



# User Interface

The screenshot shows a window titled 'Nobel DB' with a feather icon. Inside the window, the title 'New Winner' is displayed in blue. Below the title, there are four input fields: 'name =' (a single-line text box), 'country =' (a single-line text box), 'year =' (a single-line text box), and 'category =' (a dropdown menu). The dropdown menu is currently set to 'chemistry'. At the bottom of the form is a blue button with the text 'Add' in white.



# Content

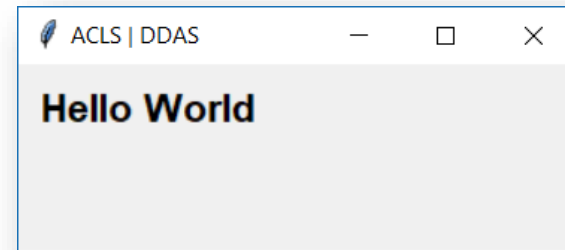
- tkinter
- Widgets
- Apply

The screenshot shows a Tkinter window titled "Nobel DB" with a standard macOS-style title bar (minimize, maximize, close buttons). The window content has a light gray background. At the top, the text "New Winner" is displayed in a bold, blue font. Below this, there are four labels with corresponding input fields: "name =" followed by a single-line text entry field; "country =" followed by a single-line text entry field; "year =" followed by a single-line text entry field; and "category =" followed by a dropdown menu currently showing "chemistry". At the bottom center of the form is a blue button with the white text "Add".



# Content

- tkinter
- Widgets
- Apply



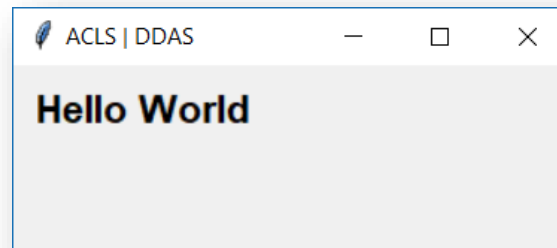


# tkinter

## Tk

**tkinter** is the Python interface to the **Tk** toolkit shipped with Python.

**Tk** is a free and open-source, cross-platform **widget** toolkit that provides a library of basic elements for building a graphical user interface (GUI) in many programming languages. It is very well suited for “quick-and-dirty” programming and prototypes.





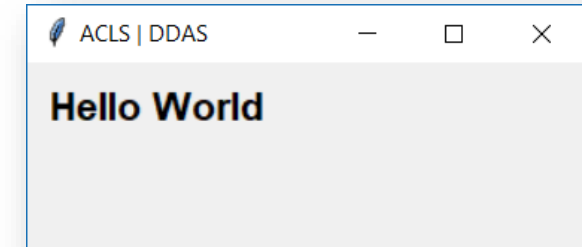
# tkinter

```
# import
import tkinter

# create new window
main_win = tkinter.Tk()
main_win.title('ACLS | DDAS')
main_win.geometry('300x100')

# add a label
lbl = tkinter.Label(main_win, text='Hello World')
lbl['font'] = 'Arial 16 bold'
lbl.config(padx=10, pady=10)
lbl.grid(row=0, column=0)

# run GUI
main_win.mainloop()
```





# Content

- tkinter
- Widgets
- Apply





# Widgets

Tkinter provides various **controls**, such as buttons, labels, and text boxes used in a GUI application. These controls are commonly called **widgets**.

Button	Is used to display buttons that can be clicked with the mouse.
Checkbutton	Is used to display a number of options as checkboxes. The user can select multiple options at a time.
Entry	Is used to display a single-line text field for accepting values from a user.
Frame	Is used as a container widget to organize other widgets.
Label	Is used to provide a single-line caption for other widgets. It can also contain images.
Listbox	Is used to provide a list of options to a user.
Radiobutton	Is used to display a number of options as radio buttons. The user can select only one option at a time.
Scrollbar	Is used to add scrolling capability to various widgets, such as list boxes.
Text	Is used to display text in multiple lines.

and more...



# Widgets

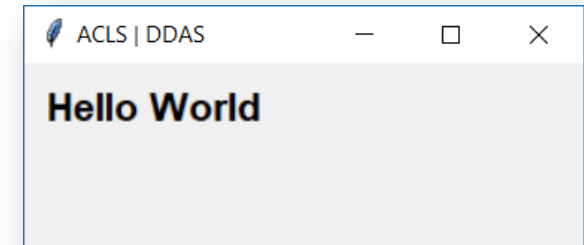
## Label

```
# import
import tkinter

# create new window
main_win = tkinter.Tk()
main_win.title('ACLS | DDAS')
main_win.geometry('300x100')

# add a label
lbl = tkinter.Label(main_win, text='Hello World')
lbl['font'] = 'Arial 16 bold'
lbl.config(padx=10, pady=10)
lbl.grid(row=0, column=0)

# run GUI
main_win.mainloop()
```







# Widgets

## Attributes

There are three ways to set the attribute values:

```
# add a label
lbl = tkinter.Label(main_win, text='Hello World')
lbl['font'] = 'Arial 16 bold'
lbl.config(padx=10, pady=10)
```

All widgets have the following general attributes:

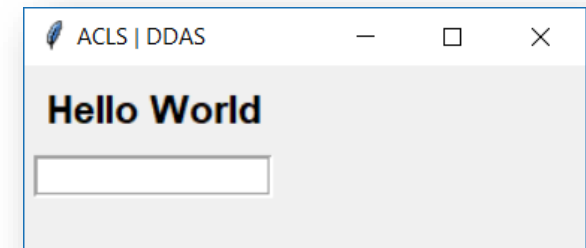
font	Used to set the font of the widget
height	Used to set the height of the widget
width	Used to set the width of the widget
borderwidth	Used to set the width of the border of the widget
relief	Used to set the style of the border of the widget
image	Used to set an image
bg	Used to set the background color
fg	Used to set the foreground color (usually font color)



# Widgets

## Entry

```
[...]  
  
# add a text entry  
etr = tkinter.Entry(main_win)  
etr['borderwidth'] = 3  
etr['relief'] = 'groove'  
etr.grid(row=1, column=0)  
  
[...]
```

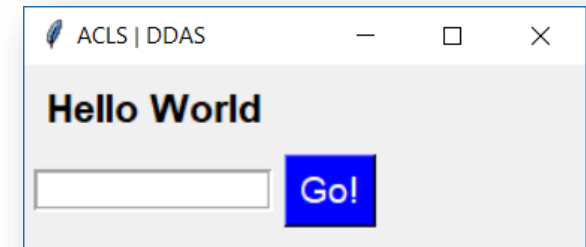




# Widgets

## Button

```
[...]  
  
# add a button  
btn = tkinter.Button(main_win, text='Go!')  
btn['command'] = change_text  
btn.config(font='Arial 14')  
btn.config(fg='#FFFFFF', bg='#0000FF')  
btn.grid(row=1, column=1)  
  
[...]
```





# Widgets

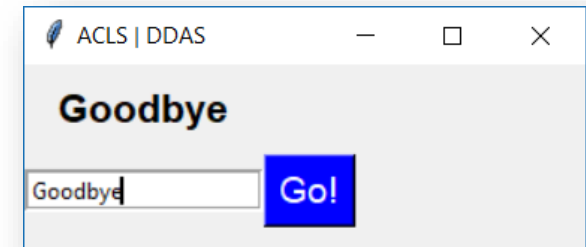
## Button - Command

```
# method to change the label text
def change_text():
    lbl['text'] = etr.get()

[...]

# add a button
btn = tkinter.Button(main_win, text='Go!')
btn['command'] = change_text
btn.config(font='Arial 14')
btn.config(fg='#FFFFFF', bg='#0000FF')
btn.grid(row=1, column=1)

[...]
```





# Widgets

## Geometry Management

```
# use the grid method to arrange widgets  
lbl.grid(row=0,column=0)  
etr.grid(row=1,column=0)  
btn.grid(row=1,column=1)
```



- pack()** – This geometry manager organizes widgets in blocks before placing them in the parent widget.
- grid()** – This geometry manager organizes widgets in a table-like structure in the parent widget.
- place()** – This geometry manager organizes widgets by placing them in a specific position in the parent widget.



# Content

- tkinter
- Widgets
- Apply

The screenshot shows a Tkinter window titled "Nobel DB" with a standard macOS-style title bar (minimize, maximize, close buttons). The window content has a light gray background and a blue title "New Winner". Below the title, there are four input fields arranged vertically, each preceded by a label: "name =", "country =", "year =", and "category =". The "name", "country", and "year" fields are empty text boxes. The "category =" field is a dropdown menu currently showing "chemistry". At the bottom center of the window is a blue button with the white text "Add".



# Apply

Let's programm a **GUI** for the sqlite database *nobel.sqlite* created in the **exercise last week** to add a new winner

Nobel DB

### New Winner

name =

country =

year =

category =

**Add**

insert

winners		
PK	id	INTEGER
FK	category_id	INTEGER
	year	INTEGER
	name	VARCHAR(50)
	country	VARCHAR(50)