Skip to main content
Back to top
`Ctrl`+`K`
[![conda 25.1.2.dev29 documentation -
Home](///_static/conda_logo_full.svg)](///index.html)
* [Conda](https://docs.conda.io/projects/conda/) * [Conda-build](https://docs.conda.io/projects/conda-build) * [Miniconda](https://docs.anaconda.com/free/miniconda/) * conda.org
* [GitHub](https://github.com/conda/conda "GitHub")
* [![Element](///_static/element_logo.svg)](http://bit.ly/conda-chat-room "Element") * [Discourse](https://conda.discourse.group/ "Discourse")
* [Conda](https://docs.conda.io/projects/conda/)
* [Conda-build](https://docs.conda.io/projects/conda-build)
* [Miniconda](https://docs.anaconda.com/free/miniconda/)
* conda.org
* [GitHub](https://github.com/conda/conda "GitHub") * [![Element](///_static/element_logo.svg)](http://bit.ly/conda-chat-room "Element") * [Discourse](https://conda.discourse.group/ "Discourse")

* [User guide](///user-guide/index.html)											
* [Getting started with conda](///user-guide/getting-started.html)											
* [Installing conda](///user-guide/install/index.html)											
* [Installing on Windows](///user-guide/install/windows.html)											
* [Installing on macOS](///user-guide/install/macos.html)											
* [Installing on Linux](///user-guide/install/linux.html)											
* [RPM and Debian Repositories for Miniconda](///user-guide/install/rpm-debian.html)											
* [Tasks](///user-guide/tasks/index.html)											
* [Managing conda](///user-guide/tasks/manage-conda.html)											
* [Managing environments](///user-guide/tasks/manage-environments.html)											
* [Managing channels](///user-guide/tasks/manage-channels.html)											
* [Managing packages](///user-guide/tasks/manage-pkgs.html)											
* [Managing Python](///user-guide/tasks/manage-python.html)											
* [Managing virtual packages](///user-guide/tasks/manage-virtual.html)											
* [Creating custom channels](///user-guide/tasks/create-custom-channels.html)											
* [Creating projects](///user-guide/tasks/creating-projects.html)											
* [Viewing command-line help](///user-guide/tasks/view-command-line-help.html)											
* [Configuration](///user-guide/configuration/index.html)											
* [Using the .condarc conda configuration											
file](///user-guide/configuration/use-condarc.html)											
* [Settings](///user-guide/configuration/settings.html)											
* [Administering a multi-user conda											
installation](///user-guide/configuration/admin-multi-user-install.html)											
* [Mirroring channels](///user-guide/configuration/mirroring.html)											

* [Disabling SSL verification](../../../user-guide/configuration/disable-ssl-verification.html)

```
* [Using non-standard certificates](../../../user-guide/configuration/non-standard-certs.html)
                         [Using
                                   Custom
                                               Locations
                                                            for
                                                                  Environment
                                                                                  and
                                                                                         Package
Cache](../../../user-guide/configuration/custom-env-and-pkg-locations.html)
   * [Improving interoperability with pip](../../../user-guide/configuration/pip-interoperability.html)
   * [Using the free channel](../../../user-guide/configuration/free-channel.html)
  * [Concepts](../../../user-guide/concepts/index.html) ___
   * [Commands](../../../user-guide/concepts/conda-commands.html)
   * [Packages](../../../user-quide/concepts/packages.html)
   * [Package specification](../../../user-guide/concepts/pkg-specs.html)
   * [Package search and install specifications](../../../user-guide/concepts/pkg-search.html)
   * [Channels](../../../user-guide/concepts/channels.html)
   * [Environments](../../../user-guide/concepts/environments.html)
   * [Installing with conda](../../../user-guide/concepts/installing-with-conda.html)
   * [Performance](../../../user-guide/concepts/conda-performance.html)
   * [Conda for data scientists](../../../user-guide/concepts/data-science.html)
   * [Plugins](../../../user-guide/concepts/conda-plugins.html)
  * [Troubleshooting](../../../user-guide/troubleshooting.html)
  * [Cheatsheet](../../../user-guide/cheatsheet.html)
 * [Configuration](../../../configuration.html)
 * [Commands](../../../commands/index.html)
  * [`conda clean`](../../../commands/clean.html)
  * [`conda compare`](../../../commands/compare.html)
  * [`conda config`](../../../commands/config.html)
  * [`conda create`](../../../commands/create.html)
  * [`conda doctor`](../../../commands/doctor.html)
  * [`conda env`](../../../commands/env/index.html)
   * [` conda env config`](../../../commands/env/config/index.html) ___
```

```
* [` conda env config vars`](../../../commands/env/config/vars/index.html) ___
    * [` conda env config vars list`](../../../commands/env/config/vars/list.html)
    * [`conda env config vars set`](../../../commands/env/config/vars/set.html)
    * [`conda env config vars unset`](../../../commands/env/config/vars/unset.html)
  * [`conda env create`](../../../commands/env/create.html)
  * [`conda env export`](../../../commands/env/export.html)
  * [`conda env list`](../../../commands/env/list.html)
  * [`conda env remove`](../../../commands/env/remove.html)
  * [`conda env update`](../../../commands/env/update.html)
* [`conda info`](../../../commands/info.html)
* [`conda init`](../../../commands/init.html)
* [`conda install`](../../../commands/install.html)
 * [`conda list`](../../../commands/list.html)
* [`conda notices`](../../../commands/notices.html)
 * [`conda package`](../../../commands/package.html)
 * [`conda remove`](../../../commands/remove.html)
 * [`conda rename`](../../../commands/rename.html)
* [`conda run`](../../../commands/run.html)
* [`conda search`](../../../commands/search.html)
 * [`conda update`](../../../commands/update.html)
* [Release notes](../../../release-notes.html)
* [Glossary](../../../glossary.html)
* [Developer guide](../../../index.html) ___
 * [Architecture](../../../architecture.html)
 * [Contributing to conda](../../../contributing.html)
 * [Development Environment](../../../development-environment.html)
 * [Deep dives](../../../deep-dives/index.html) ___
```

```
* [` conda install`](../../../deep-dives/install.html)
 * [`conda init` and `conda activate`](../../../deep-dives/activation.html)
 * [`conda config` and context](../../../deep-dives/context.html)
 * [Solvers](../../../deep-dives/solvers.html)
 * [Logging](../../../deep-dives/logging.html)
* [Writing Tests](../../../writing-tests/index.html) ___
 * [Integration Tests](../../.writing-tests/integration-tests.html)
* [Deprecations](../../../deprecations.html)
* [Releasing](../../../releasing.html)
* [Plugins](../../../plugins/index.html)
 * [Auth Handlers](../../plugins/auth_handlers.html)
 * [Health Checks](../../../plugins/health_checks.html)
 * [Request Headers](../../../plugins/request_headers.html)
 * [Post-commands](../../../plugins/post_commands.html)
 * [Pre-commands](../../../plugins/pre commands.html)
 * [Reporter Backends](../../../plugins/reporter_backends.html)
 * [Settings](../../../plugins/settings.html)
 * [Solvers](../../../plugins/solvers.html)
 * [Subcommands](../../../plugins/subcommands.html)
 * [Virtual Packages](../../../plugins/virtual packages.html)
* [Specifications](../../../specs/index.html) ___
 * [Technical specification: solver state](../../../specs/solver-state.html)
* [API](../../../api.html)
 * [` conda`](../../index.html) ___
  * [` __main__`](../../__main__/index.html)
  * [` vendor`](../../ vendor/index.html)
   * [`frozendict`](../../_vendor/frozendict/index.html)
```

```
* [`_version`](../../_version/index.html)
* [`activate`](../../activate/index.html)
* [`api`](../../api/index.html)
* [`auxlib`](../../auxlib/index.html)
 * [` collection`](../../auxlib/collection/index.html)
 * [`compat`](../../auxlib/compat/index.html)
 * [`decorators`](../../auxlib/decorators/index.html)
 * [`entity`](../../auxlib/entity/index.html)
 * [`exceptions`](../../auxlib/exceptions/index.html)
 * [`ish`](../../auxlib/ish/index.html)
 * [`logz`](../../auxlib/logz/index.html)
 * [`type_coercion`](../../auxlib/type_coercion/index.html)
* [`base`](../../base/index.html) ___
 * [`constants`](../../base/constants/index.html)
 * [`context`](../../base/context/index.html)
* [`cli`](../../cli/index.html) ___
 * [`actions`](../../cli/actions/index.html)
 * [`common`](../../cli/common/index.html)
 * [`conda_argparse`](../../cli/conda_argparse/index.html)
 * [`find_commands`](../../cli/find_commands/index.html)
 * [`helpers`](../../cli/helpers/index.html)
 * [`install`](../../cli/install/index.html)
 * [`main`](../../cli/main/index.html)
 * [`main_clean`](../../cli/main_clean/index.html)
 * [`main_commands`](../../cli/main_commands/index.html)
 * [`main compare`](../../cli/main compare/index.html)
```

* [`main_config`](../../cli/main_config/index.html)

```
* [`main_create`](../../cli/main_create/index.html)
```

- * [`main_env`](../../cli/main_env/index.html)
- * [`main_env_config`](../../cli/main_env_config/index.html)
- * [`main_env_create`](../../cli/main_env_create/index.html)
- * [`main_env_export`](../../cli/main_env_export/index.html)
- * [`main_env_list`](../../cli/main_env_list/index.html)
- * [`main_env_remove`](../../cli/main_env_remove/index.html)
- * [`main_env_update`](../../cli/main_env_update/index.html)
- * [`main env vars`](../../cli/main env vars/index.html)
- * [`main export`](../../cli/main export/index.html)
- * [`main info`](../../cli/main info/index.html)
- * [`main_init`](../../cli/main_init/index.html)
- * [`main_install`](../../cli/main_install/index.html)
- * [`main_list`](../../cli/main_list/index.html)
- * [`main_mock_activate`](../../cli/main_mock_activate/index.html)
- * [`main_mock_deactivate`](../../cli/main_mock_deactivate/index.html)
- * [`main_notices`](../../cli/main_notices/index.html)
- * [`main_package`](../../cli/main_package/index.html)
- * [`main pip`](../../cli/main pip/index.html)
- * [`main remove`](../../cli/main remove/index.html)
- * [`main_rename`](../../cli/main_rename/index.html)
- * [`main_run`](../../cli/main_run/index.html)
- * [`main_search`](../../cli/main_search/index.html)
- * [`main_update`](../../cli/main_update/index.html)
- * [`python_api`](../../cli/python_api/index.html)
- * [`common`](../index.html)
 - * [`_logic`](../_logic/index.html)

```
* [`_os`](../_os/index.html) ___
  * [` linux`](../_os/linux/index.html)
  * [`osx`](../_os/osx/index.html)
  * [`unix`](../_os/unix/index.html)
  * [`windows`](../_os/windows/index.html)
 * [`compat`](../compat/index.html)
 * [`configuration`](../configuration/index.html)
 * [`constants`](../constants/index.html)
 * [`disk`](../disk/index.html)
 * `io`
 * [`iterators`](../iterators/index.html)
 * [`logic`](../logic/index.html)
 * [`path`](../path/index.html) ___
  * [`_cygpath`](../path/_cygpath/index.html)
  * [`directories`](../path/directories/index.html)
  * [`python`](../path/python/index.html)
  * [`windows`](../path/windows/index.html)
 * [`pkg_formats`](../pkg_formats/index.html) ___
  * [`python`](../pkg_formats/python/index.html)
 * [`serialize`](../serialize/index.html)
 * [`signals`](../signals/index.html)
 * [`toposort`](../toposort/index.html)
 * [`url`](../url/index.html)
* [`core`](../../core/index.html) ___
 * [` envs_manager`](../../core/envs_manager/index.html)
 * [`index`](../../core/index/index.html)
```

* [`initialize`](../../core/initialize/index.html)

```
* [`link`](../../core/link/index.html)
 * [`package_cache_data`](../../core/package_cache_data/index.html)
 * [`path_actions`](../../core/path_actions/index.html)
 * [`portability`](../../core/portability/index.html)
 * [`prefix_data`](../../core/prefix_data/index.html)
 * [`solve`](../../core/solve/index.html)
 * [`subdir_data`](../../core/subdir_data/index.html)
* [`deprecations`](../../deprecations/index.html)
* [`env`](../../env/index.html)
 * [` env`](../../env/env/index.html)
 * [`installers`](../../env/installers/index.html) ___
  * [`base`](../../env/installers/base/index.html)
  * [`conda`](../../env/installers/conda/index.html)
  * [`pip`](../../env/installers/pip/index.html)
 * [`pip_util`](../../env/pip_util/index.html)
 * [`specs`](../../env/specs/index.html) ___
  * [`binstar`](../../env/specs/binstar/index.html)
  * [`requirements`](../../env/specs/requirements/index.html)
  * ['yaml_file'](../../env/specs/yaml_file/index.html)
* [`exception_handler`](../../exception_handler/index.html)
* [`exceptions`](../../exceptions/index.html)
* [`exports`](../../exports/index.html)
* [`gateways`](../../gateways/index.html) ___
 * [` anaconda_client`](../../gateways/anaconda_client/index.html)
 * [`connection`](../../gateways/connection/index.html) ___
  * [` adapters`](../../gateways/connection/adapters/index.html) ___
   * [`ftp`](../../gateways/connection/adapters/ftp/index.html)
```

```
* [`http`](../../gateways/connection/adapters/http/index.html)
    * [`localfs`](../../gateways/connection/adapters/localfs/index.html)
    * [`s3`](../../gateways/connection/adapters/s3/index.html)
  * [`download`](../../gateways/connection/download/index.html)
  * [`session`](../../gateways/connection/session/index.html)
 * [`disk`](../../gateways/disk/index.html) ___
  * [` create`](../../gateways/disk/create/index.html)
  * [`delete`](../../gateways/disk/delete/index.html)
  * [`link`](../../gateways/disk/link/index.html)
  * [`lock`](../../gateways/disk/lock/index.html)
  * [`permissions`](../../gateways/disk/permissions/index.html)
  * [`read`](../../gateways/disk/read/index.html)
  * [`test`](../../gateways/disk/test/index.html)
  * [`update`](../../gateways/disk/update/index.html)
 * [`logging`](../../gateways/logging/index.html)
 * [`repodata`](../../gateways/repodata/index.html) ___
  * [` jlap`](../../gateways/repodata/jlap/index.html) ___
   * [`core`](../../gateways/repodata/jlap/core/index.html)
   * [`fetch`](../../gateways/repodata/jlap/fetch/index.html)
   * [`interface`](../../gateways/repodata/jlap/interface/index.html)
  * [`lock`](../../gateways/repodata/lock/index.html)
 * [`subprocess`](../../gateways/subprocess/index.html)
* [`history`](../../history/index.html)
* [`instructions`](../../instructions/index.html)
* [`misc`](../../misc/index.html)
* [`models`](../../models/index.html)
 * [` channel`](../../models/channel/index.html)
```

```
* [`dist`](../../models/dist/index.html)
 * [`enums`](../../models/enums/index.html)
 * [`leased_path_entry`](../../models/leased_path_entry/index.html)
 * [`match_spec`](../../models/match_spec/index.html)
 * [`package_info`](../../models/package_info/index.html)
 * [`prefix_graph`](../../models/prefix_graph/index.html)
 * [`records`](../../models/records/index.html)
 * [`version`](../../models/version/index.html)
* [`notices`](../../notices/index.html)
 * [` cache`](../../notices/cache/index.html)
 * [`core`](../../notices/core/index.html)
 * [`fetch`](../../notices/fetch/index.html)
 * [`types`](../../notices/types/index.html)
 * [`views`](../../notices/views/index.html)
* [`plan`](../../plan/index.html)
* [`plugins`](../../plugins/index.html) ___
 * [`hookspec'](../../plugins/hookspec/index.html)
 * [`manager`](../../plugins/manager/index.html)
 * [`post_solves`](../../plugins/post_solves/index.html) ___
  * [` signature_verification`](../../plugins/post_solves/signature_verification/index.html)
 * [`reporter_backends`](../../plugins/reporter_backends/index.html) ___
  * [` console`](../../plugins/reporter_backends/console/index.html)
  * [`json`](../../plugins/reporter_backends/json/index.html)
 * [`solvers`](../../plugins/solvers/index.html)
 * [`subcommands`](../../plugins/subcommands/index.html) ___
  * [` doctor`](../../plugins/subcommands/doctor/index.html) ___
   * [` health_checks`](../../plugins/subcommands/doctor/health_checks/index.html)
```

```
* [`types`](../../plugins/types/index.html)
  * [`virtual_packages`](../../plugins/virtual_packages/index.html) ___
   * [`archspec`](../../plugins/virtual_packages/archspec/index.html)
   * [`conda`](../../plugins/virtual_packages/conda/index.html)
   * [`cuda`](../../plugins/virtual_packages/cuda/index.html)
   * [`freebsd`](../../plugins/virtual_packages/freebsd/index.html)
   * [`linux`](../../plugins/virtual_packages/linux/index.html)
   * [`osx`](../../plugins/virtual_packages/osx/index.html)
   * [`windows`](../../plugins/virtual_packages/windows/index.html)
 * [`reporters`](../../reporters/index.html)
 * [`resolve`](../../resolve/index.html)
 * [`testing`](../../testing/index.html) ___
  * [` cases`](../../testing/cases/index.html)
  * [`fixtures`](../../testing/fixtures/index.html)
  * [`gateways`](../../testing/gateways/index.html) ___
   * [` fixtures`](../../testing/gateways/fixtures/index.html)
  * [`helpers`](../../testing/helpers/index.html)
  * [`integration`](../../testing/integration/index.html)
  * [`notices`](../../testing/notices/index.html) ___
   * [` fixtures`](../../testing/notices/fixtures/index.html)
   * [`helpers`](../../testing/notices/helpers/index.html)
  * [`solver_helpers`](../../testing/solver_helpers/index.html)
 * [`trust`](../../trust/index.html) ___
  * [` constants`](../../trust/constants/index.html)
  * [`signature_verification`](../../trust/signature_verification/index.html)
 * [`utils`](../../utils/index.html)
* [`conda_env`](../../conda_env/index.html) ___
```

```
* [`installers`](../../conda_env/installers/index.html)
 * [ ___](../../../index.html)
 * [Developer guide](../../../index.html)
 * [` common`](../index.html)
# [`io`](https://docs.python.org/3/library/io.html#module-io "\(in Python
v3.13\)")#
Common I/O utilities.
## Classes#
`DeltaSecondsFormatter` | Logging formatter with additional attributes for run time logging.
---|---
`ContextDecorator` | Base class for a context manager class (implementing enter () and
__exit__()) that also
`SwallowBrokenPipe` | Base class for a context manager class (implementing __enter__() and
__exit__()) that also
`CaptureTarget` | Constants used for contextmanager captured.
`Spinner` |
`ProgressBar` |
`DummyExecutor` | This is an abstract base class for concrete asynchronous executors.
`ThreadLimitedThreadPoolExecutor` | This is an abstract base class for concrete asynchronous
```

* [`cli`](../../conda_env/cli/index.html)

```
`time_recorder` | Base class for a context manager class (implementing __enter__() and __exit__())
that also
## Functions#
---|---
`env vars`([var map, callback, stack callback]) |
`env_var`(name, value[, callback, stack_callback]) |
`env_unmodified`([callback]) |
`captured`([stdout, stderr]) | Capture outputs of sys.stdout and sys.stderr.
`argv`(args_list) |
`_logger_lock`() |
`disable_logger`(logger_name) |
`stderr_log_level`(level[, logger_name]) |
`attach_stderr_handler`([level, logger_name, propagate, ...]) | Attach a new stderr handler to the
given logger and configure both.
`timeout`(timeout secs, func, *args[, default return]) | Enforce a maximum time for a callable to
complete.
`get_instrumentation_record_file`() |
`print_instrumentation_data`() |
## Attributes#
`IS_INTERACTIVE` |
---|---
```

executors.

`_FORMATTER`
`swallow_broken_pipe`
`as_completed`
IS_INTERACTIVE#
_class _DeltaSecondsFormatter(_fmt =None_, _datefmt =None_)#
Bases:
[`logging.Formatter`](https://docs.python.org/3/library/logging.html#logging.Formatter
"\(in Python v3.13\)")
Logging formatter with additional attributes for run time logging.
`delta_secs`
Elapsed seconds since last log/format call (or creation of logger).
`relative_created_secs`

Like relativeCreated, time relative to the initialization of the logging module but conveniently scaled to seconds as a float value.

Initialize the formatter with specified format strings.

Initialize the formatter either with the specified format string, or a default as described above. Allow for specialized date formatting with the optional datefmt argument. If datefmt is omitted, you get an ISO8601-like (or RFC 3339-like) format.

Use a style parameter of '%', '{' or '\$' to specify that you want to use one of %-formatting,

[`str.format()`](https://docs.python.org/3/library/stdtypes.html#str.format "\(in Python v3.13\)") (`{}`) formatting or

[`string.Template`](https://docs.python.org/3/library/string.html#string.Template "\(in Python v3.13\)") formatting in your format string.

Changed in version 3.2: Added the `style` parameter.

format(_record_)#

Format the specified record as text.

The record's attribute dictionary is used as the operand to a string

formatting operation which yields the returned string. Before formatting the dictionary, a couple of preparatory steps are carried out. The message attribute of the record is computed using LogRecord.getMessage(). If the formatting string uses the time (as determined by a call to usesTime(), formatTime() is called to format the event time. If there is exception information, it is formatted using formatException() and appended to the message.

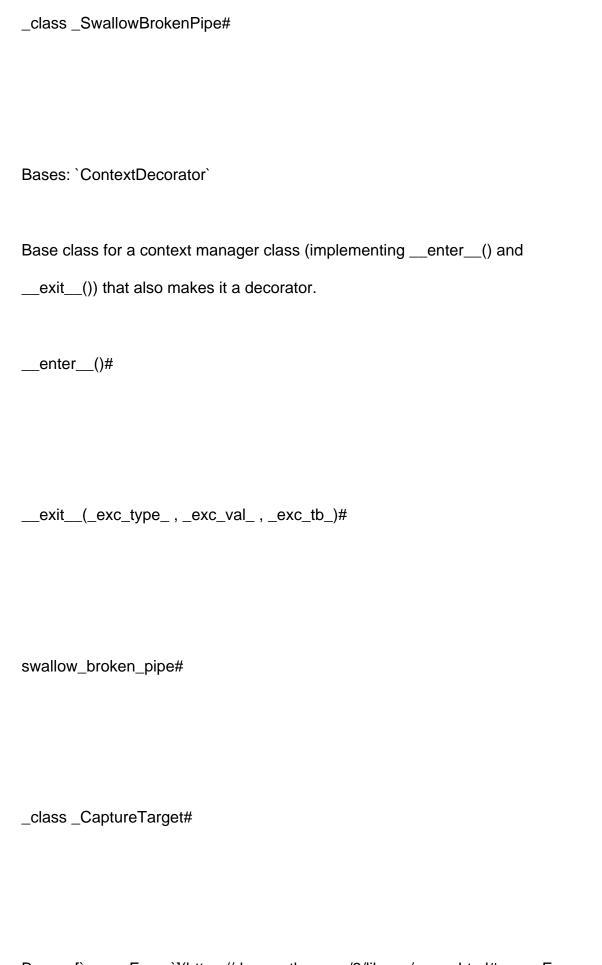
FORMATTER#

dashlist(_iterable_ , _indent =2_)#

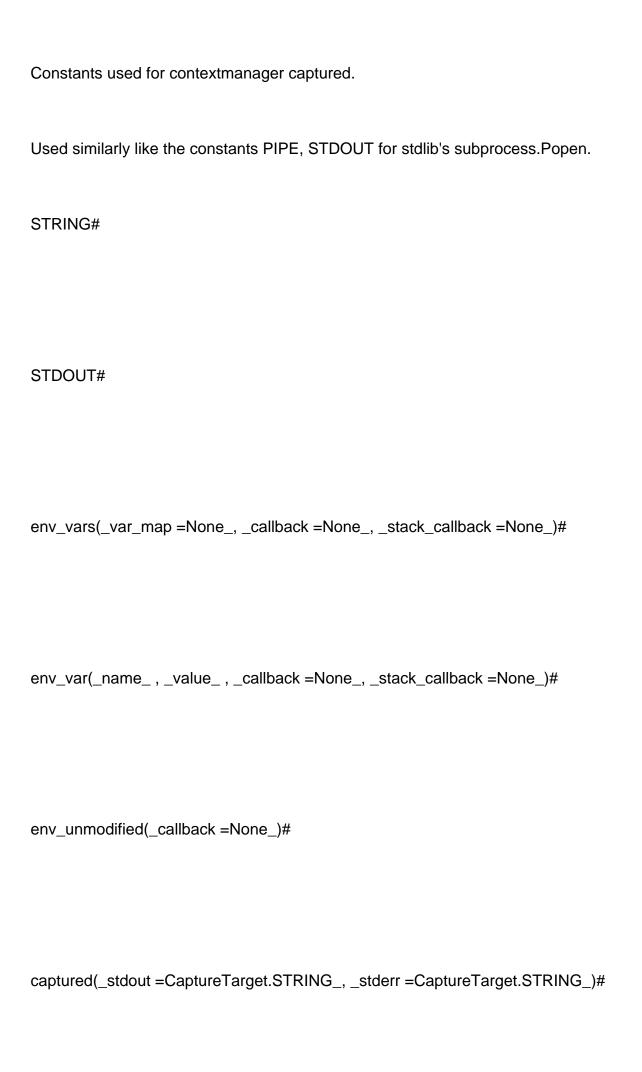
_class _ContextDecorator#

Base class for a context manager class (implementing __enter__() and __exit__()) that also makes it a decorator.

__call__(_f_)#



Bases: [`enum.Enum`](https://docs.python.org/3/library/enum.html#enum.Enum "\(in Python v3.13\)")



Capture outputs of sys.stdout and sys.stderr.										
If stdout is STRING, capture sys.stdout as a string, if stdout is None, do not										
capture sys.stdout, leaving it untouched, otherwise redirect sys.stdout to the										
file-like object given by stdout.										
Behave correspondingly for stderr with the exception that if stderr is STDOUT,										
redirect sys.stderr to stdout target and set stderr attribute of yielded										
object to None.										
>>> from conda.common.io import captured										
>>> with captured() as c:										
print("hello world!")										
····										
>>> c.stdout										
'hello world!\n'										
Parameters:										

* **stdout** \-- capture target for sys.stdout, one of STRING, None, or file-like object



stderr_log_level(_level_ , _logger_name =None_)#

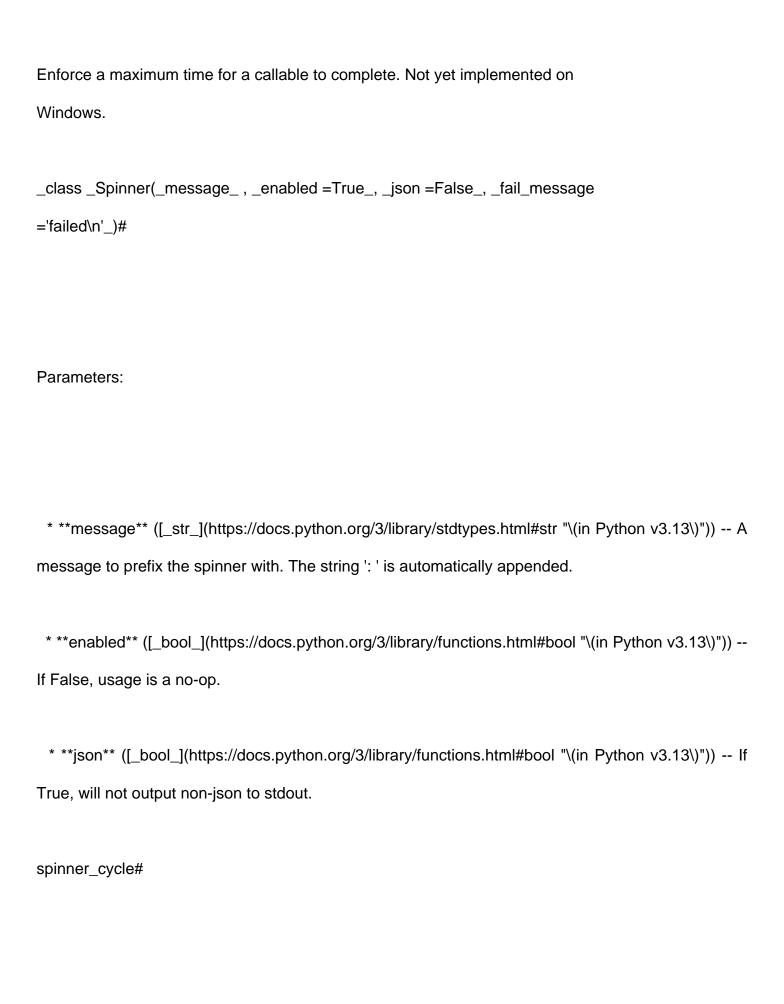
attach_stderr_handler(_level =WARN_, _logger_name =None_, _propagate =False_, _formatter =None_, _filters =None_)#

Attach a new stderr handler to the given logger and configure both.

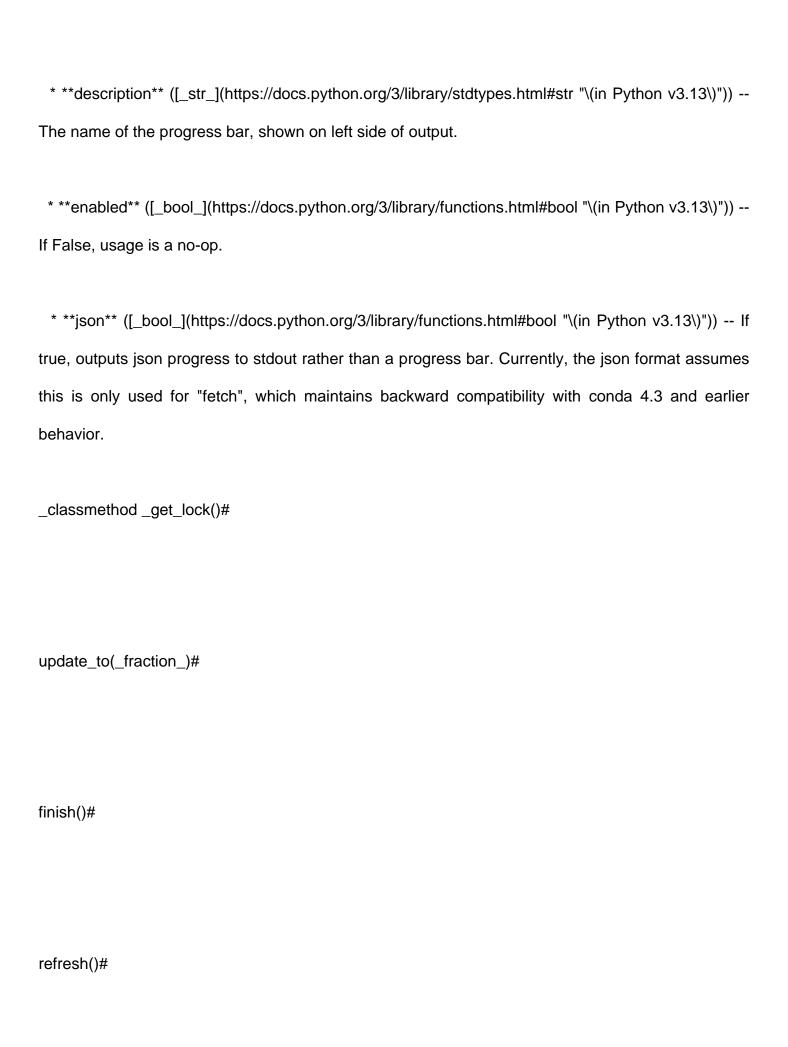
This function creates a new StreamHandler that writes to stderr and attaches it to the logger given by logger_name (which maybe None, in which case the root logger is used). If the logger already has a handler by the name of stderr, it is removed first.

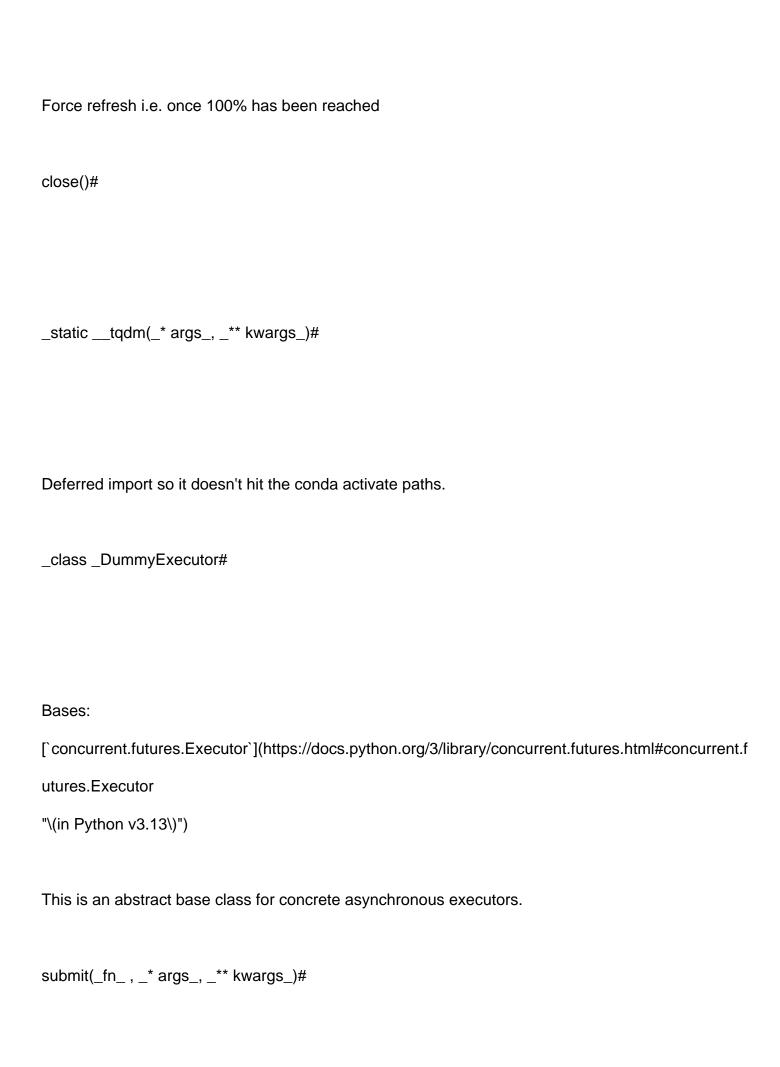
The given level is set **for the handler**, not for the logger; however, this function also sets the level of the given logger to the minimum of its current effective level and the new handler level, ensuring that the handler will receive the required log records, while minimizing the number of unnecessary log events. It also sets the loggers propagate property according to the propagate argument. The formatter argument can be used to set the formatter of the handler.

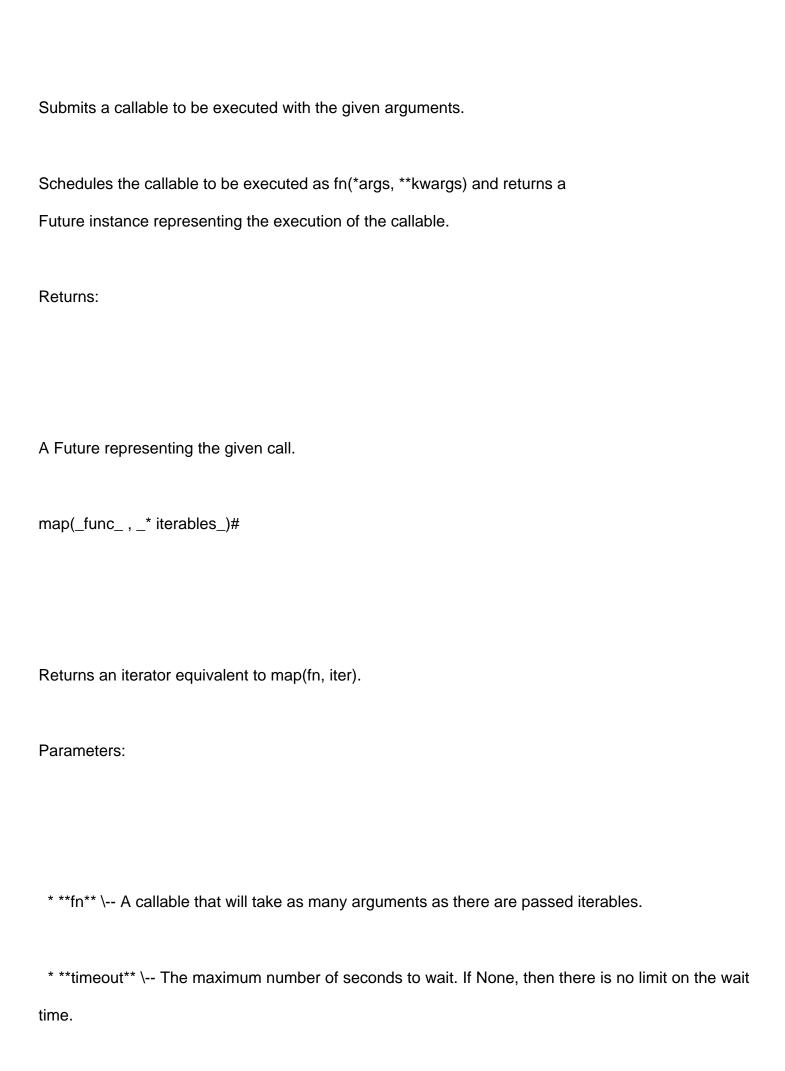
timeout(_timeout_secs_ , _func_ , _* args_, _default_return =None_, _**
kwargs_)#



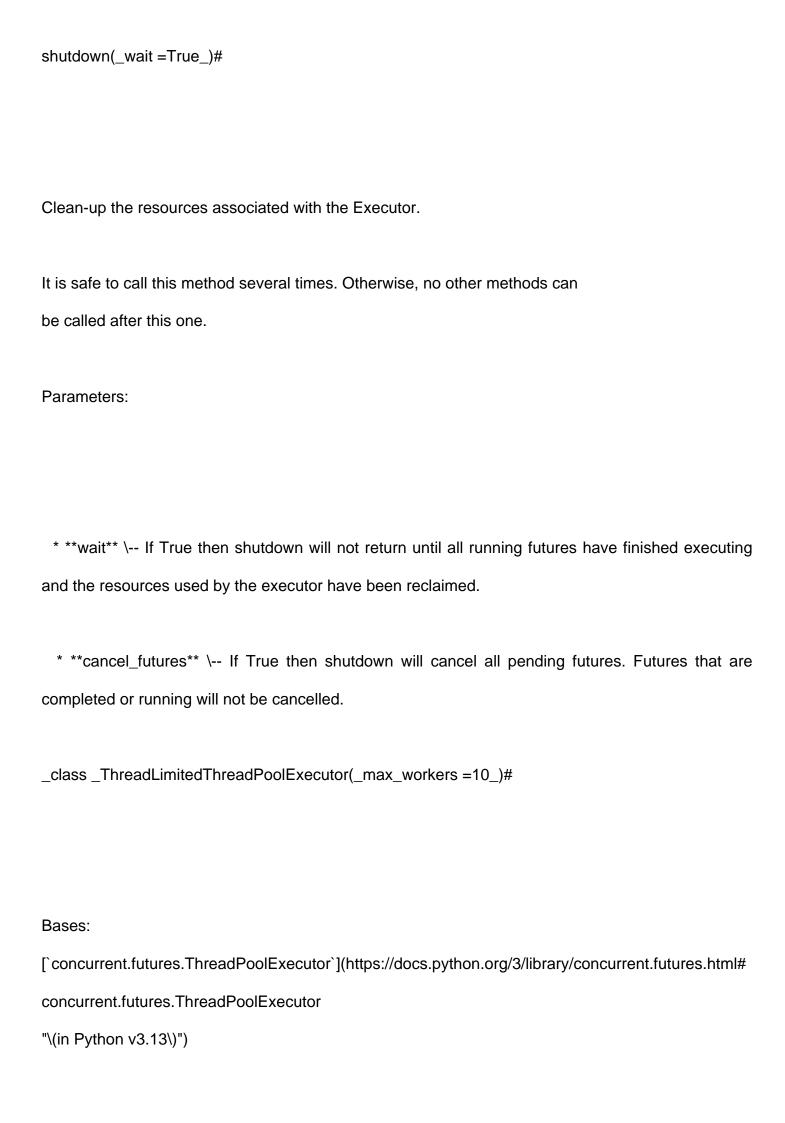


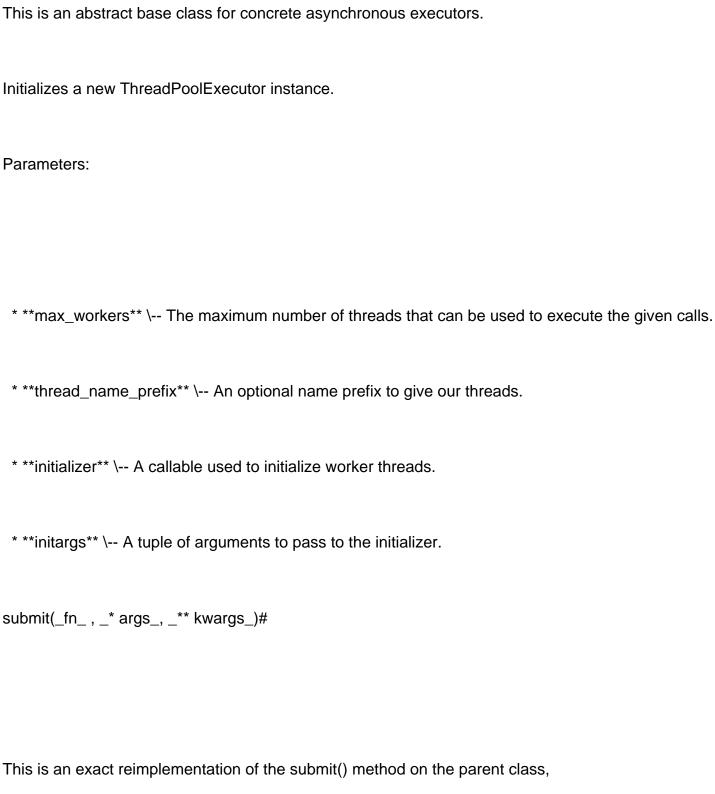






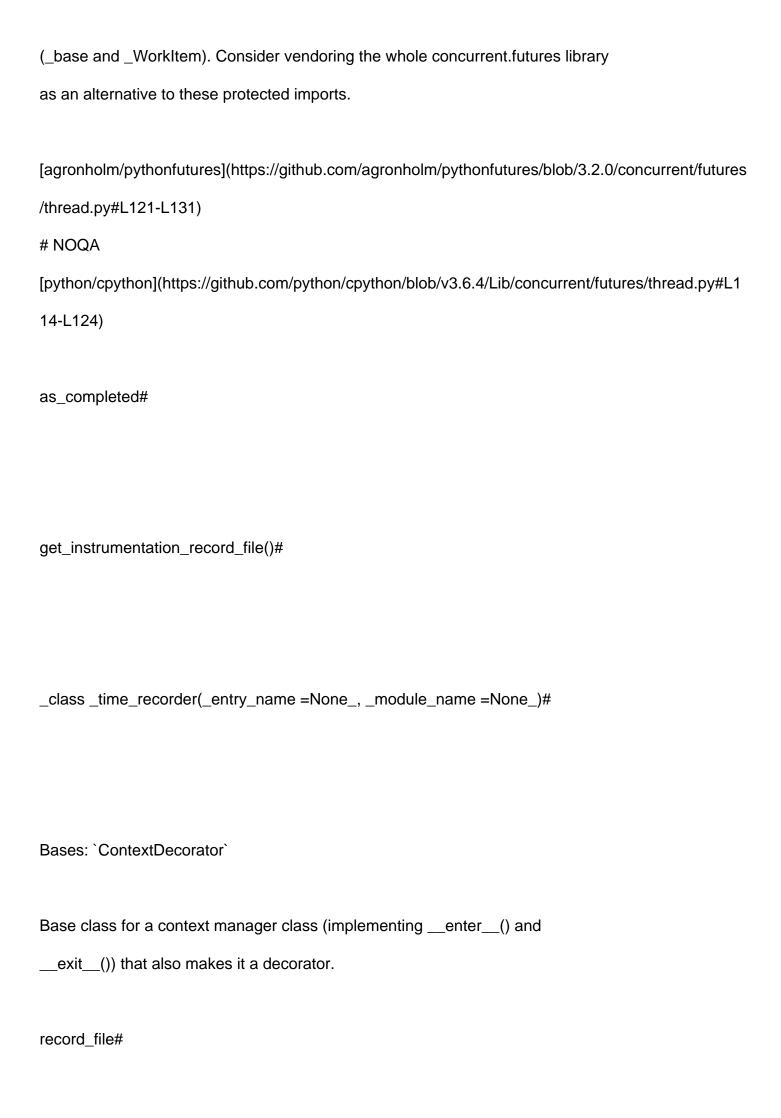
* **(chunksize**	' \ Th	e size of the	e ch	nunks	the itera	able	will be broke	en into befor	re be	eing	passe	d to a
child	process.	This	argument	is	only	used	by	ProcessPoo	olExecutor;	it	is	ignore	d by
Threa	adPoolExec	cutor.											
Retur	ns:												
,	,												
map(func, ^iteral	oles) b	ut the calls	may	be ev	/aluated	d out	-of-order.					
Retur	n type:												
An ite	erator equiv	alent t	0										
Raise	es:												
								exceptions.h				'\(in P	ython
			s://docs.pyt		.org/3/	/library/	exce	ptions.html#	Exception "	\(in I	Pyth	on v3.	13\)")





This is an exact reimplementation of the submit() method on the parent class, except with an added try/except around self._adjust_thread_count(). So long as there is at least one living thread, this thread pool will not throw an exception if threads cannot be expanded to max_workers.

In the implementation, we use "protected" attributes from concurrent.futures



start_time# total_call_num# total_run_time# _set_entry_name(_f_)# __call__(_f_)#

__enter__()#

```
__exit__(_exc_type_ , _exc_val_ , _exc_tb_)#
_classmethod _log_totals()#
_ensure_dir()#
print_instrumentation_data()#
__On this page
 * Classes
 * Functions
 * Attributes
  * `IS_INTERACTIVE`
  * `DeltaSecondsFormatter`
   * `DeltaSecondsFormatter.format()`
  * `_FORMATTER`
  * `dashlist()`
```

* `ContextDecorator`

```
* `ContextDecorator.__call__()`
* `SwallowBrokenPipe`
 * `SwallowBrokenPipe.__enter__()`
 * `SwallowBrokenPipe.__exit__()`
* `swallow_broken_pipe`
* `CaptureTarget`
 * `CaptureTarget.STRING`
 * `CaptureTarget.STDOUT`
* `env_vars()`
* `env_var()`
* `env_unmodified()`
* `captured()`
* `argv()`
* `_logger_lock()`
* `disable_logger()`
* `stderr_log_level()`
* `attach_stderr_handler()`
* `timeout()`
* `Spinner`
 * `Spinner_cycle`
 * `Spinner.start()`
 * `Spinner.stop()`
 * `Spinner._start_spinning()`
 * `Spinner.__enter__()`
 * `Spinner.__exit__()`
* `ProgressBar`
 * `ProgressBar.get_lock()`
```

```
* `ProgressBar.update_to()`
 * `ProgressBar.finish()`
 * `ProgressBar.refresh()`
 * `ProgressBar.close()`
 * `ProgressBar._tqdm()`
* `DummyExecutor`
 * `DummyExecutor.submit()`
 * `DummyExecutor.map()`
 * `DummyExecutor.shutdown()`
* `ThreadLimitedThreadPoolExecutor`
 * `ThreadLimitedThreadPoolExecutor.submit()`
* `as_completed`
* `get_instrumentation_record_file()`
* `time recorder`
 * `time_recorder.record_file`
 * `time_recorder.start_time`
 * `time_recorder.total_call_num`
 * `time_recorder.total_run_time`
 * `time recorder. set entry name()`
 * `time_recorder.__call__()`
 * `time_recorder.__enter__()`
 * `time_recorder.__exit__()`
 * `time_recorder.log_totals()`
 * `time_recorder._ensure_dir()`
* `print_instrumentation_data()`
```