Utilises this as reference:
https://realpython.com/python-gui-tkinter/#controlling-layout-with-geometry-managers

It may be best to open this page and go through it alongside this project as parts are directly copied from the page.

Your First Tkinter GUI

Taken From *Building Your First Python GUI Application with Tkinter*

Step 1. Import the Tkinter python module like so:

```
Python >>>
>>> import tkinter as tk
```

This should be at the top of your Python file and on its own line.
What does this do?
We are taking/copying already written out Python code (stored in another Python File/Module) and are going to use that in our program. As they say, "*Why reinvent the wheel?*" This specific python module is for GUIs (*Graphical User Interfaces*)

Step 2. Now we will create a Tk **object and instance** (it is both) (more importantly pay attention to object) that is assigned the name *window:*

```
Python >>>
>>> window = tk.Tk()
```

What is an object?
An object is a way of grouping code such that they have specific characteristics/properties and actions. For example, an object may be a car. We can say that a car is drivable, gets us to point A to point B, has wheels, and etc. Objects are simply a **general** and **broad** representation.The point is that objects are a way for us to program and represent what we see in the world or think in our minds. *I will expand*.

What is an instance?
An instance is merely the existence of an object. It's like how we have different types of cars that tend to look different. While it may still be a car, it's a totally **unique** car. The point is that instances have to deal with the **uniqueness** of objects.

Step 3. For now, add a Label **Widget** called *greeting*:

```
Python >>>
>>> greeting = tk.Label(text="Hello, Tkinter")
```

Step 4. We need to **pack** (put in) the Label Widget into our **GUI window** like so:

```
Python >>>

>>> greeting.pack()
```

Step 5: Now, we need to write the following line to ensure that the GUI window **runs**:

```
Python >>>

>>> window.mainloop()
```

What is this?
This is called an **event loop**. For now, all you have to understand is that it helps your program run for as long you want it to. It will continue to run unless it either **runs into an error in your code** **OR** **you close the GUI**.

The Clicker Counter Project

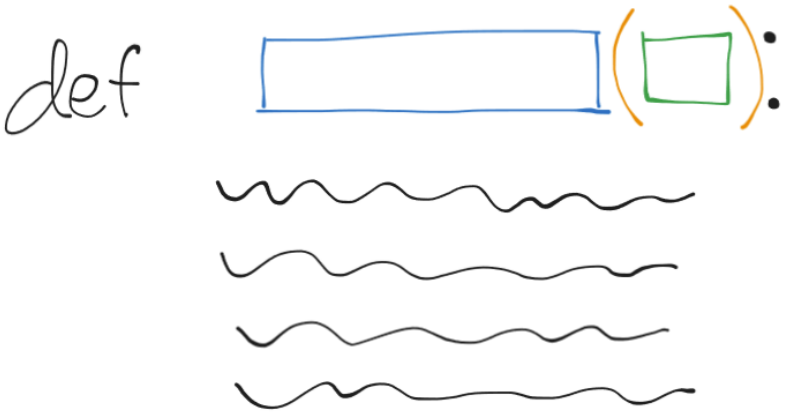
Step 1. Import the Tkinter Python module

Step 2. Create the window with the Tk object/instance

Step 3. Create a **Label Widget** stored to a variable called *counter* with the text as "0".

Step 4. Now **pack** the *counter* into the GUI Window.

Step 5. Create a function called *increase_count*:



In the **blue box** you will write the name of the function. For now, we can leave the **green box** empty.

The **green box** is where we place our parameters. Look back at *Your First Tkinter GUI* above. Go to **Step 3**. Notice how in between the parentheses () there's *text*. *text* is a **parameter/argument** (there's a difference but meh...) and we can assign to a value as seen.

Where the squiggly line is, we will continue writing our code there. The code is tabbed.

What is a function?
A function is a piece of code that we can **reuse** by calling it anywhere else in our **Python program**. It will run exactly as written out in the function.

Step 6. Inside the function, we will first create a variable called *temp*. It's an abbreviation for temporary. We will assign the *temp* variable to int(counter['text']) + 1 like so:

```
temp = int(counter['text']) + 1
```

What does this do?
counter['text'] gets us the **Label Widget**'s text. This will return a **String**. So we want to convert it into an **Integer** by casting.
Once we have converted into an **Integer**, we add one to the number and store into the variable called *temp*.

Step 7. On the next line, still inside the function, we will change the counter's text like so:

```
counter.config(text=str(temp))
```

What does this do?
The counter.config() function (also called a **method**) is a way for us to change the text of the **Label Widget**.
Inside the parenthesis we use the text **parameter/argument** and assign it to the now changed **Integer**. We then cast it to a **String** because text is always going to be **String**.

Step 8. Now, outside of the function (from now on), we will create a button:

```
btn = tk.Button(text="CLICK ME!", command=increase_count)
```

What is this?
This is a **button widget**. We have assigned it the text "CLICK ME!".
Notice the **command parameter/argument**. We give it the name of our **function**. We are giving our **function** to the **button**'s EVENT LISTENER.
Basically, whenever we **CLICK** the button, it will run our **function**.

Step 9. Pack the button.

Step 10. Run the main **event loop**.

YOU DID IT! YIPPPEEE!! 🎉:>>>>

Now, try messing around by changing the text or changing your function!