

使用 Tk 创建图形用户界面

Tk/Tcl 早已成为 Python 的一部分。它提供了一套健壮且独立于平台的窗口工具集，Python 程序员可通过 [tkinter](#) 包及其扩展 [tkinter.ttk](#) 模块来使用它。

[tkinter](#) 包是使用面向对象方式对 Tcl/Tk 进行的一层薄包装。使用 [tkinter](#)，你不需要写 Tcl 代码，但你将需要参阅 Tk 文档，有时还需要参阅 Tcl 文档。[tkinter](#) 是一组包装器，它将 Tk 的可视化部件实现为相应的 Python 类。

[tkinter](#) 的主要特点是速度很快，并且通常直接附带在 Python 中。虽然它的官方文档做得不好，但还是有许多可用的资源，包括：在线参考、教程、入门书等等。[tkinter](#) 还有众所周知的较过时的外观界面，这在 Tk 8.5 中已得到很大改进。无论如何，你还可以考虑许多其他的 GUI 库。Python wiki 例出了一些替代性的 [GUI 框架和工具](#)。

- [tkinter --- Tcl/Tk 的 Python 接口](#)
 - 架构
 - Tkinter 模块
 - Tkinter 拾遗
 - [Hello World 程序](#)
 - [重要的 Tk 概念](#)
 - [了解 Tkinter 如何封装 Tcl/Tk](#)
 - [我该如何...? 这个选项会做...?](#)
 - [浏览 Tcl/Tk 参考手册](#)
 - 线程模型
 - 快速参考
 - [可选配置项](#)
 - [包装器](#)
 - [包装器的参数](#)
 - [部件与变量的关联](#)
 - [窗口管理器](#)
 - [Tk 参数的数据类型](#)
 - [绑定和事件](#)
 - [index 参数](#)
 - [图片](#)
 - 文件处理程序
- [tkinter.colorchooser --- 颜色选择对话框](#)
- [tkinter.font --- Tkinter 字体包装器](#)
- Tkinter 对话框
 - [tkinter.simpledialog --- 标准 Tkinter 输入对话框](#)
 - [tkinter.filedialog --- 文件选择对话框.](#)
 - [原生的载入/保存对话框.](#)

- `tkinter.commondialog` --- 对话窗口模板
- `tkinter.messagebox` --- Tkinter 消息提示
- `tkinter.scrolledtext` --- 流动文本控件
- `tkinter.dnd` --- 拖放操作支持
- `tkinter.ttk` --- Tk 带主题的控件
 - `ttk` 的用法
 - `ttk` 控件
 - 控件
 - 标准属性
 - 可滚动控件的属性
 - 标签控件的属性
 - 兼容性属性
 - 控件状态
 - `ttk.Widget`
 - `Combobox`
 - 属性
 - 虚拟事件
 - `ttk.Combobox`
 - `Spinbox`
 - 属性
 - 虚拟事件
 - `ttk.Spinbox`
 - `Notebook`
 - 属性
 - Tab 属性
 - Tab ID
 - 虚拟事件
 - `ttk.Notebook`
 - `Progressbar`
 - 属性
 - `ttk.Progressbar`
 - `Separator`
 - 属性
 - `Sizegrip`
 - 与平台相关的注意事项
 - Bug
 - `Treeview`
 - 属性
 - 数据项的属性
 - `tag` 属性
 - 列标识
 - 虚拟事件
 - `ttk.Treeview`

- Tk 样式
 - 布局
- IDLE --- Python 编辑器和 shell
 - 目录
 - 文件菜单 (命令行和编辑器)
 - 编辑菜单 (命令行和编辑器)
 - 格式菜单 (仅 window 编辑器)
 - 运行菜单 (仅 window 编辑器)
 - Shell 菜单 (仅限 Shell 窗口)
 - 调试菜单 (仅限 Shell 窗口)
 - 选项菜单 (命令行和编辑器)
 - Window 菜单 (命令行和编辑器)
 - 帮助菜单 (命令行和编辑器)
 - 上下文菜单
 - 编辑和导航
 - 编辑窗口
 - 按键绑定
 - 自动缩进
 - 搜索和替换
 - 补全
 - 提示
 - 代码上下文
 - Shell 窗口
 - 文本颜色
 - 启动和代码执行
 - 命令行用法
 - 启动失败
 - 运行用户代码
 - Shell 中的用户输出
 - 开发 tkinter 应用程序
 - 在没有子进程的情况下运行
 - 帮助和首选项 Help and Preferences
 - 帮助源
 - 首选项设置
 - macOS 上的 IDLE
 - 扩展
 - idlelib --- IDLE 应用程序的实现
- turtle --- 海龟绘图
 - 概述
 - 入门
 - 教程
 - 启动海龟环境
 - 基本绘图

- 画笔控制
- 海龟的位置
- 使用算法绘制图案
- 如何...
 - 尽快地开始
 - 自动开始和结束填充
 - 使用 `turtle` 模块命名空间
 - 在脚本中使用海龟绘图
 - 使用面向对象的海龟绘图
- 海龟绘图参考
 - `Turtle` 方法
 - `TurtleScreen/Screen` 方法
- `RawTurtle/Turtle` 方法和对应函数
 - 海龟动作
 - 获取海龟的状态
 - 度量单位设置
 - 画笔控制
 - 绘图状态
 - 颜色控制
 - 填充
 - 更多绘图控制
 - 海龟状态
 - 可见性
 - 外观
 - 使用事件
 - 特殊海龟方法
 - 复合形状
- `TurtleScreen/Screen` 方法及对应函数
 - 窗口控制
 - 动画控制
 - 使用屏幕事件
 - 输入方法
 - 设置与特殊方法
 - `Screen` 专有方法, 而非继承自 `TurtleScreen`
- 公共类
- 说明
- 帮助与配置
 - 如何使用帮助
 - 文档字符串翻译为不同的语言
 - 如何配置 `Screen` 和 `Turtle`
- `turtledemo` --- 演示脚本集
- Python 2.6 之后的变化
- Python 3.0 之后的变化