# แผนบริหารการสอนประจำบทที่ 10

# ส่วนต่อประสานกราฟิกกับผู้ใช้

### หัวข้อประจำบท

- 1. เบื้องต้นเกี่ยวกับส่วนต่อประสานกราฟิกกับผู้ใช้
  - 1.1 การออกแบบส่วนต่อประสานกราฟิกกับผู้ใช้
  - 1.2 การสร้างโปรแกรมส่วนต่อประสานกราฟิกกับผู้ใช้
- 2. องค์ประกอบของส่วนต่อประสานกราฟิกกับผู้ใช้ tkinter
- 3. การสร้างหน้าต่างของโปรแกรม
- 4. การสร้างวัตถุบนหน้าต่างโปรแกรมด้วยวิดเจ็ต
  - 4.1 Label
  - 4.2 Entry
  - 4.3 Text
  - 4.4 Button
  - 4.5 Checkbutton
  - 4.6 Listbox
  - 4.7 Menubutton
  - 4.8 Menu
  - 4.9 Radiobutton
  - 4.10 Message
- 5. การจัดวางตำแหน่งวิดเจ็ตด้วย Geometry Management
  - 5.1 pack()
  - 5.2 grid()
  - 5.3 place()
- 6. การกำหนดเหตุการณ์ด้วย Event และ Callback
- 7. การประยุกต์เขียนโปรแกรมส่วนต่อประสานกราฟิกกับผู้ใช้

### วัตถุประสงค์เชิงพฤติกรรม

- 1. เพื่อให้ผู้เรียนสามารถอธิบายเกี่ยวกับเบื้องต้นเกี่ยวกับส่วนต่อประสานกราฟิกกับผู้ใช้ได้
- 2. เพื่อให้ผู้เรียนสามารถอธิบายเกี่ยวกับองค์ประกอบส่วนต่อประสานกราฟิกกับผู้ใช้ได้
- 3. เพื่อให้ผู้เรียนสามารถอธิบายเกี่ยวกับการสร้างหน้าต่างของโปรแกรมได้
- 4. เพื่อให้ผู้เรียนสามารถอธิบายเกี่ยวกับการสร้างวัตถุบนหน้าต่างโปรแกรมด้วยวิดเจ็ตได้
- 5. เพื่อให้ผู้เรียนสามารถอธิบายเกี่ยวกับการกำหนดเหตุการณ์ด้วย Event และ Callback
- 6. เพื่อให้ผู้เรียนสามารถอธิบายเกี่ยวกับการจัดวางตำแหน่งวิดเจ็ตด้วย Geometry

Management ได้

 เพื่อให้ผู้เรียนสามารถนำความรู้เรื่อง ประยุกต์เขียนโปรแกรมส่วนต่อประสานกราฟิกกับ ผู้ใช้ ประยุกต์ใช้งานในการแก้โจทย์ปัญหาได้อย่างถูกต้อง

# วิธีการสอนและกิจกรรม

 ผู้สอนบรรยายในชั้นเรียน และโปรแกรมนำเสนอ ตามหัวข้อเนื้อหาในเอกสาร ประกอบการสอน

2. ผู้เรียนทำแบบฝึกหัดท้ายบท และฝึกเขียนโปรแกรมด้วยคอมพิวเตอร์

# สื่อการเรียนการสอน

- 1. เอกสารประกอบการสอน วิชา หลักการเขียนโปรแกรมคอมพิวเตอร์
- 2. โปรแกรม Python รุ่น 3.8.10
- 3. โปรแกรม PowerPoint

# การวัดผลและการประเมินผล

- 1. สังเกตจากการอภิปราย การตอบคำถาม และซักถามระหว่างเรียน
- 2. การปฏิบัติบนเครื่องคอมพิวเตอร์
- 3. การทำแบบฝึกหัดท้ายบท

# บทที่ 10 ส่วนต่อประสานกราฟิกกับผู้ใช้

การใช้งานโปรแกรมตัวอย่างโปรแกรมในบทที่ผ่านมา พบว่าการติดต่อกับโปรแกรมจะเป็น ลักษณะของข้อความ โปรแกรมจะแสดงข้อความดำเนินการอย่างใดอย่างหนึ่ง โดยให้ผู้ใช้โปรแกรม ทำการกรอกข้อมูลลงไปในส่วนที่รับข้อมูล โดยข้อมูลที่ได้รับจะถูกนำไปดำเนินการต่อและแสดง ผลลัพธ์ที่ได้กลับมายังผู้ใช้โปรแกรม ซึ่งลักษณะของการใช้งานดังกล่าวเป็นการใช้งานโปรแกรม เบื้องต้นที่มีลักษณะของการนำไปใช้งานพื้นฐาน แต่ในปัจจุบันการใช้งานโปรแกรมจะมีลักษณะการ ใช้งานอุปกรณ์ที่นอกเหนือรับข้อมูลจากแผงแป้นอักขระและการนำเสนอของโปรแกรมใช้งานใน ลักษณะของรูปภาพกำหนดลักษณะการทำงานของโปรแกรม ซึ่งอำนวยความสะดวกต่อการ ใช้โปรแกรมที่เป็นมิตรต่อผู้ใช้

ภาษาไพธอนมีเครื่องมือที่สนับสนุนการสร้างโปรแกรมที่เป็นลักษณะของรูปภาพ ที่เป็นโมดูล ที่ติดตั้งมาพร้อมกับภาษาไพธอนและโมดูลที่ถูกพัฒนาเพิ่มเติมในการเรียกใช้งาน ทั้งนี้ เพื่อความ สะดวกและเพิ่มประสิทธิภาพการพัฒนาโปรแกรมลักษณะของส่วนต่อประสานกราฟิกกับผู้ใช้ มีรายละเอียดการใช้นำไปใช้ดังต่อไปนี้

# เบื้องต้นเกี่ยวกับส่วนต่อประสานกราฟิกกับผู้ใช้

ส่วนต่อประสานกราฟิกกับผู้ใช้ หรือ จียูไอ (GUI : Graphical User Interface) (ราชบัณฑิตย สภา, 2564) หมายถึง การใช้งานโปรแกรมที่นำเสนอในลักษณะของรูปภาพหรือไอคอน ผ่านเมาส์หรือ อุปกรณ์ที่เป็นใช้ตัวชี้บนจอภาพ ทั้งนี้เพื่อแก้ปัญหาของการใช้โปรแกรมในลักษณะของการพิมพ์ ข้อความที่ผู้ใช้โปรแกรมไม่เข้าใจถึงไวยากรณ์ที่ถูกต้องและการสะกดคำสั่งของโปรแกรมที่แท้จริง (Farrell, 2011)

เครื่องมือที่สนับสนุนใช้งานกับภาษาไพธอนลักษณะรูปแบบของส่วนต่อประสานกราฟิกกับ ผู้ใช้ มีผู้พัฒนาโมดูลให้ใช้งานร่วมกับภาษาไพธอนมากมาย แต่ละโมดูลจะมีหน้าที่ของการทำงาน แตกต่างกันออกไป เช่น PyGTK PyQt wxPython และ Kivy เป็นต้น แต่ภาษาโปรแกรมไพธอนได้ เลือกโมดูล tkinter เป็นโมดูลหลักที่ใช้ในการพัฒนาส่วนต่อประสานกราฟิกกับผู้ใช้ (สุชาติ คุ้มมณี, 2558) ซึ่งเมื่อติดตั้งภาษาโปรแกรมไพธอน จะสามารถเรียกใช้โมดูล tkinter นี้ได้ทันที

การสร้างส่วนต่อประสานกราฟิกกับผู้ใช้ จะมีลำดับขั้นตอนของการพัฒนาโปรแกรม มีรายละเอียดดังต่อไปนี้

### 1. การออกแบบส่วนต่อประสานกราฟิกกับผู้ใช้

การพัฒนาโปรแกรมให้ตรงกับจุดประสงค์ของการใช้งาน ในเบื้องต้นผู้พัฒนาโปรแกรม จะต้องทำการออกแบบ โดยการร่างต้นแบบลักษณะของโปรแกรม ก่อนที่จะนำผลลัพธ์ของต้นแบบ ไปดำเนินการสร้างเป็นโปรแกรมต่อไป ทั้งนี้ในการออกแบบต้นแบบของโปรแกรมสามารถปรับปรุง เปลี่ยนแปลงได้ตลอดเวลาเพื่อให้มีความถูกต้อง เมื่อต้นแบบมีความสมบูรณ์ตามที่ต้องการ จะสามารถ นำไปพัฒนาเป็นโปรแกรมได้ทันที แต่ถ้าหากไม่มีการร่างไว้ก่อน เมื่อนำไปพัฒนาเป็นโปรแกรมและ เกิดการเปลี่ยนแปลง จะต้องปรับปรุงชุดคำสั่งให้ได้ผลลัพธ์ที่ต้องการ จะทำเกิดความล่าช้าในการ พัฒนาโปรแกรม

การร่างต้นแบบลักษณะของโปรแกรม อาจเขียนลงกระดาษหรือโปรแกรมสำหรับ การออกแบบ เพื่อกำหนดสิ่งที่มีอยู่ในโปรแกรมและการจัดตำแหน่งของแต่ละส่วนภายในโปรแกรม ให้มีความเหมาะสมตามจุดประสงค์ของการใช้งาน ซึ่งมีลักษณะตัวอย่างของการร่างต้นแบบของ โปรแกรมดังภาพที่ 10.1



ภาพที่ 10.1 การร่างต้นแบบของโปรแกรม

การร่างต้นแบบโปรแกรมประกอบด้วยส่วนประกอบการทำงานที่อยู่ในโปรแกรม การทำงาน แต่ละส่วนจะบ่งบอกลักษณะของการทำงาน เช่น ช่องสำหรับกรอกข้อมูล ปุ่มตัวเลือก ปุ่มบันทึกและ ออกจากโปรแกรม เป็นต้น ดังภาพที่ 10.1 ต้นแบบที่ได้จากการร่างนี้จะถูกนำไปใช้ในการพัฒนา โปรแกรมรูปแบบของส่วนต่อประสานกราฟิกกับผู้ใช้ ให้สามารถทำงานตามจุดประสงค์ที่ได้ ร่างต้นแบบของโปรแกรมไว้

### 2. การสร้างโปรแกรมส่วนต่อประสานกราฟิกกับผู้ใช้

โปรแกรมลักษณะของส่วนต่อประสานกราฟิก ถูกพัฒนาขึ้นจากการร่างต้นแบบของโปรแกรม ซึ่งนำต้นแบบที่ได้มาพิจารณาร่วมกับเครื่องมือที่มีใช้ในโปรแกรม ซึ่งต้องเลือกรูปแบบที่เหมาะสมกับ ต้นแบบที่วางไว้ บางกรณีเครื่องมือที่ใช้ในโปรแกรม อาจมีหรือไม่มีให้เรียกใช้งาน ก็จะต้องเลือกเครื่อง ที่มีความใกล้เคียงและเหมาะสม แต่โดยทั่วไปตามรูปแบบพื้นฐานของส่วนต่อประสานกราฟิกกับผู้ใช้ จะมีให้ใช้งานลักษณะเดียวกัน เช่น กล่องข้อความ ช่องสำหรับป้อนข้อมูลขนาดสั้น ช่องสำหรับกรอก ข้อมูลที่มีปริมาณมาก รายการกลุ่มตัวเลือก ปุ่มการทำงาน เป็นต้น

การพัฒนาโปรแกรมตามต้นแบบที่ออกแบบไว้ข้างต้น สามารถพัฒนาส่วนต่อประสานกราฟิก กับผู้ใช้ด้วยโมดูล tkinter เมื่อพัฒนาโดยนำต้นแบบมาสร้างเป็นโปรแกรม จะมีผลลัพธ์ที่ได้ดังภาพ ต่อไปนี้

🕴 จัดการข้อมูลนัก	าศึกษา	_	- C	]	×
รหัส					
ชื่อ-นามสกุล		លេខា	🖲 ซาย	Ои	ญิง
สาขาวิชา					
ที่อยู่					
เบอร์โทร		บ้เ	เทีก		
อีเมล์		ออกจาก	โปรแกรม		

ภาพที่ 10.2 การสร้างโปรแกรมลักษณะของส่วนต่อประสานกราฟิกกับผู้ใช้

ภาพที่ 10.2 ผลลัพธ์ที่ได้จากโปรแกรมจากการเรียกใช้เครื่องมือที่มีอยู่ในโมดูล tkinter ของภาษาไพธอน จะเห็นได้ว่าโปรแกรมที่ได้ถูกออกแบบมาจากต้นแบบ มีลักษณะใกล้เคียงกับ ต้นแบบที่ได้ออกแบบไว้ ซึ่งสามารถปรับได้ตามความเหมาะสมของเครื่องมือที่นำมาใช้

การพัฒนาส่วนต่อประสานกราฟิกกับผู้ใช้จากข้างต้นด้วย tkinter มีเครื่องมือที่สามารถ นำมาใช้เป็นจำนวนมาก ซึ่งแต่ละเครื่องมือจะมีหน้าที่ในการทำงานที่มีลักษณะแตกต่างกัน ซึ่งจะ กล่าวถึงวิธีการนำมาใช้ในขั้นตอนการสร้างส่วนต่อประสานกราฟิกกับผู้ใช้ต่อไป

### องค์ประกอบของส่วนต่อประสานกราฟิกกับผู้ใช้ tkinter

การสร้างส่วนต่อประสานกราฟิกกับผู้ใช้ จากรายละเอียดที่กล่าวไว้ในข้างตน เครื่องมือ tkinter มีองค์ประกอบของการทำงานดังต่อไปนี้



**ภาพท 10.3** องคบระกอบสวนตอบระสานกราพกกบผูเซ tkint **ที่มา :** Chaudhary (2015)

จากภาพที่ 10.3 องค์ประกอบของการสร้างส่วนต่อประสานกราฟิกกับผู้ใช้โดยใช้โมดูล tkinter นั้น จะประกอบไปด้วยขั้นตอนการสร้างโปรแกรม 3 ขั้นตอน (Chaudhary, 2015) ได้แก่ การสร้างหน้าต่างของโปรแกรมและวางเครื่องมือที่จะให้โปรแกรมทำงานได้ตามวัตถุประสงค์ที่ตั้งไว้ จะอาศัยเครื่องมือที่อยู่ในโมดูล tkinter เรียกว่า "วิดเจ็ต (Widget)" (Wikipedia, 2021) ดังภาพที่ 10.4



**ภาพที่ 10.4** ลักษณะของวิดเจ็ต

การวางตำแหน่งของวิดเจ็ต ต้องใช้ส่วนประกอบที่เรียกว่า "Geometry Management" ที่ควบคุมรูปแบบการวางตำแหน่งของวิดเจ็ต เพื่อให้เครื่องมือในโปรแกรมมีระเบียบ ส่งผลต่อ ความรู้สึกที่ดีต่อการใช้โปรแกรมของผู้ใช้งาน ดังตัวอย่างภาพที่ 10.5



ภาพที่ 10.5 ลักษณะของ Geometry Management

เมื่อได้ออกโปรแกรมแบบตามรูปแบบที่กำหนดเสร็จสิ้น จะต้องทำการเขียนโปรแกรมให้ สามารถทำงานได้ตามรูปแบบที่ได้วางไว้ การประมวลผลประกอบด้วย ลักษณะการทำงาน 2 ประเภท คือ Event และ Callback ดังภาพที่ 10.6



ภาพที่ 10.6 Event และ Callback

การทำงานของโปรแกรมนั้น เมื่อเสร็จสิ้นการเขียนโปรแกรม ลักษณะขั้นตอนต่อไป คือ เริ่ม การใช้โปรแกรม โดยการป้อนข้อมูลเข้าสู่โปรแกรม จากนั้นโปรแกรมจะทำการส่งข้อมูลเมื่อมี เหตุการณ์กดที่ปุ่ม "บันทึก" ส่งข้อมูลไปบันทึกที่ฐานข้อมูล กระบวนการดังกล่าวนี้เรียกว่า Event และเมื่อผ่านกระบวนการบันทึกข้อมูลเสร็จสิ้น จะมีการแจ้งผลกลับมายังผู้ใช้งานโปรแกรม ให้ทราบ ถึงสถานการณ์ทำงาน เมื่อผ่านกระบวนของการประมวลผล ส่วนนี้เรียกว่า Callback ที่เป็นการสิ้นสุด กระบวนการของการทำงานบันทึกข้อมูล ดังตัวอย่างภาพที่ 10.6

### การสร้างหน้าต่างของโปรแกรม

หน้าต่าง (Window) (ราชบัณฑิตยสภา, 2564) คือ ส่วนต่อประสานกราฟิกกับผู้ใช้ที่เป็น ลักษณะกรอบการทำงานของโปรแกรม ซึ่งภายในจะมีเครื่องมือที่ใช้งานในหน้าต่างนั้น แต่ละหน้าต่าง ของโปรแกรม มีหน้าที่การทำงานแตกต่างกัน โดยมีการทำงานอย่างใดอย่างหนึ่ง เช่น หน้าต่างบันทึก ข้อมูลนักศึกษา หน้าต่างลงทะเบียน หน้าต่างบันทึกผลการเรียน หน้าต่างรายงานผลการเรียน เป็นต้น การสร้างหน้าต่างของโปรแกรมไพธอน มีส่วนประกอบที่ควบคุมคุณสมบติการแสดงหน้าต่างส่วนต่อ ประสานกราฟิกกับผู้ใช้ ดังรายละเอียดการใช้คำสั่งต่อไปนี้

นำเข้าโมดูล tkinter หน้าต่างโปรแกรม = คลาส tk() กลุ่มคำสั่งกำหนดคุณสมบัติ ... หน้าต่างโปรแกรม.mainloop()

คำสั่งที่เกี่ยวข้องคุณสมบัติที่จัดการคุณลักษณะของหน้าต่างของโปรแกรม มีรายละเอียดของ คำสั่งดังต่อไปนี้

คำสั่ง	คุณสมบัติ	
title	การกำหนดแถบชื่อเรื่อง	
geometry	การกำหนดขนาดของหน้าต่างโปรแกรม	
mainloop	การแสดงหน้าต่างของโปรแกรม	

รูปแบบโครงสร้างการสร้างหน้าต่างของโปรแกรม สามารถนำไปใช้ในการสร้างโปรแกรม ดังตัวอย่างที่ 10.1

บรรทัดที่	คำสั่ง
1	import tkinter
2	window = tkinter.Tk()
3	window.title("tkinter GUI")
4	window.geometry("400x150")
5	window.mainloop()
ผลลัพธ์	
	🖡 tkinter GUI — 🗆 X

ตัวอย่างที่ 10.1 การสร้างหน้าต่างของโปรแกรม

โปรแกรมตัวอย่างที่ 10.1 มีขั้นตอนของการสร้างหน้าต่างของโปรแกรม โดยเริ่มต้นจากการ นำเข้าโมดูล tkinter ที่มีคุณสมบัติในการสร้างโปรแกรมในลักษณะของส่วนประสานกราฟิกกับผู้ใช้ ดำเนินการสร้างหน้าต่างของโปรแกรมดังบรรทัดที่ 2 กำหนดขนาดของหน้าต่างโปรแกรมด้วย คุณสมบัติ geometry() โดยกำหนดขนาดความกว้างและความยาว มีหน่วยเป็นพิเซล เมื่อกำหนด รายละเอียดเสร็จสิ้นและให้แสดงผลลัพธ์ของหน้าต่างโปรแกรม จะใช้คำสั่ง mainloop() ในการ แสดงผล

### การสร้างวัตถุบนหน้าต่างโปรแกรมด้วยวิดเจ็ต

วิดเจ็ต หมายถึง องค์ประกอบที่ปรากฏอยู่บนหน้าจอ ซึ่งแต่ละชนิดจะมีลักษณะการทำงาน ที่แตกต่างกัน จะถูกเลือกเพื่อต่อประสานกับผู้ใช้ องค์ประกอบเหล่านี้เป็นลักษณะภาพ ที่จะให้ใช้งาน ในส่วนหน้ากับผู้ใช้งาน (Chaudhary, 2015)

tkinter ประกอบด้วยเครื่องมือที่ใช้ออกแบบการทำงานในโปรแกรมหลายรูปแบบ แต่ละ รูปแบบที่นำมาใช้ คือ วิดเจ็ต โดยมีวิดเจ็ตเป็นคลาสที่ถูกสร้างไว้ให้เรียกใช้ รูปแบบของการเรียกใช้วิด เจ็ต มีรายละเอียดดังต่อไปนี้ ชื่อ วิดเจ็ต = คลาส วิดเจ็ต(หน้าต่างหลัก, คุณสมบัติ, ... )

รูปแบบการเรียกใช้คำสั่งเรียกใช้งานวิดเจ็ต เมื่อนำไปใช้งานแต่ละประเภท มีรายละเอียด ดังต่อไปนี้

### 1. Label

Label หมายถึง วิดเจ็ตที่มีลักษณะของการป้ายแสดงข้อความ เพื่อให้ทราบถึงรายละเอียด การใช้งาน (สุชาติ คุ้มมณี, 2558) เช่น ป้ายรหัส ป่ายชื่อ-นามสกุล ป้ายที่อยู่ ป้ายเบอร์โทรศัพท์ เป็นรูปแบบโครงสร้างคำสั่ง Label มีรายละเอียดดังต่อไปนี้

ตัวอย่างที่ 10.2	การสร้างป้ายเ	เสดงข้อคว	ามด้วย	Label
------------------	---------------	-----------	--------	-------

บรรทัดที่	คำสั่ง
1	from tkinter import *
2	window = Tk()
3	window.geometry("300x100")
4	window.title("ข้อมูลนักศึกษา")
5	id_Lb=Label(window, text="รหัส").pack()
6	name_Lb=Label(window, text="ชื่อ-นามสกุล").pack()
7	course_Lb=Label(window, text="สาขาวิชา").pack()
8	window.mainloop()
ผลลัพธ์	
(d)	
	🖉 ข้อมูลนักศึกษา — 🗆 🗙
$\circ$	รหัส
	<b>ชื่อ-นามสกุล</b>
	สายาวิยา

โปรแกรมตัวอย่างที่ 10.2 รูปแบบการสร้าง Label ในประโยคคำสั่งบรรทัดที่ 5 ถึง 7 กำหนดการแสดงข้อความ รหัส ชื่อ-นามสกุล และสาขาวิชา เมื่อใช้คำสั่ง pack() ในการแสดงป้าย ข้อความเหล่านี้จะถูกแยกแสดงในแต่ละบรรทัด โดยจะแสดงในตำแหน่งกึ่งกลางของหน้าต่าง โปรแกรม

### 2. Entry

Entry คือ วิดเจ็ตที่มีลักษณะของกล่องข้อความสำหรับรับข้อมูลที่ได้ จากการป้อนข้อมูล แผงแป้นอักขระ เช่น รับข้อมูลรหัส ชื่อ-นามสกุล สาขาวิชา ที่อยู่ เบอร์โทรศัพท์ เป็นต้น รูปแบบ โครงสร้างคำสั่ง Entry มีรายละเอียดดังต่อไปนี้

**ตัวอย่างที่ 10.3** การสร้างกล่องข้อความด้วยคำสั่ง Entry

บรรทัดที่	คำสั่ง		
1	from tkinter import *		
2	window = Tk()		
3	window.geometry("300x100")		
4	window.title("ข้อมูลนักศึกษา")		
5	id_Lb=Label(window, text="รหัส").pack()		
6	id_Et=Entry().pack()		
7	name_Lb=Label(window, text="ชื่อ-นามสกุล").pack()		
8	name_Et=Entry().pack()		
9	window.mainloop()		
ผลลัพธ์			
	🖉 ข้อมูลนักศึกษา — 🗆 🗙		
	รหัส		
	ชื่อ-นามสกุล		
N			

โปรแกรมตัวอย่างที่ 10.3 รูปแบบการสร้าง Entry ในประโยคคำสั่งบรรทัดที่ 6 และ 8 กำหนดกล่องในการรับข้อมูล รหัส และชื่อ-นามสกุล เมื่อใช้คำสั่ง pack() ในการแสดงป้ายข้อความ และกล่องข้อความจะถูกแยกแสดงในแต่ละบรรทัด โดยจะแสดงในตำแหน่งกึ่งกลางของหน้าต่าง โปรแกรม

### 3. Text

Text คือ วิดเจ็ตที่เป็นกล่องข้อความเช่นเดียวกันกับ Entry แต่จะมีขนาดที่รองรับการกรอก ข้อมูลปริมาณมาก และรูปแบบการแสดงผลของการรับข้อมูล จะมีลักษณะของรับข้อมูลได้หลาย บรรทัด เช่น ข้อมูลที่อยู่ ข้อมูลแสดงความคิดเห็น เป็นต้น รูปแบบโครงสร้างคำสั่ง Text มีรายละเอียด ดังต่อไปนี้



v . d	ิย	ע ו	ູ	ิย		ิย	
ตวอยางท 10.4	การสราง	กลองข	อความรเ	าขอมล	ปรมาณ	มากดวย	Text
				91			

โปรแกรมตัวอย่างที่ 10.4 รูปแบบการสร้าง Text ในประโยคคำสั่งบรรทัดที่ 6 และ 8 กำหนดกล่องในการรับข้อมูล รหัส และชื่อ-นามสกุล ซึ่งลักษณะของกล่องข้อความแบบ Text นี้จะมี ขนาดใหญ่ รับปริมาณข้อมูลได้จำนวนมาก สามารถขยายกล่องรับข้อมูลได้ทั้งแนวความกว้างและ ความยาว เมื่อใช้คำสั่ง pack() ในการแสดง Text จะถูกจัดในตำแหน่งกึ่งกลางของหน้าต่างโปรแกรม

#### 4. Button

Button คือ วิดเจ็ตที่มีลักษณะตอบสนองกับผู้ใช้งานในรูปแบบของปุ่ม โดยผู้ใช้งานสามารถ ตอบสนองด้วยการกด การทำงานทั่วไปหลังจากเมื่อมีการกดที่ปุ่ม จะเกิดเหตุการณ์หรือการ ประมวลผลอย่างใดอย่างหนึ่งเกิดขึ้น รูปแบบโครงสร้างคำสั่ง Button มีรายละเอียดดังต่อไปนี้

ตัวอย่างที่ 10.5	การสร้างปุ่มด้วย	Button
------------------	------------------	--------

บรรทัดที่	คำสั่ง			
1	from tkinter import *			
2	window = Tk()			
3	window.geometry("300x200")			
4	window.title("ข้อมูลนักศึกษา")			
5	name_Lb=Label(window, text="ชื่อ-นามสกุล").pack()			
6	name_Text=Text(window, height=2,width=20).pack()			
7	address_Lb=Label(window, text="ที่อยู่").pack()			
8	address_Text=Text(window, height=3,width=30).pack()			
9	save_btn=Button(window, text="บันทึก").pack()			
10	exit_btn=Button(window, text="ออกจากโปรแกรม").pack()			
11	window.mainloop()			
ผลลัพธ์				
	🖗 ข้อมูลนักศึกษา — 🗆 🗙			
	ชื่อ-นามสกุล			
(9)				
	าบบบู			
0).				
	บันทึก			
	ออกจากไปรแกรม			

โปรแกรมตัวอย่างที่ 10.5 รูปแบบการสร้าง Button ในประโยคคำสั่งบรรทัดที่ 9 และ 10 กำหนดปุ่มที่ใช้สำหรับบันทึกและออกจากโปรแกรม เมื่อใช้คำสั่ง pack() ในการแสดง Button จะถูก จัดในตำแหน่งกึ่งกลางของหน้าต่างโปรแกรม

#### 5. Checkbutton

Checkbutton คือ วิดเจ็ตที่มีลักษณะของการทำเครื่องหมายเป็นการเลือกโดยให้ความหมาย ใช่หรือไม่ใช่ โดยทั่วไปเมื่อทำเครื่องหมายถูกในวิดเจ็ต Checkbutton รายการนั้น จะมีการนำข้อมูลที่ ได้เลือกไปใช้ทำงานหรือประมวลผลต่อไป โดยทั่วไปจะมีรายการให้เลือกหลายรายการ ซึ่งสามารถ เลือกได้มากกว่า 1 รายการ เช่น เลือกช่องทางในการติดต่อ เลือกรายการอาหารที่ชอบ เลือกสถานที่ ท่องเที่ยวสนใจ เลือกสาขาที่เรียน เป็นต้น รูปแบบโครงสร้างคำสั่ง Checkbutton มีรายละเอียด ดังต่อไปนี้

บรรทัดที่	คำสั่ง
1	from tkinter import *
2	window = Tk()
3	window.geometry("300x150")
4	window.title("ข้อมูลนักศึกษา")
5	connet_Lb=Label(window, text="สามารถติดต่อได้ที่").pack()
6	tel_chk=Checkbutton(window, text=" โทรศัพท์ ").pack()
7	email_chk=Checkbutton(window, text=" E-mail ").pack()
8	fb_chk=Checkbutton(window, text=" Line ").pack()
9	line_chk=Checkbutton(window, text=" Facebook ").pack()
10	other_chk=Checkbutton(window, text=" อื่น ๆ ").pack()
11	window.mainloop()
ผลลัพธ์	
$\left( \mathbf{Q}\right) $	
$\sim$ $^{\prime}$	🖗 ข้อมูลนักศึกษา — 🗆 🗙
$\phi$	สามารถติดต่อได้ที่
	🔽 โทรด้พท์
	E-mail
	Line
	Facebook
	อื่น ๆ

### **ตัวอย่างที่ 10.6** การสร้างตัวเลือกด้วย Checkbutton

โปรแกรมตัวอย่างที่ 10.6 รูปแบบการสร้าง Checkbutton ในประโยคคำสั่งบรรทัดที่ 6 ถึง 10 กำหนดรายการตัวเลือก โดยการเลือกจะทำเครื่องหมายถูกลงในกล่องขนาดเล็กหน้ารายการ ตัวเลือกนั้นและสามารถทำการเลือกได้หลายรายการ เมื่อใช้คำสั่ง pack() ในการแสดง Checkbutton แต่ละรายการตัวเลือกจะถูกจัดในตำแหน่งกึ่งกลางของหน้าต่างโปรแกรม

#### 6. Listbox

Listbox คือ วิดเจ็ตที่ลักษณะของรายการให้เลือก รายการนี้จะถูกเลือกเพียง 1 รายการ เท่านั้น เช่น สัญชาติ เชื้อชาติ ศาสนา สถานที่เกิด เป็นต้น รูปแบบโครงสร้างคำสั่ง Listbox มีรายละเอียดดังต่อไปนี้

### **ตัวอย่างที่ 10.7** การสร้างตัวเลือกด้วย Listbox

บรรทัดที่	คำสั่ง		
1	from tkinter import *		
2	window = Tk()		
3	window.geometry("300x150")		
4	lbl = Label(window,text = "นับถือศาสนา")		
5	listbox = Listbox(window,height=6,width=25)		
6	listbox.insert(1,"พุทธ")		
7	listbox.insert(2, "คริสต์")		
8	listbox.insert(3, "อิสลาม")		
9	listbox.insert(4, "ฮินดู")		
10	listbox.insert(5, "อื่น ๆ")		
11	lbl.pack()		
12	listbox.pack()		
13	window.mainloop()		
ผลลัพธ์			
	🖉 tk — 🗆 🗙		
	. นับกือศาสนา		
	พุทธ <u>ตริสต์</u> ภิสอาน		
	อินดู อิน ๆ		

โปรแกรมตัวอย่างที่ 10.7 รูปแบบการสร้าง Listbox ในประโยคคำสั่งบรรทัดที่ 5 กำหนดการสร้าง Listbox และคำสั่งในบรรทัดที่ 6-10 กำหนดรายการตัวเลือก โดยการเลือกจะทำ การเลือกได้เพียงรายการเดียวเท่านั้น เมื่อใช้คำสั่ง pack() ในการแสดง Listbox จะถูกจัดในตำแหน่ง กึ่งกลางของหน้าต่างโปรแกรม

#### 7. Menubutton

Menubutton คือ วิดเจ็ตที่มีลักษณะของรายการให้เลือก แต่ลักษณะของรายการเลือกจะมี การเลื่อนขึ้นและเลื่อนลงเมื่อไปคลิกที่หัวข้อรายการและสามารถเลือกได้เพียงรายการเดียวเท่านั้น เช่นเดียวกันกับ Listbox รูปแบบโครงสร้างคำสั่ง Menubutton มีรายละเอียดดังต่อไปนี้

ตัวอย่างที่ 10.8 การสร้างตัวเลือกด้วย Menubutton

บรรทัดที่	คำสั่ง	
1	from tkinter import *	
2	window = Tk()	
3	window.geometry("300x150")	
4	religion_MenuB = Menubutton(window, text = "นับถือศาสนา",)	
5	religion_MenuB.menu = Menu(religion_MenuB)	
6	religion_MenuB["menu"]=religion_MenuB.menu	
7	religion_MenuB.menu.add_checkbutton(label = "พุทธ")	
8	religion_MenuB.menu.add_checkbutton(label = "คริสต์")	
9	religion_MenuB.menu.add_checkbutton(label = "อิสลาม")	
10	religion_MenuB.menu.add_checkbutton(label = "ฮินดู")	
110	religion_MenuB.menu.add_checkbutton(label = "อื่นๆ")	
12	religion_MenuB.pack()	
13	window.mainloop()	
ผลลัพธ์		
	นับกิอศาสนา <mark>พุทธ</mark> ดริสต์ อิสคาม	
	อินตู อินๆ	

โปรแกรมตัวอย่างที่ 10.8 รูปแบบการสร้าง Menubutton ในประโยคคำสั่งบรรทัดที่ 4-6 กำหนดคุณลักษณะของ Menubutton และคำสั่งบรรทัดที่ 7-11 สร้างรายการตัวเลือกใน Menubutton โดยการใช้งานสามารถเลือกได้เพียงรายการเดียวเท่านั้น เมื่อใช้คำสั่ง pack() ในการ แสดง Menubutton จะถูกจัดในตำแหน่งกึ่งกลางของหน้าต่างโปรแกรม

#### 8. Menu

Menu คือ วิดเจ็ตที่มีลักษณะของการสร้างแถบรายการ ซึ่งแถบรายการนี้จะเป็นรายการหลัก และเมื่อรายการหลักแต่ละรายการ ถูกเลือกจะแสดงรายการย่อยที่อยู่ภายในของแต่ละรายการหลัก และแต่ละรายการย่อยสามารถมีรายการย่อยที่อยู่ภายในซ้อนต่อเนื่องกัน รูปแบบโครงสร้างคำสั่ง Menu มีรายละเอียดดังต่อไปนี้

### ตัวอย่างที่ 10.9 การสร้างกลุ่มของรายการ Menu

บรรทัดที่	คำสั่ง			
1	from tkinter import *			
2	window = Tk()			
3	menu_bar = Menu(window)			
4	data_menu = Menu(menu_bar, tearoff=0)			
5	menu_bar.add_cascade(label="ข้อมูลทั่วไป",menu=data_menu)			
6	data_menu.add_command(label="ลูกค้ำ")			
7	data_menu.add_command(label="สินค้ำ")			
8	data_menu.add_command(label="พนักงาน")			
9	data_menu.add_separator()			
10	data_menu.add_command(label="แผนก")			
11	data_menu.add_command(label="ตำแหน่ง")			
12	process_menu = Menu(menu_bar, tearoff=0)			
13	menu_bar.add_cascade(label="สั่งซื้อ-ขาย", menu=process_menu)			
14	process_menu.add_command(label="สั่งซื้อสินค้า")			
15	process_menu.add_separator()			
16	process_menu.add_command(label="ขายสิค้า")			
17	report_menu = Menu(menu_bar, tearoff=0)			
18	menu_bar.add_cascade(label="รายงาน", menu=report_menu)			



โปรแกรมตัวอย่างที่ 10.9 รูปแบบการสร้าง Menu ในประโยคคำสั่งบรรทัดที่ 4 ทำการสร้าง Menu โดยให้มีรายการหลักประกอบด้วย ข้อมูลทั่วไป สั่งซื้อ-ขาย และรายงาน ดังคำสั่งบรรทัดที่ 5 13 และ 18 แต่ละรายการหลักจะมีการกำหนดรายการย่อยอยู่ภายใน ดังคำสั่งบรรทัดที่ 5-11 14-16 และ 19-23 ผลลัพธ์ที่ได้ของโปรแกรมจะแสดงแถบรายการหลักเรียงลำดับรายการ เมื่อมีการเลือกแต่ ละรายการจะแสดงรายการย่อย ซึ่งสามารถเลื่อนเพื่อเลือกการทำงาน การใช้งานลักษณนี้เรียกว่า "Pull-Down Menu" (ธีรวัฒน์ ประกอบผล, 2558)

#### 9. Radiobutton

Radiobutton คือ วิดเจ็ตที่มีลักษณะของการเลือก โดยการกำหนดจุดข้างหน้าตัวเลือก เพื่อแสดงถึงการเลือกในรายการนั้น ซึ่งจะมีลักษณะคล้ายกับ Checkbutton แต่จะแตกต่างกันตรงที่ Checkbutton จะสามารถเลือกได้หลายรายการ แต่วิดเจ็ต Radiobutton จะสามารถเลือกได้เพียง 1 รายการเท่านั้น รูปแบบโครงสร้างคำสั่ง Radiobutton มีรายละเอียดดังต่อไปนี้



ตัวอย่างที่ 10.10 การสร้างรายการตัวเลือกด้วย Radiobutton

โปรแกรมตัวอย่างที่ 10.10 รูปแบบการสร้าง Radiobutton ในประโยคคำสั่งบรรทัดที่ 7 ถึง 11 กำหนดรายการตัวเลือก โดยการเลือกจะทำเครื่องหมายจุดสีดำลงในช่องวงกลมขนาดเล็ก ด้านหน้าของแต่ละรายการ การตัวเลือกรายการนั้น สามารถทำการเลือกได้เพียงหนึ่งรายการเท่านั้น เมื่อใช้คำสั่ง pack() ในการแสดง Radiobutton แต่ละรายการตัวเลือกจะถูกจัดในตำแหน่งกึ่งกลาง ของหน้าต่างโปรแกรม  1.2.1 Scale คือ วิดเจ็ตที่มีลักษณะการเลื่อนแถบ เพื่อกำหนดสถานการณ์แสดงข้อมูลตาม ขอบเขตที่ต้องการ เช่น ความคมชัดของสี ความสว่าง ความเข้ม เป็นต้น รูปแบบโครงสร้างคำสั่ง Scale มีรายละเอียดดังต่อไปนี้

บรรทัดที่	คำสั่ง	
1	from tkinter import *	
2	window = Tk()	
3	window.geometry("200x100")	
4	scale1=Scale(window, orient = HORIZONTAL).pack()	
5	scale2=Scale(window, orient = VERTICAL).pack()	
6	window.mainloop()	
ผลลัพธ์		
	1 tk -	

ตัวอย่างที่ 10.11 การสร้างแถบเลื่อนสถานะด้วย Scale

โปรแกรมตัวอย่างที่ 10.11 รูปแบบการสร้าง Scale ในประโยคคำสั่งบรรทัดที่ 4 และ 5 โดยการใช้งานจะเป็นลักษณะของการเลื่อนปุ่มที่อยู่บนแถบจากทางซ้ายไปทางขวาหรือจากด้านบนลง ด้านล่างขึ้นอยู่กับการตั้งแนวของ Scale เมื่อใช้คำสั่ง pack() ในการแสดง Scale จะถูกจัดในตำแหน่ง กึ่งกลางของหน้าต่างโปรแกรม

#### 10. Message

Message คือ วิดเจ็ตที่มีลักษณะของการแสดงข้อความ มีลักษณะการนำไปใช้ใกล้เคียงกัน กับ Label การนำไปใช้รูปแบบโครงสร้างคำสั่ง Message มีรายละเอียดดังต่อไปนี้

บรรทัดที่	คำสั่ง
1	from tkinter import *
2	window = Tk()
3	window.geometry("250x100")
4	msg = Message( window, text="ภาษาไพธอน",width=60)
5	msg.pack()
6	window.mainloop()
ผลลัพธ์	
	🦸 tk — 🗆 🗙
	ภาษาไพธอน

#### **ตัวอย่างที่ 10.12** การสร้างหน้าต่างแจ้งเตือนด้วย Message

โปรแกรมตัวอย่างที่ 10.12 รูปแบบการสร้าง Message ในประโยคคำสั่งบรรทัดที่ 4 โดยการ ใช้งานจะเป็นลักษณะของการแสดงข้อความและคุณสมบัติของการแสดงข้อความด้วยคำสั่ง width ในการกำหนดขนาดของข้อความ เมื่อใช้คำสั่ง pack() การแสดง Message จะถูกจัดในตำแหน่ง กึ่งกลางของหน้าต่างโปรแกรม

การกำหนดคุณลักษณะของวิดเจ็ต ให้มีคุณสมบัติตามที่ต้องการ จะต้องใช้คำสั่งที่กำหนด คุณสมบัติภายในวิดเจ็ต ซึ่งคุณสมบัติของคำสั่งวิดเจ็ต มีคำสั่งใช้งานจำนวนมากหรือน้อยขึ้นอยู่กับ ความซับซ้อนของการทำงานแต่ละวิดเจ็ต โดยมีตัวอย่างคุณสมบัติที่เกี่ยวข้องกับ Label รายละเอียด ดังต่อไปนี้

คำสั่ง	คุณสมบัติ
option	การกำหนดแถบชื่อเรื่อง
anchor	การกำหนดขนาดของหน้าต่างโปรแกรม
bg	การกำหนดสีพื้นหลัง
bitmap	การกำหนดพื้นหลังด้วยรูปภาพ

ตารางที่ 10.1 ตัวอย่างคำสั่งที่เกี่ยวกับวิดเจ็ตประเภท Label

คำสั่ง	คุณสมบัติ
bd	การกำหนดขนาดของเส้นขอบ
cursor	การกำหนดลักษณะของตัวชี้เมาส์เมื่อเลื่อนมาบนวิดเจ็ต
font	การกำหนดชนิดของตัวอักษรข้อความในวิดเจ็ต
fg	การกำหนดสีตัวอักษร
height	การกำหนดขนาดความสูงของวิดเจ็ต
image	การแสดงรูปภาพในวิดเจ็ต
justify	การจัดตำแหน่งข้อความในวิดเจ็ต เช่น LEFT RIGHT CENTER
padx	การกำหนดค่าเพื่อแสดงข้อความขอบด้านซ้ายหรือด้านขวาของพื้นที่
pady	การกำหนดค่าเพื่อแสดงข้อความขอบด้านบนหรือด้านล่างของพื้นที่
relief	ประเภทของเส้นขอบ ค่าพื้นฐานมีค่าเท่ากับ FLAT
text	การกำหนดข้อความที่จะอยู่ในบรรทัดของข้อความ
textvariable	การรับค่าจากตัวแปรที่เป็นข้อความ
underline	การกำหนดเส้นใต้ของตัวอักษร
width	การกำหนดความกว้างวิดเจ็ต เป็นการกำหนดขนาดของตัวอักษร
wraplength	การกำหนดปริมาณข้อความที่แสดงแต่ละบรรทัด

### ตารางที่ 10.1 ตัวอย่างคำสั่งที่เกี่ยวกับวิดเจ็ตประเภท Label (ต่อ)

คุณสมบติของวิดเจ็ต เมื่อนำไปใช้ จะต้องตรวจสอบคุณสมบัติที่ทำให้วิดเจ็ต มีลักษณะการ ทำงานตามต้องการ เพื่อให้การทำงานวิดเจ็ต มีความเหมาะสม เกิดประสิทธิภาพ ดังคุณสมบัติของ Label ตารางที่ 10.1 คำสั่งกำหนดคุณสมบัติมีหลายรูปแบบ ซึ่งคุณสมบัติของวิดเจ็ต อื่นก็มีรูปแบบ กำหนดคุณสมบัติมากมายเช่นกัน และบางรูปแบบก็มีลักษณะการกำหนดคุณสมบัติเหมือนกัน

# การจัดวางตำแหน่งวิดเจ็ตด้วย Geometry Management

การจัดตำแหน่งของเครื่องมือในหน้าต่างของโปรแกรม tkinter ได้มีส่วนที่ควบคุมการวาง ตำแหน่ง เรียกว่า Geometry Management ซึ่งมีลักษณะของการจัดวางตำแหน่งของวิดเจ็ต ด้วยเรขาคณิต ประกอบด้วย 3 เมธอด (สุชาติ คุ้มมณี, 2558) ซึ่งมีรูปแบบการใช้คำสั่ง ดังต่อไปนี้

#### 1. pack()

การแสดงผลของวิดเจ็ต ด้วยคำสั่ง pack() คือ รูปแบบพื้นฐานของการแสดงผลของวิดเจ็ต ใน หน้าต่างของโปรแกรม ซึ่งคำสั่งจะอยู่ถัดไปหลังจากการสร้างวิดเจ็ต ซึ่งมีรูปแบบของการใช้คำสั่ง ดังต่อไปนี้

สร้าง วิดเจ็ต() วิดเจ็ต.pack(คุณสมบัติ)

รูปแบบไวยกรณ์โครงสร้างชุดคำสั่ง pack() สามารถนำไปใช้ในการสร้างโปรแกรมดังตัวอย่าง ต่อไปนี้

### **ตัวอย่างที่ 10.13** การจัดตำแหน่งวิดเจ็ตด้วยคำสั่ง pack()

บรรทัดเ	SL-	คำสั่ง
1		from tkinter import *
2		window = Tk()
3		window.geometry("400x100")
4		Label(window, text="รหัส", bg="red").pack(expand=TRUE)
5		Label(window, text="ชื่อ-นามสกุล", bg="pink").pack(fill=X)
6		Label(window, text="ที่อยู่", bg="yellow").pack(side=RIGHT)
7		window.mainloop()
ผลลัพธ์		
		<u>X</u> •
	Ø	tk – 🗆 X
(0)		
		ชื่อ-นามสกุล
		<mark>ที่อยู่</mark>

#### 2. grid()

การวางตำแหน่งแบบ grid( ) จะมีแนวคิดของการจัดวางในรูปแบบของตาราง ที่มีมุมมอง ในแนวแถวและคอลัมน์ ทั้งนี้เพื่อให้ส่วนต่อประสานกราฟิกกับผู้ใช้เป็นระเบียบเรียบร้อยต่อการใช้งาน สามารถกำหนดรูปแบบที่เป็นอิสระในการจัดตำแหน่ง เรียกว่า Grid Layout ซึ่งแนวคิดในการกำหนด ตำแหน่งมีรายละเอียดดังภาพที่ 10.7

				Ŷ	คอลัมน์ ((	Column	)			
	0	1	2	3	4	5	6	7	5	9
แถว	1									
(Row)	2									
	3									

ภาพที่ 10.7 รูปแบบตำแหน่งของการจัดแบบ Grid Layout

ภาพที่ 10.7 การจัดตำแหน่งของวิดเจ็ต โดยใช้รูปแบบของ Grid Layout จะแบ่งออกเป็น 2 ส่วน ประกอบด้วย ส่วนที่เป็นแนวตั้ง เรียกว่า คอลัมน์ (Column) และแนวนอน เรียกว่า แถว (Row) ซึ่งมีรูปแบบของการกำหนดตำแหน่งดังต่อไปนี้

grid(Row=ตัวเลขตำแหน่งแนวคอลัมน์, Column=ตัวเลขตำแหน่งแนวแถว)

รูปแบบของการใช้คำสั่งในการจัดตำแหน่งของวิดเจ็ตคำสั่ง grid() สามารถนำไปจัดรูปแบบได้ ดังตัวอย่างที่ 10.14

**ตัวอย่างที่ 10.14** การจัดตำแหน่งวิดเจ็ตด้วยคำสั่ง grid()

	บรรทัดที่	คำสั่ง			
	1	from tkinter import *			
• (	2	window = Tk()			
	3	window.geometry("350x320")			
	4	window.title("จัดการข้อมูลนักศึกษา")			
	5	id_Lb=Label(text="")			
	6	id_Lb.grid(row=0)			
	7	id_Lb=Label(text=" รหัส ")			
	8	id_Lb.grid(row=1)			
	9	id_En=Entry()			

10	id_En.grid(row=1,column=1)
11	name_Lb=Label(text="ชื่อ-นามสกุล")
12	name_Lb.grid(row=2,column=0)
13	name_En=Entry()
14	name_En.grid(row=2,column=1)
15	gender_Rd=Radiobutton(text="เพศ : ชาย",value=0)
16	gender_Rd.grid(row=2,column=3)
17	gender_Rd=Radiobutton(text="หญิง",value=1)
18	gender_Rd.grid(row=2,column=4)
19	department_Lb=Label(text="สาขาวิชา")
20	department_Lb.grid(row=3)
21	dep_En=Entry()
22	dep_En.grid(row=3,column=1)
23	address_Lb=Label(text="ที่อยู่")
24	address_Lb.grid(row=4)
25	add_Text=Listbox()
26	add_Text.grid(row=4,column=1)
27	tel_Lb=Label(text="เบอร์โทร")
28	tel_Lb.grid(row=5)
29	tel_Et=Entry()
30	tel_Et.grid(row=5,column=1)
31	email_Lb=Label(text="อีเมล์")
32	email_Lb.grid(row=6)
33	email_Et=Entry()
34	email_Et.grid(row=6,column=1)
35	email_Lb=Label(text=" ")
36	email_Lb.grid(row=4,column=2)
37	save_Btn=Button(text="บันทึก")
38	save_Btn.grid(row=5,column=3,columnspan=2)
39	cancle_Btn=Button(text="ออกจากโปรแกรม")
40	cancle_Btn.grid(row=6,column=3,columnspan=2)
41	window.mainloop()



การจัดตำแหน่งวิดเจ็ตด้วยคำสั่ง grid() เมื่อทำการแบ่งเส้น จะเห็นได้ว่ามีวางตำแหน่งใน แนวคอลัมน์และแถว ทั้งสองแนวจะมีค่าแรกที่ตำแหน่ง 0 การระบุตำแหน่งจะต้องระบุค่าตำแหน่ง ทั้งสองแนว แต่ถ้าหากไม่ระบุตำแหน่งนั้นจะมีค่า 0 หรือเป็นตำแหน่งเริ่มต้นของแนวนั้น ดังโปรแกรม ตัวอย่างที่ 10.14

#### 3. place()

การวางตำแหน่งแบบ place() มีการนำไปใช้งานค่อนข้างน้อยที่สุด เนื่องจากการวางตำแหน่ง ด้วยรูปแบบนี้มีความละเอียดสูง ค่าตัวเลขที่กำหนดอยู่ในรูปของพิกเซล (Pixel) ซึ่งมีความยุ่งยาก ในการกำหนด ตำแหน่งจะอยู่ในรูปของแกน x และแกน y โดยมีรูปแบบของตำแหน่งดังต่อไปนี้



**ภาพที่ 10.8** โครงสร้างการกำหนดตำแหน่งคำสั่ง place()

โครงสร้างการกำหนดตำแหน่งคำสั่ง place() มีรูปแบบของการกำหนดคุณสมบัติ รายละเอียดดังต่อไปนี้

```
place(x=ตัวเลขตำแหน่งแนวแกน x, y=ตัวเลขตำแหน่งแนวแกน y)
```

รูปแบบของการใช้คำสั่งในการจัดตำแหน่งของวิดเจ็ตคำสั่ง place() สามารถนำไปจัดรูปแบบ ได้ดังตัวอย่างที่ 10.15

ตัวอย่างที่ 10.15 การจัดวางตำแหน่งด้วยคำสั่ง place()

บรรทัดที่ คำสั่ง		
1	from tkinter import *	
2	window = Tk()	
3	window.geometry("400x400")	
4	window.title("จัดการข้อมูลนักศึกษา")	
5	id_Lb=Label(text="รหัส").place(x=40,y=20)	
6	id_En=Entry().place(x=80,y=20)	
7	name_Lb=Label(text="ชื่อ-นามสกุล").place(x=15,y=50)	
8	name_En=Entry().place(x=80,y=50)	
9	name_Lb=Label(text="เพศ").place(x=240,y=50)	
10	gender_Rd=Radiobutton(text="ซาย",value=0).place(x=270,y=50)	
11	gender_Rd=Radiobutton(text="หญิง",value=1).place(x=320,y=50)	
12	department_Lb=Label(text="สาขาวิชา").place(x=30,y=80)	
13	dep_En=Entry().place(x=80,y=80)	
14	address_Lb=Label(text="ที่อยู่").place(x=40,y=110)	
15	add_Text=Listbox().place(x=80,y=110)	
16	tel_Lb=Label(text="เบอร์โทร").place(x=30,y=290)	
17	tel_Et=Entry().place(x=80,y=290)	
18	email_Lb=Label(text="อีเมล์").place(x=40,y=320)	
19	email_Et=Entry().place(x=80,y=320)	
20	save_Btn=Button(text="บันทึก").place(x=80,y=350)	



การจัดตำแหน่งวิดเจ็ตด้วยคำสั่ง place() คือ การจัดตำแหน่งในแนวแกน x และแกน y หน่วยที่กำหนดเป็นระดับของพิกเซล การกำหนดตัวเลขให้ได้ตำแหน่งที่ต้องการ อาจต้องใช้เวลา ในการกำหนด เนื่องจากมีความละเอียดสูง แต่ข้อดี คือ ความแม่นยำสูงของตำแหน่งที่วางและมีอิสระ โดยสามารถวางในจุดใดของหน้างต่างโปรแกรมได้ ซึ่งจะแตกต่างจากการใช้คำสั่ง pack() และคำสั่ง grid() ดังโปรแกรมตัวอย่างที่ 10.15

### การกำหนดเหตุการณ์ด้วย Event และ Callback

Event หมายถึง คุณลักษณะพิเศษที่ถูกวางไว้ให้รอเหตุการณ์ที่จะเกิดขึ้น โดยทั่วไปแล้วจะรอ เหตุการณ์ที่จะถูกส่ง เช่น กดที่ปุ่มเมาส์ การเลื่อนเมาส์ไปชี้ที่วัตถุ การกดที่ปุ่มบนคีย์บอร์ด เป็นต้น ซึ่งจะส่งผลต่อเหตุการณ์ที่จะเกิดขึ้นต่อไป (Meier, 2015)

Callback หมายถึง การกระทำอย่างใดอย่างหนึ่งที่เกิดขึ้น หลังจากกระทำต่อ Events นั้น ซึ่งจะแสดงให้เห็นเหตุการณ์ที่จะเกิดขึ้น เช่น กดปุ่มจะมีการคำนวณราคาสินค้า การเลื่อนเมาส์ไปบน วัตถุ จะมีการแจ้งรายละเอียด การแจ้งเตือนเมื่อกรอกข้อมูลผิด เป็นต้น (สุชาติ คุ้มมณี, 2558) ซึ่งลักษณะการทำงานดังต่อไปนี้



**ภาพที่ 10.9** การกำหนดเหตุการณ์ Event และ Callback

การกำหนดเหตุการณ์ด้วย Event และ Callback มีการใช้คำสั่งและรูปแบบการเชื่อมโยง เหตุการณ์ดังต่อไปนี้



รูปแบบการกำหนดเหตุการณ์ด้วย Event และ Callback สามารถนำไปเขียนโปรแกรม ดังโปรแกรมตัวอย่างที่ 10.16

**ตัวอย่างที่ 10.16** การกำหนดการกระทำด้วย Event และ Callback

บรรทัดที่	คำสั่ง
1	from tkinter import *
2	def show_msg():
3	Label(window,text = "หลักการเขียนโปรแกรมคอมพิวเตอร์").pack()
4	window = Tk()
5	window.geometry("300x120")
6	window.title("Event & Callback")
7	btn=Button(text=" แสดงข้อความ ",command=show_msg).pack()
8	window.mainloop()

ผลลัพธ์			
🕴 Event & Callback	_	×	🖉 Event & Callback — 🗆 🗙
	แสดงข้อความ		แสดงข้อความ หล้าการเขียนโปรแกรมดอมพิวเตอร์

โปรแกรมตัวอย่างที่ 10.16 การทำงานของ Event และ Callback ตรงตำแหน่งที่มีการสร้าง ปุ่มและเมื่อทำการกดปุ่ม จะมีการเรียกใช้คำสั่ง command() ที่ทำหน้าที่เรียกใช้ฟังก์ชัน show\_msg() ในการแสดงข้อความ ซึ่งจะเห็นได้ว่า Event และ Callback เริ่มต้นตั้งแต่ดำเนินการ รับข้อมูล นำข้อที่ได้ไประมวลผลผ่านชุดโปรแกรม ขั้นตอนนี้คือ Event และแสดงผลลัพธ์ออกมา ในรูปแบบของผลลัพธ์ในรูปแบบต่าง ๆ คือขั้นตอนของ Callback ทั้งสองส่วนมีความสัมพันธ์กันและ มีความสำคัญมากในการทำงานของโปรแกรม

### การประยุกต์เขียนโปรแกรมส่วนต่อประสานกราฟิกกับผู้ใช้

การสร้างส่วนต่อประสานกราฟิกกับผู้ใช้ สามารถนำขั้นตอนการพัฒนา tkinter ซึ่งประกอบ ไปด้วย การสร้างวิดเจ็ต การวางตำแหน่งของวิดเจ็ตและการสร้าง Event และ Callback ดังนั้น เพื่อ ดำเนินการประมวลผลและตอบสนองกลับมายังผู้ใช้โปรแกรม มาประยุกต์ใช้ในการเขียนโปรแกรม ตอบสนองการทำงานตามจุดประสงค์ที่กำหนด โดยมีรายละเอียดของตัวอย่างการประยุกต์การเขียน โปรแกรมแก้โจทย์ปัญหาด้วยส่วนต่อประสานกราฟิกกับผู้ใช้ ตามทิศทางการทำงานของโปรแกรม ดังต่อไปนี้

ตัวอย่างที่ 10.17	การสร้างส่วนต่อประสานกราฟิกกับผู้ใช้ ทิศทางโปรแกรมแบบเรียงลำดับ
	થ

บรรทัดที่	คำสั่ง
1	from tkinter import *
2	def calculate(*args):
3	val_price = float(price.get())
4	val_vat = float(vat.get())
5	money=val_price-(val_price*val_vat/100)
6	show_money.set(money)

7	window = Tk()						
8	window.geometry("300x150")						
9	window.title("คำนวณหาส่วนลดจากราคาสินค้า")						
10	price = String\	/ar()					
11	vat = StringVa	r()					
12	show_money	= StringVar()					
13	money_lbl =L	.abel(text="ราคาสินค้า").pack	0				
14	money_entry	money_entry = Entry(textvariable=price).pack()					
15	vat_lbl =Labe	vat_lbl =Label(text="ส่วนลด").pack()					
16	vat_entry = Entry(textvariable=vat).pack()						
17	money_cal=Button(text="จำนวนเงินชำระ",command=calculate).pack()						
18	money_show=Label(textvariable=show_money).pack()						
19	Label(text="บาท").pack()						
20	window.mainloop()						
ผลลัพธ์							
🖡 ดำนวณหาส่วนลดจากราดา – 🗆 X 🖡 ดำนวณหาส่วนลดจากราดา – 🗆 X 🖡 ดำนวณหาส่วนลดจากราดา – 🗆 X							
ר <del>ז</del> 10000	ราดาสินด้า ราดาสินด้า ราดาสินด้า 10000 10000 10000						
ส่วนลด ส่วนลด ส่วนลด ส่วนลด							
สำนวนผินประ คำนวนผินประ คำนวนผินประ คำนวนผินประ							
אווינע אוויעע אוויעע אוויעט							
	(9)						
1							

โปรแกรมตัวอย่างที่ 10.17 การสร้างโปรแกรมลักษณะส่วนต่อประสานกราฟิกกับผู้ใช้ โดยการทำงานของโปรแกรมให้แสดงผลลัพธ์ของจำนวนเงินที่ต้องชำระหลังหักส่วนลด มีการทำงาน 2 ส่วนประกอบด้วย ส่วนที่เป็นการป้อนข้อมูลราคาสินค้าและส่วนที่ป้อนข้อมูลเปอร์เซ็นต์ส่วนลด เมื่อทำการกดปุ่มจำนวนเงินชำระ โปรแกรมจะทำการประมวลผลและแสดงยอดเงินหลังจาก หักส่วนลด ซึ่งเป็นการประยุกต์ใช้รูปแบบทิศทางการทำงานของโปรแกรมแบบเรียงลำดับ

### ตัวอย่างที่ 10.18 การสร้างส่วนต่อประสานกราฟิกกับผู้ใช้ ทิศทางโปรแกรมแบบทางเลือก

บรรทัดที่	คำสั่ง			
1	from tkinter import *			
2	def calculate(*args):			

3	val_price = float(price.get())						
4	if(val_price>= 10000):						
5	money=val_price-(val_price*10/100)						
6	elif(val_price>= 1000):						
7	money=val_price-(val_price*5/100)						
8	else:						
9	money=val_price						
10	show_money.set(money)						
11	window = Tk()						
12	window.geometry("300x120")						
13	window.title("คำนวณหาส่วนลดจากราคาสินค้า")						
14	price = StringVar()						
15	show_money = StringVar()						
16	money_lbl =Label(text="ราคาสินค้า").pack()						
17	money_entry = Entry(textvariable=price).pack()						
18	money_cal=Button(text="จำนวนเงินชำระ",command=calculate).pack()						
19	window.bind(calculate)						
20	money_show=Label(textvariable=show_money).pack()						
21	Label(text="บาท").pack()						
22	window.mainloop()						
ผลลัพธ์							
🦸 ดำนวณหาล้านลดจากราดา — 🗆 X 🗳 ดำนวณหาล้านลดจากราดา — 🗆 X 🗳 ดำนวณหาล้านลดจากราดา — 🗆 X							
15000 4000 300 จำนวนผินช่างร จำนวนผินช่างร							
135 U	00.0 3800.0 300.0 311 UNI UNI						
07							

โปรแกรมตัวอย่างที่ 10.18 การสร้างโปรแกรมลักษณะส่วนต่อประสานกราฟิกกับผู้ใช้ โดยการทำงานของโปรแกรมให้แสดงผลลัพธ์ของจำนวนเงินที่ต้องชำระหลังหักส่วนลด ด้วยการป้อน ข้อมูลราคาสินค้า โปรแกรมจะทำการตรวจสอบเงื่อนไขในการลดภายในโปรแกรม เมื่อทำการกดปุ่ม จำนวนเงินชำระ โปรแกรมจะทำการประมวลผลและแสดงยอดเงินหลังจากหักส่วนลด ซึ่งเป็นการ ประยุกต์ใช้รูปแบบทิศทางการทำงานของโปรแกรมแบบทางเลือก

บรรทัดที่	คำสั่ง									
1	from tkinter im	from tkinter import *								
2	def calculate(*	def calculate(*args):								
3	val_num = int(num.get())									
1	for k in range(1.13):									
4		for K in range(1,13):								
5	val=k*va	val=k*val_num								
6	Label(te>	kt=(k,"x",val_num,"=",val)).p	back()							
7	window = Tk()									
8	window.geome	window.geometry("200x320")								
9	window.title("แ	window.title("แม่สูตรคูณ")								
10	num= StringVa	num= StringVar()								
11	money_lbl =Label(text="เลข").pack()									
12	money_entry = Entry(textvariable=num).pack()									
13	money_cal=Bu	money_cal=Button(text="แสดงแม่สูตรคูณ",command=calculate).pack()								
14	window.mainlo	window.mainloop()								
ผลลัพธ์										
	C	0								
0 แม่สู	– 🗆 ×	🖉 usią – 🗆 🗙	1 🕼 📖 — 🗆 🛛 🗸							
	เลข	เลข	เลข							
4		8	12							
	1 × 4 = 4	1 - 2 - 2	1 x 12 - 12							
	2 x 4 = 8	2 x 8 = 16	$2 \times 12 = 24$							
	3 x 4 = 12	3 x 8 = 24	3 x 12 = 36							
	4 x 4 = 16 4 x 8 = 32 4 x 12 = 48									
	5 x 4 = 20	5 x 8 = 40	5 x 12 = 60							
	б x 4 = 24	6 x 8 = 48	6 x 12 = 72							
R	7 x 4 = 28	7 x 8 = 56	7 x 12 = 84							
	8 x 4 = 32 0 x 4 = 26	8 x 8 = 64	8 x 12 = 90 0 x 12 = 109							
	$3 \times 4 = 30$	9 x 8 = 72 10 x 8 = 80	$10 \times 12 = 100$							
	11 x 4 = 44	11 x 8 = 88	$10 \times 12 = 120$ $11 \times 12 = 132$							
	12 x 4 = 48 12 x 8 = 96 12 x 12 = 144									

**ตัวอย่างที่ 10.19** การสร้างส่วนต่อประสานกราฟิกกับผู้ใช้ ทิศทางโปรแกรมแบบทำซ้ำ

โปรแกรมตัวอย่างที่ 10.19 การสร้างโปรแกรมลักษณะส่วนต่อประสานกราฟิกกับผู้ใช้ โดยการทำงานของโปรแกรมให้แสดงแม่สูตรคูณ ด้วยการป้อนแม่สูตรคูณที่ต้องการแสดง เมื่อทำการ กดปุ่มแสดงแม่สูตรคูณ โปรแกรมจะทำการประมวลผลและแสดงแม่สูตรคูณตามที่ป้อนข้อมูล ซึ่งเป็น การประยุกต์ใช้รูปแบบทิศทางการทำงานของโปรแกรมแบบทำซ้ำ

### สรุป

ส่วนต่อประสานกราฟิกกับผู้ใช้ หรือจียู่ไอ คือ ลักษณะของการสร้างโปรแกรมที่เป็นลักษณะ แผนภาพ เพื่อให้สะดวกและง่ายต่อผู้ไข้งาน ซึ่งภาษาไพธอน ได้มีเครื่องมือที่เป็นพื้นฐานใช้งานใน ลักษณะของจียู่ไอ เรียกว่า tkinter ซึ่งประกอบไปด้วยส่วนของการทำงาน 3 ส่วน ประกอบด้วย ส่วน ที่ 1 วิดเจ็ต คือ สิ่งที่เป็นวัตถุในโปรแกรม จะมีหน้าที่การทำงานแตกต่างกันไป เช่น Label ทำหน้าที่ เป็นป้ายแสดงข้อความ Entry ทำหน้าที่เป็นกล่องในการรับข้อมูล Button ทำหน้าที่เป็นปุ่มการ ทำงานอย่างใดอย่างหนึ่ง เป็นต้น นอกเหนือจากนี้ยังมีวิดเจ็ต ที่ให้เลือกใช้เป็นจำนวนมากให้เลือกใช้ ตามความเหมาะสมกับงาน ส่วนที่ 2 Geometry Management คือ ส่วนทำหน้าที่ในการจัดวาง ตำแหน่งบนหน้าต่างของโปรแกรมของวิดเจ็ต ซึ่งประกอบรูปแบบการจัดวาง 3 รูปแบบ ได้แก่ pack() grid() และ place() และส่วนที่ 3 คือ Events และ Callbacks ที่จะทำหน้าที่ในการสร้างเหตุการณ์ ของการกระทำของวัตถุที่อยู่ในโปรแกรม ซึ่งเมื่อเกิดการกระทำจากเหตุการณ์ เช่น การกด การพิมพ์ การเลื่อน เป็นต้น เรียกว่า Events และเมื่อหลังเหตุการณ์ จะเกิดการประมวลผล และเกิดผลลัพธ์ ตอบสนองไปยังผู้ใช้อย่างใดอย่างหนึ่ง เรียกว่า Callbacks ทั้ง 3 ส่วน เป็นองค์ประกอบที่สำคัญที่จะ ขาดส่วนใดส่วนหนึ่งไม่ได้ในการทำงานของโปรแกรมในลักษณะของจียู่ไอของภาษาไพธอน โปรแกรม จะสามารถทำงานได้อย่างมีประสิทธิภาพ เกิดจากการพิจารณาการเลือกคุณลักษณะที่เหมาะสมมา ออกแบบและพัฒนาโปรแกรม

## แบบฝึกหัด

1. ให้ผู้เรียนอธิบายลักษณะและประโยชน์ที่ได้ ส่วนต่อประสานกราฟิกกับผู้ใช้

ให้ผู้เรียนบอกเครื่องมือที่ใช้ในการพัฒนาส่วนต่อประสานกราฟิกกับผู้ใช้ในภาษาไพธอน
คืออะไร และมีองค์ประกอบในการทำงานอย่างไร

 ให้ผู้เรียนอธิบายขั้นตอนการสร้างส่วนต่อประสานกราฟิก มีขั้นตอนอย่างไร และแต่ละ ขั้นตอน มีกระบวนการของการทำงานอย่างไร

4. ให้ผู้เรียนอธิบายองค์ส่วนประกอบการทำงานของ tkinter แต่ละส่วน พร้อมทั้งอธิบาย ลักษณะและการทำงานของแต่ละส่วน

5. ให้ผู้เรียนอธิบายความหมายและหน้าที่ของวิดเจ็ต

 ให้ผู้เรียนยกตัวอย่างวิดเจ็ต ที่มีความสำคัญและใช้งานประจำ พร้อมบอกหน้าที่และการ ทำงานของแต่ละวิดเจ็ต

 ให้ผู้เรียนอธิบายถึงวิธีการจัดตำแหน่งของวิดเจ็ต ที่อยู่ในโปรแกรมแต่ละวิธีและอธิบาย ความแตกต่างกันอย่างไร

8. ให้ผู้เรียนเขียนโปรแกรมให้มีลักษณะดังภาพต่อไปนี้

🖉 ประเมินผลการเรียน		_		$\times$
ดะแนนเก็บ	ดะแนนกลางกาด	R	เกาด	
	ประเมินผล			
เกรดที่ได้ =				

9. ให้ผู้เรียนเขียนโปรแกรมแปลงจากองศาฟาเรนไฮต์เป็นองศาเซลเซียส

10. ให้ผู้เรียนเขียนโปรแกรมให้มีการคำนวณตัวเลข 2 จำนวน ตามสัญลักษณ์ที่ป้อน ดังภาพ ตัวอย่างต่อไปนี้

🖉 ดำนวณตัว	มลข				_		×
ตัวเลขที่ 1 5		*		6		ตัวเลข	ฑี่ 2
			ประเมินผล				
	ผลลัพธ์ที่ได้ =		30.0				

### เอกสารอ้างอิง

- ธีรวัฒน์ ประกอบผล. (2558). การเขียนแอพพลิเคชันด้วย Visual Basic 2010. กรุงเทพฯ: ซิมพลิฟลาย.
- ราชบัณฑิตยสภา. ศัพท์บัญญัติสำนักงานราชบัณฑิตยสภา. (ออนไลน์) 10 เมษายน 2564 (อ้างเมื่อ 10 เมษายน 2564). จาก: https://coined-word.orst.go.th/
- สุชาติ คุ้มมณี. (2558). เชี่ยวชาญการเรียนด้วยภาษาไพธอน. มหาสารคาม. คณะวิทยาการ สารสนเทศ: มหาวิทยาลัยมหาสารคาม.
- Chaudhary, B. (2015). Tkinter GUI Application Development Blueprints. Birmingham: Packt Publishing.
- Meier, Burkhard A. (2015). Python GUI Programming Cookbook. Birmingham: Packt Publishing.
- Wikipedia. (2021). Widget. (online) (cited 22 June 2021). Available from: https://th.wikipedia.org/wiki/วิจิท