

The background of the slide is a light gray gradient with several realistic water droplets of various sizes scattered across it. The droplets have highlights and shadows, giving them a three-dimensional appearance.

СОЗДАНИЕ ПРИЛОЖЕНИЙ С ГРАФИЧЕСКИМ ИНТЕРФЕЙСОМ

БИБЛИОТЕКА TKINTER

БИБЛИОТЕКА TKINTER

- ПОЧТИ ВСЕ СОВРЕМЕННЫЕ ГРАФИЧЕСКИЕ ИНТЕРФЕЙСЫ ОБЩЕГО НАЗНАЧЕНИЯ СТРОЯТСЯ ПО МОДЕЛИ WIMP - WINDOW, ICON, MENU, POINTER (ОКНО, ИКОНКА, МЕНЮ, КУРСОР [ОБЫЧНО МЫШИ]). ВНУТРИ ОКОН РИСУЮТСЯ ЭЛЕМЕНТЫ ГРАФИЧЕСКОГО ИНТЕРФЕЙСА, ИМЕНУЕМЫЕ "ВИДЖЕТАМИ" (WIDGET - ШТУЧКА).

Toplevel

Окно верхнего уровня (корневой виджет)

Button

Кнопка. Простая кнопка для выполнения команды и других действий

Canvas

Рисунок. Может использоваться для вывода графических примитивов, например, для построения графиков

Checkbutton

Флажок. Кнопка, имеющая два состояния, при нажатии изменяет состояние на противоположное

Entry

Поле ввода текста

Frame

Рамка. Содержит в себе другие визуальные компоненты

Label

Этикетка. Показывает некоторый текст или графическое изображение

Listbox

Список. Показывает список, из которых пользователь может выделить один или несколько элементов

Menu

Меню. Служит для организации всплывающих (popup) и ниспадающих (pulldown) меню

Menubutton

Кнопка-меню. Используется для организации pulldown-меню

Message

Сообщение. Аналогично Label, но позволяет заворачивать длинные строки и легко меняет свой размер

Radiobutton

Переключатель. Представляет одно из альтернативных значений некоторой переменной. Обычно действует в группе. При нажатии на одну из кнопок, все остальные кнопки автоматически "отскакивают", совсем как кнопки выбора волны на радиоприемнике

Scale

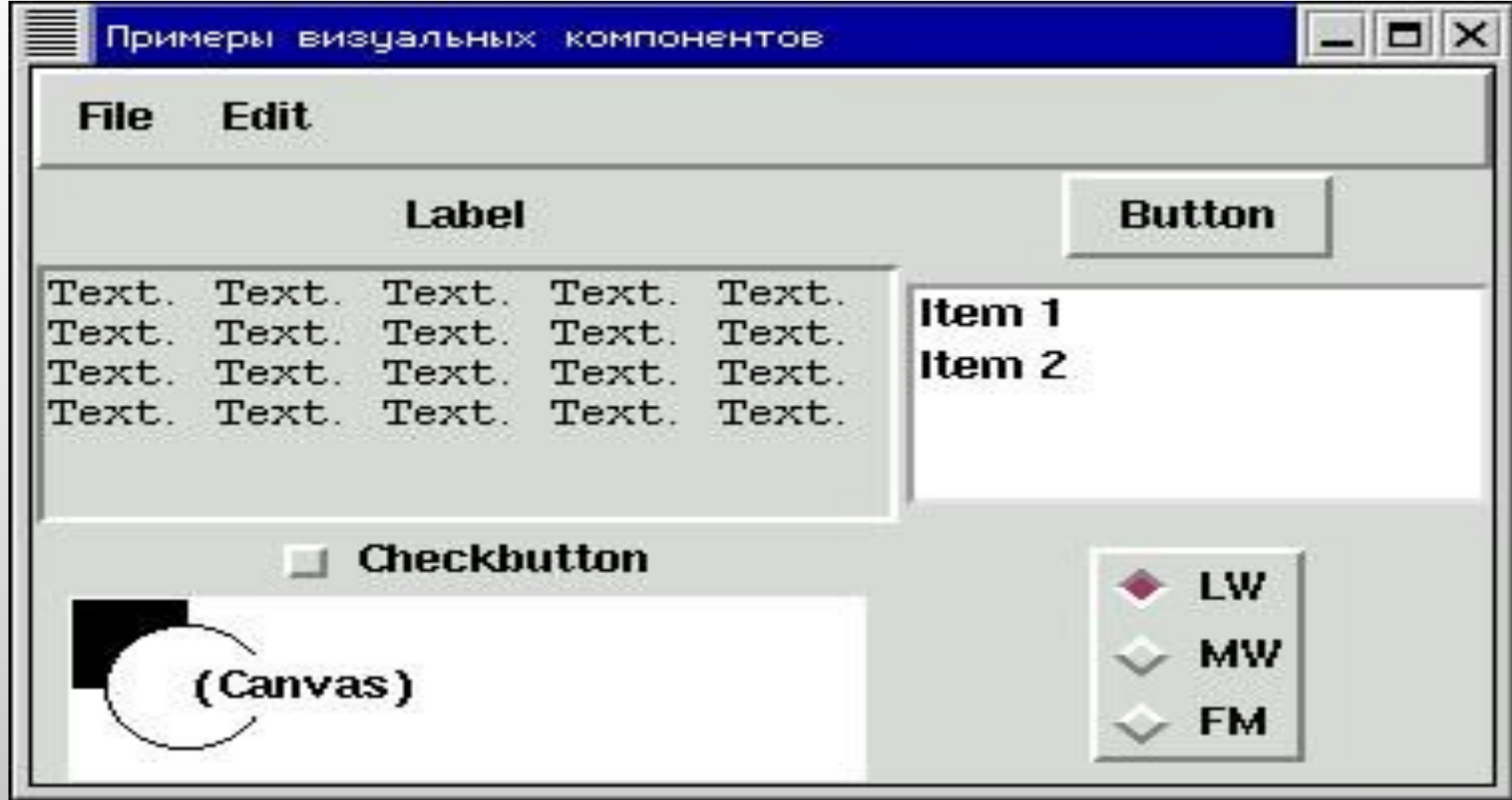
Шкала. Позволяет задать числовое значение путем перемещения движка

Scrollbar

Полоса прокрутки. Может использоваться вместе с некоторыми другими компонентами для их прокрутки

Text

Форматированный текст. Позволяет показывать, редактировать и форматировать текст с использованием различных стилей, а также внедрять в текст рисунки и окна.



- FROM TKINTER IMPORT *
- IMPORT MATH

- DEF SQUARE(RADIUS):
- RETURN MATH.PI * RADIUS**2

- DEF CALCULATE_SQUARE():
- RADIUS = FLOAT(RADIUS_ENTRY.GET())
- TRY:
- SQUARE = "%11.3F" % SQUARE(RADIUS)
- EXCEPT:
- SQUARE = "?"
- SQUARE_LABEL.CONFIGURE(TEXT=SQUARE)

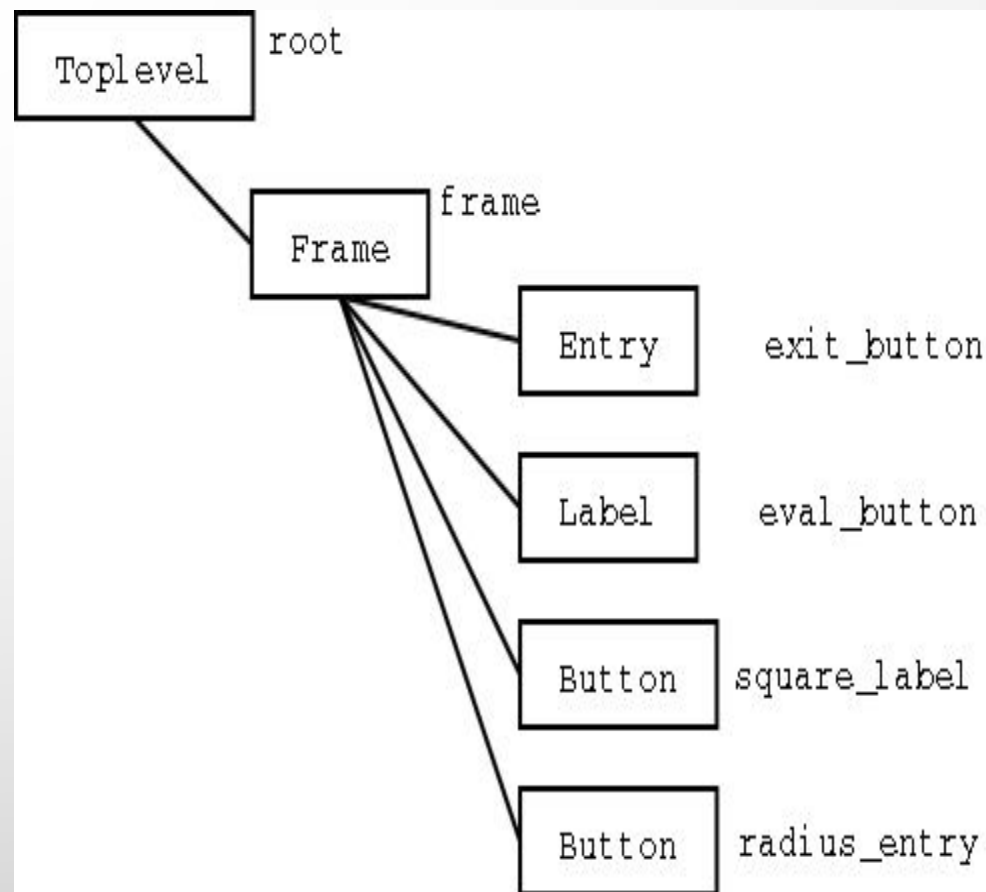


- `ROOT = TK()`
- `ROOT.TITLE("ПЛОЩАДЬ КРУГА")`
- `FRAME = FRAME(ROOT)`
- `FRAME.PACK()`
- `RADIUS_ENTRY = ENTRY(FRAME, WIDTH=10)`
- `RADIUS_ENTRY.GRID(ROW=0, COLUMN=0)`
- `SQUARE_LABEL = LABEL(FRAME, TEXT="?")`
- `SQUARE_LABEL.GRID(ROW=0, COLUMN=1)`
- `EVAL_BUTTON = BUTTON(FRAME, TEXT="CALCULATE", WIDTH=10,`
`COMMAND=CALCULATE_SQUARE)`
- `EVAL_BUTTON.GRID(ROW=1, COLUMN=0)`
- `EXIT_BUTTON = BUTTON(FRAME, TEXT="EXIT", WIDTH=10,`
`COMMAND=ROOT.DESTROY)`
- `EXIT_BUTTON.GRID(ROW=1, COLUMN=1)`
- `ROOT.MAINLOOP()`



ОКНО ГРАФИЧЕСКОГО ПРИЛОЖЕНИЯ

- РАСПОЛОЖЕНИЕМ ВИДЖЕТОВ УПРАВЛЯЮТ МЕНЕДЖЕРЫ РАСПОЛОЖЕНИЯ. В TKINTER ИХ ТРИ РАЗНОВИДНОСТИ:
- PACK
- "УПАКОВЫВАЕТ" ВИДЖЕТЫ ДРУГ К ДРУГУ ТАК, ЧТОБЫ ОНИ "ТЯГОТЕЛИ" К ОПРЕДЕЛЕННЫМ КРАЯМ. САМЫЙ ПРОСТОЙ ИЗ МЕНЕДЖЕРОВ РАСПОЛОЖЕНИЯ, НО ДОСТАТОЧЕН ДЛЯ МНОГИХ ЗАДАЧ.
- GRID
- РАСПОЛАГАЕТ ВИДЖЕТЫ В ЯЧЕЙКАХ ПРЯМОУГОЛЬНОЙ СЕТКИ (ПОЧТИ КАК В ЭЛЕКТРОННОЙ ТАБЛИЦЕ).
- PLACE
- РАСПОЛАГАЕТ ПОДЧИНЕННЫЕ ВИДЖЕТЫ ОТНОСИТЕЛЬНО ВИДЖЕТА-ХОЗЯИНА, В АБСОЛЮТНЫХ ИЛИ ОТНОСИТЕЛЬНЫХ КООРДИНАТАХ.



КАЖДЫЙ ВИДЖЕТ МОЖЕТ ОБРАБАТЫВАТЬ ВОЗНИКАЮЩИЕ В НЕМ СОБЫТИЯ

- KEYPRESS
- НАЖАТИЕ КЛАВИШИ НА КЛАВИАТУРЕ
- KEYRELEASE
- ОТПУСКАНИЕ КЛАВИШИ НА КЛАВИАТУРЕ
- BUTTONPRESS
- НАЖАТИЕ КНОПКИ МЫШИ
- BUTTONRELEASE
- ОТПУСКАНИЕ КНОПКИ МЫШИ
- MOTION
- ДВИЖЕНИЕ МЫШИ (В РАМКАХ КОМПЕТЕНЦИИ ВИДЖЕТА)
- ENTER
- ВХОЖДЕНИЕ КУРСОРА В ПРЕДЕЛЫ ВИДЖЕТА
- LEAVE
- ВЫХОД КУРСОРА ЗА ПРЕДЕЛЫ ВИДЖЕТА
- MOUSEWHEEL
- КРУЧЕНИЕ КОЛЕСИКА МЫШИ
- VISIBILITY
- ИЗМЕНЕНИЕ ВИДИМОСТИ ОКНА
- FOCUSIN
- ОКНО ПОЛУЧАЕТ ФОКУС

- FOCUSOUT
- ОКНО ТЕРЯЕТ ФОКУС
- REPARENT
- ИЗМЕНЕНИЕ РОДИТЕЛЯ ОКНА
- DESTROY
- УНИЧТОЖЕНИЕ ОКНА
- ACTIVATE
- ОКНО СТАНОВИТСЯ АКТИВНЫМ
- DEACTIVATE
- ОКНО СТАНОВИТСЯ

- ОПИСАНИЕ СОБЫТИЯ ЗАДАЕТСЯ СТРОКОЙ (ЧАСТЬ ЭЛЕМЕНТОВ МОЖЕТ БЫТЬ ОПУЩЕНА):
- <МОДИФИКАТОРЫ-ТИП-ДЕТАЛИЗАЦИЯ> НАПРИМЕР,
- "<BUTTONPRESS-3>" ИЛИ "<3>" -- НАЖАТИЕ ПРАВОЙ КНОПКИ МЫШИ
- "<DOUBLE-BUTTON-1>" -- ДВОЙНОЙ ЩЕЛЧОК ЛЕВОЙ КНОПКИ МЫШИ
- "<SHIFT-DOUBLE-BUTTON-1>" -- ДВОЙНОЙ ЩЕЛЧОК ЛЕВОЙ КНОПКИ МЫШИ ПРИ НАЖАТОЙ КЛАВИШЕ SHIFT
- "<SPACE>" -- НАЖАТИЕ КЛАВИШИ ПРОБЕЛ
- "к" -- НАЖАТИЕ КЛАВИШИ С СИМВОЛОМ "к"

- ПРИМЕР ПРОГРАММЫ, В КОТОРОЙ ТЕКСТОВОЕ ПОЛЕ ПЕРЕХВАТЫВАЕТ ДВИЖЕНИЯ МЫШИ И ПИШЕТ АТТРИБУТЫ СОБЫТИЯ (EVENT)

```
tk
char: ??
focus: 1
height: 0
keycode: 0
keysym: ??
keysym_num: 0
num: 0
send_event: 0
serial: 3947
state: 0
time: -1582016252
type: 6
widget: .135733392
width: 0
x: 563
x_root: 577
y: 173
y_root: 203
```

- FROM TKINTER IMPORT *
- TK = TK()
- TXT = TEXT(TK)
- TXT.PACK()

- DEF GIVE_INFO(EVENT):
- # EVENT.WIDGET - ВИДЖЕТ, В КОТОРОМ ПРОИЗОШЛО СОБЫТИЕ
- TXT.DELETE("1.0", END)
- ITEMS = EVENT.__DICT__.ITEMS()
- ITEMS.SORT()
- FOR K, V IN ITEMS:
- TXT.INSERT(END, "%S: %S\n" % (K, V))
-
- TXT.BIND("<MOTION>", GIVE_INFO)
- TK.MAINLOOP()

Пример 1. Минимальная программа с визуальным интерфейсом tkinter

```
from tkinter import * # подключение библиотеки tkinter
# Создаем метку и прикрепляем ее к верху формы
(окно программы):
Label(text = "Hello World").pack(side = TOP)
# Если явно в программе не создавать главное окно,
# то библиотека tkinter создаст его самостоятельно.
# Создаем функцию без параметров (обработчик
нажатия кнопки):
def hello( ):
    print("На кнопку нажали")
# Создаем кнопку, подключаем обработчик и
прикрепляем
# ее к низу формы:
Button(text = "Hello event World",
    command = hello).pack(side = BOTTOM)
# Запускаем цикл обработки сообщений:
mainloop()
```

