There seems to be a matrix of possible use cases.

Axes:

1. How the spiders are run: scrapy script; CrawlerProcess; CrawlerRunner; tests.
2. The reactor used: the asyncio reactor; a non-asyncio reactor.
3. The user code in `async def` functions: the code uses asyncio; it doesn't.
4. The `ASYNCIO_ENABLED` value: explicitly enabled; explicitly disabled; default.
5. The asyncio loop (in the main thread?): running; not running.

Axis 3

If the user code **doesn't use** asyncio features it should work for all values on other axes, this includes using a non-asyncio reactor (axis 2).

If the user code **uses** asyncio features, on the other hand, for axis 2 we require the asyncio reactor, for axis 4 we require `ASYNCIO_ENABLED=True` explicitly as the current default is False. All other values for these axes are not supported and ideally we want to show error messages but it isn't clear yet if it's possible in all cases.

Axis 1

If the **scrapy script** is used, it will install the reactor, so the only possible combinations for axes 2 and 4 are the asyncio reactor with explicit `ASYNCIO_ENABLED=True` and the default one otherwise.

In the **tests** that import the scrapy modules we control which reactor is installed via the pytest option. For the tests using the scrapy script the options are the same as above.

If the user uses **`CrawlerProcess`**, it should work just as the scrapy script. I think this is currently not implemented.

If the user uses **`CrawlerRunner`**, the user controls the reactor. The case with a non-asyncio reactor and `ASYNCIO_ENABLED=True` is possible but not supported, we should produce an error message in this case. Other cases seem obvious per what is already discussed above.

Axis 5

Makes sense only with `CrawlerProcess` and `CrawlerRunner`. The only non-trivial cases are the non-asyncio reactor (installed by Scrapy with `CrawlerProcess` or by the user with `CrawlerRunner`) but an existing asyncio loop. As the non-asyncio reactor requires `ASYNCIO_ENABLED=False`, these cases don't officially support asyncio features in the user code. What will happen in practice needs to be tested.

Simplifying

It looks like there are these basic groups of use cases:

1. No asyncio features in the user code. The simplest one, other options don't matter.
2. `ASYNCIO_ENABLED=False` (explicit or default) and asyncio features in the user code. This is not supported, the coroutines will never fire (IIRC).
3. `ASYNCIO_ENABLED=True`, asyncio features in the user code, the run mode implies no reactor is installed. Scrapy will install the asyncio one.
4. `ASYNCIO_ENABLED=True`, asyncio features in the user code, the run mode implies a reactor is installed. Scrapy will check if it's the asyncio one, and show an error otherwise. Should the error be fatal?