

Использование языка Python для создания форм

I. Введение и создание окна формы

Для создания формы на Python будем использовать библиотеку **tkinter**, из которой нам нужно все её команды (*), которые можно добавить с помощью команд **from** и **import**:

```
from tkinter import *
```

Для создания окна формы присваиваем переменной объект **Tk()**, у меня она называется **root** :

```
root = Tk ()
```

*Примечание: Для создания нескольких окон, можно присваивать к разным переменным объект **Tk()***

Чтобы выставить размер и положение окна, используем функцию **geometry()**:

```
root.geometry('200x100+100+100 ')
```

Первые два числа указывают на длину и ширину окна в пикселях.

Последние два указывают на положение окна относительно левого верхнего угла монитора в пикселях.

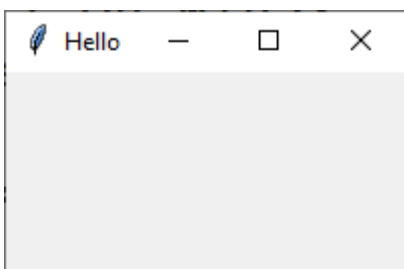
Чтобы выставить название окна можно использовать функцию **title()**. Наше окно будет называться “Hello”:

```
root.title("Hello")
```

А чтобы это окно все время прорисовывалась, напишем в конце всей программы функцию:

```
root.mainloop()
```

В итоге у нас должно получиться такое окно:



Попробуйте сами создать свое второе окно, опираясь на документацию и примечание

II. Таблички или Labels

Для начала, создадим табличку с текстом, например, всем известным словосочетанием: “Hello, World!!!”

Для этого мы должны присвоить переменной объект **Label**, пусть будет называться **lab**. Рисовать табличку будем в окне **root** с текстом “Hello, World!!!”:

```
lab = Label(root, text= "Hello, World!!!")
```

Чтобы поместить на окно какие-нибудь объекты можно использовать функцию **grid()**, которая помещает его в **таблицу мест** с учётом **столбцов (column)** и **строк(row)**, начиная с нуля.

Поместим нашу табличку **lab** на нулевую позицию:

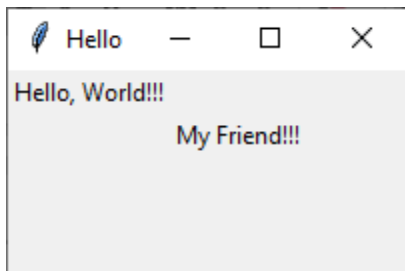
```
lab.grid(row=0, column=0)
```

Также можно создать ещё одну табличку и расположить её в другое место, например:

```
lab1 = Label(root, text= "My Friend!!!")
```

```
lab1.grid(row=1, column=1)
```

В итоге у нас должно получиться вот так:



Попробуйте сами создать свою табличку, опираясь на документацию

III. Кнопки и функции (Buttons and functions)

Для создания кнопки нужно присвоить переменной объект **Button**.

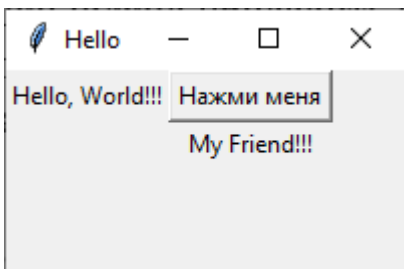
Также нужно учитывать на каком окне будет помещена кнопка, а также можно указать название на кнопке, например, “Нажми меня”. Пусть наша кнопка будет называться **btn**.

```
btn = Button(root, text= "Нажми меня")
```

Помещаем на нужное место с помощью функции **grid()**:

```
btn.grid(row=0, column=1)
```

И должно получиться вот так:



А чтобы наша кнопка могла на нажатие что-то выполнять, то нужно для неё создать функцию с помощью команды **def**. Пусть наша функция будет называться **print_hello**. И, как по названию вы поняли, она будет печатать слово “Hi” в консоль.

```
def print_hello():
```

```
    print("Hi")
```

Примечание 1: Пробелы нужно обязательно учитывать!!!

*Примечание 2: Чтобы использовать переменные из программы нужно их инициализировать в функции как **global**.*

Далее, надо настроить нашу кнопку на выполнение функции (**command**). Используем функцию **config()**.

```
btn.config(command=print_hello)
```

Теперь при нажатии на кнопку у нас будет выводиться слово “Hi”.

Попробуйте сами создать свою кнопку, опираясь на документацию и примечания.

VI. Поле для ввода или Entry

Для создания поля для ввода информации необходимо присвоить переменной объект **Entry**. Пусть наша переменная будет называться **ent** и она будет помещена на окно **root**.

```
ent = Entry(root)
```

И с помощью функции **grid** можно расположить на нужную позицию.

```
ent.grid(row=0, column=1)
```

Чтобы принимать информацию из поля, нужно использовать функцию **get()**.

Информация принимается переменной в виде строки!!!

```
info = ent.get()
```

Чтобы принимать другие виды данных можно произвести *конвертацию*.

<= это знак комментария, который не учитывает программа

```
info = int(ent.get()) # из строки в целые числа
```

```
info = float(ent.get()) # из строки в вещественные числа
```

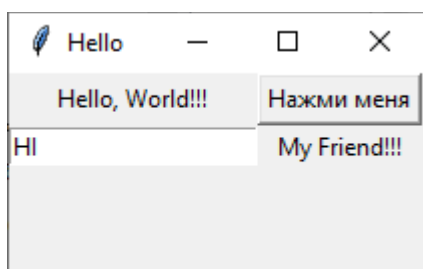
```
info = bool(ent.get()) # из строки в логический результат
```

А чтобы что-нибудь написать в поле можно написать в поле какой-нибудь текст, как, например, “HI”, то нужно использовать функцию **insert()**, обязательно учитывая что мы пишем в конец строки.

```
ent.insert(END, "HI")
```

А чтобы стереть всё из поля, то нужно использовать функцию: **ent.delete(0, END)**.

Должно получиться вот так:



Попробуйте сами создать свое поле, опираясь на документацию.