



El futuro digital
es de todos

MinTIC

«Misión
TIC2022»

Fund. de Programación

Grupos 79,80,81



UNIVERSIDAD
DE ANTIOQUIA

Facultad de Ingeniería

RESUMEN SESIÓN ANTERIOR





COMPONENTES

Tkinter cuenta con distintos elementos para construir una GUI, estos elementos también reciben el nombre de widget:

https://www.tutorialspoint.com/python/python_gui_programming.htm

- Frame
- Label
- Button
- Entry
- Text



grid()

El método grid permite posicionar los elementos en una celda en específico, indicamos la celda usando el índice de fila (**row**) y columna (**column**) correspondiente, el ancho y la altura de cada celda son configurables, además un widget puede ocupar varias celdas. Con grid se puede crear fácilmente GUI de tipo formulario.



FRAME

Los marcos nos permiten agrupar el contenido de nuestra interfaz y manipularlos más fácilmente. En un principio estos son invisibles, así que es difícil identificarlos, pero si les damos un color de fondo veremos donde están ubicados. Para crear un marco podemos instanciarlo indicándole en que ventana va a estar contenido.



MÉTODOS DE INTERES

- **frame.winfo_children():** Retorna un listado de todos los elementos que tiene asociado un frame
- **widget.destroy():** Permite destruir un elemento de la GUI
- **frame.pack_forget():** Refresca la pantalla y retira los elementos eliminados del frame

Manejo de archivos

JUAN FERNANDO GONZÁLEZ
GRUPOS 79,80,81
Semana 7 Sesión 1





ARCHIVOS

Los archivos son el medio por el cual se almacena información en un computador. Python permite leer y escribir strings en distintos tipos de archivos por medio del módulo file, el cual viene cargado por defecto por lo cual no es necesario importarlo para poder usar sus métodos.



ABRIR ARCHIVO

La forma más sencilla para leer un archivo es usando el método `open()`

`open("archivo", "modo")`

modos:

- **'r'** modo lectura (read)
- **'w'** modo escritura (write)
- **'a'** para agregar (append)

El operador `+` agrega funcionalidades al modo eg. **"r+"**



LEER ARCHIVOS

El archivo debe ser abierto (**open**) en un modo que permita la lectura

- **read():** Retorna todo el archivo como un string
- **readline():** Retorna linea por linea
- **readlines():** Retorna el archivo como una lista, donde cada elemento de la lista es una línea



ESCRIBIR Y CERRAR ARCHIVO

- Para escribir se utiliza el método **write()** El cual recibe como argumento un string

file.write("string de datos a guardar")

**Si el modo es w sobrescribe

- EL método **close()** cierra un archivo abierto, lo ideal es que después de realizar una tarea con el archivo se cierre su conexión



SENTENCIA WITH PARA ABRIR ARCHIVOS

La sentencia with se introdujo en python para, entre otras cosas, trabajar con archivos, con esta no es necesario llamar la sentencia close ya que with permite agrupar los procesos de trabajo con el archivo y se encarga de controlar el flujo.

with open('file', 'modo') as f:
 f.metodoA()
 f.metodoB()



ARCHIVOS JSON

Java**S**cript **O**bject **N**otation, es un formato de texto para guardar y modificar diversos tipos de datos (números, hileras, matrices, booleanos...). Se caracteriza por ser fácil de trabajar y comprender, tanto para los desarrolladores como para las máquinas. Su estructura se suele almacenar en archivos con extensión .json (también pueden ser guardados en str) y su formato recuerda a la construcción de un diccionario.



MÓDULO JSON

Python cuenta con un módulo para trabajar con archivos JSON, el cual permite leer uno de estos archivos y lo carga a python como un diccionario. Para el caso de escritura, toma un diccionario y lo almacena en un archivo tipo JSON



ESCRIBIR UN JSON

Se debe abrir un flujo al archivo en modo escritura y usar el método **dump()**

```
dict={1:"dato"}
```

```
with open('file.json','w') as f:  
    json.dump(dict,f)
```




CARGAR UN JSON

Se debe abrir un flujo al archivo en modo lectura y usar el método **load()**

```
with open('file.json','r') as f:  
    dict=json.load(f)
```