



สร้างแอปพลิเคชันด้วย Python  
Tkinter (GUI Application)

**Tkinter คืออะไร**

# Tkinter คืออะไร



Tkinter หรือ TK Interface เป็นไลบรารี  
สำหรับการพัฒนา GUI Application  
(GUI : Graphic User Interface) ที่ติดมา  
กับภาษา Python (Standard Library)

ต้องมีพื้นฐานอะไรบ้าง

# ต้องมีพื้นฐานอะไรบ้าง

- พื้นฐานการเขียนโปรแกรมภาษา Python
- เข้าใจโครงสร้างข้อมูลในภาษา Python (List , Dictionary)
- พื้นฐานการเขียนโปรแกรมเชิงวัตถุด้วยภาษา Python (Class , Method , Object)

# เครื่องมือ

- Python
- Visual Studio Code





ติดตั้ง Python



คำสั่งสำหรับตรวจสอบเวอร์ชัน

```
python --version
```





ติดตั้ง Visual Studio Code

# สร้างหน้าจอ GUI ด้วย Tk

# Widget คืออะไร

แนวคิดในการแบ่ง User Interface ออกเป็นชิ้นส่วนต่างๆ แล้วนำมาแสดงผลบนหน้าจอ เราจะเรียกชิ้นส่วนนี้ว่า Widget เช่น ปุ่ม ข้อความ เป็นต้น

# Widget คืออะไร

Widget แต่ละตัวจะมีส่วนที่เรียกว่า **Options** หรือ **Properties** สำหรับกำหนดคุณสมบัติให้กับ Widget นั้นๆ

Widget แต่ละตัวก็มีคุณสมบัติที่หลากหลายให้ใช้งานแตกต่างกันออกไป เช่น กำหนดสี เส้นขอบ พื้นหลัง เป็นต้น

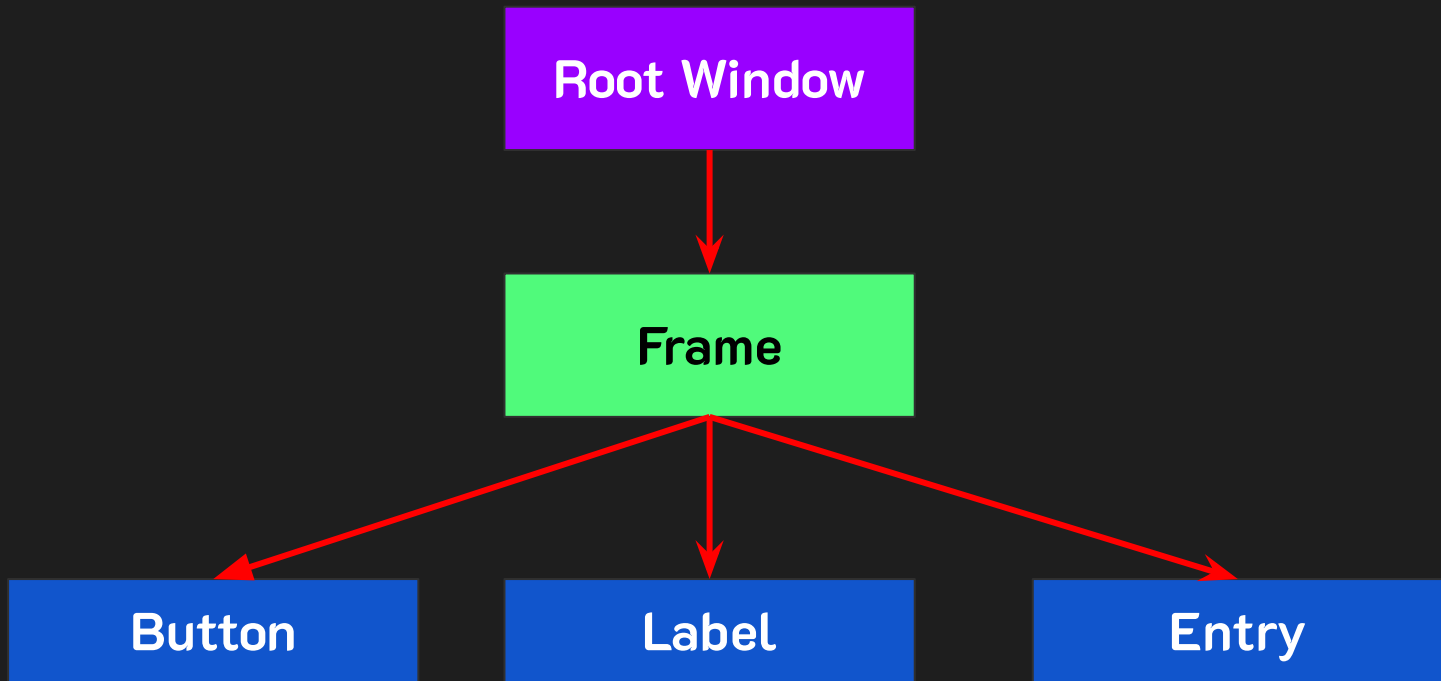
# Widget พื้นฐาน

- Label Widget สำหรับแสดงข้อความ หรือ รูปภาพ
- Button Widget สำหรับจัดการปุ่ม
- Entry Widget สำหรับสร้างกล่องรับข้อความ
- Frame Widget กล่องที่รวบรวม Widget ต่างๆ

# ลำดับขั้นตอนการสร้างหน้าจอ

การสร้างหน้าจอ GUI ให้ปรากฏขึ้นมานั้นจะต้อง  
เริ่มต้นสร้างหน้าต่างหลัก (Root Window) ขึ้นมาก่อน  
และทำการสร้าง Frame หรือ Canvas เพื่อจัดกลุ่มหรือ  
ใช้บรรจุ Widget ต่างๆลงไปในหน้าจอ สำหรับควบคุม  
การทำงานของ Widget

# ลำดับชั้นการสร้างหน้าจอ



ตั้งค่าหน้าต่างหลัก  
(Root Window)



# การตั้งค่าหน้าต่างหลัก

- **title** คือ ตั้งค่าข้อความที่ Title Bar
- **iconbitmap** คือ ตั้งค่าไอคอน
- **geometry** คือ ตั้งค่าขนาดและตำแหน่งหน้าจอ
- **resizable** คือ ตั้งค่าการเปลี่ยนขนาดหน้าจอ
- **bg** คือ กำหนดสีพื้นหลังหน้าจอ

# การตั้งค่าไอคอน

```
root.iconbitmap("ตำแหน่งไอคอน")
```

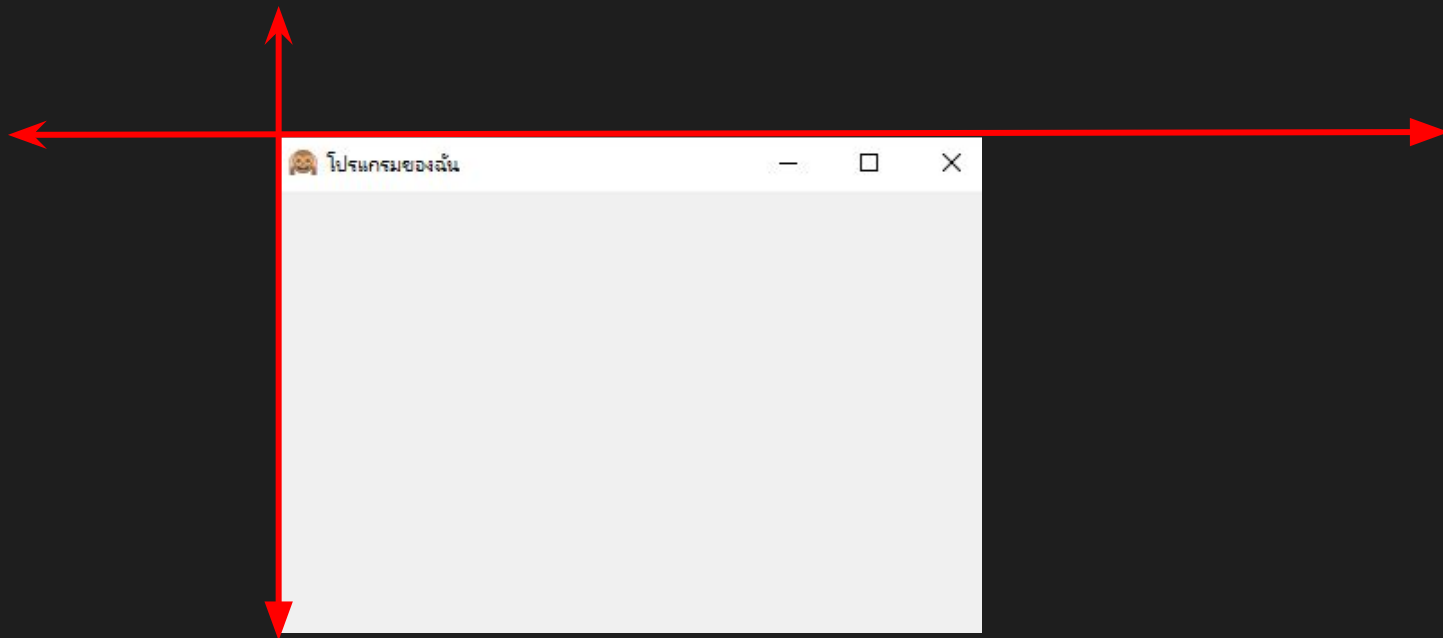
```
root.iconbitmap("icons/logo.ico")
```

# การตั้งค่าขนาดและตำแหน่งหน้าจอ

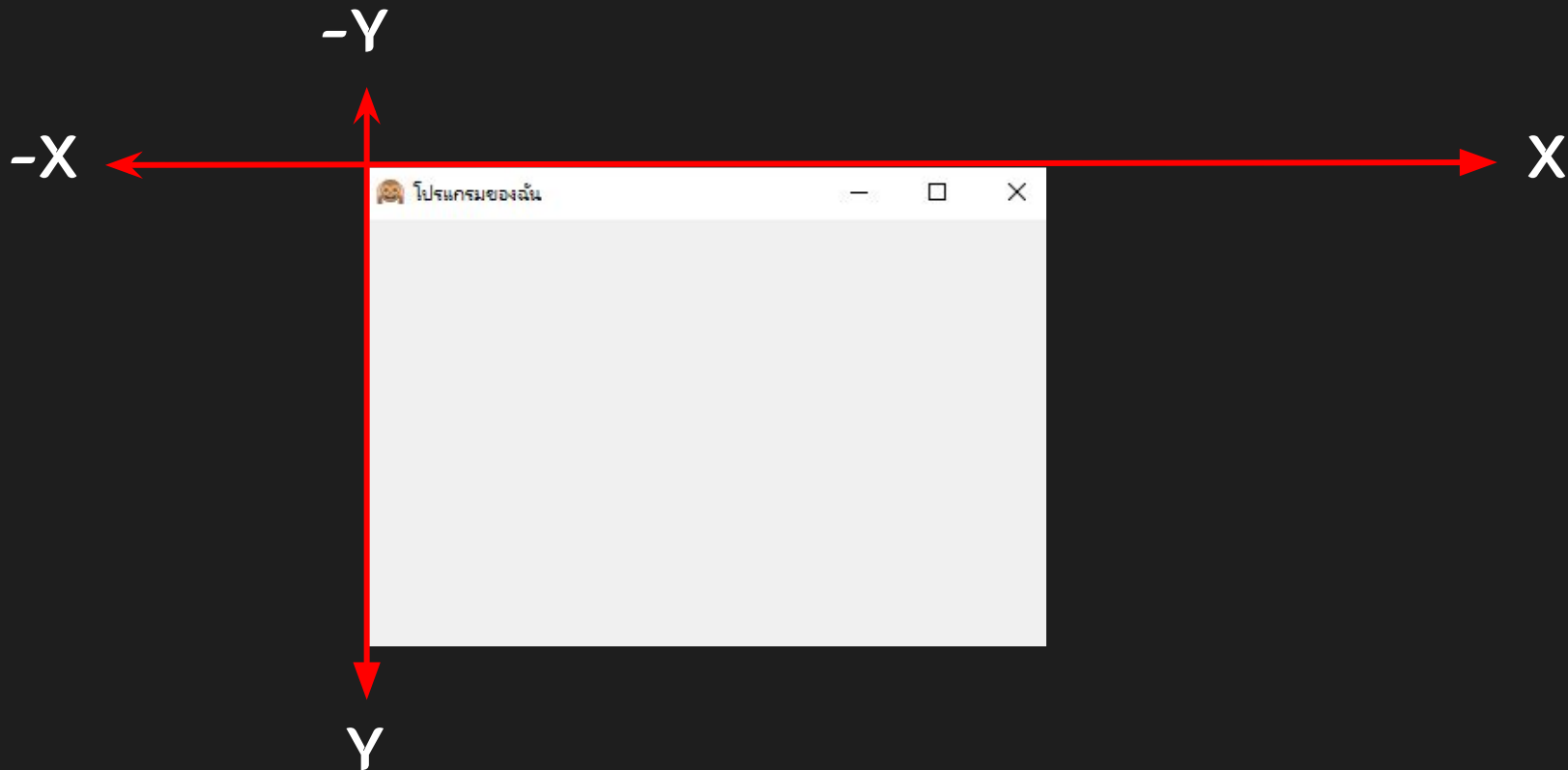
```
root.geometry("width x height + x_pos + y_pos")
```

- width คือ ขนาดความกว้างหน้าจอ
- height คือ ขนาดความสูงหน้าจอ
- x\_pos คือ ตำแหน่งพิกัดแกน x
- y\_pos คือ ตำแหน่งพิกัดแกน y

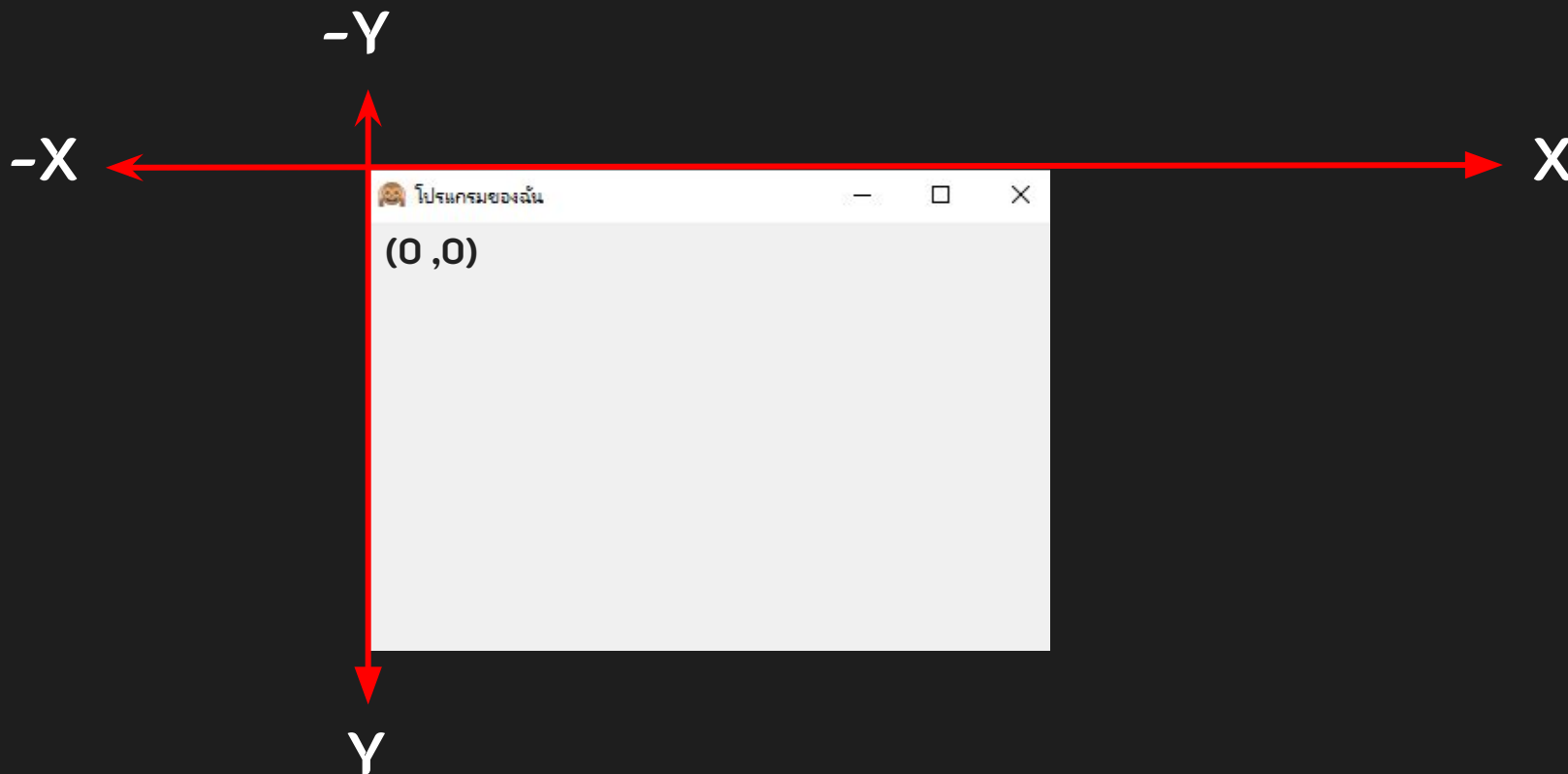
# ระบบพิกัดในคอมพิวเตอร์กราฟิก



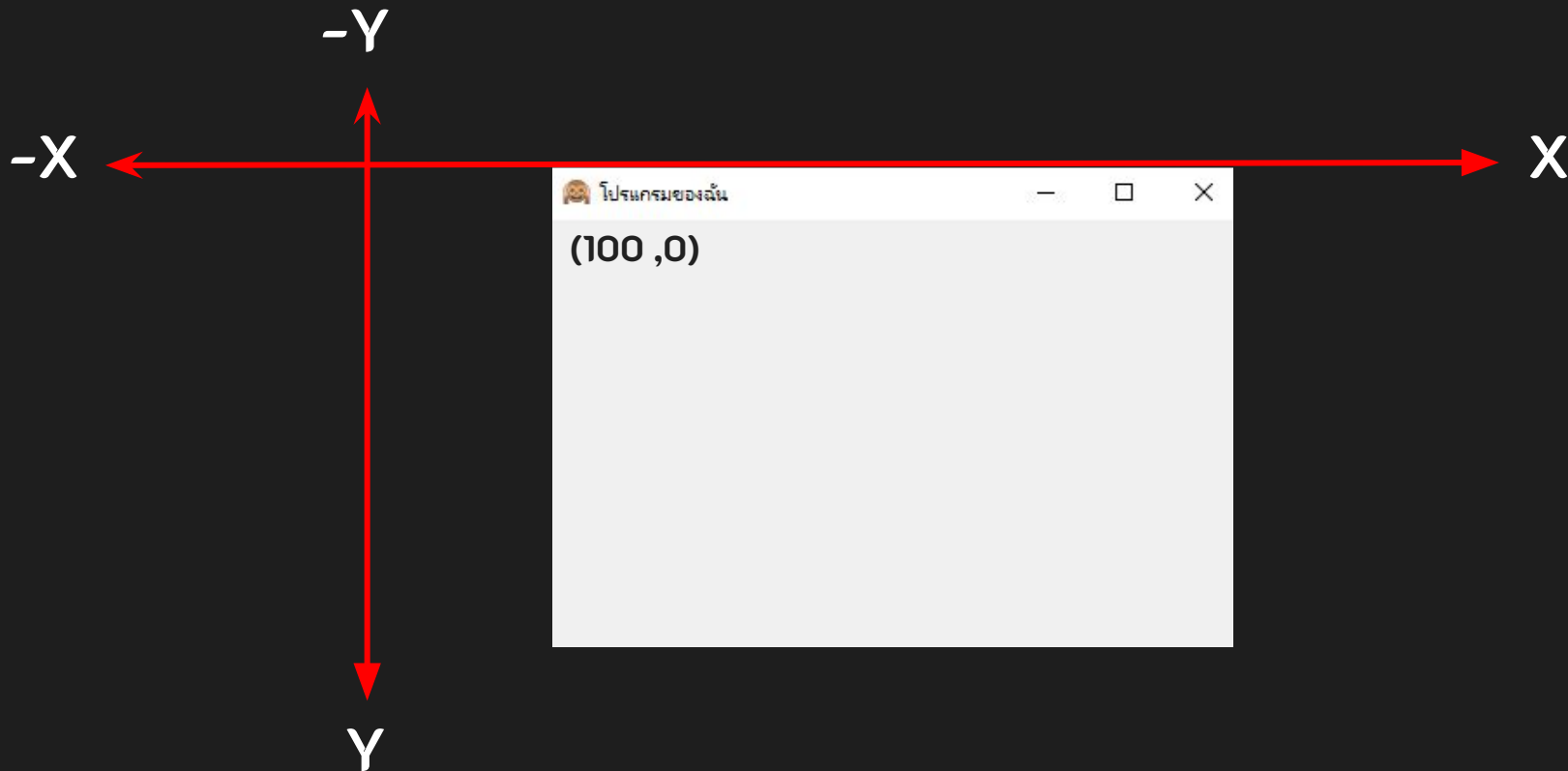
# ระบบพิกัดในคอมพิวเตอร์กราฟิก



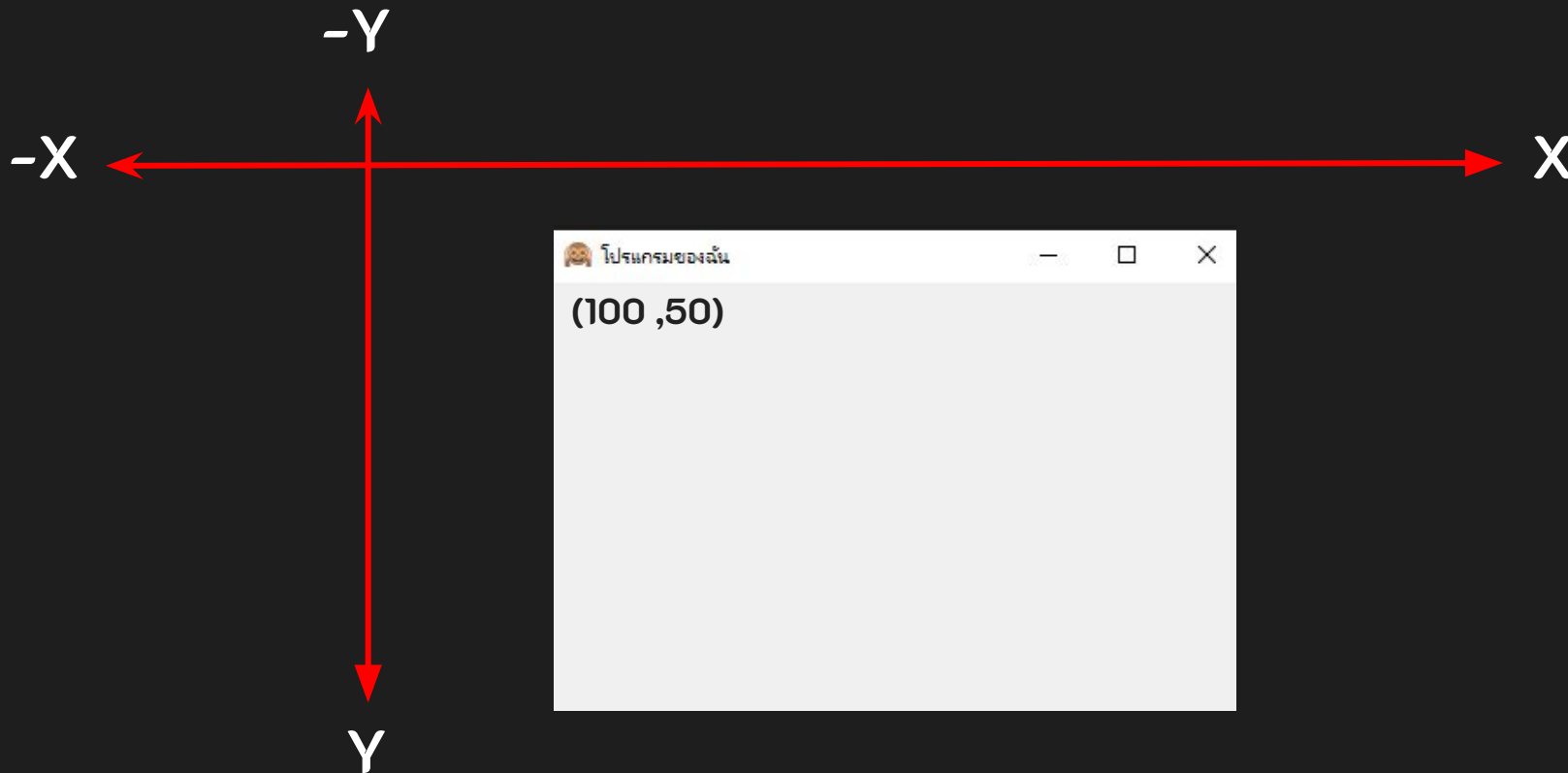
# ระบบพิกัดในคอมพิวเตอร์กราฟิก



# ระบบพิกัดในคอมพิวเตอร์กราฟิก



# ระบบพิกัดในคอมพิวเตอร์กราฟิก





# กำหนดสีพื้นหลัง

```
root.config(bg="ค่าสี")
```

## รูปแบบการกำหนดสีให้ Widget

- กำหนดค่าสีโดยใช้เลขฐาน 16 เช่น #fff = สีขาว, #000 = สีดำ
- กำหนดค่าสีแบบข้อความ เช่น red = สีแดง , blue = สีน้ำเงิน

# แสดงข้อความ (Label Widget)

# Widget พื้นฐาน

- Label Widget สำหรับแสดงข้อความ หรือ รูปภาพ
- Button Widget สำหรับจัดการปุ่ม
- Entry Widget สำหรับสร้างกล่องรับข้อความ
- Frame Widget กล่องที่รวบรวม Widget ต่างๆ

# ตัวอย่างคุณสมบัติใน Widget

- **bg (Background)** คือ กำหนดพื้นหลัง
- **fg (Foreground)** คือ กำหนดสีด้านหน้า
- **activebackground** คือ กำหนดสีพื้นหลังของ Widgets เมื่อ Widgets ทำงาน (Active)
- **font** คือ กำหนดฟอนต์ให้กับ Widget

# กำหนดฟอนต์ (Font)

`font=(name,size,style)`

- name คือ ชื่อฟอนต์ (Font Family)
- size คือ ขนาด
- style คือ รูปแบบฟอนต์ เช่น ปกติ , ตัวหนา , ตัวเอียง

# Pack (Layout)

คือ รูปแบบการจัดวางตำแหน่งของ Widget ในพื้นที่ Parent (Root Window , Frame) โดยแบ่งพื้นที่การจัดวางเท่ากันและซ้อนทับกันไปเรื่อยๆ โดยตำแหน่งเริ่มต้นจะอยู่ที่กึ่งกลางหน้าจอและเรียงลำดับ Widget จากบนลงล่าง

จัดวาง Widget ด้วย Pack

# Pack (Layout)

คือ รูปแบบการจัดวางตำแหน่งของ Widget ในพื้นที่ Parent (Root Window , Frame) โดยแบ่งพื้นที่การจัดวางเท่ากันและซ้อนทับกันไปเรื่อยๆ โดยตำแหน่งเริ่มต้นจะอยู่ที่กึ่งกลางหน้าจอและเรียงลำดับ Widget จากบนลงล่าง



# ลักษณะการจัดวาง (Options)

Options	ความหมาย
padx (LEFT , RIGHT)	กำหนดระยะห่างในแนวนอน x (แนวนอน) กับ Widget ที่อยู่ติดกัน
pady (TOP , BOTTOM)	กำหนดระยะห่างในแนวแกน y (แนวตั้ง) กับ Widget ที่อยู่ติดกัน
ipadx (LEFT , RIGHT)	กำหนดระยะห่างในแนวนอน x ระหว่างเนื้อหา กับ ขอบของ Widget
ipady (TOP , BOTTOM)	กำหนดระยะห่างในแนวแกน y ระหว่างเนื้อหา กับ ขอบของ Widget

# ลักษณะการจัดวาง (Options)

Options	ความหมาย
fill	<p>หากมีพื้นที่ว่างให้ยืดขนาด Widget ออกไปจนเต็มพื้นที่ดังกล่าว</p> <ul style="list-style-type: none"><li>• x คือ ยืดขนาดแนวนอน x</li><li>• y คือ ยืดขนาดแนวนอน y</li><li>• both คือ ยืดขนาดทั้งแนวนอน x และ y</li></ul>
expand	<p>หากมีการขยายหน้าต่างจะกระจายตำแหน่ง Widget ให้พอดีกับพื้นที่โดยอัตโนมัติ</p>

# ลักษณะการจัดวาง (Options)

Options	ความหมาย
Side	<p>กำหนดทิศทางการจัดเรียง Widget ว่าจะให้เริ่มต้นที่ด้านใด</p> <ul style="list-style-type: none"><li>• TOP คือ ด้านบน</li><li>• BOTTOM คือ ด้านล่าง</li><li>• LEFT คือ ด้านซ้าย</li><li>• RIGHT คือ ด้านขวา</li></ul>

จัดวาง Widget ด้วย Grid

# Grid (Layout)

คือ รูปแบบการจัดวางตำแหน่งของ Widget ในพื้นที่ แบบตารางที่มีลักษณะเป็นแบบแถว (แนวนอน) และ แบบคอลัมน์ (แนวตั้ง) โดยวาง Widget ลงไปในแต่ละช่องซึ่งมีเลขลำดับช่องเริ่มต้นที่ 0

# ลักษณะการจัดวาง (Options)

Options	ความหมาย
row	ลำดับแถวที่จะวาง Widget ลงไปทำงาน ในแนวนอน (เริ่มต้นที่ 0)
column	ลำดับคอลัมน์ที่จะวาง Widget ลงไปทำ งานในแนวตั้ง (เริ่มต้นที่ 0)

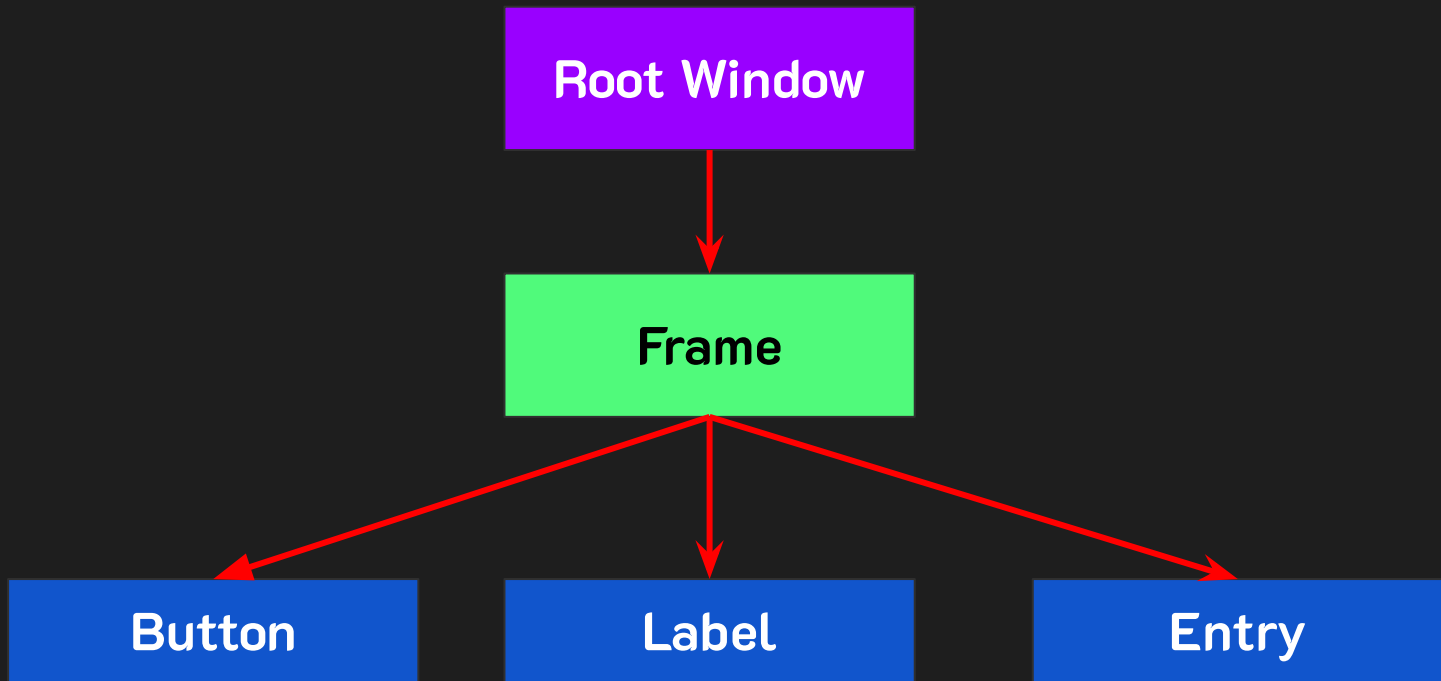
# ลักษณะการจัดวาง (Options)

Options	ความหมาย
padx (LEFT , RIGHT)	กำหนดระยะห่างในแนวนอน x (แนวนอน) กับ Widget ที่อยู่ติดกัน
pady (TOP , BOTTOM)	กำหนดระยะห่างในแนวแกน y (แนวตั้ง) กับ Widget ที่อยู่ติดกัน
ipadx (LEFT , RIGHT)	กำหนดระยะห่างในแนวนอน x ระหว่างเนื้อหา กับ ขอบของ Widget
ipady (TOP , BOTTOM)	กำหนดระยะห่างในแนวแกน y ระหว่างเนื้อหา กับ ขอบของ Widget

# การสร้างเฟรม (Frame)



# ลำดับชั้นการสร้างหน้าจอ



# คุณสมบัติของ Frame

- เพื่อสร้างกรอบแยกหน้าต่างโปรแกรมออกเป็นส่วนๆ
- เพื่อจัดกลุ่ม Widget ก่อนที่จะนำไปแสดงผลที่หน้าจอหลัก (Root Window)

# การสร้างเฟรม (Frame)

`Frame(root,options)`

- root คือ หน้าต่างหลัก
- options คือ คุณสมบัติต่างๆที่จะกำหนดให้ Frame

# Entry Widget

# Widget พื้นฐาน

- Label Widget สำหรับแสดงข้อความ หรือ รูปภาพ
- Button Widget สำหรับจัดการปุ่ม
- Entry Widget สำหรับสร้างกล่องรับข้อความ
- Frame Widget กล่องที่รวบรวม Widget ต่างๆ

# Entry Widget Method

- **delete(first , last)** ทำหน้าที่ในการลบตัวอักษรออกจาก Entry โดย first คือตำแหน่งตัวอักษรเริ่มต้น และ last ตำแหน่งตัวอักษรสุดท้าย
- **get()** ทำหน้าที่ดึงข้อมูลจาก Entry มาใช้งานในรูปแบบชุดข้อความ (String)

# Command Callback

# Command Callback

- รูปแบบการตอบสนองต่อเหตุการณ์ (Event) ที่เกิดขึ้นภายในโปรแกรม เช่น การเลือกตัวเลือก , การกดปุ่ม เป็นต้น

- `command = functionName`



**Text Variable**

# Text Variable

Text Variable คือ กำหนดตัวแปรสำหรับใช้เก็บข้อความ โดยทำงานร่วมกับคลาส `StringVar()`

- เมธอด `get()` ทำหน้าที่ดึงข้อมูลจากตัวแปรคลาส `StringVar()`
- เมธอด `set()` ทำหน้าที่กำหนดข้อความใหม่

# Message Box

# Message Box

คือ Widget ที่มีลักษณะเป็นกล่องข้อความแจ้งเตือน  
เพื่ออธิบายการทำงานบางอย่างภายในโปรแกรมและ  
เป็นข้อความที่ไม่สามารถแก้ไขได้

# Radio Button

# Radio Button

คือ Widget ที่มีลักษณะเป็นปุ่มกลมมีช่องว่าง  
อยู่ภายใน ใช้สำหรับสร้างเป็นกลุ่มของตัวเลือก  
เพื่อให้ผู้ใช้งานโปรแกรมสามารถเลือกตัวเลือกได้  
(รายการเดียวเท่านั้น)

# Radio Button

Options	ความหมาย
variable	ตัวแปรที่เก็บข้อมูลตัวเลือกสามารถเก็บค่าเป็น StringVar (ข้อความ) หรือ IntVar (ตัวเลข) ได้
value	กำหนดค่าตัวเลือกใน Radio Button

รูปภาพ (Image)



# ติดตั้ง Package จัดการรูปภาพ

- `pip install pillow`

# ทำงานกับรูปภาพ

- `from PIL import ImageTk, Image`
- `ImageTk.PhotoImage(Image.open("ตำแหน่งภาพ"))`

# Widget พื้นฐาน

- Label Widget สำหรับแสดงข้อความ หรือ รูปภาพ
- Button Widget สำหรับจัดการปุ่ม
- Entry Widget สำหรับสร้างกล่องรับข้อความ
- Frame Widget กล่องที่รวบรวม Widget ต่างๆ