

# Runners

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]asyncio-runner.rst, line 1)**

Unknown directive type "currentmodule".

```
.. currentmodule:: asyncio
```

Source code: `:source:`Lib/asyncio/runners.py``

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]asyncio-runner.rst, line 8); [backlink](#)**

Unknown interpreted text role "source".

This section outlines high-level asyncio primitives to run asyncio code.

They are built on top of an [ref` event loop <asyncio-event-loop>](#) with the aim to simplify async code usage for common widespread scenarios.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]asyncio-runner.rst, line 13); [backlink](#)**

Unknown interpreted text role "ref".

- [Running an asyncio Program](#)
- [Runner context manager](#)
- [Handling Keyboard Interruption](#)

## Running an asyncio Program

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]asyncio-runner.rst, line 25)**

Unknown directive type "function".

```
.. function:: run(coro, *, debug=None)
```

Execute the `:term:`coroutine` *coro*` and return the result.

This function runs the passed coroutine, taking care of managing the asyncio event loop, `*finalizing asynchronous generators*`, and closing the threadpool.

This function cannot be called when another asyncio event loop is running in the same thread.

If `*debug*` is ```True```, the event loop will be run in debug mode. ```False``` disables debug mode explicitly. ```None``` is used to respect the global `:ref:`asyncio-debug-mode`` settings.

This function always creates a new event loop and closes it at the end. It should be used as a main entry point for asyncio programs, and should ideally only be called once.

Example::

```
async def main():
    await asyncio.sleep(1)
    print('hello')
```

```
asyncio.run(main())
```

```
.. versionadded:: 3.7
```

```
.. versionchanged:: 3.9
    Updated to use :meth:`loop.shutdown_default_executor`.
```

```
.. versionchanged:: 3.10
```

`*debug*` is ``None`` by default to respect the global debug mode settings.

## Runner context manager

A context manager that simplifies *multiple* async function calls in the same context.

Sometimes several top-level async functions should be called in the same `ref`event loop <asyncio-event-loop>`` and `class`contextvars.Context``.

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]asyncio-runner.rst, line 70); [backlink](#)

Unknown interpreted text role "ref".

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]asyncio-runner.rst, line 70); [backlink](#)

Unknown interpreted text role "class".

If `debug` is `True`, the event loop will be run in debug mode. `False` disables debug mode explicitly. `None` is used to respect the global `ref`asyncio-debug-mode`` settings.

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]asyncio-runner.rst, line 73); [backlink](#)

Unknown interpreted text role "ref".

`loop_factory` could be used for overriding the loop creation. `func`asyncio.new_event_loop`` is used if `None`.

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]asyncio-runner.rst, line 77); [backlink](#)

Unknown interpreted text role "func".

Basically, `func`asyncio.run()`` example can be rewritten with the runner usage:

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]asyncio-runner.rst, line 80); [backlink](#)

Unknown interpreted text role "func".

```
async def main():
    await asyncio.sleep(1)
    print('hello')
```

```
with asyncio.Runner() as runner:
    runner.run(main())
```

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]asyncio-runner.rst, line 89)

Unknown directive type "versionadded".

```
.. versionadded:: 3.11
```

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]asyncio-runner.rst, line 91)

Unknown directive type "method".

```
.. method:: run(coro, *, context=None)
```

Run a `:term:`coroutine <coroutine>`` `*coro*` in the embedded loop.

Return the coroutine's result or raise its exception.

An optional keyword-only `*context*` argument allows specifying a custom `:class:`contextvars.Context`` for the `*coro*` to run in. The runner's default context is used if ``None``.

This function cannot be called when another asyncio event loop is running in the same thread.

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] asyncio-runner.rst, line 104)

Unknown directive type "method".

```
.. method:: close()
```

Close the runner.

Finalize asynchronous generators, shutdown default executor, close the event loop and release embedded `:class:`contextvars.Context``.

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] asyncio-runner.rst, line 111)

Unknown directive type "method".

```
.. method:: get_loop()
```

Return the event loop associated with the runner instance.

## Note

`:class:`Runner`` uses the lazy initialization strategy, its constructor doesn't initialize underlying low-level structures.

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] asyncio-runner.rst, line 117); [backlink](#)

Unknown interpreted text role "class".

Embedded `loop` and `context` are created at the `:keyword:`with`` body entering or the first call of `:meth:`run`` or `:meth:`get_loop``.

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] asyncio-runner.rst, line 120); [backlink](#)

Unknown interpreted text role "keyword".

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] asyncio-runner.rst, line 120); [backlink](#)

Unknown interpreted text role "meth".

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] asyncio-runner.rst, line 120); [backlink](#)

Unknown interpreted text role "meth".

## Handling Keyboard Interruption

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] asyncio-runner.rst, line 127)

Unknown directive type "versionadded".

```
.. versionadded:: 3.11
```

When `:const:`signal.SIGINT`` is raised by `:kbd:`Ctrl-C``, `:exc:`KeyboardInterrupt`` exception is raised in the main thread by default. However this doesn't work with `:mod:`asyncio`` because it can interrupt asyncio internals and can hang the program from exiting.

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]asyncio-runner.rst, line 129); [backlink](#)

Unknown interpreted text role "const".

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]asyncio-runner.rst, line 129); [backlink](#)

Unknown interpreted text role "kbd".

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]asyncio-runner.rst, line 129); [backlink](#)

Unknown interpreted text role "exc".

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]asyncio-runner.rst, line 129); [backlink](#)

Unknown interpreted text role "mod".

To mitigate this issue, `:mod:`asyncio`` handles `:const:`signal.SIGINT`` as follows:

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]asyncio-runner.rst, line 134); [backlink](#)

Unknown interpreted text role "mod".

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]asyncio-runner.rst, line 134); [backlink](#)

Unknown interpreted text role "const".

1. `:meth:`asyncio.Runner.run`` installs a custom `:const:`signal.SIGINT`` handler before any user code is executed and removes it when exiting from the function.

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]asyncio-runner.rst, line 136); [backlink](#)

Unknown interpreted text role "meth".

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]asyncio-runner.rst, line 136); [backlink](#)

Unknown interpreted text role "const".

2. The `:class:`~asyncio.Runner`` creates the main task for the passed coroutine for its execution.

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]asyncio-runner.rst, line 138); [backlink](#)

Unknown interpreted text role "class".

3. When `:const:`signal.SIGINT`` is raised by `:kbd:`Ctrl-C``, the custom signal handler cancels the main task by calling `:meth:`asyncio.Task.cancel`` which raises `:exc:`asyncio.CancelledError`` inside the the main task. This causes the Python stack to unwind, `try/except` and `try/finally` blocks can be used for resource cleanup. After the main task is cancelled, `:meth:`asyncio.Runner.run`` raises `:exc:`KeyboardInterrupt``.

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] asyncio-runner.rst, line 140); [backlink](#)

Unknown interpreted text role "const".

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] asyncio-runner.rst, line 140); [backlink](#)

Unknown interpreted text role "kbd".

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] asyncio-runner.rst, line 140); [backlink](#)

Unknown interpreted text role "meth".

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] asyncio-runner.rst, line 140); [backlink](#)

Unknown interpreted text role "exc".

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] asyncio-runner.rst, line 140); [backlink](#)

Unknown interpreted text role "meth".

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] asyncio-runner.rst, line 140); [backlink](#)

Unknown interpreted text role "exc".

4. A user could write a tight loop which cannot be interrupted by `:meth:`asyncio.Task.cancel``, in which case the second following `:kbd:`Ctrl-C`` immediately raises the `:exc:`KeyboardInterrupt`` without cancelling the main task.

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] asyncio-runner.rst, line 146); [backlink](#)

Unknown interpreted text role "meth".

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] asyncio-runner.rst, line 146); [backlink](#)

Unknown interpreted text role "kbd".

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] asyncio-runner.rst, line 146); [backlink](#)

Unknown interpreted text role "exc".