



## How-Tos

### Version 6.0.2-1

This is a collection of How-Tos designed to make it easier to get started using Wing with specific 3rd party tools and libraries for Python.

You can use these How-Tos to get set up quickly developing desktop UIs, web applications, 2D and 3D movies and games, and to learn how to use Wing with other libraries.

These How-Tos assume you are already familiar with the 3rd party library or application and with Wing IDE. To learn more about Wing IDE see the [Quick Start Guide](#) or [Tutorial](#).

---

*Wingware, the feather logo, Wing IDE, Wing IDE 101, Wing IDE Personal, Wing IDE Professional, and "The Intelligent Development Environment" 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.*

---

Copyright (c) 1999-2017 by Wingware. All rights reserved.

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

# Contents

<b>How-Tos</b>	<b>1</b>
Wing IDE Quick Start Guide	1
Install Python	1
Configure Python	1
Configuring the UI	1
Navigating Code	1
Debugging Code	2
Other Features	2
Related Documents	2

## Wing IDE Quick Start Guide

This is a minimalist guide for getting started quickly with Wing IDE. 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.

**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 IDE 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**.

**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 IDE Tutorial](#), a detailed guided tour for Wing IDE.
- [Wing IDE Reference Manual](#), which describes Wing IDE in detail.