black · PyPI

pypi.org/project/black

Project description

The Uncompromising Code Formatter

docs passing coverage 96% license MIT pypi v22.3.0 Test passing downloads 194M conda-forge 3.8M code style black

> "Any color you like."

Black is the uncompromising Python code formatter. By using it, you agree to cede control over minutiae of hand-formatting. In return, *Black* gives you speed, determinism, and freedom from `pycodestyle` nagging about formatting. You will save time and mental energy for more important matters.

Blackened code looks the same regardless of the project you're reading. Formatting becomes transparent after a while and you can focus on the content instead.

Black makes code review faster by producing the smallest diffs possible.

Try it out now using the Black Playground. Watch the PyCon 2019 talk to learn more.

Read the documentation on ReadTheDocs!

Installation and usage

Installation

Black can be installed by running `pip install black`. It requires Python 3.6.2+ to run. If you want to format Jupyter Notebooks, install with `pip install black[jupyter]`.

If you can't wait for the latest *hotness* and want to install from GitHub, use:

`pip install git+https://github.com/psf/black`

Usage

To get started right away with sensible defaults:

```
black {source_file_or_directory}
```

You can run *Black* as a package if running it as a script doesn't work:

```
python -m black {source_file_or_directory}
```

Further information can be found in our docs:

> Usage and Configuration

Black is already successfully used by many projects, small and big. *Black* has a comprehensive test suite, with efficient parallel tests, and our own auto formatting and parallel Continuous Integration runner. Now that we have become stable, you should not expect large formatting to changes in the future. Stylistic changes will mostly be responses to bug reports and support for new Python syntax. For more information please refer to the The Black Code Style.

Also, as a safety measure which slows down processing, *Black* will check that the reformatted code still produces a valid AST that is effectively equivalent to the original (see the Pragmatism section for details). If you're feeling confident, use `--fast`.

The *Black* code style

Black is a PEP 8 compliant opinionated formatter. *Black* reformats entire files in place. Style configuration options are deliberately limited and rarely added. It doesn't take previous formatting into account (see Pragmatism for exceptions).

Our documentation covers the current *Black* code style, but planned changes to it are also documented. They're both worth taking a look:

Changes to the *Black* code style are bound by the Stability Policy:

> The *Black* Code Style: Stability Policy

Please refer to this document before submitting an issue. What seems like a bug might be intended behaviour.

Pragmatism

Early versions of *Black* used to be absolutist in some respects. They took after its initial author. This was fine at the time as it made the implementation simpler and there were not many users anyway. Not many edge cases were reported. As a mature tool, *Black* does make some exceptions to rules it otherwise holds.

> The *Black* code style: Pragmatism

Please refer to this document before submitting an issue just like with the document above. What seems like a bug might be intended behaviour.

Configuration

Black is able to read project-specific default values for its command line options from a `pyproject.toml` file. This is especially useful for specifying custom `--include` and `--exclude` / `--force-exclude` / `--extend-exclude` patterns for your project.

You can find more details in our documentation:

> The basics: Configuration via a file

And if you're looking for more general configuration documentation:

> Usage and Configuration

Pro-tip: If you're asking yourself "Do I need to configure anything?" the answer is "No". *Black* is all about sensible defaults. Applying those defaults will have your code in compliance with many other *Black* formatted projects.

Used by

The following notable open-source projects trust *Black* with enforcing a consistent code style: pytest, tox, Pyramid, Django, Django Channels, Hypothesis, attrs, SQLAlchemy, Poetry, PyPA applications (Warehouse, Bandersnatch, Pipenv, virtualenv), pandas, Pillow, Twisted, LocalStack, every Datadog Agent Integration, Home Assistant, Zulip, Kedro, OpenOA, FLORIS, ORBIT, WOMBAT, and many more.

The following organizations use *Black*: Facebook, Dropbox, KeepTruckin, Mozilla, Quora, Duolingo, QuantumBlack, Tesla.

Are we missing anyone? Let us know.

Testimonials

Mike Bayer, author of `SQLAlchemy`:

> I can't think of any single tool in my entire programming career that has given me a bigger productivity increase by its introduction. I can now do refactorings in about 1% of the keystrokes that it would have taken me previously when we had no way for code to format itself.

Dusty Phillips, writer:

> *Black* is opinionated so you don't have to be.

Hynek Schlawack, creator of `attrs`, core developer of Twisted and CPython:

> An auto-formatter that doesn't suck is all I want for Xmas!

Carl Meyer, Django core developer:

> At least the name is good.

Kenneth Reitz, creator of `requests` and `pipenv`:

> This vastly improves the formatting of our code. Thanks a ton!

Show your style

Use the badge in your project's README.md:

```
[![Code style: black](https://img.shields.io/badge/code%20style-black-000000.svg)]
(https://github.com/psf/black)
```

Using the badge in README.rst:

```
.. image:: https://img.shields.io/badge/code%20style-black-000000.svg
    :target: https://github.com/psf/black
```

Looks like this:

code style black

License

MIT

Contributing

Welcome! Happy to see you willing to make the project better. You can get started by reading this:

> Contributing: The basics

You can also take a look at the rest of the contributing docs or talk with the developers:

Change log

The log has become rather long. It moved to its own file.

See CHANGES.

Authors

The author list is quite long nowadays, so it lives in its own file.

See AUTHORS.md

Code of Conduct

Everyone participating in the *Black* project, and in particular in the issue tracker, pull requests, and social media activity, is expected to treat other people with respect and more generally to follow the guidelines articulated in the Python Community Code of Conduct.

At the same time, humor is encouraged. In fact, basic familiarity with Monty Python's Flying Circus is expected. We are not savages.

And if you *really* need to slap somebody, do it with a fish while dancing.

Change Log

Unreleased

Highlights

Style

Preview style

Blackd

Configuration

Documentation

Integrations

Output

Packaging

Parser

Performance

22.3.0

Preview style

- Code cell separators `#%%` are now standardised to `# %%` (#2919)
- Remove unnecessary parentheses from `except` statements (#2939)
- Remove unnecessary parentheses from tuple unpacking in `for` loops (#2945)
- Avoid magic-trailing-comma in single-element subscripts (#2942)

Configuration

- Do not format `__pypackages__` directories by default (#2836)
- Add support for specifying stable version with `--required-version` (#2832).
- Avoid crashing when the user has no homedir (#2814)
- Avoid crashing when md5 is not available (#2905)
- Fix handling of directory junctions on Windows (#2904)

Documentation

Update pylint config documentation (#2931)

Integrations

Move test to disable plugin in Vim/Neovim, which speeds up loading (#2896)

Output

In verbose, mode, log when *Black* is using user-level config (#2861)

Packaging

- Fix Black to work with Click 8.1.0 (#2966)
- On Python 3.11 and newer, use the standard library's `tomllib` instead of `tomli` (#2903)
- `black-primer`, the deprecated internal devtool, has been removed and copied to a separate repository (#2924)

Parser

Black can now parse starred expressions in the target of `for` and `async for` statements, e.g `for item in *items_1, *items_2: pass` (#2879).

22.1.0

At long last, *Black* is no longer a beta product! This is the first non-beta release and the first release covered by our new stability policy.

Highlights

- **Remove Python 2 support** (#2740)
- Introduce the `--preview` flag (#2752)

Style

- Deprecate `--experimental-string-processing` and move the functionality under `--preview` (#2789)
- For stubs, one blank line between class attributes and methods is now kept if there's at least one pre-existing blank line (#2736)
- Black now normalizes string prefix order (#2297)
- Remove spaces around power operators if both operands are simple (#2726)
- Work around bug that causes unstable formatting in some cases in the presence of the magic trailing comma (#2807)
- Use parentheses for attribute access on decimal float and int literals (#2799)
- Don't add whitespace for attribute access on hexadecimal, binary, octal, and complex literals (#2799)
- Treat blank lines in stubs the same inside top-level `if` statements (#2820)
- Fix unstable formatting with semicolons and arithmetic expressions (#2817)
- Fix unstable formatting around magic trailing comma (#2572)

Parser

- Fix mapping cases that contain as-expressions, like `case {"key": 1 | 2 as password}` (#2686)
- Fix cases that contain multiple top-level as-expressions, like `case 1 as a, 2 as b` (#2716)
- Fix call patterns that contain as-expressions with keyword arguments, like `case Foo(bar=baz as quux)` (#2749)
- Tuple unpacking on `return` and `yield` constructs now implies 3.8+ (#2700)
- Unparenthesized tuples on annotated assignments (e.g `values: Tuple[int, ...] = 1, 2, 3`) now implies 3.8+ (#2708)
- Fix handling of standalone `match()` or `case()` when there is a trailing newline or a comment inside of the parentheses. (#2760)
- `from __future__ import annotations` statement now implies Python 3.7+ (#2690)

Performance

- Speed-up the new backtracking parser about 4X in general (enabled when `--target-version` is set to 3.10 and higher). (#2728)
- *Black* is now compiled with mypyc for an overall 2x speed-up. 64-bit Windows, MacOS, and Linux (not including musl) are supported. (#1009, #2431)

Configuration

- Do not accept bare carriage return line endings in pyproject.toml (#2408)
- Add configuration option (`python-cell-magics`) to format cells with custom magics in Jupyter Notebooks (#2744)
- Allow setting custom cache directory on all platforms with environment variable `BLACK_CACHE_DIR` (#2739).

- Enable Python 3.10+ by default, without any extra need to specify `--target-version=py310`. (#2758)
- Make passing `SRC` or `--code` mandatory and mutually exclusive (#2804)

Output

- Improve error message for invalid regular expression (#2678)
- Improve error message when parsing fails during AST safety check by embedding the underlying SyntaxError (#2693)
- No longer color diff headers white as it's unreadable in light themed terminals (#2691)
- Text coloring added in the final statistics (#2712)
- Verbose mode also now describes how a project root was discovered and which paths will be formatted. (#2526)

Packaging

- All upper version bounds on dependencies have been removed (#2718)
- `typing-extensions` is no longer a required dependency in Python 3.10+ (#2772)
- Set `click` lower bound to `8.0.0` (#2791)

Integrations

Update GitHub action to support containerized runs (#2748)

Documentation

- Change protocol in pip installation instructions to `https://` (#2761)
- Change HTML theme to Furo primarily for its responsive design and mobile support (#2793)
- Deprecate the `black-primer` tool (#2809)
- Document Python support policy (#2819)

21.12b0

Black

- Fix determination of f-string expression spans (#2654)
- Fix bad formatting of error messages about EOF in multi-line statements (#2343)
- Functions and classes in blocks now have more consistent surrounding spacing (#2472)

Jupyter Notebook support

- Cell magics are now only processed if they are known Python cell magics. Earlier, all cell magics were tokenized, leading to possible indentation errors e.g. with `%%writefile`. (#2630)
- Fix assignment to environment variables in Jupyter Notebooks (#2642)

Python 3.10 support

- Point users to using `--target-version py310` if we detect 3.10-only syntax (#2668)
- Fix `match` statements with open sequence subjects, like `match a, b:` or `match a, *b:` (#2639) (#2659)
- Fix `match` / `case` statements that contain `match` / `case` soft keywords multiple times, like `match re.match()` (#2661)
- Fix `case` statements with an inline body (#2665)
- Fix styling of starred expressions inside `match` subject (#2667)
- Fix parser error location on invalid syntax in a `match` statement (#2649)
- Fix Python 3.10 support on platforms without ProcessPoolExecutor (#2631)
- Improve parsing performance on code that uses `match` under `--target-version py310` up to ~50% (#2670)

Packaging

Remove dependency on `regex` (#2644) (#2663)

21.11b1

Black

Bumped regex version minimum to 2021.4.4 to fix Pattern class usage (#2621)

21.11b0

Black

- Warn about Python 2 deprecation in more cases by improving Python 2 only syntax detection (#2592)
- Add experimental PyPy support (#2559)
- Add partial support for the match statement. As it's experimental, it's only enabled when `--target-version py310` is explicitly specified (#2586)
- Add support for parenthesized with (#2586)
- Declare support for Python 3.10 for running Black (#2562)

Integrations

- Fixed vim plugin with Python 3.10 by removing deprecated distutils import (#2610)
- The vim plugin now parses `skip_magic_trailing_comma` from pyproject.toml (#2613)

21.10b0

Black

- Document stability policy, that will apply for non-beta releases (#2529)
- Add new `--workers` parameter (#2514)
- Fixed feature detection for positional-only arguments in lambdas (#2532)
- Bumped typed-ast version minimum to 1.4.3 for 3.10 compatibility (#2519)
- Fixed a Python 3.10 compatibility issue where the loop argument was still being passed even though it has been removed (#2580)
- Deprecate Python 2 formatting support (#2523)

Blackd

- Remove dependency on aiohttp-cors (#2500)
- Bump required aiohttp version to 3.7.4 (#2509)

Black-Primer

- Add primer support for --projects (#2555)
- Print primer summary after individual failures (#2570)

Integrations

- Allow to pass `target_version` in the vim plugin (#1319)
- Install build tools in docker file and use multi-stage build to keep the image size down (#2582)

21.9b0

Packaging

- Fix missing modules in self-contained binaries (#2466)
- Fix missing toml extra used during installation (#2475)

21.8b0

Black

- Add support for formatting Jupyter Notebook files (#2357)
- Move from `appdirs` dependency to `platformdirs` (#2375)
- Present a more user-friendly error if .gitignore is invalid (#2414)
- The failsafe for accidentally added backslashes in f-string expressions has been hardened to handle more edge cases during quote normalization (#2437)
- Avoid changing a function return type annotation's type to a tuple by adding a trailing comma (#2384)
- Parsing support has been added for unparenthesized walruses in set literals, set comprehensions, and indices (#2447).
- Pin `setuptools-scm` build-time dependency version (#2457)
- Exclude typing-extensions version 3.10.0.1 due to it being broken on Python 3.10 (#2460)

Blackd

Replace sys.exit(-1) with raise ImportError as it plays more nicely with tools that scan installed packages (#2440)

Integrations

The provided pre-commit hooks no longer specify `language_version` to avoid overriding `default_language_version` (#2430)

21.7b0

Black

- Configuration files using TOML features higher than spec v0.5.0 are now supported (#2301)
- Add primer support and test for code piped into black via STDIN (#2315)
- Fix internal error when `FORCE_OPTIONAL_PARENTHESES` feature is enabled (#2332)
- Accept empty stdin (#2346)
- Provide a more useful error when parsing fails during AST safety checks (#2304)

Docker

Add new `latest_release` tag automation to follow latest black release on docker images (#2374)

Integrations

- The vim plugin now searches upwards from the directory containing the current buffer instead of the current working directory for pyproject.toml. (#1871)
- The vim plugin now reads the correct string normalization option in pyproject.toml (#1869)
- The vim plugin no longer crashes Black when there's boolean values in pyproject.toml (#1869)

21.6b0

Black

- Fix failure caused by `fmt: skip` and indentation (#2281)
- Account for += assignment when deciding whether to split string (#2312)
- Correct max string length calculation when there are string operators (#2292)
- Fixed option usage when using the `--code` flag (#2259)
- Do not call `uvloop.install()` when *Black* is used as a library (#2303)
- Added `--required-version` option to require a specific version to be running (#2300)

- Fix incorrect custom breakpoint indices when string group contains fake f-strings (#2311)
- Fix regression where `R` prefixes would be lowercased for docstrings (#2285)
- Fix handling of named escapes (`\N{...}`) when `--experimental-string-processing` is used (#2319)

Integrations

The official Black action now supports choosing what version to use, and supports the major 3 OSes. (#1940)

21.5b2

Black

- A space is no longer inserted into empty docstrings (#2249)
- Fix handling of .gitignore files containing non-ASCII characters on Windows (#2229)
- Respect `.gitignore` files in all levels, not only `root/.gitignore` file (apply `.gitignore` rules like `git` does) (#2225)
- Restored compatibility with Click 8.0 on Python 3.6 when LANG=C used (#2227)
- Add extra uvloop install + import support if in python env (#2258)
- Fix --experimental-string-processing crash when matching parens are not found (#2283)
- Make sure to split lines that start with a string operator (#2286)
- Fix regular expression that black uses to identify f-expressions (#2287)

Blackd

Add a lower bound for the `aiohttp-cors` dependency. Only 0.4.0 or higher is supported. (#2231)

Packaging

- Release self-contained x86_64 MacOS binaries as part of the GitHub release pipeline (#2198)
- Always build binaries with the latest available Python (#2260)

Documentation

- Add discussion of magic comments to FAQ page (#2272)
- `--experimental-string-processing` will be enabled by default in the future (#2273)
- Fix typos discovered by codespell (#2228)
- Fix Vim plugin installation instructions. (#2235)
- Add new Frequently Asked Questions page (#2247)
- Fix encoding + symlink issues preventing proper build on Windows (#2262)

21.5b1

Black

Refactor `src/black/__init__.py` into many files (#2206)

Documentation

- Replaced all remaining references to the `master` branch with the `main` branch. Some additional changes in the source code were also made. (#2210)
- Sigificantly reorganized the documentation to make much more sense. Check them out by heading over to the stable docs on RTD. (#2174)

21.5b0

Black

- Set `--pyi` mode if `--stdin-filename` ends in `.pyi` (#2169)
- Stop detecting target version as Python 3.9+ with pre-PEP-614 decorators that are being called but with no arguments (#2182)

Black-Primer

Add `--no-diff` to black-primer to suppress formatting changes (#2187)

21.4b2

Black

- Fix crash if the user configuration directory is inaccessible. (#2158)

- Clarify circumstances in which *Black* may change the AST (#2159)

- Allow `.gitignore` rules to be overridden by specifying `exclude` in `pyproject.toml` or on the command line. (#2170)

Packaging

Install `primer.json` (used by `black-primer` by default) with black. (#2154)

21.4b1

Black

- Fix crash on docstrings ending with "\ ". (#2142)

- Fix crash when atypical whitespace is cleaned out of dostrings (#2120)

- Reflect the `--skip-magic-trailing-comma` and `--experimental-string-processing` flags in the name of the cache file. Without this fix, changes in these flags would not take effect if the cache had already been populated. (#2131)
- Don't remove necessary parentheses from assignment expression containing assert / return statements. (#2143)

Packaging

Bump pathspec to >= 0.8.1 to solve invalid .gitignore exclusion handling

21.4b0

Black

- Fixed a rare but annoying formatting instability created by the combination of optional trailing commas inserted by `Black` and optional parentheses looking at pre-existing "magic" trailing commas. This fixes issue #1629 and all of its many many duplicates. (#2126)
- `Black` now processes one-line docstrings by stripping leading and trailing spaces, and adding a padding space when needed to break up """". (#1740)
- `Black` now cleans up leading non-breaking spaces in comments (#2092)
- `Black` now respects `--skip-string-normalization` when normalizing multiline docstring quotes (#1637)
- `Black` no longer removes all empty lines between non-function code and decorators when formatting typing stubs. Now `Black` enforces a single empty line. (#1646)
- `Black` no longer adds an incorrect space after a parenthesized assignment expression in if/while statements (#1655)
- Added `--skip-magic-trailing-comma` / `-C` to avoid using trailing commas as a reason to split lines (#1824)
- fixed a crash when PWD=/ on POSIX (#1631)
- fixed "I/O operation on closed file" when using --diff (#1664)
- Prevent coloured diff output being interleaved with multiple files (#1673)
- Added support for PEP 614 relaxed decorator syntax on python 3.9 (#1711)
- Added parsing support for unparenthesized tuples and yield expressions in annotated assignments (#1835)

- added `--extend-exclude` argument (PR #2005)
- speed up caching by avoiding pathlib (#1950)
- `--diff` correctly indicates when a file doesn't end in a newline (#1662)
- Added `--stdin-filename` argument to allow stdin to respect `--force-exclude` rules (#1780)
- Lines ending with `fmt: skip` will now be not formatted (#1800)
- PR #2053: Black no longer relies on typed-ast for Python 3.8 and higher
- PR #2053: Python 2 support is now optional, install with `python3 -m pip install black[python2]` to maintain support.
- Exclude `venv` directory by default (#1683)
- Fixed "Black produced code that is not equivalent to the source" when formatting Python 2 docstrings (#2037)

Packaging

Self-contained native *Black* binaries are now provided for releases via GitHub Releases (#1743)

20.8b1

Packaging

explicitly depend on Click 7.1.2 or newer as `Black` no longer works with versions older than 7.0

20.8b0

Black

- re-implemented support for explicit trailing commas: now it works consistently within any bracket pair, including nested structures (#1288 and duplicates)
- `Black` now reindents docstrings when reindenting code around it (#1053)
- `Black` now shows colored diffs (#1266)
- `Black` is now packaged using 'py3' tagged wheels (#1388)
- `Black` now supports Python 3.8 code, e.g. star expressions in return statements (#1121)

- `Black` no longer normalizes capital R-string prefixes as those have a community-accepted meaning (#1244)

- `Black` now uses exit code 2 when specified configuration file doesn't exit (#1361)

- `Black` now works on AWS Lambda (#1141)

- added `--force-exclude` argument (#1032)

- removed deprecated `--py36` option (#1236)

- fixed `--diff` output when EOF is encountered (#526)

- fixed `# fmt: off` handling around decorators (#560)

- fixed unstable formatting with some `# type: ignore` comments (#1113)

- fixed invalid removal on organizing brackets followed by indexing (#1575)

- introduced `black-primer`, a CI tool that allows us to run regression tests against existing open source users of Black (#1402)

- introduced property-based fuzzing to our test suite based on Hypothesis and Hypothersmith (#1566)

- implemented experimental and disabled by default long string rewrapping (#1132), hidden under a `--experimental-string-processing` flag while it's being worked on; this is an undocumented and unsupported feature, you lose Internet points for depending on it (#1609)

Vim plugin

prefer virtualenv packages over global packages (#1383)

19.10b0

- added support for PEP 572 assignment expressions (#711)

- added support for PEP 570 positional-only arguments (#943)

- added support for async generators (#593)

- added support for pre-splitting collections by putting an explicit trailing comma inside (#826)

- added `black -c` as a way to format code passed from the command line (#761)

- --safe now works with Python 2 code (#840)

- fixed grammar selection for Python 2-specific code (#765)

- fixed feature detection for trailing commas in function definitions and call sites (#763)
- `# fmt: off` / `# fmt: on` comment pairs placed multiple times within the same block of code now behave correctly (#1005)
- *Black* no longer crashes on Windows machines with more than 61 cores (#838)
- *Black* no longer crashes on standalone comments prepended with a backslash (#767)
- *Black* no longer crashes on `from` ... `import` blocks with comments (#829)
- *Black* no longer crashes on Python 3.7 on some platform configurations (#494)
- *Black* no longer fails on comments in from-imports (#671)
- *Black* no longer fails when the file starts with a backslash (#922)
- *Black* no longer merges regular comments with type comments (#1027)
- *Black* no longer splits long lines that contain type comments (#997)
- removed unnecessary parentheses around `yield` expressions (#834)
- added parentheses around long tuples in unpacking assignments (#832)
- added parentheses around complex powers when they are prefixed by a unary operator (#646)
- fixed bug that led *Black* format some code with a line length target of 1 (#762)
- *Black* no longer introduces quotes in f-string subexpressions on string boundaries (#863)
- if *Black* puts parenthesis around a single expression, it moves comments to the wrapped expression instead of after the brackets (#872)
- `blackd` now returns the version of *Black* in the response headers (#1013)
- `blackd` can now output the diff of formats on source code when the `X-Diff` header is provided (#969)

19.3b0

- new option `--target-version` to control which Python versions *Black*-formatted code should target (#618)
- deprecated `--py36` (use `--target-version=py36` instead) (#724)
- *Black* no longer normalizes numeric literals to include `_` separators (#696)

- long `del` statements are now split into multiple lines (#698)
- type comments are no longer mangled in function signatures
- improved performance of formatting deeply nested data structures (#509)
- *Black* now properly formats multiple files in parallel on Windows (#632)
- *Black* now creates cache files atomically which allows it to be used in parallel pipelines (like `xargs -P8`) (#673)
- *Black* now correctly indents comments in files that were previously formatted with tabs (#262)
- `blackd` now supports CORS (#622)

18.9b0

- numeric literals are now formatted by *Black* (#452, #461, #464, #469):
  - numeric literals are normalized to include `_` separators on Python 3.6+ code
  - added `--skip-numeric-underscore-normalization` to disable the above behavior and leave numeric underscores as they were in the input
  - code with `_` in numeric literals is recognized as Python 3.6+
  - most letters in numeric literals are lowercased (e.g., in `1e10`, `0x01`)
  - hexadecimal digits are always uppercased (e.g. `0xBADC0DE`)
- added `blackd`, see its documentation for more info (#349)
- adjacent string literals are now correctly split into multiple lines (#463)
- trailing comma is now added to single imports that don't fit on a line (#250)
- cache is now populated when `--check` is successful for a file which speeds up consecutive checks of properly formatted unmodified files (#448)
- whitespace at the beginning of the file is now removed (#399)
- fixed mangling pweave and Spyder IDE special comments (#532)
- fixed unstable formatting when unpacking big tuples (#267)
- fixed parsing of `__future__` imports with renames (#389)
- fixed scope of `# fmt: off` when directly preceding `yield` and other nodes (#385)

- fixed formatting of lambda expressions with default arguments (#468)
- fixed `async for` statements: *Black* no longer breaks them into separate lines (#372)
- note: the Vim plugin stopped registering `,=` as a default chord as it turned out to be a bad idea (#415)

18.6b4

hotfix: don't freeze when multiple comments directly precede `# fmt: off` (#371)

18.6b3

- typing stub files (`.pyi`) now have blank lines added after constants (#340)
- `# fmt: off` and `# fmt: on` are now much more dependable:
  - they now work also within bracket pairs (#329)
  - they now correctly work across function/class boundaries (#335)
  - they now work when an indentation block starts with empty lines or misaligned comments (#334)
- made Click not fail on invalid environments; note that Click is right but the likelihood we'll need to access non-ASCII file paths when dealing with Python source code is low (#277)
- fixed improper formatting of f-strings with quotes inside interpolated expressions (#322)
- fixed unnecessary slowdown when long list literals where found in a file
- fixed unnecessary slowdown on AST nodes with very many siblings
- fixed cannibalizing backslashes during string normalization
- fixed a crash due to symbolic links pointing outside of the project directory (#338)

18.6b2

- added `--config` (#65)
- added `-h` equivalent to `--help` (#316)
- fixed improper unmodified file caching when `-S` was used
- fixed extra space in string unpacking (#305)

- fixed formatting of empty triple quoted strings (#313)
- fixed unnecessary slowdown in comment placement calculation on lines without comments

18.6b1

- hotfix: don't output human-facing information on stdout (#299)
- hotfix: don't output cake emoji on non-zero return code (#300)

18.6b0

- added `--include` and `--exclude` (#270)
- added `--skip-string-normalization` (#118)
- added `--verbose` (#283)
- the header output in `--diff` now actually conforms to the unified diff spec
- fixed long trivial assignments being wrapped in unnecessary parentheses (#273)
- fixed unnecessary parentheses when a line contained multiline strings (#232)
- fixed stdin handling not working correctly if an old version of Click was used (#276)
- *Black* now preserves line endings when formatting a file in place (#258)

18.5b1

- added `--pyi` (#249)
- added `--py36` (#249)
- Python grammar pickle caches are stored with the formatting caches, making *Black* work in environments where site-packages is not user-writable (#192)
- *Black* now enforces a PEP 257 empty line after a class-level docstring (and/or fields) and the first method
- fixed invalid code produced when standalone comments were present in a trailer that was omitted from line splitting on a large expression (#237)
- fixed optional parentheses being removed within `# fmt: off` sections (#224)
- fixed invalid code produced when stars in very long imports were incorrectly wrapped in optional parentheses (#234)

- fixed unstable formatting when inline comments were moved around in a trailer that was omitted from line splitting on a large expression (#238)
- fixed extra empty line between a class declaration and the first method if no class docstring or fields are present (#219)
- fixed extra empty line between a function signature and an inner function or inner class (#196)

18.5b0

- call chains are now formatted according to the fluent interfaces style (#67)
- data structure literals (tuples, lists, dictionaries, and sets) are now also always exploded like imports when they don't fit in a single line (#152)
- slices are now formatted according to PEP 8 (#178)
- parentheses are now also managed automatically on the right-hand side of assignments and return statements (#140)
- math operators now use their respective priorities for delimiting multiline expressions (#148)
- optional parentheses are now omitted on expressions that start or end with a bracket and only contain a single operator (#177)
- empty parentheses in a class definition are now removed (#145, #180)
- string prefixes are now standardized to lowercase and `u` is removed on Python 3.6+ only code and Python 2.7+ code with the `unicode_literals` future import (#188, #198, #199)
- typing stub files (`.pyi`) are now formatted in a style that is consistent with PEP 484 (#207, #210)
- progress when reformatting many files is now reported incrementally
- fixed trailers (content with brackets) being unnecessarily exploded into their own lines after a dedented closing bracket (#119)
- fixed an invalid trailing comma sometimes left in imports (#185)
- fixed non-deterministic formatting when multiple pairs of removable parentheses were used (#183)
- fixed multiline strings being unnecessarily wrapped in optional parentheses in long assignments (#215)
- fixed not splitting long from-imports with only a single name

- fixed Python 3.6+ file discovery by also looking at function calls with unpacking. This fixed non-deterministic formatting if trailing commas where used both in function signatures with stars and function calls with stars but the former would be reformatted to a single line.
- fixed crash on dealing with optional parentheses (#193)
- fixed "is", "is not", "in", and "not in" not considered operators for splitting purposes
- fixed crash when dead symlinks where encountered

18.4a4

don't populate the cache on `--check` (#175)

18.4a3

- added a "cache"; files already reformatted that haven't changed on disk won't be reformatted again (#109)
- `--check` and `--diff` are no longer mutually exclusive (#149)
- generalized star expression handling, including double stars; this fixes multiplication making expressions "unsafe" for trailing commas (#132)
- *Black* no longer enforces putting empty lines behind control flow statements (#90)
- *Black* now splits imports like "Mode 3 + trailing comma" of isort (#127)
- fixed comment indentation when a standalone comment closes a block (#16, #32)
- fixed standalone comments receiving extra empty lines if immediately preceding a class, def, or decorator (#56, #154)
- fixed `--diff` not showing entire path (#130)
- fixed parsing of complex expressions after star and double stars in function calls (#2)
- fixed invalid splitting on comma in lambda arguments (#133)
- fixed missing splits of ternary expressions (#141)

18.4a2

- fixed parsing of unaligned standalone comments (#99, #112)
- fixed placement of dictionary unpacking inside dictionary literals (#111)
- Vim plugin now works on Windows, too

- fixed unstable formatting when encountering unnecessarily escaped quotes in a string (#120)

18.4a1

- added `--quiet` (#78)
- added automatic parentheses management (#4)
- added pre-commit integration (#103, #104)
- fixed reporting on `--check` with multiple files (#101, #102)
- fixed removing backslash escapes from raw strings (#100, #105)

18.4a0

- added `--diff` (#87)
- add line breaks before all delimiters, except in cases like commas, to better comply with PEP 8 (#73)
- standardize string literals to use double quotes (almost) everywhere (#75)
- fixed handling of standalone comments within nested bracketed expressions; *Black* will no longer produce super long lines or put all standalone comments at the end of the expression (#22)
- fixed 18.3a4 regression: don't crash and burn on empty lines with trailing whitespace (#80)
- fixed 18.3a4 regression: `# yapf: disable` usage as trailing comment would cause *Black* to not emit the rest of the file (#95)
- when CTRL+C is pressed while formatting many files, *Black* no longer freaks out with a flurry of asyncio-related exceptions
- only allow up to two empty lines on module level and only single empty lines within functions (#74)

18.3a4

- `# fmt: off` and `# fmt: on` are implemented (#5)
- automatic detection of deprecated Python 2 forms of print statements and exec statements in the formatted file (#49)
- use proper spaces for complex expressions in default values of typed function arguments (#60)

- only return exit code 1 when --check is used (#50)
- don't remove single trailing commas from square bracket indexing (#59)
- don't omit whitespace if the previous factor leaf wasn't a math operator (#55)
- omit extra space in kwarg unpacking if it's the first argument (#46)
- omit extra space in Sphinx auto-attribute comments (#68)

18.3a3

- don't remove single empty lines outside of bracketed expressions (#19)
- added ability to pipe formatting from stdin to stdin (#25)
- restored ability to format code with legacy usage of `async` as a name (#20, #42)
- even better handling of numpy-style array indexing (#33, again)

18.3a2

- changed positioning of binary operators to occur at beginning of lines instead of at the end, following a recent change to PEP 8 (#21)
- ignore empty bracket pairs while splitting. This avoids very weirdly looking formattings (#34, #35)
- remove a trailing comma if there is a single argument to a call
- if top level functions were separated by a comment, don't put four empty lines after the upper function
- fixed unstable formatting of newlines with imports
- fixed unintentional folding of post scriptum standalone comments into last statement if it was a simple statement (#18, #28)
- fixed missing space in numpy-style array indexing (#33)
- fixed spurious space after star-based unary expressions (#31)

18.3a1

- added `--check`
- only put trailing commas in function signatures and calls if it's safe to do so. If the file is Python 3.6+ it's always safe, otherwise only safe if there are no `*args` or `**kwargs` used in the signature or call. (#8)

- fixed invalid spacing of dots in relative imports (#6, #13)
- fixed invalid splitting after comma on unpacked variables in for-loops (#23)
- fixed spurious space in parenthesized set expressions (#7)
- fixed spurious space after opening parentheses and in default arguments (#14, #17)
- fixed spurious space after unary operators when the operand was a complex expression (#15)

18.3a0

- first published version, Happy 🍰 Day 2018!
- alpha quality
- date-versioned (see: https://calver.org/)