

ПЛАН ЗАНЯТИЯ

1.	Построение диалоговых окон для работы с файлами.....	2
1.1	Постановка задачи	2
1.2	Главное окно редактора.....	2
1.3	Диалоговые окно Open File (открыть файл), Save As File (сохранить как файл)3	
1.4	Диалог подтверждения	3
1.5	Диалог сообщений	4

1. Построение диалоговых окон для работы с файлами

1.1 Постановка задачи

В **Основах Tkinter** мы узнали, как выводить на экран окно, использовать классы Python, создавать дочерние окна и налаживать взаимодействие между родительскими и дочерними окнами. Мы познакомились со следующими элементами управления:

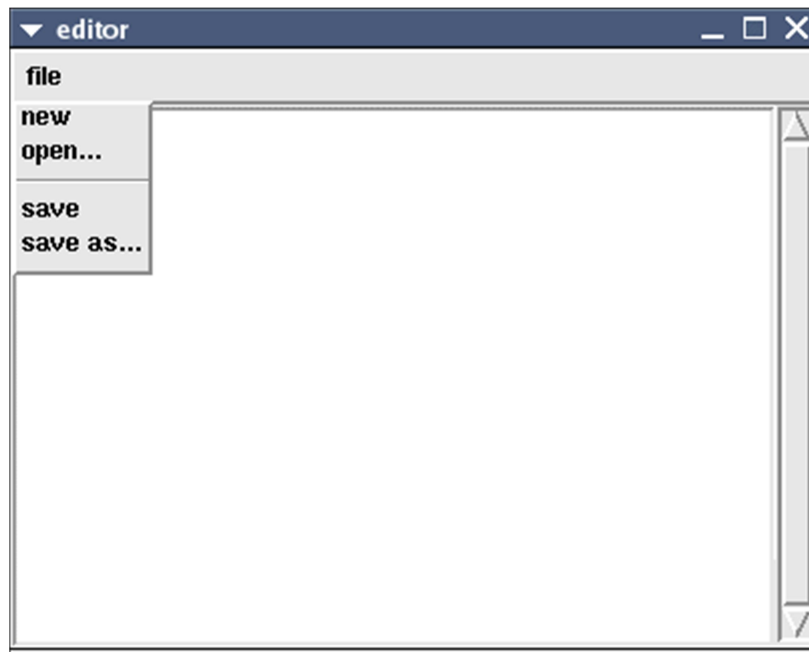
```
Tk(), Toplevel(), Button(), Text(), Label(), Frame().
```

В этом руководстве нам предстоит сконструировать простой, работающий текстовый редактор. В процессе придется поработать с некоторыми другими элементами управления Tkinter и научиться как связывать разные части интерфейса воедино.

Первый шаг в разработке графического интерфейса - получение нужных окон:

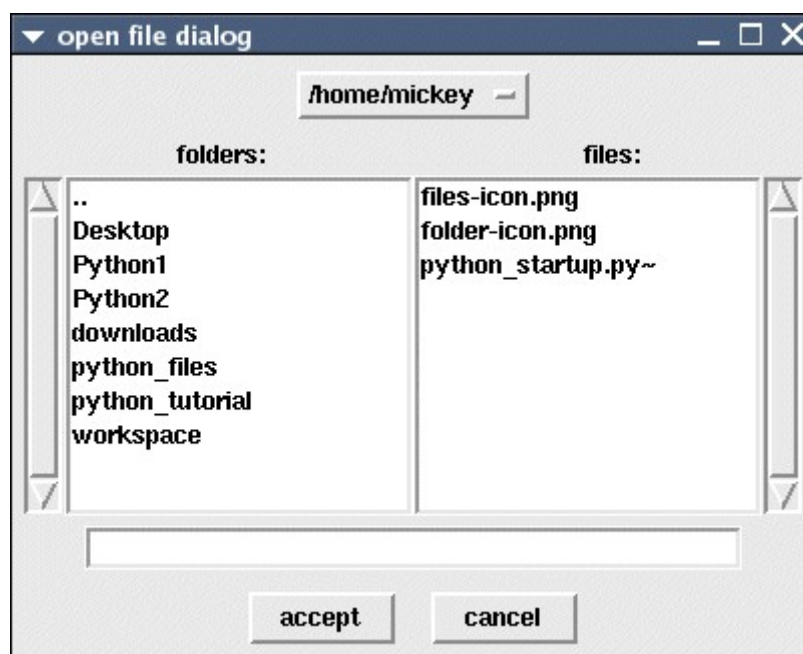
1.2 Главное окно редактора

Меню включает в себя пункты, которые, считается, должны быть в файловом меню любого редактора: *new* (новый), *open* (открыть), *save* (сохранить) и *save as* (сохранить как). Нетрудно определить, для каких из них нужен диалог - для тех, где есть "...":



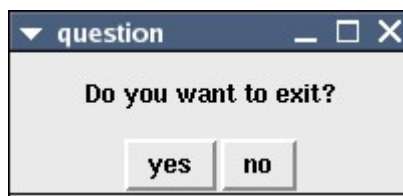
1.3 Диалоговые окно Open File (открыть файл), Save As File (сохранить как файл)

Диалоги *file.open* и *file.save* выглядят одинаково. Оба они легко могут быть описаны с помощью одного класса, если применить небольшой фокус при инициализации диалогового объекта:



1.4 Диалог подтверждения

Он нам потребуется при выходе из программы или чтобы задать вопрос: "Этот файл был изменен. Не хотите ли сохранить изменения?":



1.5 Диалог сообщений

Через диалог сообщений пользователь будет предупреждаться о возможных проблемах:



Итак, на следующих нескольких страницах мы создадим классы для этих окон [диалог *подтверждения* уже есть, например *yesno*], а в завершение свяжем эти диалоги друг с другом.