Tkinter Hello, World!

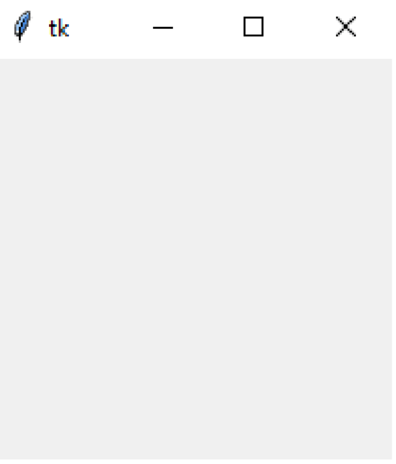
Summary: in this tutorial, you'll learn step by step how to develop the `Tkinter` "Hello, World!" program.

Creating a window

The following program shows how to display a window (https://www.pythontutorial.net/tkinter/tkinter-window/) on the screen:

```
import tkinter as tk

root = tk.Tk()
root.mainloop()
```

If you execute the program, you'll see the following window:



How it works.

First, import the `tkinter` module as `tk` to the program:

```
import tkinter as tk
```

Second, create an instance of the `tk.Tk` class that will create the application window:

```
root = tk.Tk()
```

By convention, the main window in Tkinter is called `root` . But you can use any other name like `main` .

Third, call the `mainloop()` method of the main window object:

```
root.mainloop()
```

The `mainloop()` keeps the window visible on the screen. If you don't call the `mainloop()` method, the window will display and disappear immediately. It will be so fast that you may not see its appearance.

Also, the `mainloop()` method keeps the window displaying and running until you close it.

Typically, you place the call to the `mainloop()` method as the last statement in a Tkinter program, after creating the widgets.

Troubleshooting

The `tkinter` module is a built-in Python module. But sometimes, it is not the case. For example, on Ubuntu, you may get the following error:

```
ImportError: No module named Tkinter
```

In this case, you need to install `tkinter` module using the following command line:

```
sudo apt-get install python3-tk
```

Displaying a label

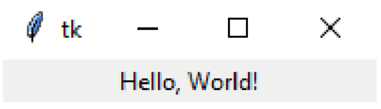
Now, it's time to place a component on the window. In `Tkinter`, components are called **widgets**.

The following adds a label (https://www.pythontutorial.net/tkinter/tkinter-label/) widget to the root window:

```
import tkinter as tk


root = tk.Tk()

# place a label on the root window
message = tk.Label(root, text="Hello, World!")
message.pack()

# keep the window displaying
root.mainloop()
```

Note that you'll learn more about the `Label` widget (https://www.pythontutorial.net/tkinter/tkinter-label/) in the upcoming tutorial.

If you run the program, you'll see the following output:



```
Hello, World!
```

How it works.

To create a widget that belongs to a container, you use the following syntax:

```
widget = WidgetName(container, **options)
```

In this syntax:

- The `container` is the parent window (https://www.pythontutorial.net/tkinter/tkinter-window/) or frame (https://www.pythontutorial.net/tkinter/tkinter-frame/) where you want to place the widget.
- The `options` is one or more keyword arguments (https://www.pythontutorial.net/python-basics/python-keyword-arguments/) that specify the configurations of the widget.

In the program, the following creates a `Label` widget placed on the `root` window:

```
message = tk.Label(root, text="Hello, World!")
```

And the following statement positions the `Label` on the main window:

```
message.pack()
```

Note that you'll learn more about the `pack()` (https://www.pythontutorial.net/tkinter/tkinter-pack/) method later. If you don't call the pack() method, the Tkinter still creates the widget. However, the widget is invisible.

Fixing the blur UI on Windows

If you find the text and UI are blurry on Windows, you can use the `ctypes` Python library to fix it.

First import the `ctypes` module:

```
from ctypes import windll
```

Second, call the `SetProcessDpiAwareness()` function:

```
windll.shcore.SetProcessDpiAwareness(1)
```

If you want the application to run across platforms such as Windows, macOS, and Linux, you can place the above code in a `try...finally` (https://www.pythontutorial.net/python-basics/python-try-except-finally/) block:

```
try:
    from ctypes import windll

    windll.shcore.SetProcessDpiAwareness(1)
finally:
    root.mainloop()
```

Summary

- Import `tkinter` module to create a Tkinter desktop application.
- Use `Tk` class to create the main window and call the `mainloop()` method to keep the window displays.
- In Tkinter, components are called widgets.