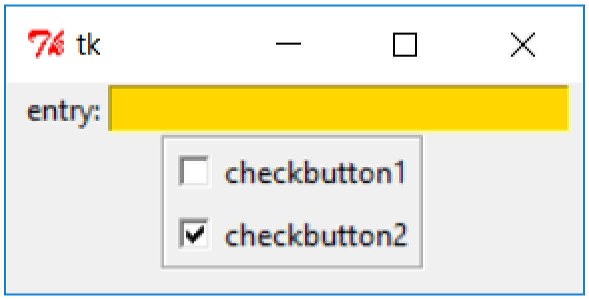
GUI Maker

This program translates an xml file into a Tkinter GUI. The app recognizes all Tkinter objects by name but also recognizes a series of special tags:

```
#Special Tags
#Top Level Tag
   FORM - The initial tag in an XML file that creates a master frame

#Grouping Tag
   GROUP - Used to group other widgets and can be used at any level.
           Typically used with radiobuttons, etc but not limited to only them.
           Group can have their own set of attributes. For example, you can define a variable that
           will be created at the group level but can be used by child widgets.

REPGROUP -  A repetition group is used to create a group of identical elements.
            For example, you can create a group of entry widgets each with their own label.

#Convenience Tags
INCLUDE - Allows you to include data from other XML files into the current file.
          You can use this functionality to reduce the complexity of an XML file by creating
          reusable XML files that can be included as needed.

#Placement Tags
   PACK - attributes specified are passed to the pack manager
   GRID - attributes specified are passed to the grid manager

#Variable Tags
VARIABLE - This tag is used to create special Python variables
           For example, you can create StringVar, IntVar, etc. that can be associated with
           radiobuttons, checkbuttons, sliders,etc.

TEXTVARIABLE - This tag is used to create a textvariable for Entry widgets.

#RepGroup Tag
```

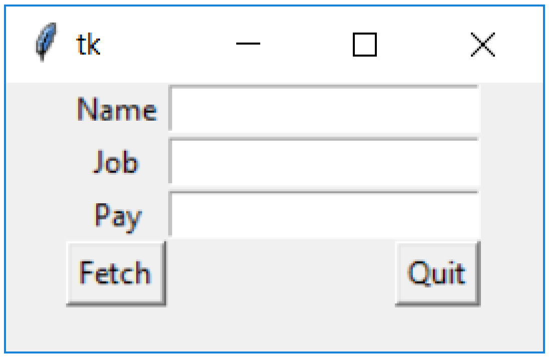
An example XML file:

```xml
<form>
  <group width="40">
    <label>
      <pack side="LEFT"/>
      <text>entry:</text>
    </label>
    <entry width='30' bg='gold'>
      <pack side="LEFT" fill="X"/>
      <textvariable name="entry1" typevar="StringVar"/>
    </entry>
  </group>
  <group name="cbgroup1" grouplist="cbgroup1list" bd="2" relief="GROOVE">
    <checkbutton text="checkbutton1">
      <pack side="TOP" anchor="W"/>
      <variable name="cb1" typevar="IntVar" onclick="noop"/>
    </checkbutton>
    <checkbutton>
      <pack side="TOP" anchor="W"/>
      <text>checkbutton2</text>
      <variable name="cb2" typevar="IntVar" default="true" onclick="noop"/>
    </checkbutton>
  </group>
</form>
```

| | |
|---|---|
|  | As you can see, you can group the label and entry widgets together and group the checkbuttons together. We specified the pack options for all the widgets.<br>Note that we used a text tag to specify the text attribute of all widgets except for the first checkbutton. For the first checkbutton, we specified the text attribute directly on the checkbutton tag. Each checkbutton has a variable tag specifying the type of variable to associate with the widget. The name attribute will be used to create a variable accessible by other parts of the program. |

An example REPGROUP XML file (`gui10.xml`):

```
<form>
  <repgroup tags="label:text, entry:textvariable" data="label:Name, Job, Pay |
entry:n1, j1, p1">
    <pack side="TOP" fill="X" expand="YES"/>
    <label width="5"><pack side="LEFT" anchor="E"/></label>
    <entry>
      <pack side="LEFT" fill="X" expand="YES"/>
      <variable typevar="StringVar" onclick="noop"/>
    </entry>
  </repgroup>
  <button text="Fetch" onclick="fetch"><pack side="LEFT"/></button>
  <button text="Quit" onclick="quit"><pack side="RIGHT"/></button>
</form>
```

| | |
|---|---|
|  | The REPGROUP produced 3 Entry widgets each with their own label. The REPGROUP requires prototype tags representing the elements in the repetition. In this case we created a label and entry prototype. We use REPGROUP tags to specify information about the group. The first tag is named "tags" and it contains a comma separated list of elements. In our example, you'll see "label:text, entry:textvariable" which corresponds to the label prototype and the entry prototype. The next tag that is required is a data tag. The data tag is used to provide the label names for each widget and the text variable name to assign. You'll use the textvariable names to access the created widget's content. |

GUI Maker

Sample Code:

```
from TkinterInterface import *
from GUI_MakerMixin import *

class Test(TkGUI_MakerMixin):
    def __init__(self, topLevelWindow=None, outputfilename=None):
        super().__init__(topLevelWindow=topLevelWindow, outputfilename=outputfilename)

    def noop(self, *arg):
        """
        Sample onclick handler
        :param arg:
        :return:
        """
        if len(arg) == 1:
            if type(arg) == tuple:
                try:
                    myarg = GetAttr(self, arg[0])
                except:
                    myarg = arg[0]

                try:
                    if isinstance(myarg, StringVarPlus) or isinstance(myarg,
mGraphicsLibName.IntVar):
                        # myarg = myarg.get()
                        print("noop called: %s:%s" % (arg[0],myarg.get()))
                        return
                except:
                    pass
            else:
                myarg = arg
        else:
            myarg = arg[0]
            print("noop called: %s:%s:%s" % (myarg, arg[1], GetAttr(self, myarg).get()))
            return
        print("noop called: %s" % myarg)


    def quit(self, *arg):
        """
        Sample onclick handler
        :param arg:
        :return:
        """
        exit()


    def fetch(self, arg):
        data={'Name':self.n1.get(),'Job':self.j1.get(), 'Pay':self.p1.get()}
        for label in data:
            value =data[label]
            print("%s:%s" % (label, value))


root = mRootWindow()
root.geometry("660x360")

m1 = Test(root)
fr = m1.makeGUI(root, "gui10.xml")
fr.pack()

root.mainloop()
```