Entry专题

何时使用 Entry 组件？

Entry 组件仅允许用于输入一行文本，如果用于输入的字符串长度比该组件可显示空间更长，那内容将被滚动。这意味着该字符串将不能被全部看到（你可以用鼠标或键盘的方向键调整文本的可见范围）。

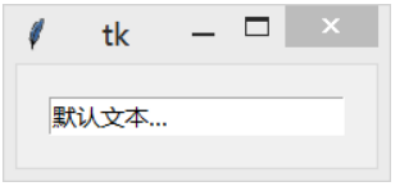
如果你希望接收多行文本的输入，可以使用 Text 组件。

用法

使用代码为 Entry 组件添加文本，可以使用 insert() 方法。如果要替换当前文本，可以先使用 delete() 方法，再使用 insert() 方法实现：

```
from tkinter import *

master = Tk()

e = Entry(master)
e.pack(padx=20, pady=20)

e.delete(0, END)
e.insert(0, "默认文本...")

mainloop()
```

复制代码



获取当前输入框的文本，可以使用 get() 方法：

```
s = e.get()
```

复制代码

你也可以绑定 Entry 组件到 Tkinter 变量（StringVar），并通过该变量设置和获取输入框的文本：

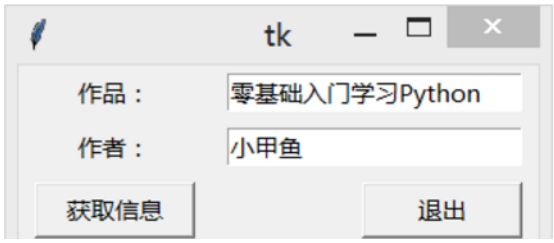
```
v = StringVar()
e = Entry(master, textvariable=v)
e.pack()

v.set("I love FishC.com!")
s = v.get()
```

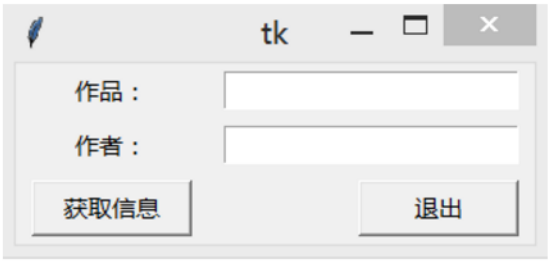
复制代码

下边的例子演示将 Entry 组件和 Button 组件配合，点击“获取信息”按钮时自动清空输入框并将内容输出：

```
from tkinter import *

master = Tk()

Label(master, text="作品：").grid(row=0)
Label(master, text="作者：").grid(row=1)

e1 = Entry(master)
e2 = Entry(master)
e1.grid(row=0, column=1, padx=10, pady=5)
e2.grid(row=1, column=1, padx=10, pady=5)

def show():
    print("作品：《%s》" % e1.get())
    print("作者：%s" % e2.get())
    e1.delete(0, END)
    e2.delete(0, END)

Button(master, text="获取信息", width=10, command=show).grid(row=3, column=0, sticky=W, padx=10, pady=5)
Button(master, text="退出", width=10, command=master.quit).grid(row=3, column=1, sticky=E, padx=10, pady=5)

mainloop()
```

复制代码






```
============
>>>
作品：《零基础入门学习Python》
作者：小甲鱼
```

最后需要提到的是 Entry 组件允许通过以下几种方式指定字符的位置：

- 数字索引号
- ANCHOR
- END
- INSERT
- 鼠标坐标（"@x"）

数字索引号：常规的 Python 索引号，从 0 开始

ANCHOR：对应第一个被选中的字符（如果有的话）

END：对应已存在文本的后一个位置

INSERT：对应插入光标的当前位置

鼠标坐标（"@x"）：x 是鼠标位置与 Entry 左侧边缘的水平距离，这样就可以通过鼠标相对地定位字符的位置

参数

Entry(master=None, **options) (class)

master -- 父组件

**options -- 组件选项，下方表格详细列举了各个选项的具体含义和用法：

选项	含义
background	1. 设置 Entry 的背景颜色 2. 默认值由系统指定
bg	跟 background 一样
borderwidth	1. 设置 Entry 的边框宽度 2. 默认值是 1 或 2 像素
bd	跟 borderwidth 一样
cursor	1. 指定当鼠标在 Entry 上飘过的时候的鼠标样式 2. 默认值由系统指定
exportselection	1. 指定选中的文本是否可以被复制到剪贴板 2. 默认值是 True 3. 可以修改为 False 表示不允许复制文本
font	1. 指定 Entry 中文本的字体 2. 默认值由系统指定
foreground	1. 设置 Entry 的文本颜色 2. 默认值由系统指定
fg	跟 foreground 一样
highlightbackground	1. 指定当 Entry 没有获得焦点的时候高亮边框的颜色 2. 默认值由系统指定

highlightcolor	1. 指定当 Entry 获得焦点的时候高亮边框的颜色 2. 默认值由系统指定
highlightthickness	1. 指定高亮边框的宽度 2. 默认值是 1 或 2 像素
insertbackground	指定输入光标的颜色
insertborderwidth	1. 指定输入光标的边框宽度 2. 如果被设置为非 0 值，光标样式会被设置为 RAISED 3. 小甲鱼温馨提示：将 insertwidth 设置大一点才能看到效果哦
insertofftime	1. 该选项控制光标的闪烁频率（灭） 2. 单位是毫秒
insertontime	1. 该选项控制光标的闪烁频率（亮） 2. 单位是毫秒
insertwidth	1. 指定光标的宽度 2. 默认值是 1 或 2 像素
invalidcommand	1. 指定当输入框输入的内容“非法”时调用的函数 2. 也就是指定当 validateCommand 选项指定的函数返回 False 时的函数 3. 详见本内容最下方小甲鱼关于验证详解
invcmd	跟 invalidcommand 一样
justify	1. 定义如何对齐输入框中的文本 2. 使用 LEFT，RIGHT 或 CENTER 3. 默认值是 LEFT
relief	1. 指定边框样式 2. 默认值是 SUNKEN 3. 其他可以选择的值是 FLAT，RAISED，GROOVE 和 RIDGE
selectbackground	1. 指定输入框的文本被选中时的背景颜色 2. 默认值由系统指定
selectborderwidth	1. 指定输入框的文本被选中时的边框宽度（选中边框） 2. 默认值由系统指定

selectforeground	1. 指定输入框的文本被选中时的字体颜色 2. 默认值由系统指定
show	1. 设置输入框如何显示文本的内容 2. 如果该值非空，则输入框会显示指定字符串代替真正的内容 3. 将该选项设置为 "*"，则是密码输入框
state	1. Entry 组件可以设置的状态：NORMAL，DISABLED 或 "readonly"（注意，这个是字符串。它跟 DISABLED 相似，但它支持选中和拷贝，只是不能修改，而 DISABLED 是完全禁止） 2. 默认值是 NORMAL 3. 注意，如果此选项设置为 DISABLED 或 "readonly"，那么调用 insert() 和 delete() 方法都会被忽略
takefocus	1. 指定使用 Tab 键可以将焦点移动到输入框中 2. 默认是开启的，可以将该选项设置为 False 避免焦点在此输入框中
textvariable	1. 指定一个与输入框的内容相关联的 Tkinter 变量（通常是 StringVar） 2. 当输入框的内容发生改变时，该变量的值也会相应发生改变
validate	1. 该选项设置是否启用内容验证 2. 详见本内容最下方小甲鱼关于验证详解
validatecommand	1. 该选项指定一个验证函数，用于验证输入框内容是否合法 2. 验证函数需要返回 True 或 False 表示验证结果 3. 注意，该选项只有当 validate 的值非 "none" 时才有效 3. 详见本内容最下方小甲鱼关于验证详解
vcmd	跟 validatecommand 一样
width	1. 设置输入框的宽度，以字符为单位 2. 默认值是 20 3. 对于变宽字体来说，组件的实际宽度等于字体的平均宽度乘以 width 选项的值
xscrollcommand	1. 与 scrollbar（滚动条）组件相关联 2. 如果你觉得用户输入的内容会超过该组件的输入框宽度，那么可以考虑设置该选项 3. 使用方法可以参考：Scrollbar 组件

方法

delete(first, last=None)
-- 删除参数 first 到 last 范围内（包含 first 和 last）的所有内容
-- 如果忽略 last 参数，表示删除 first 参数指定的选项
-- 使用 delete(0, END) 实现删除输入框的所有内容

get()
-- 获得当前输入框的内容

icursor(index)
-- 将光标移动到 index 参数指定的位置
-- 这同时也会设置 INSERT 的值

index(index)
-- 返回与 index 参数相应的选项的序号（例如 e.index(END)）

insert(index, text)
-- 将 text 参数的内容插入到 index 参数指定的位置
-- 使用 insert(INSERT, text) 将 text 参数指定的字符串插入到光标的位置
-- 使用 insert(END, text) 将 text 参数指定的字符串插入到输入框的末尾

scan_dragto(x)
-- 见下方 scan_mark(x)

scan_mark(x)
-- 使用这种方式来实现输入框内容的滚动
-- 需要将鼠标按下事件绑定到 scan_mark(x) 方法（x 是鼠标当前的水平位置），然后再将 <motion> 事件绑定到 scan_dragto(x) 方法（x 是鼠标当前的水平位置），就可以实现输入框在当前位置和 sacn_mack(x) 指定位置之间的水平滚动

select_adjust(index)
-- 与 selection_adjust(index) 相同，见下方解释

select_clear()
-- 与 selection_clear() 相同，见下方解释

select_from(index)
-- 与 selection_from(index) 相同，见下方解释

select_present()
-- 与 selection_present() 相同，见下方解释

select_range(start, end)
-- 与 selection_range(start, end) 相同，见下方解释

select_to(index)
-- 与 selection_to(index) 相同，见下方解释

selection_adjust(index)
-- 该方法是为了确保输入框中选中的范围包含 index 参数所指定的字符
-- 如果选中的范围已经包含了该字符，那么什么事情也不会发生
-- 如果选中的范围不包含该字符，那么会从光标的位置将选中的范围扩展至该字符

selection_clear()
-- 取消选中状态

selection_from(index)
-- 开始一个新的选中范围
-- 会设置 ANCHOR 的值

selection_present()
-- 返回输入框是否有处于选中状态的文本
-- 如果有则返回 True，否则返回 False

---

selection_range(start, end)
-- 设置选中范围
-- start 参数必须必 end 参数小
-- 使用 selection_range(0, END) 选中整个输入框的所有内容

selection_to(index)
-- 选中 ANCHOR 到 index 参数的间的所有内容

xview(index)
-- 该方法用于确保给定的 index 参数所指定的字符可见
-- 如有必要，会滚动输入框的内容

xview_moveto(fraction)
-- 根据 fraction 参数给定的比率调整输入框内容的可见范围
-- fraction 参数的范围是 0.0 ~ 1.0，0.0 表示输入框的开始位置，1.0 表示输入框的结束位置

xview_scroll(number, what)
-- 根据给定的参数水平滚动输入框的可见范围
-- number 参数指定滚动的数量，如果是负数则表示反向滚动
-- what 参数指定滚动的单位，可以是 UNITS 或 PAGES（UNITS 表示一个字符单元，PAGES 表示一页）

关于验证详解

由于小甲鱼查看了不少资料，很多在这里都没有解释清楚，所以这里单独列出来详细讲解下。

Entry 组件是支持验证输入内容的合法性的，比如要求输入数字，你输入了字母那就是非法。实现该功能，需要通过设置 validate、validatecommand 和 invalidcommand 选项。

首先启用验证的“开关”是 validate 选项，该选项可以设置的值有：

值	含义
'focus'	当 Entry 组件获得或失去焦点的时候验证
'focusin'	当 Entry 组件获得焦点的时候验证
'focusout'	当 Entry 组件失去焦点的时候验证
'key'	当输入框被编辑的时候验证
'all'	当出现上边任何一种情况的时候验证
'none'	1. 关闭验证功能 2. 默认设置该选项（即不启用验证） 3. 注意，是字符串的 'none'，而非 None

其次是为 validatecommand 选项指定一个验证函数，该函数只能返回 True 或 False 表示验证的结果。一般情况下验证函数只需要知道输入框的内容即可，可以通过 Entry 组件的 get() 方法获得该字符串。

其次是为 validatecommand 选项指定一个验证函数，该函数只能返回 True 或 False 表示验证的结果。一般情况下验证函数只需要知道输入框的内容即可，可以通过 Entry 组件的 get() 方法获得该字符串。

下边的例子中，在第一个输入框输入“小甲鱼”并通过 Tab 键将焦点转移到第二个输入框的时候，验证功能被成功触发：

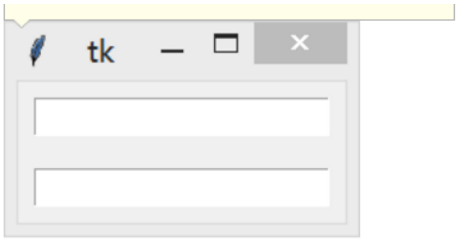
```
from tkinter import *

master = Tk()

def test():
    if e1.get() == "小甲鱼":
        print("正确！")
        return True
    else:
        print("错误！")
        e1.delete(0, END)
        return False

v = StringVar()

e1 = Entry(master, textvariable=v, validate="focusout", validatecommand=test)
e2 = Entry(master)
e1.pack(padx=10, pady=10)
e2.pack(padx=10, pady=10)

mainloop()
```

然后，invalidcommand 选项指定的函数只有在 validatecommand 的返回值为 False 的时候才被调用。

下边的例子中，在第一个输入框输入“小鱿鱼”，并通过 Tab 键将焦点转移到第二个输入框，validatecommand 指定的验证函数被触发并返回 False，接着 invalidcommand 被触发：

```
from tkinter import *

master = Tk()

v = StringVar()

def test1():
    if v.get() == "小甲鱼":
        print("正确！")
        return True
    else:
        print("错误！")
        e1.delete(0, END)
        return False

def test2():
    print("我被调用了......")
    return True

e1 = Entry(master, textvariable=v, validate="focusout", validatecommand=test1, invalidcommand=test2)
e2 = Entry(master)
e1.pack(padx=10, pady=10)
e2.pack(padx=10, pady=10)

mainloop()
```

复制代码



==========

>>>

错误！

我被调用了.....

最后，其实 Tkinter 还有隐藏技能，不过需要冷却才能触发，请听小甲鱼——道来.....

Tkinter 为验证函数提供一些额外的选项：

额外选项	含义
'%d'	操作代码：0 表示删除操作；1 表示插入操作；2 表示获得、失去焦点或 textvariable 变量的值被修改
'%i'	1. 当用户尝试插入或删除操作的时候，该选线表示插入或删除的位置（索引号） 2. 如果是由于获得、失去焦点或 textvariable 变量的值被修改而调用验证函数，那么该值是 -1
'%P'	1. 当输入框的值允许改变的时候，该值有效 2. 该值为输入框的最新文本内容
'%s'	该值为调用验证函数前输入框的文本内容
'%S'	1. 当插入或删除操作触发验证函数的时候，该值有效 2. 该选项表示文本被插入和删除的内容
'%v'	该组件当前的 validate 选项的值
'%V'	1. 调用验证函数的原因 2. 该值是 'focusin'，'focusout'，'key' 或 'forced'（textvariable 选项指定的变量值被修改）中的一个
'%W'	该组件的名字

为了使用这些选项，你可以这样写：validatecommand=(f, s1, s2, ...)

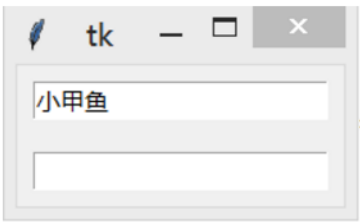
为了使用这些选项，你可以这样写：validatecommand=(f, s1, s2, ...)

其中，f 就是你“冷却后”的验证函数名，s1、s2、s3 这些是额外的选项，这些选项会作为参数依次传给 f 函数。我们刚刚说了，使用隐藏技能前需要冷却，其实就是调用 register() 方法将验证函数包装起来：

```
from tkinter import *

master = Tk()

v = StringVar()

def test(content, reason, name):
    if content == "小甲鱼":
        print("正确！")
        print(content, reason, name)
        return True
    else:
        print("错误！")
        print(content, reason, name)
        return False

testCMD = master.register(test)
e1 = Entry(master, textvariable=v, validate="focusout", validatecommand=(testCMD, '%P', '%v', '%W'))
e2 = Entry(master)
e1.pack(padx=10, pady=10)
e2.pack(padx=10, pady=10)

mainloop()
```

复制代码

当我故意输入“小甲鱼我爱你”的时候，DUANG的一下，是错误的，后来我果断删除了“我爱你”，嘿，又正确了：



========== RESTART

错误！
小甲鱼我爱你 focusout .64324776
正确！
小甲鱼 focusout .64324776