

Introduzione a Python



by Martina Bellio - 2020



- Per i partecipanti on line:
compilare il modulo inviato dal resp. Locale
(scheda di valutazione del corso)
- Copia delle trasparenze sono disponibili anche qui:
http://www2.lnl.infn.it/~canella/RASPBERRY_PYTHON
- Nel medesimo nodo web trovate anche copia dei
programmi python di esempio usati nel corso:
http://www2.lnl.infn.it/~canella/RASPBERRY_PYTHON/PYTHON_CODE
nelle 3 cartelle pycode1, pycode2, pycode3



1 - Per cominciare ...

- Introduzione a Python.
- Installazione ed uso dell'ambiente IDLE.
- Primi esempi (“Ciao Mondo!”).
- Uso interattivo e tramite script.
- Uso diretto dell'interprete Python.



Nuts and Bolts

- I costrutti base per i programmi procedurali.
- Le variabili e i valori.
- Input/Output.
- Le strutture di controllo.
- Esempi ed esercizi.

Esempi: 1



```
File Edit Format Run Options Window Help

subject = input("What is the best subject? ")
if not(subject=="Computing"):
    print ("Wrong, it's Computing")
else:
    print ("Right! It is just that")

Ln: 7 Col: 0
```

Esempio di Input/Output ed uso di IF-THEN-ELSE

Esempi: 2



```
File Edit Format Run Options Window Help

from time import sleep, time
sleep(2)
start_time = time()
print("quick,quick ... print ENTER")|
input()
reaction_time = time() - start_time
print("your reaction time is :", '%.2f' % reaction_time, "seconds")

Ln: 6 Col: 36
```

Esempio di accesso a funzioni di libreria (time) e loro uso : sleep(2) e time()
Calcolo del tempo di reazione.

Esempi: 3



```
File Edit Format Run Options Window Help
# time, sleep
from time import sleep, time
# to have random int in range 0-5
from random import randint

tot = 0

for i in range(5):
    sleep_time = randint(0,5)
    sleep(sleep_time)
    print("\nquick,quick ... hit ENTER")
    start_time = time()
    input()
    reaction_time = time() - start_time
    print(i+1, "your reaction time is :", '%.2f' % reaction_time, "seconds")
# print('That took: ', '%.2f' % timeTaken, ' secs') from Tim
    tot = tot + reaction_time

print()
print("your total reaction time is :", tot, "seconds")
print("your average reaction time is :", tot/5, "seconds")

Ln: 14 Col: 0
```

Calcolo del tempo di reazione medio, iterando 5 volte la misura (con il costrutto for ...)



I programmi object-oriented (OOP)

- Introduzione alla programmazione OO.
- Predisposizione di IDLE per usare GUIZERO.
- Elementi base per programmi OO con Guizero.
- Primi esempi di programmi Guizero.
- Idee per proseguire con Guizero.

I programmi object-oriented



La programmazione object-oriented (OOP) nasce negli anni '60 del secolo scorso negli ambienti che lavorano ai primi progetti di Intelligenza Artificiale (con LISP, SIMULA).

Prende davvero spazio al di fuori delle nicchie della ricerca per la realizzazione di giochi elettronici e, soprattutto, per la produzione di interfacce grafiche orientate all'utente (GUI), con linguaggi di programmazione di alto livello come C++, JAVA, Python.

IDLE con GUIZERO



Per vedere qualche esempio pratico di programmazione object-oriented (OOP) useremo il settore della produzione di interfacce grafiche orientate all'utente (GUI), con Python, ovviamente.

Siete riusciti a installare Python3 (e IDLE) ?

Spero di si, ora vedremo come installare anche GUIZERO.

ISTRUZIONI DI INSTALLAZIONE e MANUALI della libreria GUIZERO qui: <https://lawsie.github.io/guizero>

GUIZERO per linux



Per installare GUIZERo su linux:

1. Installare tkinter (se non già installato) usando il package manager, oppure, da terminale, con:

```
sudo apt install python3-tk
```

2. Installare pip3 con :

```
sudo apt-get install python3-pip
```

3. Installare guizero con pip3, da terminale:

```
sudo pip3 install guizero --upgrade
```

GUIZERO per Win7 1



Per installare GUIZERo su Win7 (dopo aver installato python3 (pip è già incluso):

Da command line (Prompt dei comandi):

1. Installare guizero con pip:

```
Python -m pip install --upgrade guizero
```

Ovviamente il computer deve essere in rete (– è un doppio -) .

GUIZERO per Win 2



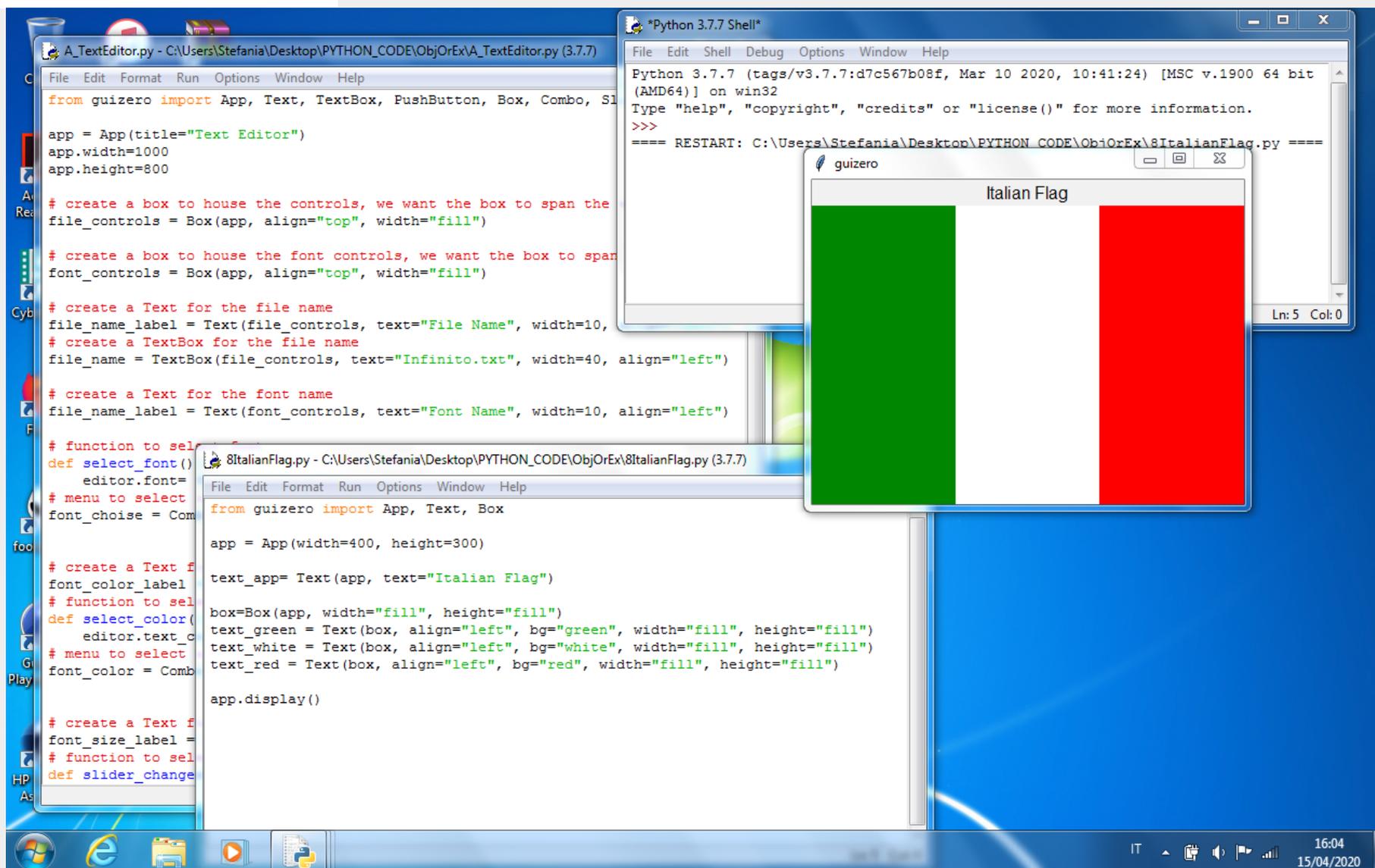
```
cmd Prompt dei comandi
Microsoft Windows [Versione 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Tutti i diritti riservati.

C:\Users\Stefania>python --version
Python 3.7.7

C:\Users\Stefania>python -m pip install --upgrade guizero
Requirement already up-to-date: guizero in c:\users\stefania\appdata\local\programs\python\python37\lib\site-packages (1.1.0)

C:\Users\Stefania>
```

GUIZERO per Win7 !!!



The screenshot shows a Windows 7 desktop environment. In the foreground, there are three windows:

- A_TextEditor.py**: A Python script using the `guizero` library to create a text editor interface. The code includes imports for `App`, `Text`, `TextBox`, `PushButton`, `Box`, and `Combo`. It sets up an application window titled "Text Editor" with dimensions 1000x800. It creates two boxes for controls, adds labels and text boxes for file and font names, and defines a `select_font` function.
- Python 3.7.7 Shell**: A terminal window showing the execution of the `8ItalianFlag.py` script. The output indicates a successful restart of the application.
- 8ItalianFlag.py**: A Python script that uses `guizero` to create a window titled "Italian Flag" with a width of 400 and height of 300. It displays the text "Italian Flag" and a box containing three vertical bars of green, white, and red, representing the Italian flag.

The taskbar at the bottom shows the system clock as 16:04 on 15/04/2020.



Esempi (di programmi object-oriented - OO)

- GUI : una finestra con un testo.
- Una finestra con un pulsante che modifica la finestra stessa.
- Un text editor.
- Esempi per proseguire.

GUI: Esempi 1

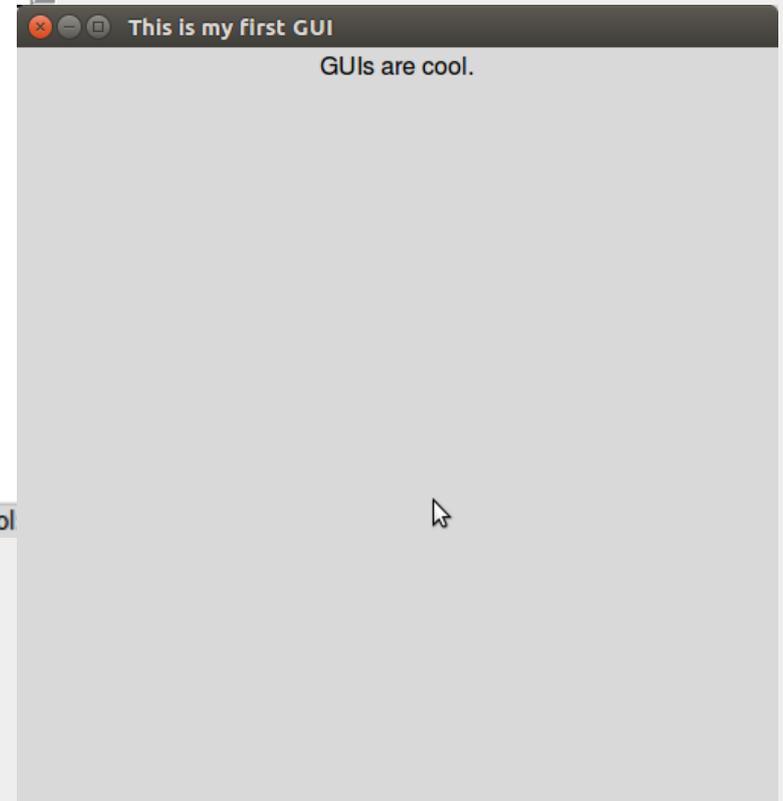


File Edit Format Run Options Window Help

```
from guizero import App, Text
app = App(title="This is my first GUI")

message = Text(app, text="GUIs are cool.")
app.display()
|
```

Ln: 6 Col



Esempio di produzione di finestra con un testo

GUI: Esempi 2



```
File Edit Format Run Options Window Help
# Import the GUI widgets that you'll be using, and create the 'app' for your program.
from guizero import App, TextBox, PushButton, Text, info

app = App()

# Function definitions for your events go here.
def btn_go_clicked():
    info("Greetings", "Hello, " + txt_name.value + " - I hope you're having a nice day")

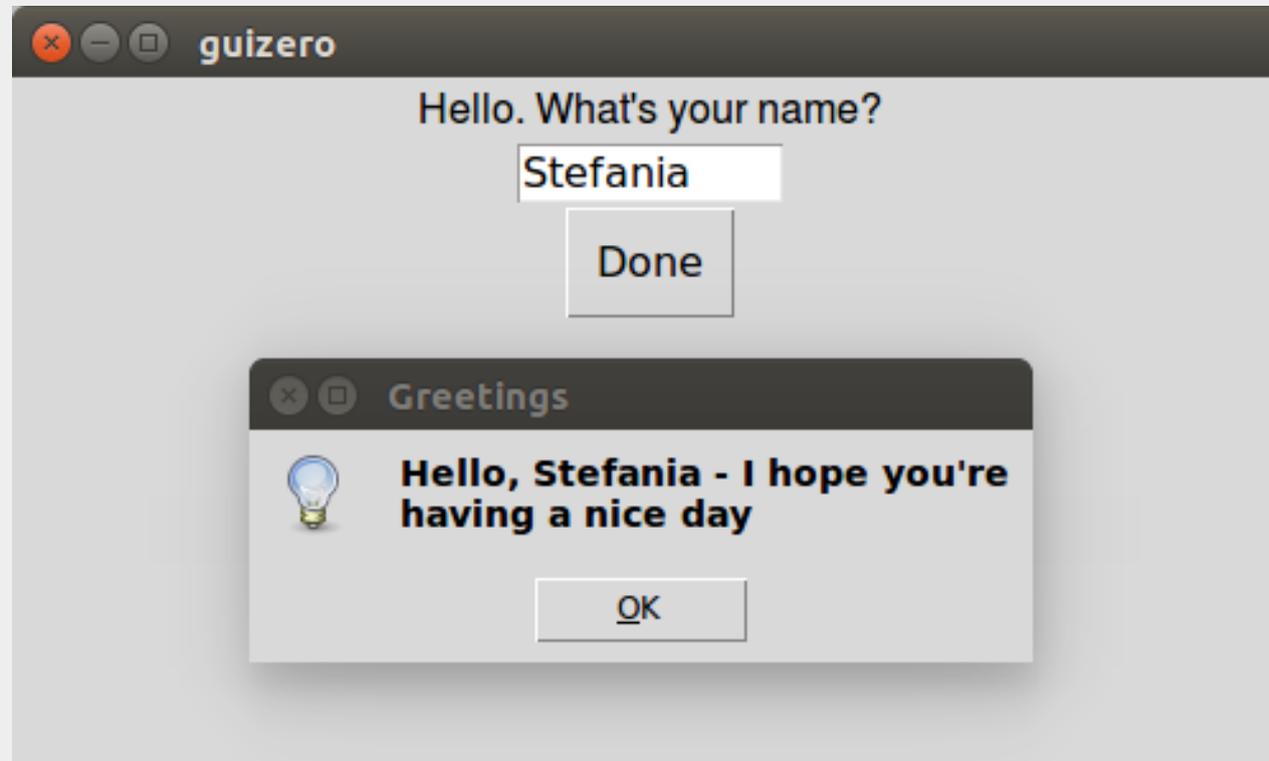
# Your GUI widgets go here
lbl_name = Text(app, text="Hello. What's your name?")
txt_name = TextBox(app)
btn_go = PushButton(app, command=btn_go_clicked, text="Done")

# Show the GUI on the screen
app.display()
|
```

Ln: 18 Col: 0

Esempio di produzione di finestra con un testo, un'area di testo scrivibile, e un pulsante sensibile al clic (e azione associata)

Esempio 2



Esempio di attivazione della GUI dell'esempio 2

Esempio 3: un editor



File Edit Format Run Options Window Help

```
from guizero import App, Text, TextBox, PushButton, Box, Combo, Slider

app = App(title="Text Editor")
app.width=1000
app.height=800

# create a box to house the controls, we want the box to span the entire width of the app
file_controls = Box(app, align="top", width="fill")

# create a box to house the font controls, we want the box to span the entire width of the app
font_controls = Box(app, align="top", width="fill")

# create a Text for the file name
file_name_label = Text(file_controls, text="File Name", width=10, align="left")
# create a TextBox for the file name
file_name = TextBox(file_controls, text="Infinito.txt", width=40, align="left")

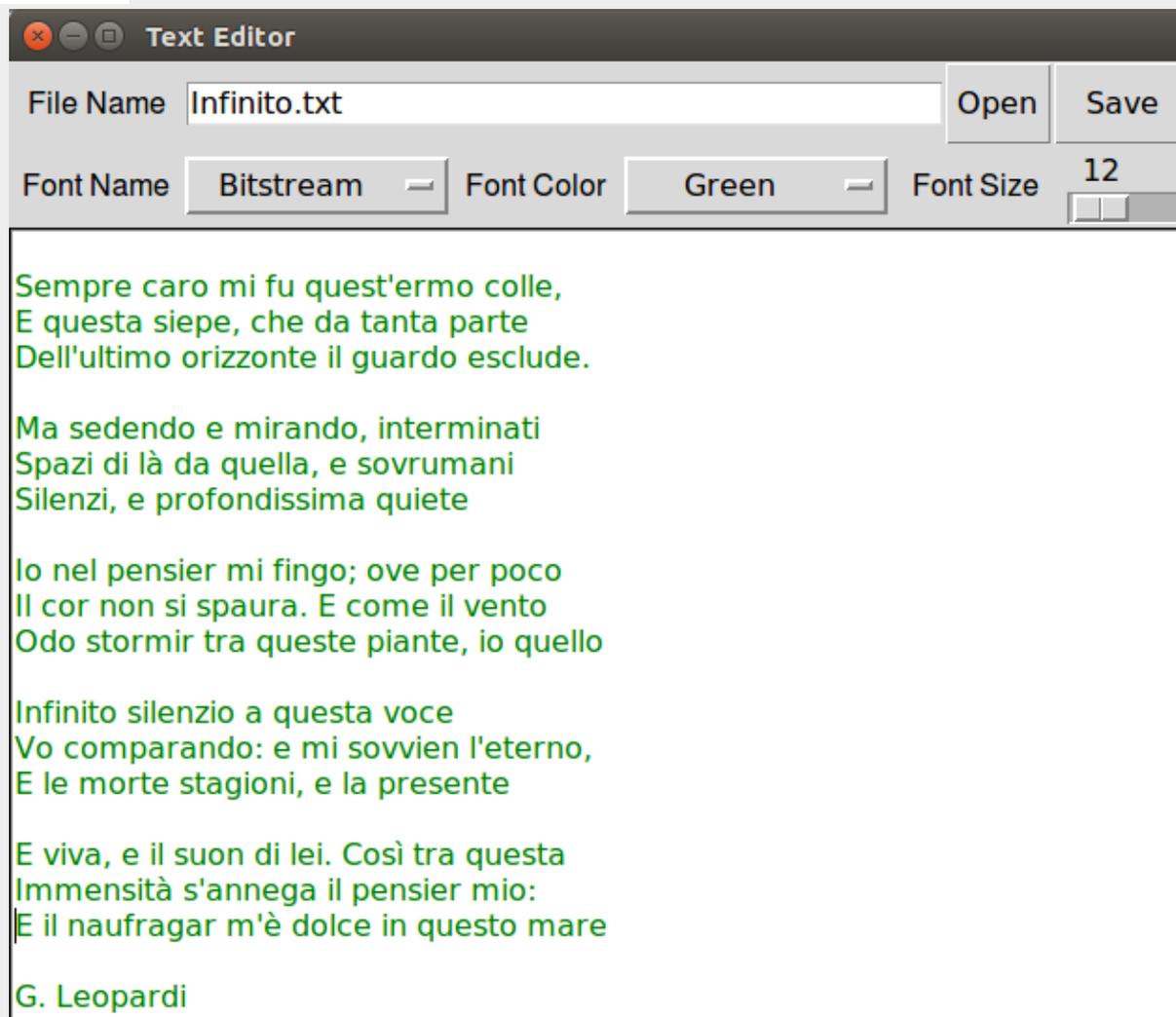
# create a Text for the font name
font_name_label = Text(font_controls, text="Font Name", width=10, align="left")

# function to select font
def select_font():
    editor.font= font_choise.value
# menu to select a font
font_choise = Combo(font_controls, options=["Bitstream", "Courier", "eufm10", "Helvetica", "Purisa", "Times"],
                    selected="Bitstream", command=select_font, width=10, align="left")

# create a Text for the font color
font_color_label = Text(font_controls, text="Font Color", width=10, align="left")
# function to select font color
def select_color():
    editor.text_color= font_color.value
# menu to select a font
font_color = Combo(font_controls, options=["Black", "Blue", "Green", "Grey", "Red", "Violet"],
                    selected="Black", command=select_color, width=10, align="left")
```

Ln: 33 Col: 0

Esempio 3



Esempio di “editor”, legge e salva un testo, cambia font e suoi attributi (70 linee)

Esempio 4 : App + thread



```
File Edit Format Run Options Window Help
from guizero import App, PushButton

def blink():
    if (blinkingbutton.bg == "black"):
        blinkingbutton.bg = "white"
        blinkingbutton.text_color = "black"
    else:
        blinkingbutton.bg = "black"
        blinkingbutton.text_color = "white"

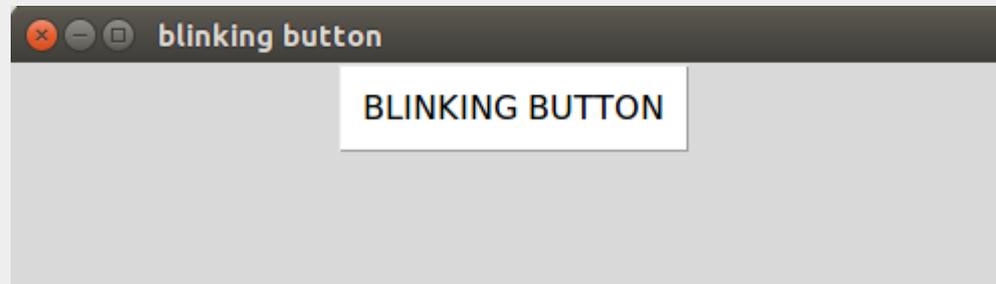
app = App("blinking button")

blinkingbutton = PushButton(app, text="BLINKING BUTTON")
blinkingbutton.bg = "white"
blinkingbutton.text_color = "black"
blinkingbutton.repeat(1000, blink)

app.display()
|
Ln: 19 Col: 0
```

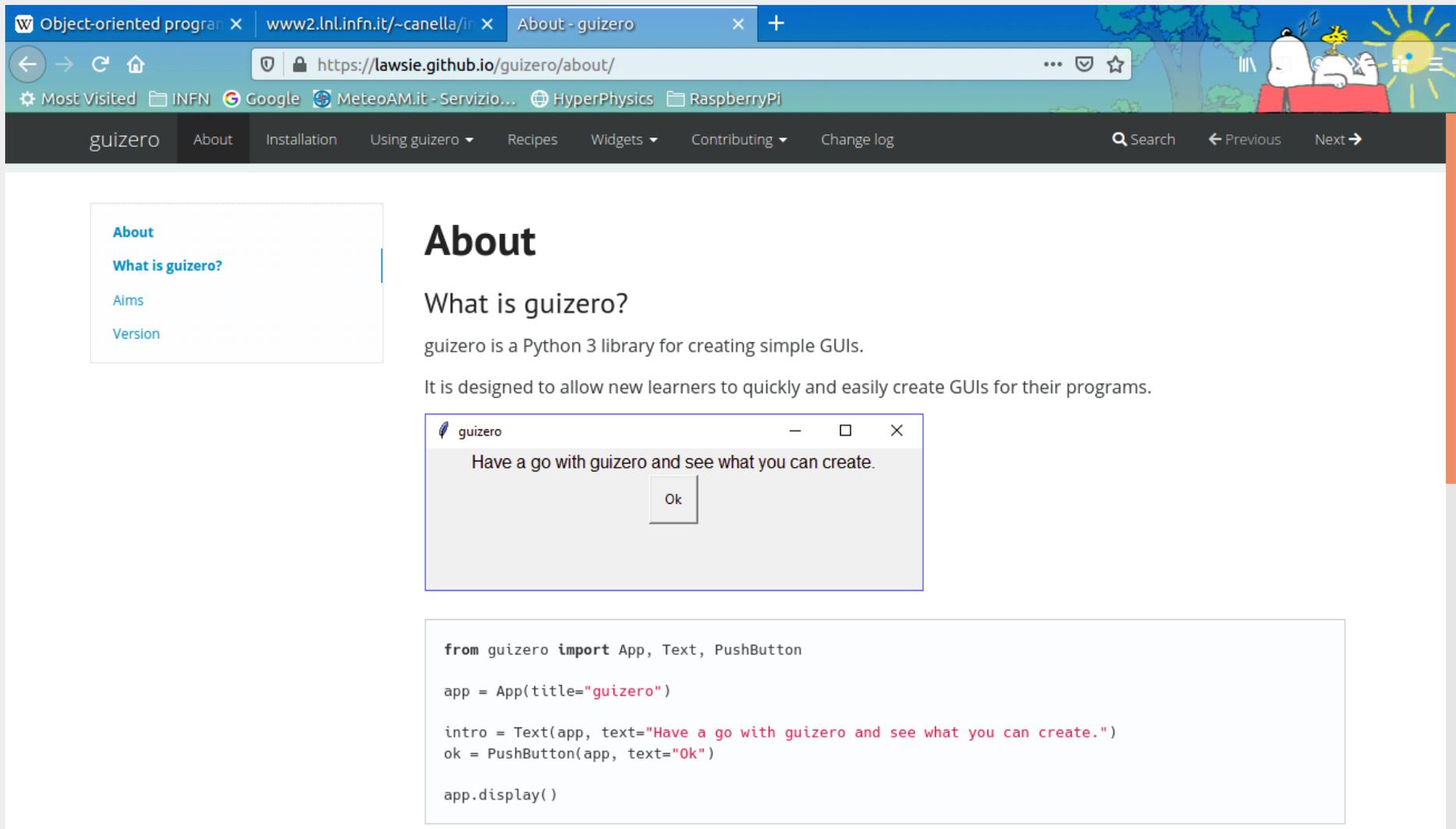
Esempio di “App”, con animazione (blinking), come esempio di “thread”

App + thread



Esempio di “App”, con animazione (blinking), come esempio di “thread”:
Ogni 1000 ms (1 volta al secondo) il pulsante con il testo BLINKING
BUTTON va in negativo e poi in positivo, come indicato nella funzione
blink().

Le thread di questo tipo possono essere usate, ad esempio per
l’aggiornamento continuo di un quadro sinottico.



The screenshot shows a web browser window displaying the 'About' page of the guizero library. The browser's address bar shows the URL `https://lawsie.github.io/guizero/about/`. The website has a dark navigation bar with links for 'guizero', 'About', 'Installation', 'Using guizero', 'Recipes', 'Widgets', 'Contributing', and 'Change log'. A search bar and navigation arrows are also present. On the left, a sidebar menu lists 'About', 'What is guizero?', 'Aims', and 'Version'. The main content area features a large 'About' heading, followed by the question 'What is guizero?'. The text explains that guizero is a Python 3 library for creating simple GUIs, designed for new learners. Below this text is a screenshot of a simple GUI window titled 'guizero' with the message 'Have a go with guizero and see what you can create.' and an 'Ok' button. At the bottom, a code block shows the Python code used to create this GUI.

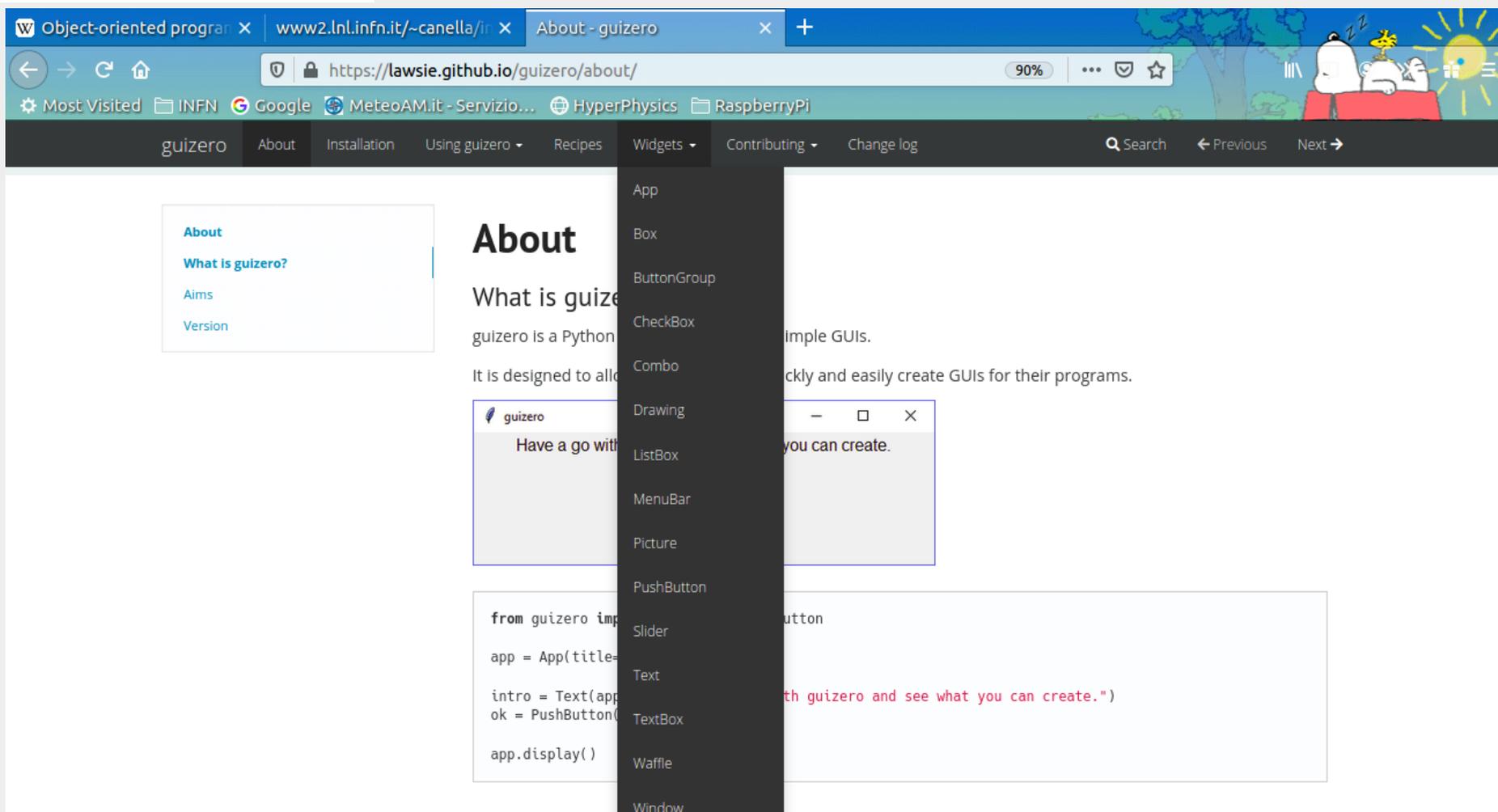
```
from guizero import App, Text, PushButton

app = App(title="guizero")

intro = Text(app, text="Have a go with guizero and see what you can create.")
ok = PushButton(app, text="Ok")

app.display()
```

Widget di GUIZERO



The screenshot shows a web browser displaying the Guizero website. The browser's address bar shows the URL `https://lawsie.github.io/guizero/about/`. The website's navigation menu includes links for `guizero`, `About`, `Installation`, `Using guizero`, `Recipes`, `Widgets`, `Contributing`, and `Change log`. A dropdown menu is open under the `Widgets` link, listing the following widget classes: `App`, `Box`, `ButtonGroup`, `CheckBox`, `Combo`, `Drawing`, `ListBox`, `MenuBar`, `Picture`, `PushButton`, `Slider`, `Text`, `TextBox`, `Waffle`, and `Window`. The main content area of the page is titled `About` and contains the text `What is guizero?` and `guizero is a Python`. A code block is visible, showing the following Python code:

```
from guizero import App, Text, PushButton, Window
app = App(title="Guizero")
intro = Text(app, "Hello, this is a simple GUI with guizero and see what you can create.")
ok = PushButton(app, "OK")
app.display()
```

Le Widget in libreria sono: App, Box, ButtonGroup, CheckBox, Combo, Drawing, ListBox, MenuBar, Picture, PushButton, Slider, Text, TextBox, Waffle, Window

Widget di GUIZERO



Sono caratterizzate da:

- un set di parametri iniziali, con valori di default,
(Starting parameters)
- un set di metodi,
(Methods)
- un set di proprietà, leggibili e scrivibili
(Properties)



Per Finire ...

- Cos'altro si può fare con Python ?
- Gestione di GUI e grafica in generale.
- Tools di comunicazione in rete.
- Tools di supporto alla gestione di siti web.
- Elaborazioni di immagini.