`!rlcompleter` --- Completion function for GNU readline

rlcompleter

Source code: `Lib/rlcompleter.py`

The `!rlcompleter` module defines a completion function suitable to be passed to `~readline.set_completer` in the `readline` module.

When this module is imported on a Unix platform with the `readline` module available, an instance of the `Completer` class is automatically created and its `~Completer.complete` method is set as the `readline completer <readline-completion>`. The method provides completion of valid Python `identifiers and keywords <identifiers>`.

Example:

```
>>> import rlcompleter
>>> import readline
>>> readline.parse_and_bind("tab: complete")
>>> readline. <TAB PRESSED>
readline.__doc__          readline.get_line_buffer(  readline.read_init_file(
readline.__file__         readline.insert_text(      readline.set_completer(
readline.__name__         readline.parse_and_bind(
>>> readline.
```

The `!rlcompleter` module is designed for use with Python's `interactive mode <tut-interactive>`. Unless Python is run with the `-S` option, the module is automatically imported and configured (see `rlcompleter-config`).

On platforms without `readline`, the `Completer` class defined by this module can still be used for custom purposes.

Completer objects have the following method:

Completer.complete(text, state)

Return the next possible completion for *text*.

When called by the `readline` module, this method is called successively with `state == 0, 1, 2, ...` until the method returns `None`.

If called for *text* that doesn't include a period character (`'.'`), it will complete from names currently defined in `__main__`, `builtins` and keywords (as defined by the `keyword` module).

If called for a dotted name, it will try to evaluate anything without obvious side-effects (functions will not be evaluated, but it can generate calls to `~object.__getattr__`) up to the last part, and find matches for the rest via the `dir` function. Any exception raised during the evaluation of the expression is caught, silenced and `None` is returned.

□□□ #2
□ □□□ □□□ 2026-10-31 17:18:23 UTC
□ □□□ □□□ 2026-10-31 17:18:24 UTC