □ metadata fixes
  - `.zenodo.json`
  - `CODEOWNERS`
- □ GitHub package registry docker deployment
- □ minor linting
- □ documentation updates (#773)

v4.32.2 (2019-06-18)

whl|asc

- □ fix 'Set changed size during iteration' (#481 -> #700, #754)
- □ add `tqdm.autonotebook` check for `$VSCODE_PID` (#747 -> #751)
- □ fix missing `multiprocessing` in Jython (#698)
- □ make `bool(tqdm(iter))` consistent with `bool(iter)` (#353 -> #694)
- □ add and update unit tests
- □ add and update documentation & framework
  - update notable contributors list
  - update usage docstrings (#714, #715)
  - add funding
  - add `awesome-python` badge
  - trove classifier update

  - update demo notebook badges
  - add JOSS DOI
  - add Zenodo metadata
  - move unnecessary root clutter

v4.32.1 (2019-05-13)

whl|asc

- □ fix `notebook` with unknown `total` (#743)

v4.32.0 (2019-05-13)

whl|asc

- □ support `unit_scale` in `notebook`
- □ support negative update (#432, #545)
- □ add `reset()` function (#547, #545)
- □ add `[python setup.py] make run`
- □ add and update documentation
  - example of dynamic usage (#735, #545, #547, #432, #374)
  - note writing issues (#737)
  - update badges
  - add PyData2019 slides link
  - add JOSS paper
  - update manpages
  - add docker install
  - add snapcraft install
  - notebooks: add binder, rename RMOTR => notebooks.ai (#679)
  - prettify and unify contributors/maintainers/authors
- □ CI and release framework updates
  - add snapcraft snaps (#647)
  - add travis auto-deployment (#685)
    - PyPI releases
    - docker devel/releases
  - update deployment dev docs
  - fix travis deploy pymake
  - update .gitinore
  - add & update unit tests
  - automate more documentation

v4.31.1 (2019-02-10)

whl|asc

- □ fix `file.encoding==None` caused by #598 (#673 -> #676)
  - add tests
- □ tidy code, update authors and tests
  - add `setup.py` test to CI

v4.31.0 (2019-02-09)

whl|asc

- □ Write bytes to `stdout`/`stderr` in py2 (https://bugs.python.org/issue21363, #589 -> #598)
  - Add `write_bytes` parameter for py2/3 default override
- □ support custom bar characters vias `ascii` parameter (#223 -> #227)
  - implement, document and test custom bar spinners
- □ support custom `bar_format` with unknown `total` (#282 -> #284)
- □ fix `TypeError` when `unit_scale` and unknown `total`
- □ document `format_dict`
- □ expose `elapsed_s`, `remaining_s` to `bar_format`
- □ add `unit` option to `bar_format` (#368)
- □ auto `README.rst` generation framework
  - add notes in `CONTRIBUTING.md`
- □ update tests
- □ update documentation
- □ misc code tidy

v4.30.0 (2019-01-26)

whl|asc

- □ avoid moving when `leave=False`
- □ generalise nested bar tests
- □ add `display()` helper function
- □ add and expose `format_dict` (#562, #482, #494, #656) + allow arbitrary keyword arguments for `format_meter` + document `format_dict` overriding + add tests for overriding + related: #660 #587
- □ add and tidy developer documentation
- □ fix py3 CLI pipe decoding error (#663)
- □ framework overhaul (developing, testing, Travis CI updates)
- □ tidy code and style (flake8)
- ⚒ `setup.py` refactor + `setup.py make` * add support for multi-line commands * tidy and efficiency + replace `setup.py`'s built-in `make` -> `py-make>=0.1.9` (#290) + add `requirements-dev.txt` and `extras_require[dev]`
- □ add, update and tidy tests
- □ include interactive Jupyter demonstration (#652 -> #659)
- □ update documentation + add FAQ about py2 pipes (#359) + add badges + update examples + update

authors

v4.29.1 (2019-01-11)

whl|asc

- □ fix `classmethod` lock bug (#617: #457 -> #658)
- □ add unit test to prevent regression

v4.29.0 (2019-01-06)

whl|asc

- □ Avoid global multiprocessing locks (#611 -> #617)
- □ Add support for infinite iterables (#651)
- □ Fix missing attr pos when used in multi-threaded environment (#573)
- □ Do not join `TMonitor` if it is the current thread (#613 -> #641)
- □ Add OpenBSD NIX support (#638)
- □ Unit tests, general documentation fixes and tidying (e.g. #642)
- □ CI travis improvements
  - `py37-dev` -> `py37` (#622)
  - fix `py26`

v4.28.1 (2018-10-21)

whl|asc

- □ fix `--manpath` file not found
- □ expose `main():argv`
- □ add unit tests
- □ release framework tidy
  - update `.gitattributes`
  - remove deprecated `make` / `dist` options

v4.28.0 (2018-10-20)

whl|asc

- □ remove installation of man pages by default (#460, #628)
- □ CLI:add `--manpath` option (#629)
- □ documentation additions and fixes

v4.27.0 (2018-10-15)

whl|asc

- □ fix `str.isnumeric` #605
- □ fix `WeakSet` `KeyError` #548, #553, #596 -> #607
- □ stop `tqdm_notebook` description truncation #582 -> #599
- □ include `unit_scale` for `rate` #608
- □ add `auto` -> nowarning `autonotebook`
- □ add better postfix numeric formatting #621
- ⚒ minor refactoring #609 -> #616
- □ update documentation
- □ add unit tests
- □ fix py26 CI

v4.26.0 (2018-09-11)

whl|asc

- □ fix `smoothing` (#566 -> #601)
- □ `pandas` updates + address the `FutureWarning` in `pandas`, drop `pandas` test in py34, add in py37 (#603) + support `pandas` `axis='index' or 'columns'` (#570)
- □ minor documentation updates (#597, #600, #606) + developer notes

v4.25.0 (2018-08-20)

whl|asc

- □ stop monitor on shutdown (#571, #572)
- □ fix `find_packages()` for submodules (#593) + properly add `autonotebook` (#586, 1cf3393)
- □ ignore ANSI escape codes in the bar length calculation (#450, #591 -> #592)
- □ update documentation (#243)

v4.24.0 (2018-07-26)

whl|asc

- □ `autonotebook` submodule for automatic selection of notebook/CLI bar (#443, #508) + update/add `pandas()` example documentation (#474)
- □ `NameError:IntProgress` changed to a more helpful `ImportError` (#187, #451, #558)
- □ support `bool()` cast when `disable=True` (#574)
- □ fix `format_sizeof` hundreds rounding (#579 -> #581)
- □ ensure URLs in documentation are secure (`https`)

v4.23.4 (2018-05-22)

whl|asc

- □ Support pandas 0.23.0 `core.groupby` module layout (#555 -> #554)

- □ Add python_requires to help pip (#557)
- □ minor maintenance updates
  - CI updates: drop travis py33 due to `tox` (tox-dev/tox#648)
  - minor code tidy

v4.23.3 (2018-05-22)

whl|asc

- ⚠ suppress 10 second warning (refix #323)

v4.23.2 (2018-05-02)

whl|asc

- □ minor import syntax fix #496
- □ re-fix nested overlaps #477
- □ update documentation and examples
- □ code tidy and abstraction

v4.23.1 (2018-04-25)

whl|asc

- □ fix `AttributeError`s (#546) + `pos` on initialisation (#323, #510 -> #544) + `fp` on `write()` (#383)
- □ fix py34 CI
- □ update documentation

v4.23.0 (2018-04-15)

whl|asc

- □ Fix `disable=True` where `iterable` has no `len()` and is not `None` (#539)
- □ Add `ncols` to specify `tqdm_notebook` bar width (#276 -> #292)
- □ allow custom `pandas` `total` (#364 -> #535)
- □ Add `progress_apply` for `pandas.(Series|DataFrame).(rolling|expanding)` (#530 -> #537)
- □ unit tests, pep8 tidy
- □ `postfix` non-`dict` documentation note

v4.22.0 (2018-04-11)

whl|asc

- □ allow direct non-string assignment to `postfix` member (lists, dicts, etc) (#382 -> #534)
- □ documentation updates
- □ unit tests

v4.21.0 (2018-04-08)

whl|asc

- □ fix #532 `write()` with manual `position`
  - more robust free position finding
- □ add `TqdmWarning` base class
- □ fix GUI `__del__()`
  - add `TqdmExperimentalWarning`
- □ add, tidy & fix unit tests
- □ documentation
  - wiki release notes
  - update contributors
  - fix `7zx` example

v4.20.0 (2018-04-03)

whl|asc

- □ `pandas` wrapper: remove `*args` and add better `total` handling (#244, #299, #322, #366 -> #524)
- □ document windows unicode know issues (#454)
- □ suppress `RuntimeError: Set changed size during iteration` -> `TqdmSynchronisationWarning` (#481)

v4.19.9 (2018-03-27)

whl|asc

- □ fix monitor thread termination and update tests (#527)

v4.19.8 (2018-03-27)

whl|asc

- □ fix monitoring thread issues
  - avoid uninitialised instance manipulation (#493)
  - fix thread starting `RuntimeError` (#522 -> #523)
- □ these release notes (#529)

v4.19.6 (2018-02-27)

whl|asc

- □ CLI improvements + `--bytes` implies `--unit B --unit_scale --unit_divisor 1024` (#503) + allow hyphens in CLI arguments (#511)
- □ synchronisation: fix for `sem_open` on `aarch64` (#513)
- □ framework update + CI/tests, year 2018, py37, badges, documentation

v4.19.5 (2017-12-10)

whl|asc

- □ multiprocess/lock fixes (#457)
- □ `set_description` in notebook (#345 -> #475)
- □ minor tidy (#476)
- □ documentation updates

v4.19.4 (2017-10-15)

whl|asc|gz|asc

- □ fix `Lock:NotImplementedError` on certain systems (#466 -> #468)
- □ use recursive locks (#469 -> #468) - fix deadlocks
- □ tidy (#448)
- □ `flush()` on `moveto()` (#398 -> #399, #420, #467)
- □ update tests and benchmarks

v4.19.2 (2017-10-08)

whl|asc|gz|asc

- □ `flush()` on `refresh()` (#459 from #317)
- □ status printer updates (#331) + use `sp()` in `refresh()` + remove redundant `clear():nomove=False`
- □ misc minor documentation updates
- □ unit tests

v4.19.1 (2017-10-03)

whl|asc|gz|asc

- □ `rate_(no)inv(_fmt)` (#72, b228bc3)
- □ `__repr__()` tidy (#389)
- □ fix `write()` before initialisation `AttributeError: _lock` (#457)
- □ `man` pages
- □ documentation updates

v4.18.0 (2017-09-30)

whl|asc|gz|asc

- □ Thread safety! (#285 -> #291 -> #329, #407, #417)
- □ Ease redirection of `sys.stdout`/`stderr` (#422)
- □ Minor internal stream bugfix (#439)
- □ `AttributeError` fixes (#323, #324, #418)

Related to:

- □ Misc bugs (#334)
- □ `concurrent.futures` (#97)
- □ Multi-`tqdm` (#143)
- □ `flush()` and `refresh()` (#331)
- □ Newline on `refresh()` (#361)
- □ Nested trees (#384)
- □ Manually positioned nested bars clearing (#385)

v4.17.1 (2017-09-26)

whl|asc|gz|asc

- □ initialise `n` even when disabled (#438)
- □ better unicode detection (#437)
- □ `__repr__()` bugfix (#441)

v4.17.0 (2017-09-20)

whl|asc|gz|asc

- □ easier redirect of `stdout` (#426 -> #431) - `cls.external_write_mode() @contextmanager`
- □ `refresh=True` for `set_description`/`update`(`_str`) (#317 -> #377)

v4.16.0 (2017-09-20)

whl|asc|gz|asc

- □ more formatting control (#347 -> #362) - `set_postfix_str()`, `set_description_str()` - move `": "` to `bar_format` - complements #266, #270
- □ better CLI support (e.g. hyphenated paths) (#421 -> #424)
- □ more unit tests (#411) - upgrade to pypy2/3-5.8.0 - add py37-dev
- □ doc updates

v4.15.0 (2017-07-29)

whl|asc

- □ minor fixes (#395)
- □ minor documentation updates (#400, #401, #412)
- □ better `pandas` support (#351 -> #409)
- □ Jupyter notebook fix (#352, #369 -> #373, #405)
- ⚡ optimise `setcheckinterval`/`setswitchinterval` (#376)
- □ add `unit_scale` (#273, #295 -> #378)

v4.14.0 (2017-07-29)

- add `unit_divisor` #354 -> #356
- update examples

v4.13.0 (2017-05-29)

whl|asc

- support unknown systems https://github.com/warner/magic-wormhole/issues/158, #76, #344 -> #390
- support nested/multi-bars better #384, #285, #329 -> #385
- ease redirection of output (default: `sys.stderr` -> `None`) https://github.com/xonsh/xonsh/issues/2058, #329, #275 -> #293 + add examples/redirect_print.py
- minor documentation updates

v4.12.0 (2017-05-29)

whl|asc

- fix monitor race condition #338 -> #339
- add explicit NetBSD support #344
- documentation tidy

v4.11.2 (2017-01-24)

whl|asc|gz|asc

- thread safety #332
- address `DeprecationWarning` #319 -> #326
- version extra #328 -> #330
    - security fix (CVE-2016-10075)
    - add non-master branch name
- minor safety
- minor documentation and comments

v4.11.1 (2017-01-23)

whl|asc|gz|asc

Officially support Python 3.6, addressing #333

v4.11.0 (2017-01-12)

whl|gz

- `postfix` argument added #266->#270
- fix tests for

  - `numpy` #272->#302
  - `PyPy3` #318->#325
- □ `7z` extraction example script
- □ new logo #315
- □ hit counter with logo colours and stats
- □ documentation updates
- □ safer `setup.py` `Makefile` emulation
- □ better github contributions integrations and Code of Conduct #310

v4.10.0 (2016-11-12)

gz|zip

- □ fix maxinterval to adjust miniters to mininterval #249
- □ minor bugfixes (eg #288)
- ⚡ misc optimisations, unit tests, and benchmarks (eg #298)
- □ documentation updates

v4.9.0 (2016-10-31)

gz|zip

- □ monitoring thread to automatically to adjust `miniters` and avoid exceeding `maxinterval`
  - test race conditions
  - kill on KeyboardInterrupt + instant refresh
  - low performance overhead, option to disable
- □ new CLI argument: `--bytes` to count chars
- □ disable automatically on non-TTY (#281)
- □ fixed `refresh()` and `clear()` if disabled (#286)
- □ misc fixes
- □ documentation
- □ unit tests
- ⚡ optimisations, style fixes
- □ codacy badge
- ⚠ safety and warning suppression
- □ ignore coverall failures (#283)
- □ fix broken link in README (#262)
- ⚡ Fix `time()` in `refresh()` + avoid `write()` races (#264)
- □ Fix `__repr__()` #269
- □ Fix `update(0)` causing `ZeroDivisionError` (on first call) #268
- □ Faster simple bar for perf test (#257)
- □ cleaner `tqdm_notebook` exit when not run in a notebook (#267)

- □ Disable coveralls because of duplicate comments on github (#263)
- □ Migrate codecov config -> codecov.yml

v4.8.4 (2016-08-17)

whl|gz

- ⚡ 14% overhead reduction
- □ fix exception when closing `tqdm_notebook` without a `total`
- □ support more pandas modules
  - pandas.Series.progress_map
  - pandas.DataFrame.progress_applymap
  - Panel.progress_apply
  - PanelGroupBy.progress_apply
  - GroupBy.progress_aggregate
  - GroupBy.progress_transform
- □ faster CI builds
- □ misc documentation update and link to wiki

v4.8.1 (2016-07-25)

- ⚠ custom tqdm warning/exceptions
- □ tests now multiprocessed and include pandas

v4.8.0 (2016-07-25)

- ⚠ `tqdm.pandas` classmethod replaces `tqdm_pandas` function (deprecated)
- □ `ipython` aliased imports fixes
- □ `write` without any instances
- □ readme update
- □ examples update
- □ test fixes and updates

v4.7.6 (2016-06-30)

- □ support for pandas series
- □ support for numpy arrays
- □ tidy & coverage

v4.7.4 (2016-06-04)

whl|gz

- □ `pandas.DataFrame.apply` support

- □ `tqdm_notebook.write()`
- □ `tqdm_gui.write()`
- □ misc notebook fixes
- ⚡ delay notebook imports (>80% reduction in import time when unused)
- □ optional `colorama` only on windows
- □ documentation tidy
- □ more & faster tests

v4.7.0 (2016-05-15)

whl|gz

- □ CLI custom delimiter support
- □ More robust CLI argument parsing
  - either one or two `-` preceding kwargs (eg `-ascii` or `--ascii`)
  - no positional args (so no need to quote multi-word `-desc` for example)
  - `bool` kwargs with implicit value=`True` still supported
  - optional `=` sign after kwargs
- □ tests
- □ doc updates, badges
  - pyversions, doi, licence, github

v4.6.2 (2016-05-15)

whl|gz|exe|zip

- □ Fix warnings due to IPython/Jupyter widget

v4.6.1 (2016-05-15)

whl|gz|exe|zip

- □ add IPython/Jupyter support
- □ fix `{bar}` length breaking display when using custom `bar_format`
- □ add a no inversion rate formatting for `bar_format` (i.e., always show `it/s` and never `s/it`)

v4.5.0 (2016-04-25)

whl|gz|zip

- □ print messages within (nested) loops using `tqdm.write(...)`

v4.4.3 (2016-04-24)

whl|gz|zip

- ▯ remove `docopt` dependency for CLI
- ▯ re-added `update(0)` ability to trigger refresh
- ▯ minor fixes
  - re-added (OSI approved) licence to source
  - dev versioning auto-fallback
- ⚡ slight speed optimisation
- ▯ tests
- ▯ documentation

v4.4.1 (2016-04-22)

whl|gz|zip

- ▯ Module execution in CLI pipes!
  - `tqdm` executable
  - documentation
  - tests
- ▯ MSYS aded to supported envs
- ▯ better VTE support
- ▯ dev versioning
- ▯ tests

v4.2.0 (2016-04-22)

whl|gz|zip

- ▯ MPL licence!
- ▯ slightly updated logo
- ▯ update examples
- ▯ `update(0)` allowed
- ▯ better VTE support
  - ascii fallback when unicode not available

v4.0.0 (2016-02-04)

whl|gz|zip

- ▯ automate `nested`
  - deprecate `nested`
  - add manual override control with `position`
- ▯ documentation and example updates, coverage and tests
- ▯ make `leave==True` by default (was `False`)

v3.8.0 (2016-01-31)

whl|gz|zip

- □ update(n<1) raises error
- □ sanitise and update examples and README
- □ fix setup.py encoding error
- □ remove py32 from Travis
- □ changelog more accessible from README

v3.7.1 (2016-01-13)

whl|gz|exe|zip

- □ Added more OS support (FreeBSD, Solaris/SunOS)
- □ Added Python3.5 support and tests
- □ Added IronPython2.7 support
- □ Fix Makefile management for Python3
- □ Added relative timing for tests
- □ More stable smoothed ETA
- ⚒ Refactoring *everything* into one big `tqdm` class for even more subclassing fun

v3.6.0 (2015-12-31)

whl|gz|zip

Added support for "restarting" a progressbar after a long pause without making the iteration rate go haywire. Simply call the `unpause()` method on the `tqdm` instance that is about to be restarted. Can also start `tqdm` skipping a number of steps specified by `initial` (without bias-ing ETA).

v3.5.0 (2015-12-31)

whl|gz|zip

- □ Fully customisable output format using the `bar_format` keyword argument.
  - Valid strings: bar, n, n_fmt, total, total_fmt, percentage, rate, rate_fmt, elapsed, remaining, l_bar, r_bar, desc
  - Example: `bar_format='{desc}{bar}{rate:03.3d}'`
  - Default: '{l_bar}{bar}{r_bar}', where l_bar is '{desc}{percentage:3.0f}%|' and r_bar is '|{n_fmt}/{total_fmt} [{elapsed_str}<{remaining_str}, {rate_fmt}]'.
- □ Solaris, FreeBSD support mentioned
- □ Link redirects circumvented

v3.3.0 (2015-12-31)

whl|gz|zip

- □ Support added for pandas `apply` => `progress_apply`
- □ 80-char width when running interactive `help()`
- □ update examples using `with` syntax

v3.1.4 (2015-11-28)

whl|gz|exe|zip

Added `smoothing` and `maxinterval` features for smooth progress bar and rate for unconstant, variable changes in update times, and abstracted gui to its own class and file.

Also added a performance test to prevent performance regressions.

v2.2.4 (2015-11-08)

whl|gz|exe|zip

Added the dynamic management of the bar on window resize, plus a better internal testing/build workflow using `python setup.py make [alias]`.

v2.0.0 (2015-10-11)

whl|gz|exe|zip

First public release of the next generation of tqdm, with lots of new features, more speed, more unit testing, more coverage (100% yeah!), more beer and coffee.

This release can also be downloaded on PyPi here.

Licence

`tqdm` is a product of collaborative work. Unless otherwise stated, all authors (see commit logs) retain copyright for their respective work, and release the work under the MIT licence (text below).

Exceptions or notable authors are listed below in reverse chronological order:

- files: * MPL-2.0 2015-2023 (c) Casper da Costa-Luis casperdcl.
- files: tqdm/_tqdm.py MIT 2016 (c) PR #96 on behalf of Google Inc.
- files: tqdm/_tqdm.py README.rst .gitignore MIT 2013 (c) Noam Yorav-Raphael, original author.

Mozilla Public Licence (MPL) v. 2.0 - Exhibit A

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this project, You can obtain one at https://mozilla.org/MPL/2.0/.

MIT License (MIT)

Copyright (c) 2013 noamraph

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

tqdm.notebook

IPython/Jupyter Notebook progressbar decorator for iterators. Includes a default `range` iterator printing to `stderr`.

Usage:

```
>>> from tqdm.notebook import trange, tqdm
>>> for i in trange(10):
...     ...
```

tqdm_notebook Objects

```
class tqdm_notebook(std_tqdm)
```

[view source]

Experimental IPython/Jupyter Notebook widget using tqdm!

status_printer

```
@staticmethod
def status_printer(_, total=None, desc=None, ncols=None)
```

[view source]

Manage the printing of an IPython/Jupyter Notebook progress bar widget.

__init__

```
def __init__(*args, **kwargs)
```

[view source]

Supports the usual `tqdm.tqdm` parameters as well as those listed below.

Parameters

- **display**: *Whether to call `display(self.container)` immediately* [default: True].

reset

```
def reset(total=None)
```

[view source]

Resets to 0 iterations for repeated use.

Consider combining with `leave=True`.

Parameters

- **total**: *int or float, optional. Total to use for the new bar.*

tnrange

```
def tnrange(*args, **kwargs)
```

[view source]

Shortcut for `tqdm.notebook.tqdm(range(*args), **kwargs)` .

tqdm/tqdm Releases

v4.66.1 (2023-08-10)

whl|asc

- □ fix `utils.envwrap` types (#1493 <- #1491, #1320 <- #966, #1319)
  - e.g. cloudwatch & kubernetes workaround: `export TQDM_POSITION=-1`
- ⚠ drop mentions of unsupported Python versions

v4.66.0 (2023-08-09)

whl|asc

- □ environment variables to override defaults (`TQDM_*`) (#1491 <- #1061, #950 <- #614, #1318, #619, #612, #370)
  - e.g. in CI jobs, `export TQDM_MININTERVAL=5` to avoid log spam
  - add tests & docs for `tqdm.utils.envwrap`
- □ fix & update CLI completion
- □ fix & update API docs
- □ minor code tidy: replace `os.path` => `pathlib.Path`
- □ fix docs image hosting
- □ release with CI bot account again (https://github.com/cli/cli/issues/6680)

v4.65.2 (2023-08-08)

whl|asc

- □ exclude `examples` from distributed wheel (#1492)

v4.65.1 (2023-08-08)

whl|asc

- □ migrate `setup.{cfg,py}` => `pyproject.toml` (#1490)
  - fix `asv` benchmarks
  - update docs
- □ fix snap build (#1490)
- □ fix & update tests (#1490)
  - fix flaky notebook tests

  - bump `pre-commit`
  - bump workflow actions

v4.65.0 (2023-03-03)

whl|asc

- ⚠ add Python 3.11 and drop Python 3.6 support (#1439, #1419, #502 <- #720, #620)
- □ misc code & docs tidy
- □ fix & update CI workflows & tests

v4.64.1 (2022-09-03)

whl|asc

- □ support `ipywidgets>=8` (#1366, #1361 <- #1310, #1359, #1360, #1364)
  - fix jupyter lab display
  - update notebook tests

v4.64.0 (2022-04-04)

whl|asc

- □ add `contrib.slack` (#1313)

v4.63.2 (2022-04-04)

whl|asc

- □ `rich`: expose `options` kwargs (#1282)
- □ `autonotebook`: re-enable VSCode (#1309)
- □ misc docs typos (#1301, #1299)
- □ update dev dependencies (#1311)

v4.63.1 (2022-03-23)

whl|asc

- □ fix stderr/stdout missing `flush()` (#1248 <- #1177)
- ⚡ misc speed improvements/optimisations

v4.63.0 (2022-02-28)

whl|asc

- □ add `__reversed__()`
- □ add efficient `__contains__()`

- ⚡ improve CLI startup time (replace `pkg_resources` => `importlib`)
- ⚠ `tqdm.autonotebook` warning & `std` fallback on missing `ipywidgets` (#1218 <- #1082, #1217)
- ⚠ warn on positional CLI arguments
- □ misc build/test framework updates
  - enable `py3.10` tests
  - add `conda` dependencies
  - update pre-commit hooks
  - fix `pytest` config (`nbval`, `asyncio`)
  - fix dependencies & tests
  - fix site deployment

v4.62.3 (2021-09-20)

whl|asc

- □ fix minor typo (#1246)
- □ minor example fix (#1246)
- ⚒ misc tidying & refactoring
- □ misc build/dev framework updates
  - update dependencies
  - update linters
  - update docs deployment branches
- □ misc test/ci updates
  - test forks
  - tidy OS & Python version tests
  - bump primary python version 3.7 => 3.8
  - beta py3.10 testing
  - fix py2.7 tests
  - better timeout handling

v4.62.2 (2021-08-23)

whl|asc

- □ fix notebook memory leak (#1216)
- □ fix `contrib.concurrent` with generators (#1233 <- #1231)

v4.62.1 (2021-08-14)

whl|asc

- □ `contrib.logging`: inherit existing handler output stream (#1191)
- □ fix `PermissionError` by using `weakref` in `DisableOnWriteError` (#1207)
- □ fix `contrib.telegram` creation rate limit handling (#1223, #1221 <- #1220, #1076)

- □ tests: fix py27 `keras` dependencies (#1222)
- □ misc tidy: use relative imports (#1222)
- □ minor documentation updates (#1222)

v4.62.0 (2021-07-30)

whl|asc

- □ `asyncio.gather` API consistency with stdlib (#1212)
- □ fix shutdown exception (#1209 <- #1198)
- □ misc build framework updates (#1209)
- ⚡ add GH Sponsors & merch links

v4.61.2 (2021-07-16)

whl|asc

- □ install `colorama` on Windows (#1139, #454)
- □ add telegram support for `leave=False` (#1189)
- □ support `pandas==1.3.0` (#1199)
- □ fix `keras` potential `AttributeError` (#1184 <- #1183)
- □ fix py3.10 `asyncio` tests (#1176)
- □ flush `stdout`/`err` before first render (#1177)
- □ misc minor build & test framework updates (#1180)

v4.61.1 (2021-06-12)

whl|asc

- □ fix `utils._screen_shape_linux()` sometimes raising `ValueError` (#1174)
- □ minor build/CI framework updates (#1175)
- □ minor documentation updates
  - fix typo (#1178)
  - link to merch! :billed_cap: :shirt:

v4.61.0 (2021-05-24)

whl|asc

- □ `keras` support for `initial_epoch` (#1150 <- #1138)
- □ misc documentation updates
  - update & shorten URLs (#1163)
  - fix typos (#1162)
- □ fix & update tests (#1163)
- □ minor framework updates (#1163)

v4.60.0 (2021-04-06)

whl|asc

- □ add `contrib.logging` helpers for redirecting to `tqdm.write()` (#1155 <- #786)
- □ support `delay` in `notebook` (#1142)
- □ fix `contrib.tmap`, `tzip` not using `tqdm_class` (#1148)
- □ add `notebook` tests (#1143)
- □ updates & misc minor fixes for documentation

v4.59.0 (2021-03-05)

whl|asc

- □ add `tqdm.dask.TqdmCallback` (#1079, #279 <- #278)
- □ add `asyncio.gather()` (#1136)
- □ add basic support for `length_hint` (#1068)
- □ add & update tests
- □ misc documentation updates (#1132)
  - update contributing guide
  - update URLs
  - bash completion: add missing `--delay`
- □ misc code tidy
  - add `[notebook]` extra (#1135)

v4.58.0 (2021-02-25)

whl|asc

- □ add start `delay` in seconds (#836 <- #1069, #704)
- □ add tests
- □ misc code tidy (#1130)
- □ misc documentation updates

v4.57.0 (2021-02-18)

whl|asc

- □ add line buffering for `DummyTqdmFile` (#960)
- □ fix & update demo notebook (#1127)
- □ fix py3 urllib examples (#1127)
- ⚠ suppress deprecated `pandas` warnings (#824, #1094)
- □ misc framework updates
- □ misc tests updates

- □ misc code tidy

v4.56.2 (2021-02-10)

whl|asc

- □ fix attribute errors when disabled (#1126)
  - `reset()` (#1125)
  - `unpause()`
- □ add tests

v4.56.1 (2021-02-09)

whl|asc

- □ fix `repr()` & `format_dict` when disabled (#1113 <- #624)
- □ rename `__repr__()` => `__str__()`
- □ minor documentation updates (#1113)
  - fix Binder demo notebook (#1119)
  - remove explicit Dockerfile
  - move some images to external repo
- □ add & update tests

v4.56.0 (2021-01-11)

whl|asc

- □ add `tqdm.tk` (#1006)
- □ add `tqdm.rich`
- □ minor formatting improvements for `tqdm.gui`
- □ fix `display()` inheritance/override
- □ add tests
- □ add documentation

v4.55.2 (2021-01-09)

whl|asc

- □ update tests (#1108)
  - make pre-commit `pytest` quicker
  - switch pre-commit from `make` to `python`
  - add and update (auto) formatters & CI (#1108, #1093)
- □ update contributing guidelines (#1108)
  - fix formatting

  - test dependencies (#1109)
- □ update `.gitignore`
- □ fix (auto & manual) formatting
- □ fix minor detected bugs
- □ misc build/CI framework upgrades

v4.55.1 (2021-01-02)

whl|asc

- □ fix `(Rolling|Expanding).progress_apply()` on `pandas==1.2.0` (#1106)
- □ minor documentation updates

v4.55.0 (2020-12-25)

whl|asc

- □ fix ASCII notebook export (#937, #1035, #1098)
- □ fix notebook gui-mode extra spaces (#433, #479, #550, #935)
- □ better ETA for early iterations (#1101)
- □ better ETA for wildly varying iteration rates (#1102)
- □ update submodule inheritance
  - `tqdm.gui`
  - `tqdm.notebook`
  - `tqdm.contrib.telegram`
  - `tqdm.contrib.discord`
- □ documentation updates
- ⚡ misc code optimisations
- □ add tests
- □ framework updates
  - build
  - CI & test
- □ misc code linting/formatting

v4.54.1 (2020-12-05)

whl|asc

- ⚠ drop `py3.4` (no longer tested) (#1091)
- □ misc CI updates (#1091)
  - update `snap` build & deploy method
  - bot releases

v4.54.0 (2020-11-27)

whl|asc

- □ get rid of `get_new` (#1085 <- #1084, #509)
- ⚡ minor CI framework optimisations

v4.53.0 (2020-11-21)

whl|asc

- □ provide `get_new()` helper for mixed subclasses in nested mode (#509)
- □ fix nested `asyncio` (#1074)
  - document async `break` hazard
- □ add tests
- ⚠ drop py2.6/3.2/3.3 and distutils (no longer tested)
  - drop py2.6 (#502 <- #620, #127)
  - drop `distutils` in favour of `setuptools` / `setup.cfg` (#723, #721)
- □ CI framework overhaul
  - drop appveyor (Windows already tested by GHA)
  - skip `devel` PRs
  - automate linting comments on failure
- □ use `setuptools_scm` (#722)
  - fix & update tests
  - fix & upgrade snap build
  - update CONTRIBUTING docs

v4.52.0 (2020-11-16)

whl|asc

- □ allow delaying `display()` to a different notebook cell (#1059 <- #909, #954)
  - add `notebook` argument `display=True` (use `display=False` with `display(tqdm_object.container)`)
  - add `keras.TqdmCallback` support for initialiser arguments (use `display=False` with `tqdm_callback_object.display()`) (#1059 <- #1065)
  - add documentation
- □ add CI on windows (#507)
- □ enable CI on OSX
- □ migrate CI Travis => GHA
  - add tests for MacOS & Windows
  - add tests for py3.9 (#1073)
  - update documentation
- □ minify docker build
- □ update tests

- □ misc tidy

v4.51.0 (2020-10-25)

whl|asc

- □ add `{eta}` datetime `bar_format` argument (#1055 <- #1051)
  - e.g. `bar_format='{l_bar}{bar}| {n_fmt}/{total_fmt} [{rate_fmt} ETA:{eta:%y-%m-%d %H:%M}{postfix}]'`
- □ fix py3 CLI `--update` & `--update_to`
- □ replace `nosetests` with `pytest` (#1052, #1045)
- □ add & update tests

v4.50.2 (2020-10-08)

whl|asc

- □ fixed `platform.system()` causing `fork()` warnings (#691)
- □ fixed `contrib.concurrent.process_map` pickling error with `threading.RLock` (#920)
- □ updated documentation & examples
- □ updated CI framework
- □ updated tests
- □ misc code tidy

v4.50.1 (2020-10-07)

whl|asc

- □ fix multiprocessing lock creation leak (#982, #936, #759)
  - fixes #617 which introduced this bug (v4.29.0, released 2019-01-06, undiagnosed until now) where multiple threads could concurrently create and append process locks to a global list, then try to release them without first acquiring :imp:
- ⚡ major test overhaul: fix, update, and speed up
- □ misc CI framework updates
- □ code linting
- □ minor documentation tidy

v4.50.0 (2020-09-28)

whl|asc

- □ add bar `colour` option (#1040 <- #450)
  - for `notebook` as well as `std`/CLI
  - plain text as well as hex (e.g. 'green' or '#00ff00')
- □ `notebook` improvements (#1032 <- #1031)
  - split bar description into separate widget

  - improve `reset()` (handle width and colour)
- □ fix exceptions when `file` is closed (#1041, #1036 <- #1033)
- □ add & update tests
- □ minor documentation updates
- □ move and configure issue/pr templates

v4.49.0 (2020-09-12)

whl|asc

- □ CLI: add `--tee` (#1014 <- #1013)
- □ CLI: add `--update` and `--update_to` (#996 <- #975)
- □ CLI: add `--null` (#996)
- □ CLI: accept `-` instead of `_` in options (e.g. `--update-to`)
- □ make `update()` return `True` on `display` to ease efficient use of custom callbacks (#845)
- □ fix `py>=3` CLI `--delim` encoding error
- □ fix `py>=3.5` version detection in `tqdm.auto` (#1029 <- #1028)
- □ fix final ETA when using `initial` (#1021 <- #689)
- □ update documentation
  - add & update custom callback examples
  - improve help formatting of boolean CLI options
- □ add & update tests

v4.48.2 (2020-08-03)

whl|asc

- □ fix `notebook` formatting

v4.48.1 (2020-08-03)

whl|asc

- □ fix ANSI escape codes breaking `tqdm.write` (#692, #777)
- □ fix long-lived strongref (#1007, https://bugs.python.org/issue39093)
- □ fix cli `--version` crash on missing `git/refs/heads` (#635)
- □ fix `contrib.tenumerate` ignoring `total` (#1017)
- □ fix potential deep => shallow `kwargs` copy issues
- □ improve `notebook` display for unknown `total` (#1015)
- □ make `asyncio` inherit from `std` rather than `auto`
- □ make `auto` multi-inherit from `autonotebook`, `asyncio` on Python3.5+
- □ misc documentation & examples updates
  - mention `docker-compose` requirements (#771)
- □ misc linting & tidy

- misc minor testing framework updates

v4.48.0 (2020-07-16)

whl|asc

- add `tqdm.asyncio` (#1004, #772, #91 <- #65)
    - supports async iterables (`async for row in tqdm(iterable)`)
    - supports iterables
    - supports `coroutine.send` (`tqdm(coroutine).send`)
    - add `as_completed` wrapper (`for f in tqdm.as_completed(fs)`)
- reduce `contrib.discord` logging verbosity
- fix potential `pandas()` modifying `**kwargs` bug
- update documentation
    - add missing inline docstrings
    - update contributing notes
    - update readme
    - add to `examples/` folder
        - async_coroutines
        - coroutine_pipe
    - standardise variable naming
- add tests
    - add benchmark tests against popular alternative libraries
    - test new functionality
    - more thorough `miniters` test (#1003)
- misc framework updates
    - tidy changelog helper
    - add helper bots
        - comment-bot `/tag`
        - post-release-bot
            - wiki updater
            - website updater
        - benchmark regressions
            - full/thorough test on schedule & release
            - quick test on PR

v4.47.0 (2020-06-28)

whl|asc

- add `contrib.discord` (similar to `contrib.telegram`) (#976)
- add `contrib.bells` to auto-enable all extras

- □ add `contrib.utils_worker` for common slow tasks (e.g. web I/O)
  - fix lazy large memory usage & discard unsent messages (unprocessed tasks)
- □ fix slow notebook imports (#955 <- #709)
- □ fix `gui` `TypeError` on unknown `len()` (#971)
- □ misc documentation/error message updates
  - more succinct ImportError on missing `ipywidgets` (#872)
  - fix broken/deprecated link (#981)
  - add inline usage for `contrib.discord` and `contrib.telegram`
- □ misc framework updates
  - add official `py3.8` support (#986)
  - fix `snap` builds

v4.46.1 (2020-06-03)

whl|asc

- □ fix missing `sys.setcheckinterval` in py3.9 (#978)
- □ fix `keras.TqdmCallback` compatibility with `tensorflow==2.2.0` (#979)
- □ update documentation
  - correct `contrib.concurrent` correct `max_workers` (#977)
  - drop prominent mention of `xrange` (#965)
- □ minor linting

v4.46.0 (2020-05-03)

whl|asc

- □ add `contrib.telegram` #949 <- #948
- □ add `bash` tab completion and `--comppath` #946, #947 <- #858
- □ fix exception safety in `external_write_mode` #940
- □ add `requests` examples (#242)
- □ update documentation

v4.45.0 (2020-04-02)

whl|asc

- □ propagate and `close()` on exceptions (#926 <- #548)
- □ fix nested `clear()`
- ⚒ rework nested `close()` (#930 <- #924 <- #918, #677)
  - reduces screen flicker/blank space at the cost of ordering

- □ update all documentation & demos
- □ add and update tests
- □ dev framework: add pre-commit helper

v4.44.1 (2020-03-29)

whl|asc

- □ fix `_utils` `ImportError` => `DeprecationWarning` (#928 <- #927)

v4.44.0 (2020-03-28)

whl|asc

- □ add automatic `nrows` and expose for manual override (#918 -> #924)
- ⚠ expose and warn about small `chunksize` in `tqdm.contrib.concurrent.process_map` (#912)
- □ fix py2 file stream comparison (#727 -> #730)
- ⚠ deprecate `utils._environ_cols_wrapper`
- □ add and update tests
- □ add documentation

v4.43.0 (2020-02-19)

whl

- □ fix `notebook` exceptions (#669 <- #650)
  - `set_description()` when `disable=True`
- □ `contrib.concurrent`: avoid unnecessary `len(iterable)` (#893)
- □ update documentation
  - clarify etymology (#895)
  - fix minor typo (#890)
  - update contributors and years
- □ update tests
- □ fix CI tests for PRs

v4.42.1 (2020-02-03)

whl|asc

- □ support `pandas==1.0.0` (#780, #555)
- □ minor metadata updates
- □ minor CI test update

v4.42.0 (2020-01-25)

whl|asc

- `notebook`: update `disable=None` behaviour (#880)
- `tqdm.keras`: support `tensorflow.keras` as well as `keras` (#885)
- add `contrib` (#882)
  - `tenumerate` (#840 <- #480, #402)
  - `tzip` <= `zip`
  - `tmap` <= `map`
  - add `concurrent` module
    - `thread_map` <= `concurrent.futures.ThreadPoolExecutor.map`
    - `process_map` <= `concurrent.futures.ProcessPoolExecutor.map`
  - add `itertools` module (#225) stub
    - `product`
- add & update tests
- add & update documentation
  - README
  - examples/
- update CI framework

v4.41.1 (2020-01-05)

whl|asc

- `keras` module tidy and fixes
  - fix `batch_size=None` (#869)
  - add missing `pop('size')`

v4.41.0 (2019-12-20)

whl|asc

- trim on `ncols` overflow with ANSI handling (#850, #716 <- #690)
- add `notebook.reset()` (#864)
- add `keras.TqdmCallback` (#867 <- #835)
- documentation updates
  - document newly added features (above)
  - notebook `ncols` percentage/pixels (#276)
- test updates
  - test newly added features (above)
  - add CI for `win` and `osx` (#841)
  - `py2` threading

v4.40.2 (2019-12-11)

whl|asc

- fix `pandas` unhashable func (#862 -> #863)
  - add tests

v4.40.1 (2019-12-11)

whl|asc

- fix floating point imprecision raising errors (#859)
- fix auto gh releases (travis-ci/dpl#914)
- update CONTRIBUTING (#84 (comment))

v4.40.0 (2019-12-01)

whl|asc

- officially support `float` for `n` and `total` (#802)
  - `notebook`: use `FloatProgress` <= `IntProgress` (#471, #456)
  - allow imprecision (`n <= total + epsilon`) (#849)
- fix unicode bar format arguments (#803 -> #851)
- add `contrib` submodule (#815)
- add `wrapattr`, `utils.CallbackIOWrapper`, `contrib.DummyTqdmFile` (#84 -> #844)
- update tests
- update documentation
- tidy automatic `snap` deployments
- minor doc update (#854)

v4.39.0 (2019-11-22)

whl|asc

- add `pandas` builtin operations check (#843 <- #697)
- avoid unnecessary `dedent` (#837)
- remove unneeded bar logic
- misc code tidy
- update documentation
  - document default argument overriding (#370)
  - add missing `isatty()` (#713)
  - update badges
  - add code of conduct
- update framework
  - clean CI stages
  - update CI default `py3.6` => `py3.7`

  - automate snap releases (#686 <- #685)

v4.38.0 (2019-11-09)

whl|asc

- □ support `lock_args` for e.g. non-blocking intermediate writes (#838 -> #839)
  - use `self.refresh()` in more places
  - add and update performance tests
  - support args for `TqdmDefaultWriteLock.acquire()`
- □ fix colorama on win (#678, #764)
- □ framework updates
  - CI cleanup
    - move to stages (check/test/deploy)
    - auto deploy notes
    - deploy only on parent repo
    - separate docker deploy
  - update documentation sync/generation
- □ update examples
- □ update documentation
- □ add and update tests
- □ misc tidy and linting

v4.37.0 (2019-10-31)

whl|asc

- □ potential future pandas fix (#824)
- □ better unicode widechar support (#803 -> #410, #805)
- □ fix example/tqdm_wget unknown total (#826)
- □ add `stacklevel=2` to warnings (#821)
- □ misc framework updates
  - snapcraft version fix
  - update issue/pr templates (#830)
  - update unit tests
- □ misc documentation updates
  - update parallel (multiprocessing, ThreadPoolExecutor) example (#407)
  - add slides, video images
  - add CII Best Practices badge
  - add repology badge
  - update badge icons

v4.36.1 (2019-09-20)

whl|asc

- □ fix CLI entrypoint

v4.36.0 (2019-09-17)

whl|asc

- □ expose more custom format params
- □ potential thread safety fix (#548)
- ⚒ update submodule architecture (#198 -> #800)
  - backward-compatibility
  - expose `utils`
  - rename `main` -> `cli`
  - add/fix tests
  - fix minor cached var optimisation
  - `gui` partial upgrade (#790)
  - `notebook` upgrade (#790)
    - support `{bar}` in `bar_format` (#594)
    - inherit methods including `set_*(refresh)` (#740 -> #741)
  - ready for `contrib` releases (#252)
- □ support custom `bar_format` with unknown `total`
- □ fix `tqdm_notebook` red block on explicit `file=None` (#791)
- □ update notebook to fully support custom `bar_format` (#594, #649)
- □ misc tidy
  - warnings
  - linting
  - update submodule contribution guide (#252)
  - parallel `(py)make test`
  - fix Zenodo metadata
- □ update documentation
  - `pandas`, `notebook`, `gui` (#245)
  - indentation
  - update multiprocessing and multithreading examples (#796)

v4.35.0 (2019-08-24)

whl|asc

- □ add `{bar}` format specifier (#623 -> #799)
  - `[width][type]`
- □ add tests and documentation

- ⚡ update performance tests

v4.34.0 (2019-08-18)

whl|asc

- □ add `leave=None` convenience option for `leave = position == 0`
- □ ensure nested completed bars respect `leave=True` (#230)
- □ ensure nested bars are cleared before being moved up (#795)
- □ support both and `=` syntax for CLI options (#761 -> #774)
- □ misc documentation updates (#778, #782)
- □ fix/update tests
- □ update GitHub docker package deployment

v4.33.0 (2019-08-08)

whl|asc

- □ fix `pandas==0.25` API change (#780)
- □ add contributor badges (`sourcerer`)
- □ fix py26 travis CI
- □ metadata fixes
  - `.zenodo.json`
  - `CODEOWNERS`
- □ GitHub package registry docker deployment
- □ minor linting
- □ documentation updates (#773)

v4.32.2 (2019-06-18)

whl|asc

- □ fix 'Set changed size during iteration' (#481 -> #700, #754)
- □ add `tqdm.autonotebook` check for `$VSCODE_PID` (#747 -> #751)
- □ fix missing `multiprocessing` in Jython (#698)
- □ make `bool(tqdm(iter))` consistent with `bool(iter)` (#353 -> #694)
- □ add and update unit tests
- □ add and update documentation & framework
  - update notable contributors list
  - update usage docstrings (#714, #715)
  - add funding
  - add `awesome-python` badge
  - trove classifier update

  - update demo notebook badges
  - add JOSS DOI
  - add Zenodo metadata
  - move unnecessary root clutter

v4.32.1 (2019-05-13)

whl|asc

- □ fix `notebook` with unknown `total` (#743)

v4.32.0 (2019-05-13)

whl|asc

- □ support `unit_scale` in `notebook`
- □ support negative update (#432, #545)
- □ add `reset()` function (#547, #545)
- □ add `[python setup.py] make run`
- □ add and update documentation
  - example of dynamic usage (#735, #545, #547, #432, #374)
  - note writing issues (#737)
  - update badges
  - add PyData2019 slides link
  - add JOSS paper
  - update manpages
  - add docker install
  - add snapcraft install
  - notebooks: add binder, rename RMOTR => notebooks.ai (#679)
  - prettify and unify contributors/maintainers/authors
- □ CI and release framework updates
  - add snapcraft snaps (#647)
  - add travis auto-deployment (#685)
    - PyPI releases
    - docker devel/releases
  - update deployment dev docs
  - fix travis deploy pymake
  - update .gitinore
  - add & update unit tests
  - automate more documentation

v4.31.1 (2019-02-10)

whl|asc

- □ fix `file.encoding==None` caused by #598 (#673 -> #676)
  - add tests
- □ tidy code, update authors and tests
  - add `setup.py` test to CI

v4.31.0 (2019-02-09)

whl|asc

- □ Write bytes to `stdout`/`stderr` in py2 (https://bugs.python.org/issue21363, #589 -> #598)
  - Add `write_bytes` parameter for py2/3 default override
- □ support custom bar characters vias `ascii` parameter (#223 -> #227)
  - implement, document and test custom bar spinners
- □ support custom `bar_format` with unknown `total` (#282 -> #284)
- □ fix `TypeError` when `unit_scale` and unknown `total`
- □ document `format_dict`
- □ expose `elapsed_s`, `remaining_s` to `bar_format`
- □ add `unit` option to `bar_format` (#368)
- □ auto `README.rst` generation framework
  - add notes in `CONTRIBUTING.md`
- □ update tests
- □ update documentation
- □ misc code tidy

v4.30.0 (2019-01-26)

whl|asc

- □ avoid moving when `leave=False`
- □ generalise nested bar tests
- □ add `display()` helper function
- □ add and expose `format_dict` (#562, #482, #494, #656) + allow arbitrary keyword arguments for `format_meter` + document `format_dict` overriding + add tests for overriding + related: #660 #587
- □ add and tidy developer documentation
- □ fix py3 CLI pipe decoding error (#663)
- □ framework overhaul (developing, testing, Travis CI updates)
- □ tidy code and style (flake8)
- ⚒ `setup.py` refactor + `setup.py make` * add support for multi-line commands * tidy and efficiency + replace `setup.py`'s built-in `make` -> `py-make>=0.1.9` (#290) + add `requirements-dev.txt` and `extras_require[dev]`
- □ add, update and tidy tests
- □ include interactive Jupyter demonstration (#652 -> #659)
- □ update documentation + add FAQ about py2 pipes (#359) + add badges + update examples + update

authors

v4.29.1 (2019-01-11)

whl|asc

- □ fix `classmethod` lock bug (#617: #457 -> #658)
- □ add unit test to prevent regression

v4.29.0 (2019-01-06)

whl|asc

- □ Avoid global multiprocessing locks (#611 -> #617)
- □ Add support for infinite iterables (#651)
- □ Fix missing attr pos when used in multi-threaded environment (#573)
- □ Do not join `TMonitor` if it is the current thread (#613 -> #641)
- □ Add OpenBSD NIX support (#638)
- □ Unit tests, general documentation fixes and tidying (e.g. #642)
- □ CI travis improvements
  - `py37-dev` -> `py37` (#622)
  - fix `py26`

v4.28.1 (2018-10-21)

whl|asc

- □ fix `--manpath` file not found
- □ expose `main():argv`
- □ add unit tests
- □ release framework tidy
  - update `.gitattributes`
  - remove deprecated `make`/`dist` options

v4.28.0 (2018-10-20)

whl|asc

- □ remove installation of man pages by default (#460, #628)
- □ CLI:add `--manpath` option (#629)
- □ documentation additions and fixes

v4.27.0 (2018-10-15)

whl|asc

- □ fix `str.isnumeric` #605
- □ fix `WeakSet` `KeyError` #548, #553, #596 -> #607
- □ stop `tqdm_notebook` description truncation #582 -> #599
- □ include `unit_scale` for `rate` #608
- □ add `auto` -> nowarning `autonotebook`
- □ add better postfix numeric formatting #621
- ⚒ minor refactoring #609 -> #616
- □ update documentation
- □ add unit tests
- □ fix py26 CI

v4.26.0 (2018-09-11)

whl|asc

- □ fix `smoothing` (#566 -> #601)
- □ `pandas` updates + address the `FutureWarning` in `pandas`, drop `pandas` test in py34, add in py37 (#603) + support `pandas` `axis='index' or 'columns'` (#570)
- □ minor documentation updates (#597, #600, #606) + developer notes

v4.25.0 (2018-08-20)

whl|asc

- □ stop monitor on shutdown (#571, #572)
- □ fix `find_packages()` for submodules (#593) + properly add `autonotebook` (#586, 1cf3393)
- □ ignore ANSI escape codes in the bar length calculation (#450, #591 -> #592)
- □ update documentation (#243)

v4.24.0 (2018-07-26)

whl|asc

- □ `autonotebook` submodule for automatic selection of notebook/CLI bar (#443, #508) + update/add `pandas()` example documentation (#474)
- □ `NameError:IntProgress` changed to a more helpful `ImportError` (#187, #451, #558)
- □ support `bool()` cast when `disable=True` (#574)
- □ fix `format_sizeof` hundreds rounding (#579 -> #581)
- □ ensure URLs in documentation are secure (`https`)

v4.23.4 (2018-05-22)

whl|asc

- □ Support pandas 0.23.0 `core.groupby` module layout (#555 -> #554)

- □ Add python_requires to help pip (#557)
- □ minor maintenance updates
  - CI updates: drop travis py33 due to `tox` (tox-dev/tox#648)
  - minor code tidy

v4.23.3 (2018-05-22)

whl|asc

- ⚠ suppress 10 second warning (refix #323)

v4.23.2 (2018-05-02)

whl|asc

- □ minor import syntax fix #496
- □ re-fix nested overlaps #477
- □ update documentation and examples
- □ code tidy and abstraction

v4.23.1 (2018-04-25)

whl|asc

- □ fix `AttributeError`s (#546) + `pos` on initialisation (#323, #510 -> #544) + `fp` on `write()` (#383)
- □ fix py34 CI
- □ update documentation

v4.23.0 (2018-04-15)

whl|asc

- □ Fix `disable=True` where `iterable` has no `len()` and is not `None` (#539)
- □ Add `ncols` to specify `tqdm_notebook` bar width (#276 -> #292)
- □ allow custom `pandas` `total` (#364 -> #535)
- □ Add `progress_apply` for `pandas.(Series|DataFrame).(rolling|expanding)` (#530 -> #537)
- □ unit tests, pep8 tidy
- □ `postfix` non-`dict` documentation note

v4.22.0 (2018-04-11)

whl|asc

- □ allow direct non-string assignment to `postfix` member (lists, dicts, etc) (#382 -> #534)
- □ documentation updates
- □ unit tests

v4.21.0 (2018-04-08)

whl|asc

- □ fix #532 `write()` with manual `position`
  - more robust free position finding
- □ add `TqdmWarning` base class
- □ fix GUI `__del__()`
  - add `TqdmExperimentalWarning`
- □ add, tidy & fix unit tests
- □ documentation
  - wiki release notes
  - update contributors
  - fix `7zx` example

v4.20.0 (2018-04-03)

whl|asc

- □ `pandas` wrapper: remove `*args` and add better `total` handling (#244, #299, #322, #366 -> #524)
- □ document windows unicode know issues (#454)
- □ suppress `RuntimeError: Set changed size during iteration` -> `TqdmSynchronisationWarning` (#481)

v4.19.9 (2018-03-27)

whl|asc

- □ fix monitor thread termination and update tests (#527)

v4.19.8 (2018-03-27)

whl|asc

- □ fix monitoring thread issues
  - avoid uninitialised instance manipulation (#493)
  - fix thread starting `RuntimeError` (#522 -> #523)
- □ these release notes (#529)

v4.19.6 (2018-02-27)

whl|asc

- □ CLI improvements + `--bytes` implies `--unit B --unit_scale --unit_divisor 1024` (#503) + allow hyphens in CLI arguments (#511)
- □ synchronisation: fix for `sem_open` on `aarch64` (#513)
- □ framework update + CI/tests, year 2018, py37, badges, documentation

v4.19.5 (2017-12-10)

whl|asc

- □ multiprocess/lock fixes (#457)
- □ `set_description` in notebook (#345 -> #475)
- □ minor tidy (#476)
- □ documentation updates

v4.19.4 (2017-10-15)

whl|asc|gz|asc

- □ fix `Lock:NotImplementedError` on certain systems (#466 -> #468)
- □ use recursive locks (#469 -> #468) - fix deadlocks
- □ tidy (#448)
- □ `flush()` on `moveto()` (#398 -> #399, #420, #467)
- □ update tests and benchmarks

v4.19.2 (2017-10-08)

whl|asc|gz|asc

- □ `flush()` on `refresh()` (#459 from #317)
- □ status printer updates (#331) + use `sp()` in `refresh()` + remove redundant `clear():nomove=False`
- □ misc minor documentation updates
- □ unit tests

v4.19.1 (2017-10-03)

whl|asc|gz|asc

- □ `rate_(no)inv(_fmt)` (#72, b228bc3)
- □ `__repr__()` tidy (#389)
- □ fix `write()` before initialisation `AttributeError: _lock` (#457)
- □ `man` pages
- □ documentation updates

v4.18.0 (2017-09-30)

whl|asc|gz|asc

- □ Thread safety! (#285 -> #291 -> #329, #407, #417)
- □ Ease redirection of `sys.stdout`/`stderr` (#422)
- □ Minor internal stream bugfix (#439)
- □ `AttributeError` fixes (#323, #324, #418)

Related to:

- □ Misc bugs (#334)
- □ `concurrent.futures` (#97)
- □ Multi-`tqdm` (#143)
- □ `flush()` and `refresh()` (#331)
- □ Newline on `refresh()` (#361)
- □ Nested trees (#384)
- □ Manually positioned nested bars clearing (#385)

v4.17.1 (2017-09-26)

whl|asc|gz|asc

- □ initialise `n` even when disabled (#438)
- □ better unicode detection (#437)
- □ `__repr__()` bugfix (#441)

v4.17.0 (2017-09-20)

whl|asc|gz|asc

- □ easier redirect of `stdout` (#426 -> #431) - `cls.external_write_mode() @contextmanager`
- □ `refresh=True` for `set_description`/`update`(`_str`) (#317 -> #377)

v4.16.0 (2017-09-20)

whl|asc|gz|asc

- □ more formatting control (#347 -> #362) - `set_postfix_str()`, `set_description_str()` - move `": "` to `bar_format` - complements #266, #270
- □ better CLI support (e.g. hyphenated paths) (#421 -> #424)
- □ more unit tests (#411) - upgrade to pypy2/3-5.8.0 - add py37-dev
- □ doc updates

v4.15.0 (2017-07-29)

whl|asc

- □ minor fixes (#395)
- □ minor documentation updates (#400, #401, #412)
- □ better `pandas` support (#351 -> #409)
- □ Jupyter notebook fix (#352, #369 -> #373, #405)
- ⚡ optimise `setcheckinterval`/`setswitchinterval` (#376)
- □ add `unit_scale` (#273, #295 -> #378)

v4.14.0 (2017-07-29)

- □ add `unit_divisor` #354 -> #356
- □ update examples

v4.13.0 (2017-05-29)

whl|asc

- □ support unknown systems https://github.com/warner/magic-wormhole/issues/158, #76, #344 -> #390
- □ support nested/multi-bars better #384, #285, #329 -> #385
- □ ease redirection of output (default: `sys.stderr` -> `None`) https://github.com/xonsh/xonsh/issues/2058, #329, #275 -> #293 + add examples/redirect_print.py
- □ minor documentation updates

v4.12.0 (2017-05-29)

whl|asc

- □ fix monitor race condition #338 -> #339
- □ add explicit NetBSD support #344
- □ documentation tidy

v4.11.2 (2017-01-24)

whl|asc|gz|asc

- □ thread safety #332
- □ address `DeprecationWarning` #319 -> #326
- □ version extra #328 -> #330
  - security fix (CVE-2016-10075)
  - add non-master branch name
- □ minor safety
- □ minor documentation and comments

v4.11.1 (2017-01-23)

whl|asc|gz|asc

Officially support Python 3.6, addressing #333

v4.11.0 (2017-01-12)

whl|gz

- □ `postfix` argument added #266->#270
- □ fix tests for

    - `numpy` #272->#302
    - `PyPy3` #318->#325
- □ `7z` extraction example script
- □ new logo #315
- □ hit counter with logo colours and stats
- □ documentation updates
- □ safer `setup.py` `Makefile` emulation
- □ better github contributions integrations and Code of Conduct #310

v4.10.0 (2016-11-12)

gz|zip

- □ fix maxinterval to adjust miniters to mininterval #249
- □ minor bugfixes (eg #288)
- ⚡ misc optimisations, unit tests, and benchmarks (eg #298)
- □ documentation updates

v4.9.0 (2016-10-31)

gz|zip

- □ monitoring thread to automatically to adjust `miniters` and avoid exceeding `maxinterval`
    - test race conditions
    - kill on KeyboardInterrupt + instant refresh
    - low performance overhead, option to disable
- □ new CLI argument: `--bytes` to count chars
- □ disable automatically on non-TTY (#281)
- □ fixed `refresh()` and `clear()` if disabled (#286)
- □ misc fixes
- □ documentation
- □ unit tests
- ⚡ optimisations, style fixes
- □ codacy badge
- ⚠ safety and warning suppression
- □ ignore coverall failures (#283)
- □ fix broken link in README (#262)
- ⚡ Fix `time()` in `refresh()` + avoid `write()` races (#264)
- □ Fix `__repr__()` #269
- □ Fix `update(0)` causing `ZeroDivisionError` (on first call) #268
- □ Faster simple bar for perf test (#257)
- □ cleaner `tqdm_notebook` exit when not run in a notebook (#267)

- □ Disable coveralls because of duplicate comments on github (#263)
- □ Migrate codecov config -> codecov.yml

v4.8.4 (2016-08-17)

whl|gz

- ⚡ 14% overhead reduction
- □ fix exception when closing `tqdm_notebook` without a `total`
- □ support more pandas modules
  - pandas.Series.progress_map
  - pandas.DataFrame.progress_applymap
  - Panel.progress_apply
  - PanelGroupBy.progress_apply
  - GroupBy.progress_aggregate
  - GroupBy.progress_transform
- □ faster CI builds
- □ misc documentation update and link to wiki

v4.8.1 (2016-07-25)

- ⚠ custom tqdm warning/exceptions
- □ tests now multiprocessed and include pandas

v4.8.0 (2016-07-25)

- ⚠ `tqdm.pandas` classmethod replaces `tqdm_pandas` function (deprecated)
- □ `ipython` aliased imports fixes
- □ `write` without any instances
- □ readme update
- □ examples update
- □ test fixes and updates

v4.7.6 (2016-06-30)

- □ support for pandas series
- □ support for numpy arrays
- □ tidy & coverage

v4.7.4 (2016-06-04)

whl|gz

- □ `pandas.DataFrame.apply` support

- □ `tqdm_notebook.write()`
- □ `tqdm_gui.write()`
- □ misc notebook fixes
- ⚡ delay notebook imports (>80% reduction in import time when unused)
- □ optional `colorama` only on windows
- □ documentation tidy
- □ more & faster tests

v4.7.0 (2016-05-15)

whl|gz

- □ CLI custom delimiter support
- □ More robust CLI argument parsing
  - either one or two `-` preceding kwargs (eg `-ascii` or `--ascii`)
  - no positional args (so no need to quote multi-word `-desc` for example)
  - `bool` kwargs with implicit value=`True` still supported
  - optional `=` sign after kwargs
- □ tests
- □ doc updates, badges
  - pyversions, doi, licence, github

v4.6.2 (2016-05-15)

whl|gz|exe|zip

- □ Fix warnings due to IPython/Jupyter widget

v4.6.1 (2016-05-15)

whl|gz|exe|zip

- □ add IPython/Jupyter support
- □ fix `{bar}` length breaking display when using custom `bar_format`
- □ add a no inversion rate formatting for `bar_format` (i.e., always show `it/s` and never `s/it`)

v4.5.0 (2016-04-25)

whl|gz|zip

- □ print messages within (nested) loops using `tqdm.write(...)`

v4.4.3 (2016-04-24)

whl|gz|zip

- ▯ remove `docopt` dependency for CLI
- ▯ re-added `update(0)` ability to trigger refresh
- ▯ minor fixes
  - re-added (OSI approved) licence to source
  - dev versioning auto-fallback
- ⚡ slight speed optimisation
- ▯ tests
- ▯ documentation

v4.4.1 (2016-04-22)

whl|gz|zip

- ▯ Module execution in CLI pipes!
  - `tqdm` executable
  - documentation
  - tests
- ▯ MSYS aded to supported envs
- ▯ better VTE support
- ▯ dev versioning
- ▯ tests

v4.2.0 (2016-04-22)

whl|gz|zip

- ▯ MPL licence!
- ▯ slightly updated logo
- ▯ update examples
- ▯ `update(0)` allowed
- ▯ better VTE support
  - ascii fallback when unicode not available

v4.0.0 (2016-02-04)

whl|gz|zip

- ▯ automate `nested`
  - deprecate `nested`
  - add manual override control with `position`
- ▯ documentation and example updates, coverage and tests
- ▯ make `leave==True` by default (was `False`)

v3.8.0 (2016-01-31)

whl|gz|zip

- □ update(n<1) raises error
- □ sanitise and update examples and README
- □ fix setup.py encoding error
- □ remove py32 from Travis
- □ changelog more accessible from README

v3.7.1 (2016-01-13)

whl|gz|exe|zip

- □ Added more OS support (FreeBSD, Solaris/SunOS)
- □ Added Python3.5 support and tests
- □ Added IronPython2.7 support
- □ Fix Makefile management for Python3
- □ Added relative timing for tests
- □ More stable smoothed ETA
- ⚒ Refactoring *everything* into one big `tqdm` class for even more subclassing fun

v3.6.0 (2015-12-31)

whl|gz|zip

Added support for "restarting" a progressbar after a long pause without making the iteration rate go haywire. Simply call the `unpause()` method on the `tqdm` instance that is about to be restarted. Can also start `tqdm` skipping a number of steps specified by `initial` (without bias-ing ETA).

v3.5.0 (2015-12-31)

whl|gz|zip

- □ Fully customisable output format using the `bar_format` keyword argument.
  - Valid strings: bar, n, n_fmt, total, total_fmt, percentage, rate, rate_fmt, elapsed, remaining, l_bar, r_bar, desc
  - Example: `bar_format='{desc}{bar}{rate:03.3d}'`
  - Default: '{l_bar}{bar}{r_bar}', where l_bar is '{desc}{percentage:3.0f}%|' and r_bar is '|{n_fmt}/{total_fmt} [{elapsed_str}<{remaining_str}, {rate_fmt}]'.
- □ Solaris, FreeBSD support mentioned
- □ Link redirects circumvented

v3.3.0 (2015-12-31)

whl|gz|zip

- □ Support added for pandas `apply` => `progress_apply`
- □ 80-char width when running interactive `help()`
- □ update examples using `with` syntax

v3.1.4 (2015-11-28)

whl|gz|exe|zip

Added `smoothing` and `maxinterval` features for smooth progress bar and rate for unconstant, variable changes in update times, and abstracted gui to its own class and file.

Also added a performance test to prevent performance regressions.

v2.2.4 (2015-11-08)

whl|gz|exe|zip

Added the dynamic management of the bar on window resize, plus a better internal testing/build workflow using `python setup.py make [alias]`.

v2.0.0 (2015-10-11)

whl|gz|exe|zip

First public release of the next generation of tqdm, with lots of new features, more speed, more unit testing, more coverage (100% yeah!), more beer and coffee.

This release can also be downloaded on PyPi here.

tqdm documentation

tqdm.tk

Tkinter GUI progressbar decorator for iterators.

Usage:

```
>>> from tqdm.tk import trange, tqdm
>>> for i in trange(10):
...     ...
```

tqdm_tk Objects

```
class tqdm_tk(std_tqdm)
```

[view source]

Experimental Tkinter GUI version of tqdm!

Note: Window interactivity suffers if `tqdm_tk` is not running within a Tkinter mainloop and values are generated infrequently. In this case, consider calling `tqdm_tk.refresh()` frequently in the Tk thread.

__init__

```
def __init__(*args, **kwargs)
```

[view source]

This class accepts the following parameters *in addition* to the parameters accepted by `tqdm`.

Parameters

- **grab***: bool, optional*
  Grab the input across all windows of the process.
- **tk_parent***: `tkinter.Wm`, optional*
  Parent Tk window.
- **cancel_callback***: Callable, optional*
  Create a cancel button and set `cancel_callback` to be called when the cancel or window close button is clicked.

cancel

```
def cancel()
```

[view source]

`cancel_callback()` followed by `close()` when close/cancel buttons clicked.

reset

```
def reset(total=None)
```

[view source]

Resets to 0 iterations for repeated use.

Parameters

- **total**: *int or float, optional. Total to use for the new bar.*

ttkrange

```
def ttkrange(*args, **kwargs)
```

[view source]

Shortcut for `tqdm.tk.tqdm(range(*args), **kwargs)` .

tqdm.utils

General helpers required for `tqdm.std`.

envwrap

[view source]

```
def envwrap(prefix, types=None, is_method=False)
```

Override parameter defaults via `os.environ[prefix + param_name]`. Maps UPPER_CASE env vars map to lower_case param names. camelCase isn't supported (because Windows ignores case).

Precedence (highest first): - call (`foo(a=3)`) - environ (`FOO_A=2`) - signature (`def foo(a=1)`)

Parameters

- **prefix***: str*
  Env var prefix, e.g. "FOO_"
- **types***: dict, optional*
  Fallback mappings `{'param_name': type, ...}` if types cannot be inferred from function signature. Consider using `types=collections.defaultdict(lambda: ast.literal_eval)`.
- **is_method***: bool, optional*
  Whether to use `functools.partialmethod`. If (default: False) use `functools.partial`.

Examples

```
$ cat foo.py
from tqdm.utils import envwrap
@envwrap("FOO_")
def test(a=1, b=2, c=3):
    print(f"received: a={a}, b={b}, c={c}")

$ FOO_A=42 FOO_C=1337 python -c 'import foo; foo.test(c=99)'
received: a=42, b=2, c=99
```

CallbackIOWrapper Objects

[view source]

```
class CallbackIOWrapper(ObjectWrapper)
```

disp_len

[view source]

```
def disp_len(data)
```

Returns the real on-screen length of a string which may contain ANSI control codes and wide chars.

disp_trim

[view source]

```
def disp_trim(data, length)
```

Trim a string which may contain ANSI control characters.

tqdm.gui

Matplotlib GUI progressbar decorator for iterators.

Usage:

```
>>> from tqdm.gui import trange, tqdm
>>> for i in trange(10):
...     ...
```

tqdm_gui Objects

```
class tqdm_gui(std_tqdm)
```

[view source]

Experimental Matplotlib GUI version of tqdm!

tgrange

```
def tgrange(*args, **kwargs)
```

[view source]

Shortcut for `tqdm.gui.tqdm(range(*args), **kwargs)` .

tqdm.std

Customisable progressbar decorator for iterators. Includes a default `range` iterator printing to `stderr`.

Usage:

```
>>> from tqdm import trange, tqdm
>>> for i in trange(10):
...     ...
```

tqdm Objects

```
class tqdm(Comparable)
```

[view source]

Decorate an iterable object, returning an iterator which acts exactly like the original iterable, but prints a dynamically updating progressbar every time a value is requested.

Parameters

- **iterable**: *iterable, optional*
  Iterable to decorate with a progressbar. Leave blank to manually manage the updates.
- **desc**: *str, optional*
  Prefix for the progressbar.
- **total**: *int or float, optional*
  The number of expected iterations. If unspecified, len(iterable) is used if possible. If float("inf") or as a last resort, only basic progress statistics are displayed (no ETA, no progressbar). If `gui` is True and this parameter needs subsequent updating, specify an initial arbitrary large positive number, e.g. 9e9.
- **leave**: *bool, optional*
  If [default: True], keeps all traces of the progressbar upon termination of iteration. If `None`, will leave only if `position` is `0`.
- **file**: *`io.TextIOWrapper` or `io.StringIO`, optional*
  Specifies where to output the progress messages (default: sys.stderr). Uses `file.write(str)` and `file.flush()` methods. For encoding, see `write_bytes`.
- **ncols**: *int, optional*
  The width of the entire output message. If specified, dynamically resizes the progressbar to stay within this bound. If unspecified, attempts to use environment width. The fallback is a meter width of 10 and no limit for the counter and statistics. If 0, will not print any meter (only stats).
- **mininterval**: *float, optional*
  Minimum progress display update interval [default: 0.1] seconds.
- **maxinterval**: *float, optional*
  Maximum progress display update interval [default: 10] seconds. Automatically adjusts `miniters` to correspond to `mininterval` after long display update lag. Only works if `dynamic_miniters` or monitor thread is enabled.
- **miniters**: *int or float, optional*

Minimum progress display update interval, in iterations. If 0 and `dynamic_miniters`, will automatically adjust to equal `mininterval` (more CPU efficient, good for tight loops). If > 0, will skip display of specified number of iterations. Tweak this and `mininterval` to get very efficient loops. If your progress is erratic with both fast and slow iterations (network, skipping items, etc) you should set miniters=1.

- **ascii***: bool or str, optional*
  If unspecified or False, use unicode (smooth blocks) to fill the meter. The fallback is to use ASCII characters " 123456789#".
- **disable***: bool, optional*
  Whether to disable the entire progressbar wrapper [default: False]. If set to None, disable on non-TTY.
- **unit***: str, optional*
  String that will be used to define the unit of each iteration [default: it].
- **unit_scale***: bool or int or float, optional*
  If 1 or True, the number of iterations will be reduced/scaled automatically and a metric prefix following the International System of Units standard will be added (kilo, mega, etc.) [default: False]. If any other non-zero number, will scale `total` and `n`.
- **dynamic_ncols***: bool, optional*
  If set, constantly alters `ncols` and `nrows` to the environment (allowing for window resizes) [default: False].
- **smoothing***: float, optional*
  Exponential moving average smoothing factor for speed estimates (ignored in GUI mode). Ranges from 0 (average speed) to 1 (current/instantaneous speed) [default: 0.3].
- **bar_format***: str, optional*
  Specify a custom bar string formatting. May impact performance. [default: '{l_bar}{bar}{r_bar}'], where l_bar='{desc}: {percentage:3.0f}%|' and r_bar='| {n_fmt}/{total_fmt} [{elapsed}<{remaining}, ' '{rate_fmt}{postfix}]' Possible vars: l_bar, bar, r_bar, n, n_fmt, total, total_fmt, percentage, elapsed, elapsed_s, ncols, nrows, desc, unit, rate, rate_fmt, rate_noinv, rate_noinv_fmt, rate_inv, rate_inv_fmt, postfix, unit_divisor, remaining, remaining_s, eta. Note that a trailing ": " is automatically removed after {desc} if the latter is empty.
- **initial***: int or float, optional*
  The initial counter value. Useful when restarting a progress bar [default: 0]. If using float, consider specifying `{n:.3f}` or similar in `bar_format`, or specifying `unit_scale`.
- **position***: int, optional*
  Specify the line offset to print this bar (starting from 0) Automatic if unspecified. Useful to manage multiple bars at once (eg, from threads).
- **postfix***: dict or *, optional*
  Specify additional stats to display at the end of the bar. Calls `set_postfix(**postfix)` if possible (dict).
- **unit_divisor***: float, optional*
  [default: 1000], ignored unless `unit_scale` is True.
- **write_bytes***: bool, optional*
  Whether to write bytes. If (default: False) will write unicode.
- **lock_args***: tuple, optional*
  Passed to `refresh` for intermediate output (initialisation, iterating, and updating).
- **nrows***: int, optional*
  The screen height. If specified, hides nested bars outside this bound. If unspecified, attempts to use environment height. The fallback is 20.
- **colour***: str, optional*
  Bar colour (e.g. 'green', '`00ff00`').
- **delay***: float, optional*

Don't display until [default: 0] seconds have elapsed.

- **gui***: bool, optional*
  WARNING: internal parameter - do not use. Use tqdm.gui.tqdm(...) instead. If set, will attempt to use matplotlib animations for a graphical output [default: False].

Returns

- **out***: decorated iterator.*

monitor_interval

set to 0 to disable the thread

format_sizeof

[view source]

```
@staticmethod
def format_sizeof(num, suffix='', divisor=1000)
```

Formats a number (greater than unity) with SI Order of Magnitude prefixes.

Parameters

- **num***: float*
  Number (>= 1) to format.
- **suffix***: str, optional*
  Post-postfix [default: ''].
- **divisor***: float, optional*
  Divisor between prefixes [default: 1000].

Returns

- **out***: str*
  Number with Order of Magnitude SI unit postfix.

format_interval

[view source]

```
@staticmethod
def format_interval(t)
```

Formats a number of seconds as a clock time, [H:]MM:SS

Parameters

- **t***: int*
  Number of seconds.

Returns

- **out***: str*
  [H:]MM:SS

format_num

[view source]

```
@staticmethod
def format_num(n)
```

Intelligent scientific notation (.3g).

Parameters

- **n**: *int or float or Numeric*
  A Number.

Returns

- **out**: *str*
  Formatted number.

status_printer

[view source]

```
@staticmethod
def status_printer(file)
```

Manage the printing and in-place updating of a line of characters. Note that if the string is longer than a line, then in-place updating may not work (it will print a new line at each refresh).

format_meter

[view source]

```
@staticmethod
def format_meter(n,
                 total,
                 elapsed,
                 ncols=None,
                 prefix='',
                 ascii=False,
                 unit='it',
                 unit_scale=False,
                 rate=None,
                 bar_format=None,
                 postfix=None,
                 unit_divisor=1000,
                 initial=0,
                 colour=None,
                 **extra_kwargs)
```

Return a string-based progress bar given some parameters

Parameters

- **n**: *int or float*
  Number of finished iterations.
- **total**: *int or float*
  The expected total number of iterations. If meaningless (None), only basic progress statistics are displayed (no ETA).
- **elapsed**: *float*
  Number of seconds passed since start.
- **ncols**: *int, optional*
  The width of the entire output message. If specified, dynamically resizes `{bar}` to stay within this bound [default: None]. If `0`, will not print any bar (only stats). The fallback is `{bar:10}`.
- **prefix**: *str, optional*
  Prefix message (included in total width) [default: '']. Use as {desc} in bar_format string.
- **ascii**: *bool, optional or str, optional*
  If not set, use unicode (smooth blocks) to fill the meter [default: False]. The fallback is to use ASCII characters " 123456789#".
- **unit**: *str, optional*
  The iteration unit [default: 'it'].
- **unit_scale**: *bool or int or float, optional*
  If 1 or True, the number of iterations will be printed with an appropriate SI metric prefix (k = 10^3, M = 10^6, etc.) [default: False]. If any other non-zero number, will scale `total` and `n`.

- **rate***: float, optional*
  Manual override for iteration rate. If [default: None], uses n/elapsed.
- **bar_format***: str, optional*
  Specify a custom bar string formatting. May impact performance. [default: '{l_bar}{bar}{r_bar}'], where l_bar='{desc}: {percentage:3.0f}%|' and r_bar='| {n_fmt}/{total_fmt} [{elapsed}<{remaining}, ' '{rate_fmt}{postfix}]' Possible vars: l_bar, bar, r_bar, n, n_fmt, total, total_fmt, percentage, elapsed, elapsed_s, ncols, nrows, desc, unit, rate, rate_fmt, rate_noinv, rate_noinv_fmt, rate_inv, rate_inv_fmt, postfix, unit_divisor, remaining, remaining_s, eta. Note that a trailing ": " is automatically removed after {desc} if the latter is empty.
- **postfix***: *, optional*
  Similar to `prefix` , but placed at the end (e.g. for additional stats). Note: postfix is usually a string (not a dict) for this method, and will if possible be set to postfix = ', ' + postfix. However other types are supported (`382`).
- **unit_divisor***: float, optional*
  [default: 1000], ignored unless `unit_scale` is True.
- **initial***: int or float, optional*
  The initial counter value [default: 0].
- **colour***: str, optional*
  Bar colour (e.g. 'green', ' `00ff00` ').

Returns

- **out***: Formatted meter and stats, ready to display.*

write

[view source]

```
@classmethod
def write(cls, s, file=None, end="\n", nolock=False)
```

Print a message via tqdm (without overlap with bars).

external_write_mode

[view source]

```
@classmethod
@contextmanager
def external_write_mode(cls, file=None, nolock=False)
```

Disable tqdm within context and refresh tqdm when exits. Useful when writing to standard output stream

set_lock

[view source]

```
@classmethod
def set_lock(cls, lock)
```

Set the global lock.

get_lock

[view source]

```
@classmethod
def get_lock(cls)
```

Get the global lock. Construct it if it does not exist.

pandas

[view source]

```
@classmethod
def pandas(cls, **tqdm_kwargs)
```

Registers the current `tqdm` class with pandas.core. (frame.DataFrame | series.Series | groupby.(generic.)DataFrameGroupBy | groupby.(generic.)SeriesGroupBy).progress_apply

A new instance will be created every time `progress_apply` is called, and each instance will automatically `close()` upon completion.

Parameters

- **tqdm_kwargs**: *arguments for the tqdm instance*

Examples

```
>>> import pandas as pd
>>> import numpy as np
>>> from tqdm import tqdm
>>> from tqdm.gui import tqdm as tqdm_gui
>>>
>>> df = pd.DataFrame(np.random.randint(0, 100, (100000, 6)))
>>> tqdm.pandas(ncols=50)  # can use tqdm_gui, optional kwargs, etc
>>> # Now you can use `progress_apply` instead of `apply`
>>> df.groupby(0).progress_apply(lambda x: x**2)
```

References

https://stackoverflow.com/questions/18603270/ progress-indicator-during-pandas-operations-python

__init__

[view source]

```
@envwrap("TQDM_",
         is_method=True,
         types={
             'total': float,
             'ncols': int,
             'miniters': float,
             'position': int,
             'nrows': int
         })
def __init__(iterable=None,
             desc=None,
             total=None,
             leave=True,
             file=None,
             ncols=None,
             mininterval=0.1,
             maxinterval=10.0,
             miniters=None,
             ascii=None,
             disable=False,
             unit='it',
             unit_scale=False,
             dynamic_ncols=False,
             smoothing=0.3,
             bar_format=None,
             initial=0,
             position=None,
             postfix=None,
             unit_divisor=1000,
             write_bytes=False,
             lock_args=None,
             nrows=None,
             colour=None,
             delay=0.0,
             gui=False,
             **kwargs)
```

see tqdm.tqdm for arguments

__iter__

[view source]

```
def __iter__()
```

Backward-compatibility to use: for x in tqdm(iterable)

update

```
def update(n=1)
```

[view source]

Manually update the progress bar, useful for streams such as reading files. E.g.:

```
>>> t = tqdm(total=filesize) # Initialise
>>> for current_buffer in stream:
...    ...
...    t.update(len(current_buffer))
>>> t.close()
```

The last line is highly recommended, but possibly not necessary if `t.update()` will be called in such a way that `filesize` will be exactly reached and printed.

Parameters

- **n***: int or float, optional*
  Increment to add to the internal counter of iterations [default: 1]. If using float, consider specifying `{n:.3f}` or similar in `bar_format`, or specifying `unit_scale`.

Returns

- **out***: bool or None*
  True if a `display()` was triggered.

close

```
def close()
```

[view source]

Cleanup and (if leave=False) close the progressbar.

clear

```
def clear(nolock=False)
```

[view source]

Clear current bar display.

refresh

```
def refresh(nolock=False, lock_args=None)
```

[view source]

Force refresh the display of this bar.

Parameters

- **nolock***: bool, optional*
  If `True`, does not lock. If [default: `False`]: calls `acquire()` on internal lock.
- **lock_args***: tuple, optional*
  Passed to internal lock's `acquire()`. If specified, will only `display()` if `acquire()` returns `True`.

unpause

```
def unpause()
```

[view source]

Restart tqdm timer from last print time.

reset

```
def reset(total=None)
```

[view source]

Resets to 0 iterations for repeated use.

Consider combining with `leave=True`.

Parameters

- **total**: *int or float, optional. Total to use for the new bar.*

set_description

```
def set_description(desc=None, refresh=True)
```

[view source]

Set/modify description of the progress bar.

Parameters

- **desc**: *str, optional*
- **refresh**: *bool, optional*
  Forces refresh [default: True].

set_description_str

```
def set_description_str(desc=None, refresh=True)
```

[view source]

Set/modify description without ': ' appended.

set_postfix

```
def set_postfix(ordered_dict=None, refresh=True, **kwargs)
```

[view source]

Set/modify postfix (additional stats) with automatic formatting based on datatype.

Parameters

- **ordered_dict**: *dict or OrderedDict, optional*
- **refresh**: *bool, optional*
  Forces refresh [default: True].
- **kwargs**: *dict, optional*

set_postfix_str

```
def set_postfix_str(s='', refresh=True)
```

[view source]

Postfix without dictionary expansion, similar to prefix handling.

format_dict

```
@property
def format_dict()
```

[view source]

Public API for read-only member access.

display

```
def display(msg=None, pos=None)
```

[view source]

Use `self.sp` to display `msg` in the specified `pos`.

Consider overloading this function when inheriting to use e.g.: `self.some_frontend(**self.format_dict)` instead of `self.sp`.

Parameters

- **msg***: str, optional. What to display (default: `repr(self)`).*
- **pos***: int, optional. Position to `moveto`*
  (default: `abs(self.pos)`).

wrapattr

[view source]

```
@classmethod
@contextmanager
def wrapattr(cls, stream, method, total=None, bytes=True, **tqdm_kwargs)
```

- **stream***: file-like object.*
- **method***: str, "read" or "write". The result of `read()` and*
  the first argument of `write()` should have a `len()` .

```
>>> with tqdm.wrapattr(file_obj, "read", total=file_obj.size) as fobj:
...     while True:
...         chunk = fobj.read(chunk_size)
...         if not chunk:
...             break
```

trange

[view source]

```
def trange(*args, **kwargs)
```

Shortcut for tqdm(range(*args), **kwargs).

tqdm.autonotebook

Automatically choose between `tqdm.notebook` and `tqdm.std`.

Usage:

```
>>> from tqdm.autonotebook import trange, tqdm
>>> for i in trange(10):
...     ...
```

tqdm.contrib.bells

Even more features than `tqdm.auto` (all the bells & whistles):

- `tqdm.auto`
- `tqdm.tqdm.pandas`
- `tqdm.contrib.telegram`
  - uses `${TQDM_TELEGRAM_TOKEN}` and `${TQDM_TELEGRAM_CHAT_ID}`
- `tqdm.contrib.discord`
  - uses `${TQDM_DISCORD_TOKEN}` and `${TQDM_DISCORD_CHANNEL_ID}`

tqdm.auto

Enables multiple commonly used features.

Method resolution order:

- `tqdm.autonotebook` without import warnings
- `tqdm.asyncio`
- `tqdm.std` base class

Usage:

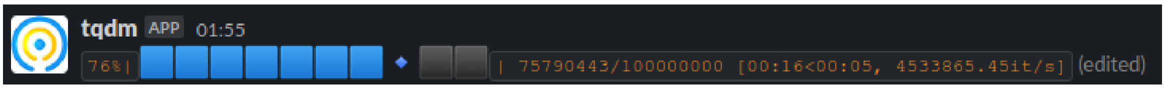
```
>>> from tqdm.auto import trange, tqdm
>>> for i in trange(10):
...     ...
```

tqdm.contrib.slack

Sends updates to a Slack app.

Usage:

```
>>> from tqdm.contrib.slack import tqdm, trange
>>> for i in trange(10, token='{token}', channel='{channel}'):
...     ...
```



```
76%|███████ ◆ ▒▒| 75790443/100000000 [00:16<00:05, 4533865.45it/s] (edited)
```

tqdm_slack Objects

```
class tqdm_slack(tqdm_auto)
```

[view source]

Standard `tqdm.auto.tqdm` but also sends updates to a Slack app. May take a few seconds to create (`__init__`).

- create a Slack app with the `chat:write` scope & invite it to a channel: https://api.slack.com/authentication/basics
- copy the bot `{token}` & `{channel}` and paste below

  ```
  >>> from tqdm.contrib.slack import tqdm, trange
  >>> for i in tqdm(iterable, token='{token}', channel='{channel}'):
  ...     ...
  ```

__init__

```
def __init__(*args, **kwargs)
```

[view source]

Parameters

- **token**: *str, required. Slack token* [default: ${TQDM_SLACK_TOKEN}].
- **channel**: *int, required. Slack channel* [default: ${TQDM_SLACK_CHANNEL}].
- **mininterval**: *float, optional.* Minimum of [default: 1.5] to avoid rate limit.

See `tqdm.auto.tqdm.__init__` for other parameters.

tsrange

```
def tsrange(*args, **kwargs)
```

[view source]

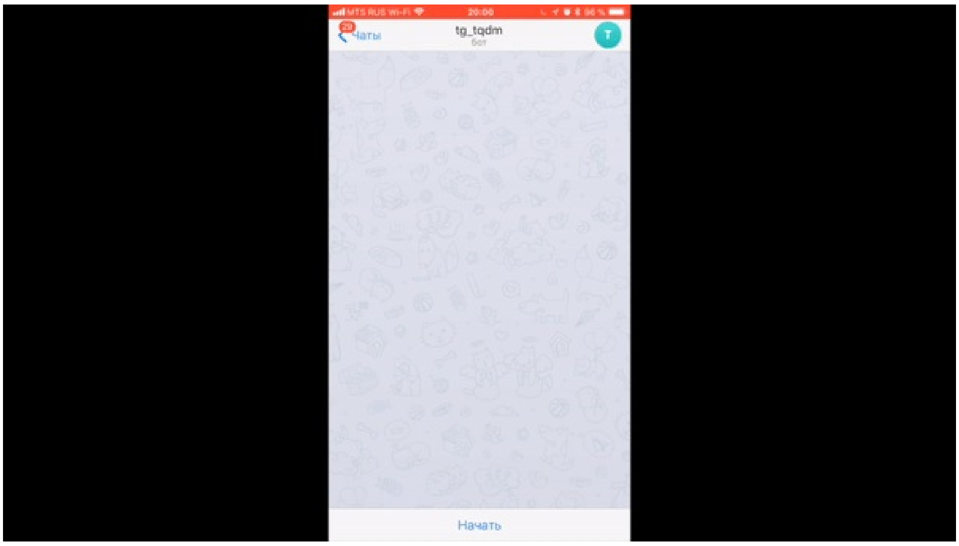
Shortcut for `tqdm.contrib.slack.tqdm(range(*args), **kwargs)`.

tqdm.contrib.telegram

Sends updates to a Telegram bot.

Usage:

```
>>> from tqdm.contrib.telegram import tqdm, trange
>>> for i in trange(10, token='{token}', chat_id='{chat_id}'):
...     ...
```



tqdm_telegram Objects

```
class tqdm_telegram(tqdm_auto)
```

[view source]

Standard `tqdm.auto.tqdm` but also sends updates to a Telegram Bot. May take a few seconds to create (`__init__`).

- create a bot https://core.telegram.org/bots#6-botfather
- copy its `{token}`
- add the bot to a chat and send it a message such as `/start`
- go to https://api.telegram.org/bot`{token}`/getUpdates to find out the `{chat_id}`
- paste the `{token}` & `{chat_id}` below

```
>>> from tqdm.contrib.telegram import tqdm, trange
>>> for i in tqdm(iterable, token='{token}', chat_id='{chat_id}'):
...     ...
```

__init__

```
def __init__(*args, **kwargs)
```

[view source]

Parameters

- **token***: str, required. Telegram token*
  [default: ${TQDM_TELEGRAM_TOKEN}].
- **chat_id***: str, required. Telegram chat ID*
  [default: ${TQDM_TELEGRAM_CHAT_ID}].

See `tqdm.auto.tqdm.__init__` for other parameters.

ttgrange

```
def ttgrange(*args, **kwargs)
```

[view source]

Shortcut for `tqdm.contrib.telegram.tqdm(range(*args), **kwargs)` .

tqdm.keras

TqdmCallback Objects

```
class TqdmCallback(keras.callbacks.Callback)
```

[view source]

Keras callback for epoch and batch progress.

__init__

```
def __init__(epochs=None,
             data_size=None,
             batch_size=None,
             verbose=1,
             tqdm_class=tqdm_auto,
             **tqdm_kwargs)
```

[view source]

Parameters

- **epochs***: int, optional*
- **data_size***: int, optional*
  Number of training pairs.
- **batch_size***: int, optional*
  Number of training pairs per batch.
- **verbose***: int*
  0: epoch, 1: batch (transient), 2: batch. [default: 1]. Will be set to `0` unless both `data_size` and `batch_size` are given.
- **tqdm_class***: optional*
  `tqdm` class to use for bars [default: `tqdm.auto.tqdm`].
- **tqdm_kwargs***: optional*
  Any other arguments used for all bars.

display

```
def display()
```

[view source]

Displays in the current cell in Notebooks.

tqdm

python 3.7 | 3.8 | 3.9 | 3.10 | 3.11 pypi v4.66.1 conda-forge v4.66.1
docker pull snap install launch binder

tqdm failing coverage 91% codecov 88% code quality A
pypi downloads 77M/month sourcerank 26 in repositories 31 mentioned in awesome Hits 2,829,366

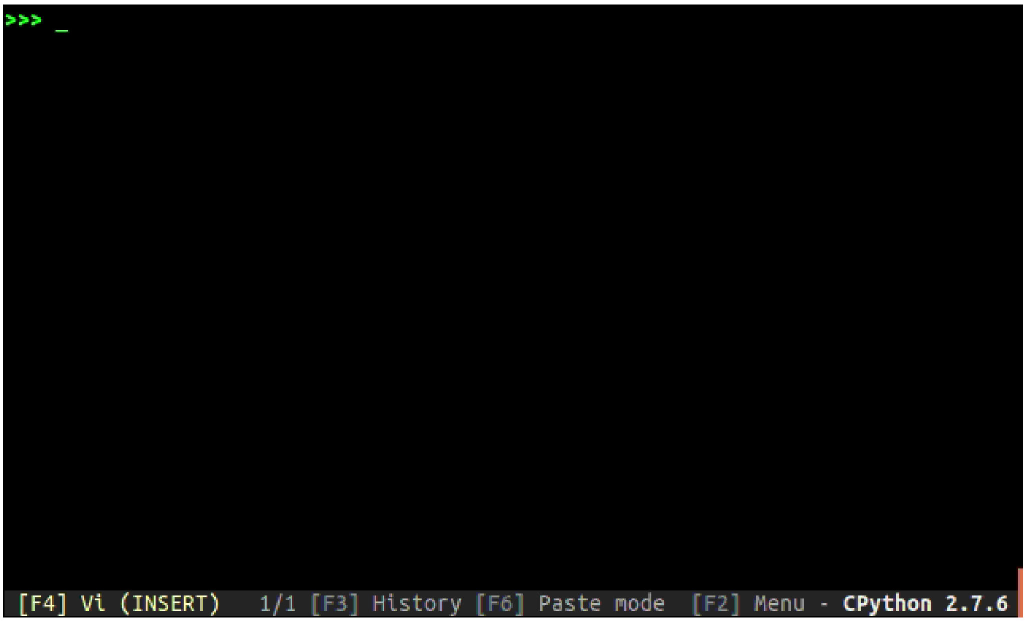
`tqdm` means "progress" in Arabic (*taqadum*, تقدّم) and is an abbreviation for "I love you so much" in Spanish (*te quiero demasiado*).

Instantly make your loops show a smart progress meter - just wrap any iterable with `tqdm(iterable)`, and you're done!

```
from tqdm import tqdm
for i in tqdm(range(10000)):
    ...
```

```
76%|████████████████████████        | 7568/10000 [00:33<00:10, 229.00it/s]
```

`trange(N)` can be also used as a convenient shortcut for `tqdm(range(N))`.



```
>>> _
```

```
 [F4] Vi (INSERT)   1/1 [F3] History [F6] Paste mode  [F2] Menu - CPython 2.7.6
```

It can also be executed as a module with pipes:

```
$ seq 9999999 | tqdm --bytes | wc -l
75.2MB [00:00, 217MB/s]
9999999
$ 7z a -bd -r backup.7z docs/ | grep Compressing | \
    tqdm --total $(find docs/ -type f | wc -l) --unit files >> backup.log
100%|███████████████████████████████| 8014/8014 [01:37<00:00, 82.29files/s]
```

Overhead is low -- about 60ns per iteration (80ns with `tqdm_gui`), and is unit tested against performance regression. By comparison, the well-established ProgressBar has an 800ns/iter overhead.

In addition to its low overhead, `tqdm` uses smart algorithms to predict the remaining time and to skip unnecessary iteration displays, which allows for a negligible overhead in most cases.

`tqdm` works on any platform (Linux, Windows, Mac, FreeBSD, NetBSD, Solaris/SunOS), in any console or in a GUI, and is also friendly with IPython/Jupyter notebooks.

`tqdm` does not require any dependencies (not even `curses`!), just Python and an environment supporting `carriage return \r` and `line feed \n` control characters.

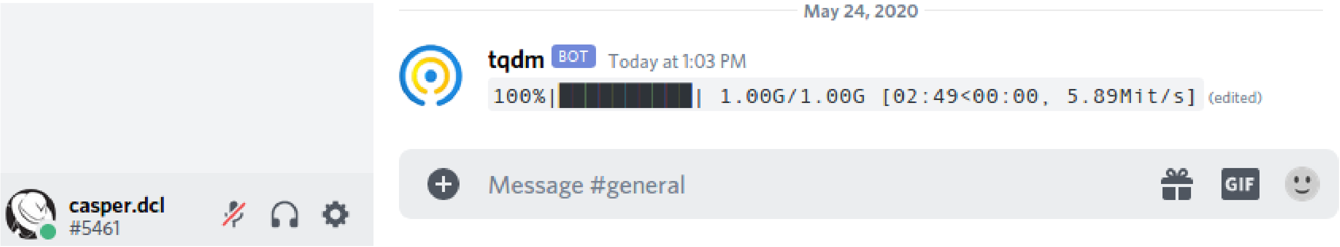
DOI 10.5281/zenodo.595120 license MPL-2.0 AND MIT Open Hub tqdm openssf best practices passing

tqdm.contrib.discord

Sends updates to a Discord bot.

Usage:

```
>>> from tqdm.contrib.discord import tqdm, trange
>>> for i in trange(10, token='{token}', channel_id='{channel_id}'):
...     ...
```



May 24, 2020

tqdm BOT Today at 1:03 PM

```
100%|██████████| 1.00G/1.00G [02:49<00:00, 5.89Mit/s]
```

(edited)

tqdm_discord Objects

class tqdm_discord(tqdm_auto) [view source]

Standard `tqdm.auto.tqdm` but also sends updates to a Discord Bot. May take a few seconds to create (`__init__`).

- create a discord bot (not public, no requirement of OAuth2 code grant, only send message permissions) & invite it to a channel: https://discordpy.readthedocs.io/en/latest/discord.html
- copy the bot `{token}` & `{channel_id}` and paste below

```
>>> from tqdm.contrib.discord import tqdm, trange
>>> for i in tqdm(iterable, token='{token}', channel_id='{channel_id}'):
...     ...
```

__init__

def __init__(*args, **kwargs) [view source]

Parameters

- **token**: *str, required. Discord token*
  [default: ${TQDM_DISCORD_TOKEN}].
- **channel_id**: *int, required. Discord channel ID*
  [default: ${TQDM_DISCORD_CHANNEL_ID}].
- **mininterval**: *float, optional.*
  Minimum of [default: 1.5] to avoid rate limit.

See `tqdm.auto.tqdm.__init__` for other parameters.

tdrange

def tdrange(*args, **kwargs) [view source]

Shortcut for `tqdm.contrib.discord.tqdm(range(*args), **kwargs)`.

tqdm.contrib.logging

Helper functionality for interoperability with stdlib `logging` .

logging_redirect_tqdm

[view source]

```
@contextmanager
def logging_redirect_tqdm(loggers=None, tqdm_class=std_tqdm)
```

Context manager redirecting console logging to `tqdm.write()` , leaving other logging handlers (e.g. log files) unaffected.

Parameters

- **loggers***: list, optional*
  Which handlers to redirect (default: [logging.root]).
- **tqdm_class***: optional*

Example

```
import logging
from tqdm import trange
from tqdm.contrib.logging import logging_redirect_tqdm

LOG = logging.getLogger(__name__)

if __name__ == '__main__':
    logging.basicConfig(level=logging.INFO)
    with logging_redirect_tqdm():
        for i in trange(9):
            if i == 4:
                LOG.info("console logging redirected to `tqdm.write()`")
    # logging restored
```

tqdm_logging_redirect

[view source]

```
@contextmanager
def tqdm_logging_redirect(*args, **kwargs)
```

Convenience shortcut for:

```
with tqdm_class(*args, **tqdm_kwargs) as pbar:
    with logging_redirect_tqdm(loggers=loggers, tqdm_class=tqdm_class):
        yield pbar
```

Parameters

- **tqdm_class***: optional, (default: tqdm.std.tqdm).*
- **loggers***: optional, list.*
  **tqdm_kwargs : passed to `tqdm_class` .

HOW TO CONTRIBUTE TO TQDM

TL;DR: Skip to [QUICK DEV SUMMARY]

This file describes how to

- contribute changes to the project, and
- upload released to the PyPI repository.

Most of the management commands have been directly placed inside the Makefile:

```
make [<alias>]  # on UNIX-like environments
python -m pymake [<alias>]  # if make is unavailable
```

The latter depends on `py-make>=0.1.0`.

Use the alias `help` (or leave blank) to list all available aliases.

HOW TO COMMIT CONTRIBUTIONS

Contributions to the project are made using the "Fork & Pull" model. The typical steps would be:

1. create an account on github
2. fork `tqdm`
3. make a local clone: `git clone https://github.com/your_account/tqdm.git`
4. make changes on the local copy
5. test (see below) and commit changes `git commit -a -m "my message"`
6. `push` to your GitHub account: `git push origin`
7. create a Pull Request (PR) from your GitHub fork (go to your fork's webpage and click on "Pull Request." You can then add a message to describe your proposal.)

WHAT CODE LAYOUT SHOULD I FOLLOW?

Don't worry too much - maintainers can help reorganise contributions. However it would be helpful to bear in mind:

- The standard core of `tqdm`, i.e. `tqdm.std.tqdm`
  - must have no dependencies apart from pure python built-in standard libraries
  - must have negligible impact on performance
  - should have 100% coverage by unit tests
  - should be appropriately commented
  - should have well-formatted docstrings for functions
    - under 76 chars (incl. initial spaces) to avoid linebreaks in terminal pagers

        - use two spaces between variable name and colon, specify a type, and most likely state that it's optional: `VAR<space><space>:<space>TYPE[, optional]`
        - use [default: ...] for default values of keyword arguments
    - will not break backward compatibility unless there is a very good reason
        - e.g. breaking py26 compatibility purely in favour of minor readability changes (such as converting `dict(a=1)` to `{'a': 1}`) is not a good enough reason
    - API changes should be discussed carefully
    - remember, with millions of downloads per month, `tqdm` must be extremely fast and reliable
- Any other kind of change may be included in a (possibly new) submodule
    - submodules are likely single python files under the main tqdm/ directory
    - submodules extending `tqdm.std.tqdm` or any other module (e.g. `tqdm.notebook.tqdm`, `tqdm.gui.tqdm`)
    - CLI wrapper `tqdm.cli`
        - if a newly added `tqdm.std.tqdm` option is not supported by the CLI, append to `tqdm.cli.UNSUPPORTED_OPTS`
    - can implement anything from experimental new features to support for third-party libraries such as `pandas`, `numpy`, etc.
    - submodule maturity
        - alpha: experimental; missing unit tests, comments, and/or feedback; raises `tqdm.TqdmExperimentalWarning`
        - beta: well-used; commented, perhaps still missing tests
        - stable: >10 users; commented, 80% coverage
- `.meta/`
    - A "hidden" folder containing helper utilities not strictly part of the `tqdm` distribution itself

TESTING

Once again, don't worry too much - tests are automated online, and maintainers can also help.

To test functionality (such as before submitting a Pull Request), there are a number of unit tests.

Standard unit tests

The standard way to run the tests:

- install `tox`
- `cd` to the root of the `tqdm` directory (in the same folder as this file)
- run the following command:

```
[python -m py]make test
# or:
tox --skip-missing-interpreters
```

This will build the module and run the tests in a virtual environment. Errors and coverage rates will be output to the console/log. (Ignore missing interpreters errors - these are due to the local machine missing certain versions of Python.)

Note: to install all versions of the Python interpreter that are specified in tox.ini, you can use `MiniConda` to install a minimal setup. You must also ensure that each distribution has an alias to call the Python interpreter (e.g. `python311` for Python 3.11's interpreter).

Alternative unit tests with pytest

Alternatively, use `pytest` to run the tests just for the current Python version:

- install test requirements: `[python -m py]make install_test`
- run the following command:

```
[python -m py]make alltests
```

MANAGE A NEW RELEASE

This section is intended for the project's maintainers and describes how to build and upload a new release. Once again, `[python -m py]make [<alias>]` will help. Also consider `pip install`ing development utilities: `[python -m py]make install_build` at a minimum, or a more thorough `conda env create`.

Pre-commit Hook

It's probably a good idea to use the `pre-commit` (`pip install pre-commit`) helper.

Run `pre-commit install` for convenient local sanity-checking.

Semantic Versioning

The `tqdm` repository managers should:

- follow the Semantic Versioning convention for tagging

Checking `pyproject.toml`

To check that the `pyproject.toml` file is compliant with PyPI requirements (e.g. version number; reStructuredText in `README.rst`) use:

```
[python -m py]make testsetup
```

To upload just metadata (including overwriting mistakenly uploaded metadata) to PyPI, use:

```
[python -m py]make pypimeta
```

Merging Pull Requests

This section describes how to cleanly merge PRs.

1 Rebase

From your project repository, merge and test (replace `pr-branch-name` as appropriate):

```
git fetch origin
git checkout -b pr-branch-name origin/pr-branch-name
git rebase master
```

If there are conflicts:

```
git mergetool
git rebase --continue
```

2 Push

Update branch with the rebased history:

```
git push origin pr-branch-name --force
```

Non maintainers can stop here.

Note: NEVER just `git push --force` (this will push all local branches, overwriting remotes).

3 Merge

```
git checkout master
git merge --no-ff pr-branch-name
```

4 Test

```
[python -m py]make alltests
```

5 Push to master

```
git push origin master
```

Building a Release and Uploading to PyPI

Formally publishing requires additional steps: testing and tagging.

Test

Ensure that all online CI tests have passed.

Tag

- ensure the version has been tagged. The tag format is `v{major}.{minor}.{patch}`, for example: `v4.4.1`. The current commit's tag is used in the version checking process. If the current commit is not tagged appropriately, the version will display as `v{major}.{minor}.{patch}.dev{N}+g{commit_hash}`.

Upload

GitHub Actions (GHA) CI should automatically do this after pushing tags. Manual instructions are given below in case of failure.

Build `tqdm` into a distributable python package:

```
[python -m py]make build
```

This will generate several builds in the `dist/` folder. On non-windows machines the windows `exe` installer may fail to build. This is normal.

Finally, upload everything to PyPI. This can be done easily using the twine module:

```
[python -m py]make pypi
```

Also, the new release can (should) be added to GitHub by creating a new release from the web interface; uploading packages from the `dist/` folder created by `[python -m py]make build`. The wiki can be automatically updated with GitHub release notes by running `make` within the wiki repository.

Docker images may be uploaded to https://hub.docker.com/r/tqdm/tqdm. Assuming `docker` is installed:

```
make -B docker
docker login
docker push tqdm/tqdm:latest
docker push tqdm/tqdm:$(docker run -i --rm tqdm/tqdm -v)
```

Snaps may be uploaded to https://snapcraft.io/tqdm. Assuming `snapcraft` is installed (`snap install snapcraft --classic --beta`):

```
make snap
snapcraft login
snapcraft push tqdm*.snap --release stable
```

Notes

- you can also test on the PyPI test servers `test.pypi.org` before the real deployment
- in case of a mistake, you can delete an uploaded release on PyPI, but you cannot re-upload another with the same version number
- in case of a mistake in the metadata on PyPI (e.g. bad README), updating just the metadata is possible: `[python -m py]make pypimeta`

Updating Websites

The most important file is `.readme.rst`, which should always be kept up-to-date and in sync with the in-line source documentation. This will affect all of the following:

- `README.rst` (generated by `mkdocs.py` during `make build`)
- The main repository site which automatically serves the latest `README.rst` as well as links to all of GitHub's features. This is the preferred online referral link for `tqdm`.
- The PyPI mirror which automatically serves the latest release built from `README.rst` as well as links to past releases.
- Many external web crawlers.

Additionally (less maintained), there exists:

- A wiki which is publicly editable.
- The gh-pages project which is built from the gh-pages branch, which is built using asv.
- The gh-pages root which is built from a separate github.io repo.

Helper Bots

There are some helpers in .github/workflows to assist with maintenance.

- Comment Bot
  - allows maintainers to write `/tag vM.m.p commit_hash` in an issue/PR to create a tag
- Post Release

  - automatically updates the wiki
  - automatically updates the gh-pages root
- Benchmark
  - automatically updates the gh-pages project

QUICK DEV SUMMARY

For experienced devs, once happy with local master, follow the steps below. Much is automated so really it's steps 1-5, then 11(a).

1. test (`[python -m py]make alltests` or rely on `pre-commit`)
2. `git commit [--amend] # -m "bump version"`
3. `git push`
4. wait for tests to pass a) in case of failure, fix and go back to (1)
5. `git tag vM.m.p && git push --tags` or comment `/tag vM.m.p commit_hash`
6. **[AUTO:GHA]** `[python -m py]make distclean`
7. **[AUTO:GHA]** `[python -m py]make build`
8. **[AUTO:GHA]** upload to PyPI. either: a) `[python -m py]make pypi`, or b) `twine upload -s -i $(git config user.signingkey) dist/tqdm-*`
9. **[AUTO:GHA]** upload to docker hub: a) `make -B docker` b) `docker push tqdm/tqdm:latest` c) `docker push tqdm/tqdm:$(docker run -i --rm tqdm/tqdm -v)`
10. **[AUTO:GHA]** upload to snapcraft: a) `make snap`, and b) `snapcraft push tqdm*.snap --release stable`
11. Wait for GHA to draft a new release on https://github.com/tqdm/tqdm/releases a) replace the commit history with helpful release notes, and click publish b) **[AUTO:GHA]** attach `dist/tqdm-*` binaries (usually only `*.whl*`)
12. **[SUB][AUTO:GHA-rel]** run `make` in the `wiki` submodule to update release notes
13. **[SUB][AUTO:GHA-rel]** run `make deploy` in the `docs` submodule to update website
14. **[SUB][AUTO:GHA-rel]** accept the automated PR in the `feedstock` submodule to update conda
15. **[AUTO:GHA-rel]** update the gh-pages project benchmarks a) `[python -m py]make testasvfull` b) `asv gh-pages`

Key:

- **[AUTO:GHA]**: GitHub Actions CI should automatically do this after `git push --tags` (5)
- **[AUTO:GHA-rel]**: GitHub Actions CI should automatically do this after release (11a)
- **[SUB]**: Requires one-time `make submodules` to clone `docs`, `wiki`, and `feedstock`

☰

Ports to Other Languages

Here is a non-exhaustive list of `tqdm` ports in various languages, with a partial mention if official and/or in development stage:

Language	**Source**
Python	tqdm/tqdm Official
C++	tqdm/tqdm.cpp Official Pre-Alpha
.NET	damageboy/yaap
C++	aminnj/cpptqdm
C++	sizmailov/tqdm-cxx
D	ShigekiKarita/snck
Elixir	antipax/tqdm_elixir
Go	kinsey40/tqdm
Go	sbwhitepack/tqdm
Julia	cloud-oak/Tqdm.jl
Node.js	jhedin/ntqdm
Perl	perlancar/perl-tqdm
Ruby	powerpak/tqdm-ruby
Ruby	rscarvalho/tqdm-ruby

tqdm.std

Customisable progressbar decorator for iterators. Includes a default `range` iterator printing to `stderr` .

Usage:

```
>>> from tqdm import trange, tqdm
>>> for i in trange(10):
...     ...
```

TqdmWarning Objects

```
class TqdmWarning(Warning)
```

[view source]

base class for all tqdm warnings.

Used for non-external-code-breaking errors, such as garbled printing.

TqdmExperimentalWarning Objects

```
class TqdmExperimentalWarning(TqdmWarning, FutureWarning)
```

[view source]

beta feature, unstable API and behaviour

TqdmMonitorWarning Objects

```
class TqdmMonitorWarning(TqdmWarning, RuntimeWarning)
```

[view source]

tqdm monitor errors which do not affect external functionality

TqdmDefaultWriteLock Objects

```
class TqdmDefaultWriteLock(object)
```

[view source]

Provide a default write lock for thread and multiprocessing safety. Works only on platforms supporting `fork` (so Windows is excluded). You must initialise a `tqdm` or `TqdmDefaultWriteLock` instance before forking in order for the write lock to work. On Windows, you need to supply the lock from the parent to the children as an argument to joblib or the parallelism lib you use.

tqdm._monitor

TMonitor Objects

class TMonitor(Thread) [view source]

Monitoring thread for tqdm bars. Monitors if tqdm bars are taking too much time to display and readjusts miniters automatically if necessary.

Parameters

- **tqdm_cls***: class*
  tqdm class to use (can be core tqdm or a submodule).
- **sleep_interval***: float*
  Time to sleep between monitoring checks.

tqdm.asyncio

Asynchronous progressbar decorator for iterators. Includes a default `range` iterator printing to `stderr`.

Usage:

```
>>> from tqdm.asyncio import trange, tqdm
>>> async for i in trange(10):
...     ...
```

tqdm_asyncio Objects

```
class tqdm_asyncio(std_tqdm)
```

[view source]

Asynchronous-friendly version of tqdm.

as_completed

```
@classmethod
def as_completed(cls,
                 fs,
                 *,
                 loop=None,
                 timeout=None,
                 total=None,
                 **tqdm_kwargs)
```

[view source]

Wrapper for `asyncio.as_completed`.

gather

```
@classmethod
async def gather(cls, *fs, loop=None, timeout=None, total=None, **tqdm_kwargs)
```

[view source]

Wrapper for `asyncio.gather`.

tarange

```
def tarange(*args, **kwargs)
```

[view source]

A shortcut for `tqdm.asyncio.tqdm(range(*args), **kwargs)`.

tqdm.dask

TqdmCallback Objects

```
class TqdmCallback(Callback)
```

[view source]

Dask callback for task progress.

__init__

```
def __init__(start=None, pretask=None, tqdm_class=tqdm_auto, **tqdm_kwargs)
```

[view source]

Parameters

- **tqdm_class**: *optional*
  `tqdm` class to use for bars [default: `tqdm.auto.tqdm`].
- **tqdm_kwargs**: *optional*
  Any other arguments used for all bars.

display

```
def display()
```

[view source]

Displays in the current cell in Notebooks.

CLI

`tqdm`'s command line interface (CLI) can be used in a script or on the terminal/console. Simply inserting `tqdm` (or `python -m tqdm`) between pipes will pass through all `stdin` to `stdout` while printing progress to `stderr`.

Run `tqdm --help` for a full list of options.

The example below demonstrate counting the number of lines in all Python files in the current directory, with timing information included.

```
$ time find . -name '*.py' -type f -exec cat \{} \; | wc -l
857365

real    0m3.458s
user    0m0.274s
sys     0m3.325s

$ time find . -name '*.py' -type f -exec cat \{} \; | tqdm | wc -l
857366it [00:03, 246471.31it/s]
857365

real    0m3.585s
user    0m0.862s
sys     0m3.358s
```

Note that the usual arguments for `tqdm` can also be specified.

```
$ find . -name '*.py' -type f -exec cat \{} \; \
  | tqdm --unit loc --unit_scale --total 857366 >> /dev/null
100%|█████████████████████████████████| 857K/857K [00:04<00:00, 246Kloc/s]
```

Backing up a large directory?

```
$ tar -zcf - docs/ | tqdm --bytes --total `du -sb docs/ | cut -f1` \
  > backup.tgz
44%|██████████████▊                   | 153M/352M [00:14<00:18, 11.0MB/s]
```

This can be beautified further:

```
$ BYTES="$(du -sb docs/ | cut -f1)"
$ tar -cf - docs/ \
  | tqdm --bytes --total "$BYTES" --desc Processing | gzip \
  | tqdm --bytes --total "$BYTES" --desc Compressed --position 1 \
  > ~/backup.tgz
Processing: 100%|██████████████████████| 352M/352M [00:14<00:00, 30.2MB/s]
Compressed:  42%|█████████▎            | 148M/352M [00:14<00:19, 10.9MB/s]
```

Or done on a file level using 7-zip:

```
$ 7z a -bd -r backup.7z docs/ | grep Compressing \
  | tqdm --total $(find docs/ -type f | wc -l) --unit files \
  | grep -v Compressing
100%|███████████████████████████| 15327/15327 [01:00<00:00, 712.96files/s]
```

Pre-existing CLI programs already outputting basic progress information will benefit from `tqdm`'s `--update` and `--update_to` flags:

```
$ seq 3 0.1 5 | tqdm --total 5 --update_to --null
100%|████████████████████████████████████| 5.0/5 [00:00<00:00, 9673.21it/s]
$ seq 10 | tqdm --update --null  # 1 + 2 + ... + 10 = 55 iterations
55it [00:00, 90006.52it/s]
```

tqdm.contrib.itertools

Thin wrappers around `itertools`.

product

```
def product(*iterables, **tqdm_kwargs)
```

[view source]

Equivalent of `itertools.product`.

Parameters

- **tqdm_class***: [default: tqdm.auto.tqdm].*

tqdm.rich

`rich.progress` decorator for iterators.

Usage:

```
>>> from tqdm.rich import trange, tqdm
>>> for i in trange(10):
...     ...
```

tqdm_rich Objects

```
class tqdm_rich(std_tqdm)
```

[view source]

Experimental rich.progress GUI version of tqdm!

__init__

```
def __init__(*args, **kwargs)
```

[view source]

This class accepts the following parameters *in addition* to the parameters accepted by `tqdm`.

Parameters

- **progress***: tuple, optional*
  arguments for `rich.progress.Progress()`.
- **options***: dict, optional*
  keyword arguments for `rich.progress.Progress()`.

reset

```
def reset(total=None)
```

[view source]

Resets to 0 iterations for repeated use.

Parameters

- **total***: int or float, optional. Total to use for the new bar.*

trrange

```
def trrange(*args, **kwargs)
```

[view source]

Shortcut for `tqdm.rich.tqdm(range(*args), **kwargs)`.