IDLE

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)idle.rst, line 6)

Unknown directive type "moduleauthor".

```
.. moduleauthor:: Guido van Rossum <guido@python.org>
```

Source code: :source:`Lib/idlelib/`

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

Unknown interpreted text role "source".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)idle.rst, line 10)

Unknown directive type "index".

```
.. index::
   single: IDLE
   single: Python Editor
   single: Integrated Development Environment
```

IDLE is Python's Integrated Development and Learning Environment.

IDLE has the following features:

- coded in 100% pure Python, using the :mod:`tkinter` GUI toolkit

  System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)idle.rst, line 21); *backlink*

  Unknown interpreted text role "mod".

- cross-platform: works mostly the same on Windows, Unix, and macOS
- Python shell window (interactive interpreter) with colorizing of code input, output, and error messages
- multi-window text editor with multiple undo, Python colorizing, smart indent, call tips, auto completion, and other features
- search within any window, replace within editor windows, and search through multiple files (grep)
- debugger with persistent breakpoints, stepping, and viewing of global and local namespaces
- configuration, browsers, and other dialogs

Menus

IDLE has two main window types, the Shell window and the Editor window. It is possible to have multiple editor windows simultaneously. On Windows and Linux, each has its own top menu. Each menu documented below indicates which window type it is associated with.

Output windows, such as used for Edit => Find in Files, are a subtype of editor window. They currently have the same top menu but a different default title and context menu.

On macOS, there is one application menu. It dynamically changes according to the window currently selected. It has an IDLE menu, and some entries described below are moved around to conform to Apple guidelines.

File menu (Shell and Editor)

New File
: Create a new file editing window.

Open...
: Open an existing file with an Open dialog.

Recent Files
: Open a list of recent files. Click one to open it.

Open Module...
: Open an existing module (searches sys.path).

> **System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)idle.rst, line 70)**
>
> Unknown directive type "index".
>
> ```
> .. index::
>    single: Class browser
>    single: Path browser
> ```

Class Browser
: Show functions, classes, and methods in the current Editor file in a tree structure. In the shell, open a module first.

Path Browser
: Show sys.path directories, modules, functions, classes and methods in a tree structure.

Save
: Save the current window to the associated file, if there is one. Windows that have been changed since being opened or last saved have a * before and after the window title. If there is no associated file, do Save As instead.

Save As...
: Save the current window with a Save As dialog. The file saved becomes the new associated file for the window.

Save Copy As...
: Save the current window to different file without changing the associated file.

Print Window
: Print the current window to the default printer.

Close Window
: Close the current window (if an unsaved editor, ask to save; if an unsaved Shell, ask to quit execution). Calling `exit()` or `close()` in the Shell window also closes Shell. If this is the only window, also exit IDLE.

Exit IDLE
: Close all windows and quit IDLE (ask to save unsaved edit windows).

Edit menu (Shell and Editor)

Undo
: Undo the last change to the current window. A maximum of 1000 changes may be undone.

Redo
: Redo the last undone change to the current window.

Cut
: Copy selection into the system-wide clipboard; then delete the selection.

Copy
: Copy selection into the system-wide clipboard.

Paste
: Insert contents of the system-wide clipboard into the current window.

The clipboard functions are also available in context menus.

Select All
: Select the entire contents of the current window.

Find...
: Open a search dialog with many options

Find Again
: Repeat the last search, if there is one.

Find Selection
: Search for the currently selected string, if there is one.

Find in Files...
: Open a file search dialog. Put results in a new output window.

Replace...
: Open a search-and-replace dialog.

Go to Line
: Move the cursor to the beginning of the line requested and make that line visible. A request past the end of the file goes to

the end. Clear any selection and update the line and column status.

Show Completions

Open a scrollable list allowing selection of existing names. See :ref:`Completions <completions>` in the Editing and navigation section below.

> **System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)idle.rst, line 152); *backlink***
>
> Unknown interpreted text role "ref".

Expand Word

Expand a prefix you have typed to match a full word in the same window; repeat to get a different expansion.

Show call tip

After an unclosed parenthesis for a function, open a small window with function parameter hints. See :ref:`Calltips <calltips>` in the Editing and navigation section below.

> **System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)idle.rst, line 160); *backlink***
>
> Unknown interpreted text role "ref".

Show surrounding parens

Highlight the surrounding parenthesis.

Format menu (Editor window only)

Indent Region
: Shift selected lines right by the indent width (default 4 spaces).

Dedent Region
: Shift selected lines left by the indent width (default 4 spaces).

Comment Out Region
: Insert ## in front of selected lines.

Uncomment Region
: Remove leading # or ## from selected lines.

Tabify Region
: Turn *leading* stretches of spaces into tabs. (Note: We recommend using 4 space blocks to indent Python code.)

Untabify Region
: Turn *all* tabs into the correct number of spaces.

Toggle Tabs
: Open a dialog to switch between indenting with spaces and tabs.

New Indent Width
: Open a dialog to change indent width. The accepted default by the Python community is 4 spaces.

Format Paragraph
: Reformat the current blank-line-delimited paragraph in comment block or multiline string or selected line in a string. All lines in the paragraph will be formatted to less than N columns, where N defaults to 72.

Strip trailing whitespace
: Remove trailing space and other whitespace characters after the last non-whitespace character of a line by applying str.rstrip to each line, including lines within multiline strings. Except for Shell windows, remove extra newlines at the end of the file.

> **System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)idle.rst, line 209)**
>
> Unknown directive type "index".
>
> ```
> .. index::
>    single: Run script
> ```

Run menu (Editor window only)

Run Module

Do :ref:`Check Module <check-module>`. If no error, restart the shell to clean the environment, then execute the module.

Output is displayed in the Shell window. Note that output requires use of `print` or `write`. When execution is complete, the Shell retains focus and displays a prompt. At this point, one may interactively explore the result of execution. This is similar to executing a file with `python -i file` at a command line.

> **System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)idle.rst, line 218); *backlink***
>
> Unknown interpreted text role "ref".

Run... Customized

Same as :ref`Run Module <run-module>`, but run the module with customized settings. *Command Line Arguments* extend :data:`sys.argv` as if passed on a command line. The module can be run in the Shell without restarting.

> **System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)idle.rst, line 229); *backlink***
>
> Unknown interpreted text role "ref".

> **System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)idle.rst, line 229); *backlink***
>
> Unknown interpreted text role "data".

Check Module

Check the syntax of the module currently open in the Editor window. If the module has not been saved IDLE will either prompt the user to save or autosave, as selected in the General tab of the Idle Settings dialog. If there is a syntax error, the approximate location is indicated in the Editor window.

Python Shell

Open or wake up the Python Shell window.

Shell menu (Shell window only)

View Last Restart

Scroll the shell window to the last Shell restart.

Restart Shell

Restart the shell to clean the environment and reset display and exception handling.

Previous History

Cycle through earlier commands in history which match the current entry.

Next History

Cycle through later commands in history which match the current entry.

Interrupt Execution

Stop a running program.

Debug menu (Shell window only)

Go to File/Line

Look on the current line. with the cursor, and the line above for a filename and line number. If found, open the file if not already open, and show the line. Use this to view source lines referenced in an exception traceback and lines found by Find in Files. Also available in the context menu of the Shell window and Output windows.

> **System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)idle.rst, line 276)**
>
> Unknown directive type "index".
>
> ```
> .. index::
>    single: debugger
>    single: stack viewer
> ```

Debugger (toggle)

When activated, code entered in the Shell or run from an Editor will run under the debugger. In the Editor, breakpoints can be set with the context menu. This feature is still incomplete and somewhat experimental.

Stack Viewer

Show the stack traceback of the last exception in a tree widget, with access to locals and globals.

Auto-open Stack Viewer
: Toggle automatically opening the stack viewer on an unhandled exception.

Options menu (Shell and Editor)

Configure IDLE
: Open a configuration dialog and change preferences for the following: fonts, indentation, keybindings, text color themes, startup windows and size, additional help sources, and extensions. On macOS, open the configuration dialog by selecting Preferences in the application menu. For more details, see :ref:`Setting preferences <preferences>` under Help and preferences.

> **System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)idle.rst, line 296); *backlink***
>
> Unknown interpreted text role "ref".

Most configuration options apply to all windows or all future windows. The option items below only apply to the active window.

Show/Hide Code Context (Editor Window only)
: Open a pane at the top of the edit window which shows the block context of the code which has scrolled above the top of the window. See :ref:`Code Context <code-context>` in the Editing and Navigation section below.

> **System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)idle.rst, line 307); *backlink***
>
> Unknown interpreted text role "ref".

Show/Hide Line Numbers (Editor Window only)
: Open a column to the left of the edit window which shows the number of each line of text. The default is off, which may be changed in the preferences (see :ref:`Setting preferences <preferences>`).

> **System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)idle.rst, line 313); *backlink***
>
> Unknown interpreted text role "ref".

Zoom/Restore Height
: Toggles the window between normal size and maximum height. The initial size defaults to 40 lines by 80 chars unless changed on the General tab of the Configure IDLE dialog. The maximum height for a screen is determined by momentarily maximizing a window the first time one is zoomed on the screen. Changing screen settings may invalidate the saved height. This toggle has no effect when a window is maximized.

Window menu (Shell and Editor)

Lists the names of all open windows; select one to bring it to the foreground (deiconifying it if necessary).

Help menu (Shell and Editor)

About IDLE
: Display version, copyright, license, credits, and more.

IDLE Help
: Display this IDLE document, detailing the menu options, basic editing and navigation, and other tips.

Python Docs
: Access local Python documentation, if installed, or start a web browser and open docs.python.org showing the latest Python documentation.

Turtle Demo
: Run the turtledemo module with example Python code and turtle drawings.

Additional help sources may be added here with the Configure IDLE dialog under the General tab. See the :ref:`Help sources <help-sources>` subsection below for more on Help menu choices.

> **System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-**

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)idle.rst, line 348); *backlink*

Unknown interpreted text role "ref".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)idle.rst, line 352)

Unknown directive type "index".

```
.. index::
   single: Cut
   single: Copy
   single: Paste
   single: Set Breakpoint
   single: Clear Breakpoint
   single: breakpoints
```

Context Menus

Open a context menu by right-clicking in a window (Control-click on macOS). Context menus have the standard clipboard functions also on the Edit menu.

Cut
: Copy selection into the system-wide clipboard; then delete the selection.

Copy
: Copy selection into the system-wide clipboard.

Paste
: Insert contents of the system-wide clipboard into the current window.

Editor windows also have breakpoint functions. Lines with a breakpoint set are specially marked. Breakpoints only have an effect when running under the debugger. Breakpoints for a file are saved in the user's `.idlerc` directory.

Set Breakpoint
: Set a breakpoint on the current line.

Clear Breakpoint
: Clear the breakpoint on that line.

Shell and Output windows also have the following.

Go to file/line
: Same as in Debug menu.

The Shell window also has an output squeezing facility explained in the *Python Shell window* subsection below.

Squeeze
: If the cursor is over an output line, squeeze all the output between the code above and the prompt below down to a 'Squeezed text' label.

Editing and navigation

Editor windows

IDLE may open editor windows when it starts, depending on settings and how you start IDLE. Thereafter, use the File menu. There can be only one open editor window for a given file.

The title bar contains the name of the file, the full path, and the version of Python and IDLE running the window. The status bar contains the line number ('Ln') and column number ('Col'). Line numbers start with 1; column numbers with 0.

IDLE assumes that files with a known .py* extension contain Python code and that other files do not. Run Python code with the Run menu.

Key bindings

In this section, 'C' refers to the :kbd:`Control` key on Windows and Unix and the :kbd:`Command` key on macOS.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)idle.rst, line 422); *backlink*

Unknown interpreted text role "kbd".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)idle.rst, line 422); *backlink*

Unknown interpreted text role "kbd".

- :kbd:`Backspace` deletes to the left; :kbd:`Del` deletes to the right

  System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)idle.rst, line 425); *backlink*

  Unknown interpreted text role "kbd".

  System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)idle.rst, line 425); *backlink*

  Unknown interpreted text role "kbd".

- :kbd:`C-Backspace` delete word left; :kbd:`C-Del` delete word to the right

  System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)idle.rst, line 427); *backlink*

  Unknown interpreted text role "kbd".

  System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)idle.rst, line 427); *backlink*

  Unknown interpreted text role "kbd".

- Arrow keys and :kbd:`Page Up`/:kbd:`Page Down` to move around

  System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)idle.rst, line 429); *backlink*

  Unknown interpreted text role "kbd".

  System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)idle.rst, line 429); *backlink*

  Unknown interpreted text role "kbd".

- :kbd:`C-LeftArrow` and :kbd:`C-RightArrow` moves by words

  System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)idle.rst, line 431); *backlink*

  Unknown interpreted text role "kbd".

  System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)idle.rst, line 431); *backlink*

  Unknown interpreted text role "kbd".

- :kbd:`Home`/:kbd:`End` go to begin/end of line

  System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)idle.rst, line 433); *backlink*

  Unknown interpreted text role "kbd".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)idle.rst, line 433); *backlink*

Unknown interpreted text role "kbd".

- :kbd:`C-Home`/:kbd:`C-End` go to begin/end of file

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)idle.rst, line 435); *backlink*

Unknown interpreted text role "kbd".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)idle.rst, line 435); *backlink*

Unknown interpreted text role "kbd".

- Some useful Emacs bindings are inherited from Tcl/Tk:
  - :kbd:`C-a` beginning of line

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)idle.rst, line 439); *backlink*

Unknown interpreted text role "kbd".

  - :kbd:`C-e` end of line

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)idle.rst, line 441); *backlink*

Unknown interpreted text role "kbd".

  - :kbd:`C-k` kill line (but doesn't put it in clipboard)

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)idle.rst, line 443); *backlink*

Unknown interpreted text role "kbd".

  - :kbd:`C-l` center window around the insertion point

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)idle.rst, line 445); *backlink*

Unknown interpreted text role "kbd".

  - :kbd:`C-b` go backward one character without deleting (usually you can also use the cursor key for this)

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)idle.rst, line 447); *backlink*

Unknown interpreted text role "kbd".

  - :kbd:`C-f` go forward one character without deleting (usually you can also use the cursor key for this)

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)idle.rst, line 450); *backlink*

Unknown interpreted text role "kbd".

- :kbd:`C-p` go up one line (usually you can also use the cursor key for this)

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)idle.rst, line 453); *backlink*

Unknown interpreted text role "kbd".

- :kbd:`C-d` delete next character

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)idle.rst, line 456); *backlink*

Unknown interpreted text role "kbd".

Standard keybindings (like :kbd:`C-c` to copy and :kbd:`C-v` to paste) may work. Keybindings are selected in the Configure IDLE dialog.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)idle.rst, line 458); *backlink*

Unknown interpreted text role "kbd".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)idle.rst, line 458); *backlink*

Unknown interpreted text role "kbd".

Automatic indentation

After a block-opening statement, the next line is indented by 4 spaces (in the Python Shell window by one tab). After certain keywords (break, return etc.) the next line is dedented. In leading indentation, :kbd:`Backspace` deletes up to 4 spaces if they are there. :kbd:`Tab` inserts spaces (in the Python Shell window one tab), number depends on Indent width. Currently, tabs are restricted to four spaces due to Tcl/Tk limitations.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)idle.rst, line 464); *backlink*

Unknown interpreted text role "kbd".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)idle.rst, line 464); *backlink*

Unknown interpreted text role "kbd".

See also the indent/dedent region commands on the :ref:`Format menu <format-menu>`.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)idle.rst, line 471); *backlink*

Unknown interpreted text role "ref".

Completions

Completions are supplied, when requested and available, for module names, attributes of classes or functions, or filenames. Each request method displays a completion box with existing names. (See tab completions below for an exception.) For any box, change the name being completed and the item highlighted in the box by typing and deleting characters; by hitting :kbd:`Up`, :kbd:`Down`, :kbd:`PageUp`, :kbd:`PageDown`, :kbd:`Home`, and :kbd:`End` keys; and by a single click within the box. Close the box with :kbd:`Escape`, :kbd:`Enter`, and double :kbd:`Tab` keys or clicks outside the box. A double click within the box selects and closes.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)idle.rst, line 479); *backlink*

Unknown interpreted text role "kbd".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)idle.rst, line 479); *backlink*

Unknown interpreted text role "kbd".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)idle.rst, line 479); *backlink*

Unknown interpreted text role "kbd".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)idle.rst, line 479); *backlink*

Unknown interpreted text role "kbd".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)idle.rst, line 479); *backlink*

Unknown interpreted text role "kbd".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)idle.rst, line 479); *backlink*

Unknown interpreted text role "kbd".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)idle.rst, line 479); *backlink*

Unknown interpreted text role "kbd".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)idle.rst, line 479); *backlink*

Unknown interpreted text role "kbd".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)idle.rst, line 479); *backlink*

Unknown interpreted text role "kbd".

One way to open a box is to type a key character and wait for a predefined interval. This defaults to 2 seconds; customize it in the settings dialog. (To prevent auto popups, set the delay to a large number of milliseconds, such as 100000000.) For imported module names or class or function attributes, type '.'. For filenames in the root directory, type :data:`os.sep` or :data:`os.altsep` immediately after an opening quote. (On Windows, one can specify a drive first.) Move into subdirectories by typing a directory name and a separator.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)idle.rst, line 490); *backlink*

Unknown interpreted text role "data".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)idle.rst, line 490); *backlink*

Unknown interpreted text role "data".

Instead of waiting, or after a box is closed, open a completion box immediately with Show Completions on the Edit menu. The default hot key is :kbd:`C-space`. If one types a prefix for the desired name before opening the box, the first match or near miss is made visible. The result is the same as if one enters a prefix after the box is displayed. Show Completions after a quote completes

filenames in the current directory instead of a root directory.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)idle.rst, line 500); *backlink*

Unknown interpreted text role "kbd".

Hitting :kbd:`Tab` after a prefix usually has the same effect as Show Completions. (With no prefix, it indents.) However, if there is only one match to the prefix, that match is immediately added to the editor text without opening a box.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)idle.rst, line 508); *backlink*

Unknown interpreted text role "kbd".

Invoking 'Show Completions', or hitting :kbd:`Tab` after a prefix, outside of a string and without a preceding '.' opens a box with keywords, builtin names, and available module-level names.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)idle.rst, line 513); *backlink*

Unknown interpreted text role "kbd".

When editing code in an editor (as oppose to Shell), increase the available module-level names by running your code and not restarting the Shell thereafter. This is especially useful after adding imports at the top of a file. This also increases possible attribute completions.

Completion boxes initially exclude names beginning with '_' or, for modules, not included in '__all__'. The hidden names can be accessed by typing '_' after '.', either before or after the box is opened.

Calltips

A calltip is shown automatically when one types :kbd:`(` after the name of an *accessible* function. A function name expression may include dots and subscripts. A calltip remains until it is clicked, the cursor is moved out of the argument area, or :kbd:`)` is typed. Whenever the cursor is in the argument part of a definition, select Edit and "Show Call Tip" on the menu or enter its shortcut to display a calltip.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)idle.rst, line 532); *backlink*

Unknown interpreted text role "kbd".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)idle.rst, line 532); *backlink*

Unknown interpreted text role "kbd".

The calltip consists of the function's signature and docstring up to the latter's first blank line or the fifth non-blank line. (Some builtin functions lack an accessible signature.) A '/' or '*' in the signature indicates that the preceding or following arguments are passed by position or name (keyword) only. Details are subject to change.

In Shell, the accessible functions depends on what modules have been imported into the user process, including those imported by Idle itself, and which definitions have been run, all since the last restart.

For example, restart the Shell and enter `itertools.count(`. A calltip appears because Idle imports itertools into the user process for its own use. (This could change.) Enter `turtle.write(` and nothing appears. Idle does not itself import turtle. The menu entry and shortcut also do nothing. Enter `import turtle`. Thereafter, `turtle.write(` will display a calltip.

In an editor, import statements have no effect until one runs the file. One might want to run a file after writing import statements, after adding function definitions, or after opening an existing file.

Code Context

Within an editor window containing Python code, code context can be toggled in order to show or hide a pane at the top of the window. When shown, this pane freezes the opening lines for block code, such as those beginning with `class`, `def`, or `if` keywords, that would have otherwise scrolled out of view. The size of the pane will be expanded and contracted as needed to show the all current levels of context, up to the maximum number of lines defined in the Configure IDLE dialog (which defaults to 15). If there are no current context lines and the feature is toggled on, a single blank line will display. Clicking on a line in the context pane will move that line to the top of the editor.

The text and background colors for the context pane can be configured under the Highlights tab in the Configure IDLE dialog.

Python Shell window

With IDLE's Shell, one enters, edits, and recalls complete statements. Most consoles and terminals only work with a single physical line at a time.

When one pastes code into Shell, it is not compiled and possibly executed until one hits :kbd:`Return`. One may edit pasted code first. If one pastes more that one statement into Shell, the result will be a :exc:`SyntaxError` when multiple statements are compiled as if they were one.

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)idle.rst`, **line 585**); *backlink*
>
> Unknown interpreted text role "kbd".

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)idle.rst`, **line 585**); *backlink*
>
> Unknown interpreted text role "exc".

The editing features described in previous subsections work when entering code interactively. IDLE's Shell window also responds to the following keys.

- :kbd:`C-c` interrupts executing command

  > **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)idle.rst`, **line 593**); *backlink*
  >
  > Unknown interpreted text role "kbd".

- :kbd:`C-d` sends end-of-file; closes window if typed at a >>> prompt

  > **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)idle.rst`, **line 595**); *backlink*
  >
  > Unknown interpreted text role "kbd".

- :kbd:`Alt-/` (Expand word) is also useful to reduce typing

  > **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)idle.rst`, **line 597**); *backlink*
  >
  > Unknown interpreted text role "kbd".

  Command history

  - :kbd:`Alt-p` retrieves previous command matching what you have typed. On macOS use :kbd:`C-p`.

    > **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)idle.rst`, **line 601**); *backlink*
    >
    > Unknown interpreted text role "kbd".

    > **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)idle.rst`, **line 601**); *backlink*
    >
    > Unknown interpreted text role "kbd".

  - :kbd:`Alt-n` retrieves next. On macOS use :kbd:`C-n`.

    > **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-`

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)idle.rst, line 604); *backlink*

Unknown interpreted text role "kbd".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)idle.rst, line 604); *backlink*

Unknown interpreted text role "kbd".

- :kbd:`Return` while on any previous command retrieves that command

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)idle.rst, line 606); *backlink*

Unknown interpreted text role "kbd".

Text colors

Idle defaults to black on white text, but colors text with special meanings. For the shell, these are shell output, shell error, user output, and user error. For Python code, at the shell prompt or in an editor, these are keywords, builtin class and function names, names following `class` and `def`, strings, and comments. For any text window, these are the cursor (when present), found text (when possible), and selected text.

IDLE also highlights the :ref:`soft keywords <soft-keywords>` :keyword:`match`, :keyword:`case <match>`, and :keyword:`_ <wildcard-patterns>` in pattern-matching statements. However, this highlighting is not perfect and will be incorrect in some rare cases, including some `_`-s in `case` patterns.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)idle.rst, line 618); *backlink*

Unknown interpreted text role "ref".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)idle.rst, line 618); *backlink*

Unknown interpreted text role "keyword".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)idle.rst, line 618); *backlink*

Unknown interpreted text role "keyword".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)idle.rst, line 618); *backlink*

Unknown interpreted text role "keyword".

Text coloring is done in the background, so uncolorized text is occasionally visible. To change the color scheme, use the Configure IDLE dialog Highlighting tab. The marking of debugger breakpoint lines in the editor and text in popups and dialogs is not user-configurable.

Startup and code execution

Upon startup with the `-s` option, IDLE will execute the file referenced by the environment variables :envvar:`IDLESTARTUP` or :envvar:`PYTHONSTARTUP`. IDLE first checks for `IDLESTARTUP`; if `IDLESTARTUP` is present the file referenced is run. If `IDLESTARTUP` is not present, IDLE checks for `PYTHONSTARTUP`. Files referenced by these environment variables are convenient places to store functions that are used frequently from the IDLE shell, or for executing import statements to import common modules.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)idle.rst, line 633); *backlink*

Unknown interpreted text role "envvar".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)idle.rst, line 633); *backlink*

Unknown interpreted text role "envvar".

In addition, Tk also loads a startup file if it is present. Note that the Tk file is loaded unconditionally. This additional file is .Idle.py and is looked for in the user's home directory. Statements in this file will be executed in the Tk namespace, so this file is not useful for importing functions to be used from IDLE's Python shell.

Command line usage

System Message: WARNING/2 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)idle.rst, line 650)

Cannot analyze code. No Pygments lexer found for "none".

```
.. code-block:: none

   idle.py [-c command] [-d] [-e] [-h] [-i] [-r file] [-s] [-t title] [-] [arg] ...

   -c command  run command in the shell window
   -d          enable debugger and open shell window
   -e          open editor window
   -h          print help message with legal combinations and exit
   -i          open shell window
   -r file     run file in shell window
   -s          run $IDLESTARTUP or $PYTHONSTARTUP first, in shell window
   -t title    set title of shell window
   -           run stdin in shell (- must be last option before args)
```

If there are arguments:

- If -, -c, or r is used, all arguments are placed in sys.argv[1:...] and sys.argv[0] is set to '', '-c', or '-r'. No editor window is opened, even if that is the default set in the Options dialog.
- Otherwise, arguments are files opened for editing and sys.argv reflects the arguments passed to IDLE itself.

Startup failure

IDLE uses a socket to communicate between the IDLE GUI process and the user code execution process. A connection must be established whenever the Shell starts or restarts. (The latter is indicated by a divider line that says 'RESTART'). If the user process fails to connect to the GUI process, it usually displays a Tk error box with a 'cannot connect' message that directs the user here. It then exits.

One specific connection failure on Unix systems results from misconfigured masquerading rules somewhere in a system's network setup. When IDLE is started from a terminal, one will see a message starting with ** Invalid host:. The valid value is 127.0.0.1 (idlelib.rpc.LOCALHOST). One can diagnose with tcpconnect -irv 127.0.0.1 6543 in one terminal window and tcplisten <same args> in another.

A common cause of failure is a user-written file with the same name as a standard library module, such as *random.py* and *tkinter.py*. When such a file is located in the same directory as a file that is about to be run, IDLE cannot import the stdlib file. The current fix is to rename the user file.

Though less common than in the past, an antivirus or firewall program may stop the connection. If the program cannot be taught to allow the connection, then it must be turned off for IDLE to work. It is safe to allow this internal connection because no data is visible on external ports. A similar problem is a network mis-configuration that blocks connections.

Python installation issues occasionally stop IDLE: multiple versions can clash, or a single installation might need admin access. If one undo the clash, or cannot or does not want to run as admin, it might be easiest to completely remove Python and start over.

A zombie pythonw.exe process could be a problem. On Windows, use Task Manager to check for one and stop it if there is. Sometimes a restart initiated by a program crash or Keyboard Interrupt (control-C) may fail to connect. Dismissing the error box or using Restart Shell on the Shell menu may fix a temporary problem.

When IDLE first starts, it attempts to read user configuration files in ~/.idlerc/ (~ is one's home directory). If there is a problem, an error message should be displayed. Leaving aside random disk glitches, this can be prevented by never editing the files by hand. Instead, use the configuration dialog, under Options. Once there is an error in a user configuration file, the best solution may be to delete it and start over with the settings dialog.

If IDLE quits with no message, and it was not started from a console, try starting it from a console or terminal (python -m idlelib) and see if this results in an error message.

On Unix-based systems with tcl/tk older than 8.6.11 (see About IDLE) certain characters of certain fonts can cause a tk failure with a message to the terminal. This can happen either if one starts IDLE to edit a file with such a character or later when entering

such a character. If one cannot upgrade tcl/tk, then re-configure IDLE to use a font that works better.

Running user code

With rare exceptions, the result of executing Python code with IDLE is intended to be the same as executing the same code by the default method, directly with Python in a text-mode system console or terminal window. However, the different interface and operation occasionally affect visible results. For instance, `sys.modules` starts with more entries, and `threading.active_count()` returns 2 instead of 1.

By default, IDLE runs user code in a separate OS process rather than in the user interface process that runs the shell and editor. In the execution process, it replaces `sys.stdin`, `sys.stdout`, and `sys.stderr` with objects that get input from and send output to the Shell window. The original values stored in `sys.__stdin__`, `sys.__stdout__`, and `sys.__stderr__` are not touched, but may be `None`.

Sending print output from one process to a text widget in another is slower than printing to a system terminal in the same process. This has the most effect when printing multiple arguments, as the string for each argument, each separator, the newline are sent separately. For development, this is usually not a problem, but if one wants to print faster in IDLE, format and join together everything one wants displayed together and then print a single string. Both format strings and :meth:`str.join` can help combine fields and lines.

> **System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)idle.rst, line 752); *backlink***
>
> Unknown interpreted text role "meth".

IDLE's standard stream replacements are not inherited by subprocesses created in the execution process, whether directly by user code or by modules such as multiprocessing. If such subprocess use `input` from sys.stdin or `print` or `write` to sys.stdout or sys.stderr, IDLE should be started in a command line window. The secondary subprocess will then be attached to that window for input and output.

If `sys` is reset by user code, such as with `importlib.reload(sys)`, IDLE's changes are lost and input from the keyboard and output to the screen will not work correctly.

When Shell has the focus, it controls the keyboard and screen. This is normally transparent, but functions that directly access the keyboard and screen will not work. These include system-specific functions that determine whether a key has been pressed and if so, which.

The IDLE code running in the execution process adds frames to the call stack that would not be there otherwise. IDLE wraps `sys.getrecursionlimit` and `sys.setrecursionlimit` to reduce the effect of the additional stack frames.

When user code raises SystemExit either directly or by calling sys.exit, IDLE returns to a Shell prompt instead of exiting.

User output in Shell

When a program outputs text, the result is determined by the corresponding output device. When IDLE executes user code, `sys.stdout` and `sys.stderr` are connected to the display area of IDLE's Shell. Some of its features are inherited from the underlying Tk Text widget. Others are programmed additions. Where it matters, Shell is designed for development rather than production runs.

For instance, Shell never throws away output. A program that sends unlimited output to Shell will eventually fill memory, resulting in a memory error. In contrast, some system text windows only keep the last n lines of output. A Windows console, for instance, keeps a user-settable 1 to 9999 lines, with 300 the default.

A Tk Text widget, and hence IDLE's Shell, displays characters (codepoints) in the BMP (Basic Multilingual Plane) subset of Unicode. Which characters are displayed with a proper glyph and which with a replacement box depends on the operating system and installed fonts. Tab characters cause the following text to begin after the next tab stop. (They occur every 8 'characters'). Newline characters cause following text to appear on a new line. Other control characters are ignored or displayed as a space, box, or something else, depending on the operating system and font. (Moving the text cursor through such output with arrow keys may exhibit some surprising spacing behavior.)

```
>>> s = 'a\tb\a<\x02><\r>\bc\nd'  # Enter 22 chars.
>>> len(s)
14
>>> s  # Display repr(s)
'a\tb\x07<\x02><\r>\x08c\nd'
>>> print(s, end='')  # Display s as is.
# Result varies by OS and font.  Try it.
```

The `repr` function is used for interactive echo of expression values. It returns an altered version of the input string in which control codes, some BMP codepoints, and all non-BMP codepoints are replaced with escape codes. As demonstrated above, it allows one to identify the characters in a string, regardless of how they are displayed.

Normal and error output are generally kept separate (on separate lines) from code input and each other. They each get different

highlight colors.

For SyntaxError tracebacks, the normal '^' marking where the error was detected is replaced by coloring the text with an error highlight. When code run from a file causes other exceptions, one may right click on a traceback line to jump to the corresponding line in an IDLE editor. The file will be opened if necessary.

Shell has a special facility for squeezing output lines down to a 'Squeezed text' label. This is done automatically for output over N lines (N = 50 by default). N can be changed in the PyShell section of the General page of the Settings dialog. Output with fewer lines can be squeezed by right clicking on the output. This can be useful lines long enough to slow down scrolling.

Squeezed output is expanded in place by double-clicking the label. It can also be sent to the clipboard or a separate view window by right-clicking the label.

Developing tkinter applications

IDLE is intentionally different from standard Python in order to facilitate development of tkinter programs. Enter `import tkinter as tk; root = tk.Tk()` in standard Python and nothing appears. Enter the same in IDLE and a tk window appears. In standard Python, one must also enter `root.update()` to see the window. IDLE does the equivalent in the background, about 20 times a second, which is about every 50 milliseconds. Next enter `b = tk.Button(root, text='button'); b.pack()`. Again, nothing visibly changes in standard Python until one enters `root.update()`.

Most tkinter programs run `root.mainloop()`, which usually does not return until the tk app is destroyed. If the program is run with `python -i` or from an IDLE editor, a `>>>` shell prompt does not appear until `mainloop()` returns, at which time there is nothing left to interact with.

When running a tkinter program from an IDLE editor, one can comment out the mainloop call. One then gets a shell prompt immediately and can interact with the live application. One just has to remember to re-enable the mainloop call when running in standard Python.

Running without a subprocess

By default, IDLE executes user code in a separate subprocess via a socket, which uses the internal loopback interface. This connection is not externally visible and no data is sent to or received from the internet. If firewall software complains anyway, you can ignore it.

If the attempt to make the socket connection fails, Idle will notify you. Such failures are sometimes transient, but if persistent, the problem may be either a firewall blocking the connection or misconfiguration of a particular system. Until the problem is fixed, one can run Idle with the -n command line switch.

If IDLE is started with the -n command line switch it will run in a single process and will not create the subprocess which runs the RPC Python execution server. This can be useful if Python cannot create the subprocess or the RPC socket interface on your platform. However, in this mode user code is not isolated from IDLE itself. Also, the environment is not restarted when Run/Run Module (F5) is selected. If your code has been modified, you must reload() the affected modules and re-import any specific items (e.g. from foo import baz) if the changes are to take effect. For these reasons, it is preferable to run IDLE with the default subprocess if at all possible.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)idle.rst, line 894)

Unknown directive type "deprecated".

```
.. deprecated:: 3.4
```

Help and preferences

Help sources

Help menu entry "IDLE Help" displays a formatted html version of the IDLE chapter of the Library Reference. The result, in a read-only tkinter text window, is close to what one sees in a web browser. Navigate through the text with a mousewheel, the scrollbar, or up and down arrow keys held down. Or click the TOC (Table of Contents) button and select a section header in the opened box.

Help menu entry "Python Docs" opens the extensive sources of help, including tutorials, available at `docs.python.org/x.y`, where 'x.y' is the currently running Python version. If your system has an off-line copy of the docs (this may be an installation option), that will be opened instead.

Selected URLs can be added or removed from the help menu at any time using the General tab of the Configure IDLE dialog.

Setting preferences

The font preferences, highlighting, keys, and general preferences can be changed via Configure IDLE on the Option menu. Non-default user settings are saved in a `.idlerc` directory in the user's home directory. Problems caused by bad user configuration files

are solved by editing or deleting one or more of the files in `.idlerc`.

On the Font tab, see the text sample for the effect of font face and size on multiple characters in multiple languages. Edit the sample to add other characters of personal interest. Use the sample to select monospaced fonts. If particular characters have problems in Shell or an editor, add them to the top of the sample and try changing first size and then font.

On the Highlights and Keys tab, select a built-in or custom color theme and key set. To use a newer built-in color theme or key set with older IDLEs, save it as a new custom theme or key set and it well be accessible to older IDLEs.

IDLE on macOS

Under System Preferences: Dock, one can set "Prefer tabs when opening documents" to "Always". This setting is not compatible with the tk/tkinter GUI framework used by IDLE, and it breaks a few IDLE features.

Extensions

IDLE contains an extension facility. Preferences for extensions can be changed with the Extensions tab of the preferences dialog. See the beginning of config-extensions.def in the idlelib directory for further information. The only current default extension is zzdummy, an example also used for testing.