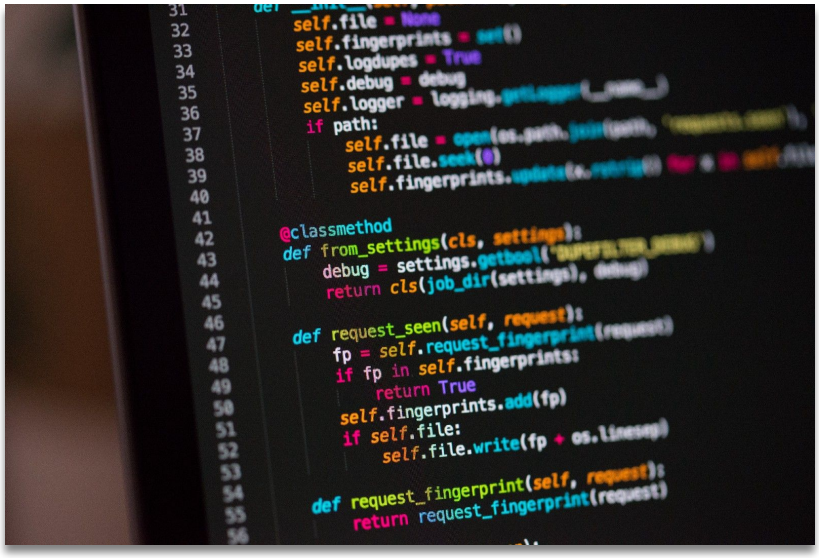
MS003 Programació bàsica



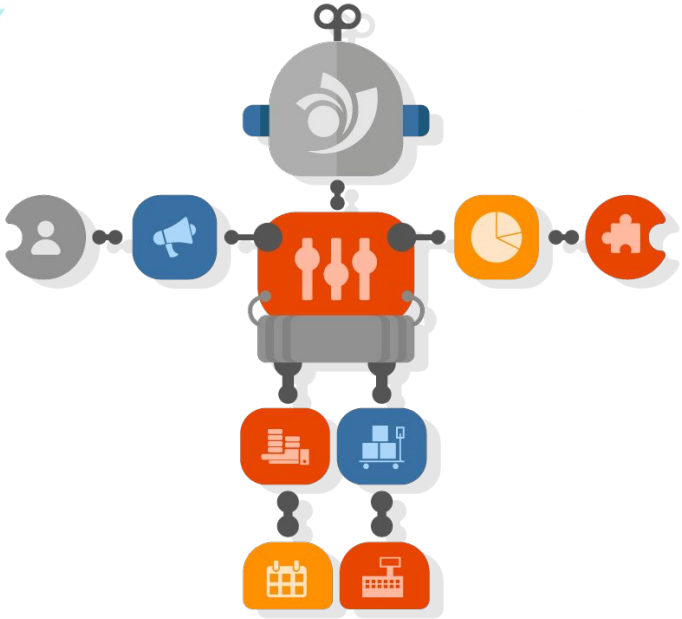
1r ASIX curs 24/25

Durada: 99 hores

5 Resultats d'Aprenentatge

institut sa palomera

Programació estructurada



RA 3. Escriu i prova programes senzills reconeixent i aplicant els fonaments de la programació modular.

Durada: 21 hores aprox.

Continguts treballats

1 Analitza els conceptes relacionats amb la programació modular.

2 Analitza els avantatges i la necessitat de la programació modular.

3 Aplica el concepte d'anàlisi descendent en l'elaboració de programes.

4 Modula correctament els programes realitzats.

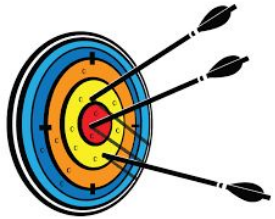
5 Realitza correctament les crides a funcions i la seva parametrització.

6 Té en compte l'àmbit de les variables en les crides a les funcions.

7 Prova, depura, comenta i documenta els programes.

8 Defineix el concepte de llibreries i la seva utilitat.

9 Utilitza llibreries en l'elaboració de programes.



Tkinter

És la llibreria estàndard de Python per crear interfícies gràfiques d'usuari (GUI) amb les següents característiques:

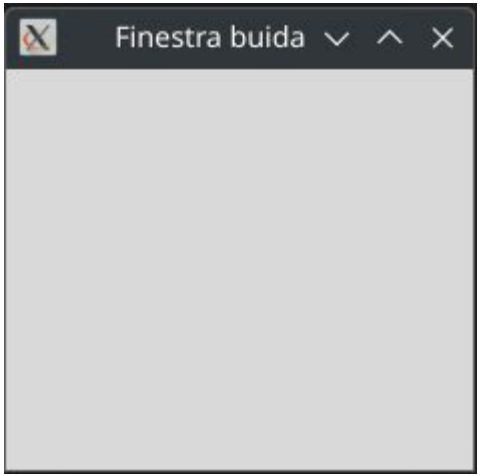
- Simple
- Integrada
- Multiplataforma
- Permet crear tots els elements necessaris:
  - Finestres
  - Botons
  - Etiquetes…

Podeu mirar-vos la documentació per detalls més específics

Tkinter: finestra principal

Sempre cal començar el programa creant la finestra principal:

```
import tkinter as tk

#Crear la finestra principal
root = tk.Tk()
root.title("Finestra buida")

#Mantenir la finestra oberta
# i escoltar esdeveniments
root.mainloop()
```

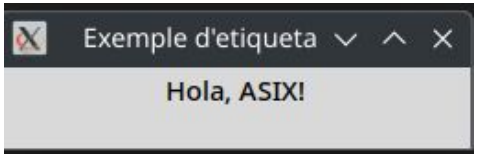


Widgets: etiquetes

```
import tkinter as tk

root = tk.Tk()
root.title("Exemple d'etiqueta")

lbl_greeting = tk.Label(root, text="Hola, ASIX!")
lbl_greeting.pack()

root.mainloop()
```

El nom de la finestra principal

Modifiquem la propietat text



La col·loquem dins la finestra

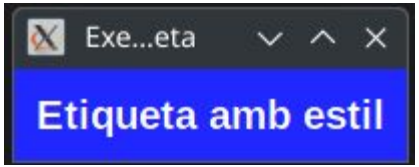
Widgets: etiquetes

Algunes de les propietats més habituals que es modifiquen a les etiquetes són:

- text: el text que mostra l'etiqueta
- font: la tipografia que utilitza (font, mida, estil)
- fg: color de la lletra
- bg: color de fons
- padx: marge intern horitzontal de l'etiqueta
- pady: marge intern vertical de l'etiqueta

Widgets: etiquetes

```
import tkinter as tk

root = tk.Tk()
root.title("Exemple d'etiqueta")

lbl_example = tk.Label(root,
    text="Etiqueta amb estil",
    font=("Arial", 14, "bold"),
    fg="white",
    bg="blue",
    padx=10,
    pady=10)
lbl_example.pack()

root.mainloop()
```



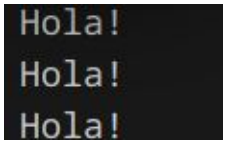
Widgets: etiquetes

Es pot canviar dinàmicament el contingut d'una etiqueta accedint a la seva configuració:

- lbl_nom.config(opcions)
- Habitualment es fa amb una funció que es crida quan passa algun esdeveniment, per exemple, si es clica un botó

Widgets: botons

```
import tkinter as tk

def say_hi():
   print("Hola!")

window = tk.Tk()

#Crear un botó
#Compte! per definir la funció no fem servir els parèntesis!
btn_hi = tk.Button(window, text="Fes clic aquí", command=say_hi)

#Afegir-lo a la finestra
btn_hi.pack()

window.mainloop()
```



Fixeu-vos com, quan passem la funció a cridar, no hi posem els parèntesis!

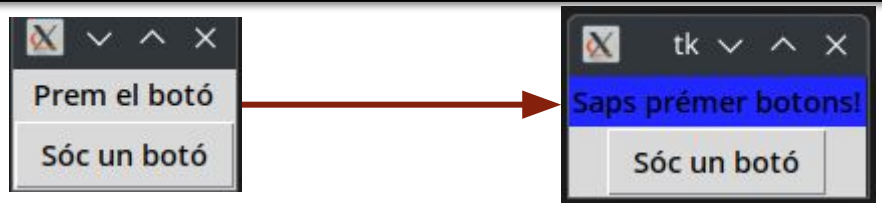
Widgets: botons

Algunes de les propietats més habituals que es modifiquen als botons són

- state: **disabled** desactiva el botó, **active** l'activa
- text: el text que mostra el botó
- font: la tipografia que utilitza (font, mida, estil)
- fg: color de la lletra
- bg: color de fons
- padx: marge intern horitzontal de l'etiqueta
- pady: marge intern vertical de l'etiqueta

Widgets: etiquetes i botons

```
import tkinter as tk

def change_text():
    lbl_message.config(text="Saps prémer botons!", bg="blue")

root = tk.Tk()
lbl_message = tk.Label(root, text="Prem el botó")
lbl_message.pack()

btn_update = tk.Button(root, text="Sóc un botó", command=change_text)
btn_update.pack()

root.mainloop()
```



Tkinter: quadres de text

```
import tkinter as tk

root = tk.Tk()

#Crear un quadre de text
txt_entry = tk.Text(root, height=5, width=40)
txt_entry.insert("1.0","Sóc un text")
txt_entry.pack()

root.mainloop()
```

Quadre de text de 5 files i 40 columnes

Text per defecte

Posició on comença el text

Widgets: quadres de text

Permeten que l'usuari introdueixi text

- Es pot afegir text des del programa amb insert("posició inicial", "text")
- Es pot obtenir el text amb get("posició inicial", "posició final")
  - text = txt_name.get("1.0", tk.END)
    - Agafa des del primer caràcter fins l'últim
- Es pot evitar que l'usuari introdueixi text
  - txt_name.config(state="disabled")

Tkinter: quadres de text

```
import tkinter as tk

root = tk.Tk()

#Crear un quadre de text
txt_entry = tk.Text(root, height=5, width=40)
txt_entry.insert("1.0","Sóc un text")
txt_entry.pack()

root.mainloop()
```

Quadre de text de 5 files i 40 columnes

Text per defecte

Posició on comença el text

Nomenclatura

És molt **recomanable** agafar un estàndard de nomenclatura dels diferents *widgets*, per exemple:

- Les etiquetes sempre siguin lbl_*nom_identificatiu*
- Els botons sempre siguin btn_*nom_identificatiu*
- Els quadres de text sempre siguin txt_*nom_identificatiu*

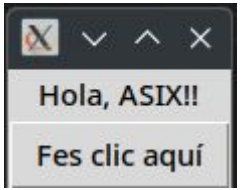
Facilitarà molt la llegibilitat del codi.

Disposició dels elements: pack

pack() és un dels mètodes més simples per organitzar widgetspermet disposar els elements verticalment o horitzontalment.

Pot ser útil en aplicacions molt simples, però és força limitat.

```python
import tkinter as tk

def say_hi():
    print("Hola!")

window = tk.Tk()

lbl_greeting = tk.Label(window, text="Hola, ASIX!!")
lbl_greeting.pack()

btn_hi = tk.Button(window, text="Fes clic aquí", command=say_hi)
btn_hi.pack()

window.mainloop()
```



Disposició dels elements: pack

La sintaxi bàsica és:

- widget.pack(opció1=valor, opció2=valor…)
- Per defecte, col·loca els widgets un sota l'altre, ocupant tot l'ample de la finestra

Disposició dels elements: pack

```
import tkinter as tk

window = tk.Tk()

etiqueta1 = tk.Label(window, text="Etiqueta 1", bg="red")
etiqueta2 = tk.Label(window, text="Etiqueta 2", bg="green")
etiqueta3 = tk.Label(window, text="Etiqueta 3", bg="blue")

etiqueta1.pack()
etiqueta2.pack()
etiqueta3.pack()

window.mainloop()
```

Disposició dels elements: pack

- Es pot indicar en quin costat es vol el widget amb l'opció side:
  - top (per defecte)
  - bottom
  - left
  - right

Disposició dels elements: pack

```
import tkinter as tk

window = tk.Tk()

etiqueta1 = tk.Label(window, text="TOP", bg="red")
etiqueta2 = tk.Label(window, text="BOTTOM", bg="green")
etiqueta3 = tk.Label(window, text="LEFT", bg="blue")
etiqueta4 = tk.Label(window, text="RIGHT", bg="yellow")

etiqueta1.pack(side="top")
etiqueta2.pack(side="bottom")
etiqueta3.pack(side="left")
etiqueta4.pack(side="right")

window.mainloop()
```

Disposició dels elements: pack

- Es pot establir espai entre els widgets amb padx (espai horitzontal) i pady (espai vertical)
- Es pot controlar l'expansió amb l'opció fill (x per horitzontal, y per vertical, both per tots dos)
- Amb la opció expand=True, el widget agafarà tot l'espai disponible

Disposició dels elements: pack

```
import tkinter as tk

window = tk.Tk()

etiqueta1 = tk.Label(window, text="Expand X", bg="red")
etiqueta2 = tk.Label(window, text="Expand Y", bg="green")
etiqueta3 = tk.Label(window, text="Expand Both", bg="blue")

etiqueta1.pack(fill="x")  # Omple tot l'ample
etiqueta2.pack(fill="y", expand=True)  # Omple tota l'altura
etiqueta3.pack(fill="both", expand=True)  # Omple tot l'espai
disponible

window.mainloop()
```

Sense pad

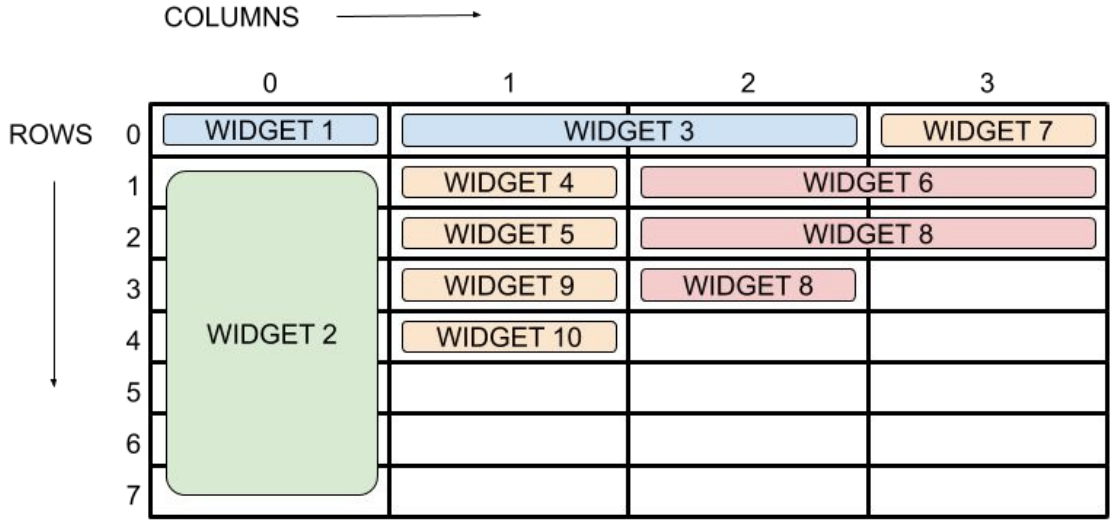
Amb pad

Disposició dels elements: pack

Disposició dels elements: grid

grid() és un sistema de distribució més potent que pack()

Permet posar els widgets en una quadrícula amb files (rows) i columnes (columns)



Disposició dels elements: grid

La sintaxi bàsica és:

- widget.grid(row=n, column=m)
- Tant les files com les columnes comencen per 0
- Atenció! Si no feu una mica de planificació prèvia, haureu de canviar els nombres moltes vegades!
  - Una bona eina és un full de càlcul!

	0	1
0	lbl_titol	
1	lbl_name	
2	txt_name	
3	lbl_age	txt_age
4	lbl_studies	drp_studies
5	btn_accept	
6		

Disposició dels elements: grid

```
import tkinter as tk

window = tk.Tk()

etiqueta1 = tk.Label(window, text="Etiqueta 1", bg="red")
etiqueta2 = tk.Label(window, text="Etiqueta 2", bg="green")
etiqueta3 = tk.Label(window, text="Etiqueta 3", bg="blue")

etiqueta1.grid(row=0, column=0)
etiqueta2.grid(row=0, column=1)
etiqueta3.grid(row=1, column=0)

window.mainloop()
```

Disposició dels elements: grid

- Es poden expandir cel·les perquè ocupin més d'una fila o columna:
  - columnspan=n expandeix un widget en diverses columnes
  - rowspan=n expandeix un widget en diverses files
- També es pot utilitzar padx i pady, com en el cas de pack()

Disposició dels elements: grid


```python
import tkinter as tk

window = tk.Tk()

etiqueta1 = tk.Label(window, text="Etiqueta 1", bg="red")
etiqueta2 = tk.Label(window, text="Etiqueta 2", bg="green")
etiqueta3 = tk.Label(window, text="Etiqueta 3 (ocupa 2 columnes)", bg="blue")

etiqueta1.grid(row=0, column=0)
etiqueta2.grid(row=0, column=1)
etiqueta3.grid(row=1, column=0, columnspan=2)  # Ocupa dues columnes

window.mainloop()
```

Interactivitat


```python
import tkinter as tk

window = tk.Tk()

etiqueta1 = tk.Label(window, text="Etiqueta 1", bg="red")
etiqueta2 = tk.Label(window, text="Etiqueta 2", bg="green")
etiqueta3 = tk.Label(window, text="Etiqueta 3 (ocupa 2 columnes)", bg="blue")

etiqueta1.grid(row=0, column=0)
etiqueta2.grid(row=0, column=1)
etiqueta3.grid(row=1, column=0, columnspan=2)  # Ocupa dues columnes

window.mainloop()
```

Exercicis

Teniu alguns exercicis d'aquest tema a l'AEA3.

Recordeu que cada setmana s'han d'entregar els obligatoris!