How-Tos

Version 6.0.11-1

This collection of How-Tos explains how to get started using Wing with specific Python frameworks, tools, and libraries for web and GUI development, 2D and 3D modeling, compositing, and rendering, game development, scientific analysis, and much more.

These How-Tos assume that you are familiar with the Python framework or tool being discussed and that you already know how to use Wing. To learn more about Wing see the Quick Start Guide or Tutorial.

Wingware, the feather logo, Wing Python IDE, Wing Pro, Wing Personal, Wing 101, Wing IDE, Wing IDE 101, Wing IDE Personal, Wing IDE Professional, Wing IDE Pro, Wing Debugger, and "The Intelligent Development Environment for Python Programmers" are trademarks or registered trademarks of Wingware in the United States and other countries.

Disclaimers: The information contained in this document is subject to change without notice. Wingware shall not be liable for technical or editorial errors or omissions contained in this document; nor for incidental or consequential damages resulting from furnishing, performance, or use of this material.

Hardware and software products mentioned herein are named for identification purposes only and may be trademarks of their respective owners.

```
Wingware
P.O. Box 400527
Cambridge, MA  02140-0006
United States of America
```

Contents

Wing Python IDE Quick Start Guide

This is a minimalist guide for getting started quickly with Wing's Python IDE features. For a more in-depth introduction, try the Tutorial.

Install Python

If you don't already have it on your system, install Python. You may need to restart Wing after doing so.

Configure Python

Wing finds the latest installed Python and uses that with your code. If you want to change which Python is used, use `Configure Python` in the `Edit` menu.

Configuring the UI

You are now ready to start working with code, but may want to make a few configuration changes first:

Key Bindings - Wing can emulate VI/Vim, Visual Studio, Emacs, Eclipse, and Brief editors, selected with the `User Interface > Keyboard > Personality` preference.

Tab Key - The default tab key action depends on file type, context, and whether or not there is a selection. This can be changed from the `User Interface > Keyboard > Tab Key Action` preference.

There are many other options in `Preferences`.

Navigating Code

Wing provides many ways to get around your code quickly:

Goto-definition is available from the toolbar, `Source` menu, and by right-clicking on symbols in the editor or shells. Use the browser-like forward/back history buttons at the top left of the editor to return from visiting a point of definition.

Source Index menus at the top of the editor provide quick access to other parts of a source file.

Search in the `Tools` menu provides incremental text, wildcard, and regular expression search and replace in selections and the current file.

Toolbar search is another quick way to search the current file.

Editing Code

Wing's editor focuses on fast error-free Python coding:

Auto-completion in Wing's editor speeds up typing and reduces coding errors. The auto-completer uses Tab by default for completion, but this can be changed in the `Editor > Auto-Completion > Completion Keys` preference. This feature is disabled by default in Wing 101.

Multiple Selections can be made with the `Editor > Multiple Selections` menu or the multiple selections toolbar item, and by pressing `Ctrl` (or `Command` on the Mac) while making a selection with the mouse. Once multiple selections have been made, edits made will be applied to all the selections at once.

Code Selection from the `Edit > Select` menu makes selecting whole statements, blocks, or scopes a snap, before copying, editing, or searching through them.

Debugging Code

Wing's debugger is a powerful tool for finding and fixing bugs, understanding unfamiliar code, and writing new code interactively. You can launch code from the `Debug` menu or toolbar, or from the `Python Shell` (click on the bug icon in the top right of the shell to enable debugging there).

Breakpoints can be set by clicking on the breakpoint margin of the editor and debugging is started from the toolbar or `Debug` menu. The `Stack Data` tool is used to inspect or change program data. Debug process I/O is shown in the `Debug I/O` tool, or optionally in an external console.

Other Features

Wing contains many other features, including:

Python Shell -- Wing's `Python Shell` lets you try out code in a sandbox process kept isolated from Wing and your debug process. Code run here can optionally be debugged. To enable this, click the bug icon in the top right of the `Python Shell`. The shell provides auto-completion if it has been enabled in preferences.

User Interface Customization in `Preferences` gives you control of the overall layout and color of the IDE, among many other options. Right click on the tabs for layout options, or drag tool and editor tabs to move them or create new splits. Right click on the toolbar to configure which tools are visible or to add your own. Wing also supports defining sharable color palettes and syntax colors.

Related Documents

For more information see:

- Wing Tutorial, a detailed guided tour for Wing.
- Wing Reference Manual, which describes Wing in detail.