

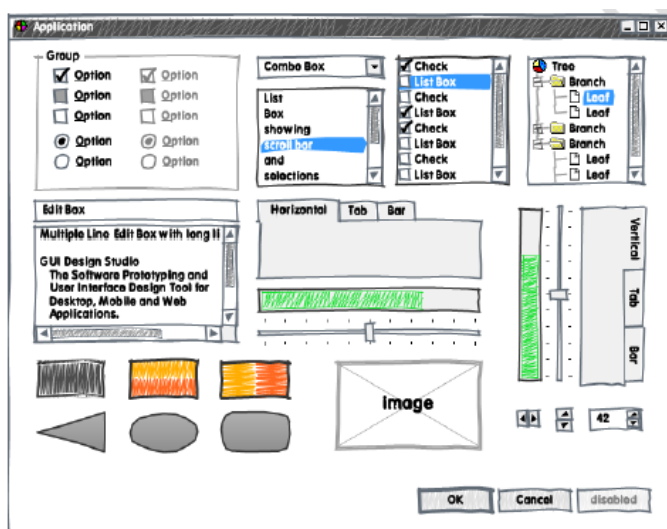


Python : GUI

การเขียนโปรแกรมให้แสดงผลแบบ Graphic User Interface ด้วยภาษา Python

การเขียนโปรแกรมด้วยภาษา Python เมื่อคุณเขียนโปรแกรมเสร็จแล้ว โดยปกติคุณจะต้องสั่งให้โปรแกรมทำงานและแสดงผลได้บนหน้าต่าง Terminal หรือหน้าต่าง PowerShell ซึ่งมักจะแสดงผลออกมาในรูปแบบข้อความตัวอักษรเป็นหลัก หากต้องการพัฒนาโปรแกรมให้แสดงผลแบบหน้าต่างโปรแกรมทั่วไปก็สามารถทำได้ เพราะใน Python มีโมดูลสำหรับพัฒนา GUI (Graphic User Interface) อยู่มากมาย เช่น tkinter, wxPython, JPython, wxWidgets, Qt, Gtk+, FLTK, FOX และ OpenGL เป็นต้น โดยในการพัฒนาครั้งนี้จะเลือกใช้โมดูลที่ชื่อว่า tkinter ซึ่งย่อมาจาก TK Interface จากผู้พัฒนา TK GUI Toolkit ซึ่งพัฒนาเครื่องมือบนระบบปฏิบัติการ UNIX มาก่อน ต่อมาได้พัฒนา tkinter เพื่อให้การสร้าง GUI ใน Python ทำได้ง่ายและสะดวกขึ้น อีกทั้ง tkinter เป็นโมดูลที่มีคำสั่งในการสร้างหน้าต่างโปรแกรม และแสดงผล widget ชนิดต่าง ๆ เช่น Label, Text box, Image, Button ได้หลากหลาย จึงทำให้กลายเป็นโมดูลที่นิยมใช้กันแพร่หลายในหมู่นักพัฒนาโปรแกรมด้วยภาษา Python ทั่วโลก

ก่อนที่จะทำการเขียนโปรแกรมให้แสดงผลแบบ GUI นั้น นักพัฒนาควรจะต้องออกแบบหน้าตาของโปรแกรมก่อน ว่าในโปรแกรมจะมีการแสดงผลข้อมูลหรือ Widget ใด ๆ ในหน้าต่างโปรแกรมบ้าง ซึ่งการออกแบบโปรแกรมนั้นสามารถร่างลงบนกระดาษ หรือ ใช้โปรแกรมในการออกแบบหรือทำตัวอย่างหน้าตาโปรแกรมก่อนก็ได้ เพื่อให้ได้โครงร่างหรือแนวทางในการพัฒนาโปรแกรม จากนั้นค่อยเริ่มพัฒนาในส่วนของการ Coding ต่อไป ซึ่งในขั้นตอนการพัฒนาจริง อาจจะมีการขยับหรือปรับเปลี่ยนรูปแบบ รวมถึงการเพิ่มการแสดงผลหรือ Function ต่าง ๆ เข้ามาด้วยก็ได้



การสร้างโปรแกรมคำนวณดัชนีมวลกายแบบ GUI ด้วยภาษา Python

Workshop ที่ 1 เริ่มต้นสร้างหน้าจอโปรแกรม

```
BMI.py
1  from tkinter import *
2
3  root = Tk()
4
5  root.mainloop()
6
```

1. สร้างไฟล์เดอร์ใหม่ใน Drive C: ตั้งชื่อว่า BMI จากนั้นให้สร้างไฟล์ใหม่ขึ้นมาตั้งชื่อว่า BMI.py
2. ทำการ import โมดูลที่ชื่อว่า tkinter (ชื่อเต็มว่า tk interface) เพื่อเรียกใช้งานคำสั่งเกี่ยวกับ GUI

```
from tkinter import *
```

3. ตั้งชื่อตัวแปรขึ้นมา 1 ตัวเพื่อใช้เรียกแทนชื่อหน้าต่างโปรแกรมหลักหรือหน้าต่าง Windows form หรือหน้าต่าง Main container ว่า root, main, windows, container หรือจะตั้งชื่อว่าอะไรก็ได้ เพื่อให้สื่อถึงลำดับชั้นในการแสดงผลของหน้าต่างโปรแกรมที่เป็นหน้าต่างลำดับแรกสุด ในตัวอย่างนี้จะตั้งชื่อว่า root มีค่าเท่ากับการเรียกใช้งานคำสั่ง Tk() เพื่อสร้างหน้าต่างโปรแกรมขึ้นมา

```
root = Tk()
```

4. เรียกใช้คำสั่ง mainloop เพื่อให้หน้าต่างโปรแกรมเปิดทำงานตลอดเวลา
โดยคำสั่งนี้จะต้องพิมพ์ไว้ด้านล่างสุดของชุดคำสั่งทั้งหมด

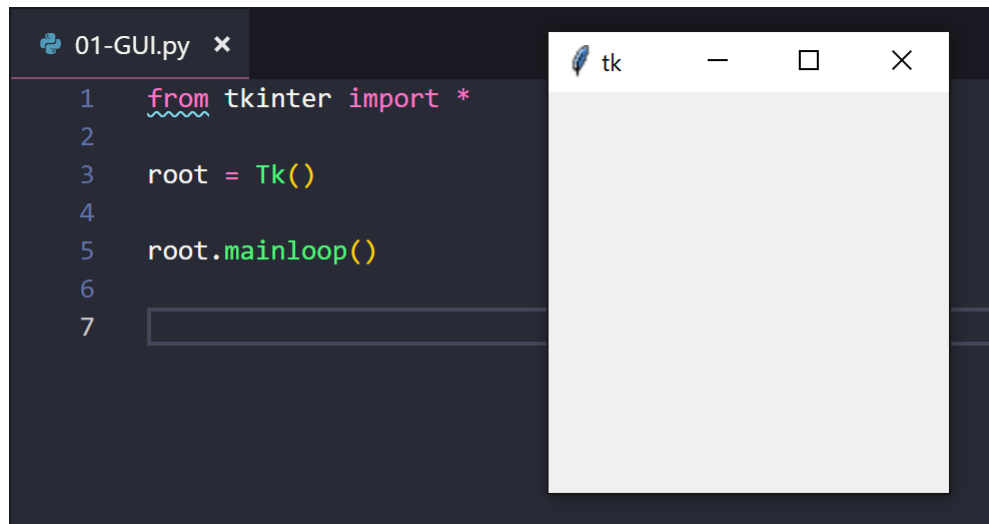
```
root.mainloop()
```

5. ทดลองเรียกหน้าต่างโปรแกรม โดยการคลิกเมาส์ขวา
แล้วคลิกเลือกคำสั่ง Run Python File in Terminal

```
01-GUI.py
1  from tkinter import *
2
3  root = Tk()
4
5  root.mainloop()
6
7
```

Go to Definition	F12
Peek Definition	Alt+F12
Find All References	Shift+Alt+F12
Peek References	Shift+F12
Rename Symbol	F2
Change All Occurrences	Ctrl+F2
Format Document	Shift+Alt+F
Source Action...	
Cut	Ctrl+X
Copy	Ctrl+C
Paste	Ctrl+V
Run Current Unit Test File	
Run Python File in Terminal	
Run Selection/Line in Python Terminal	Shift+Enter

6. จะปรากฏหน้าต่างโปรแกรม tk ขึ้นมาเป็นหน้าต่างว่างเปล่า จากนั้นให้กด X เพื่อปิดโปรแกรม



7. ทำการตั้งชื่อโปรแกรมโดยใช้คำสั่ง title แล้วระบุชื่อโปรแกรมที่ต้องการ

```
from tkinter import *

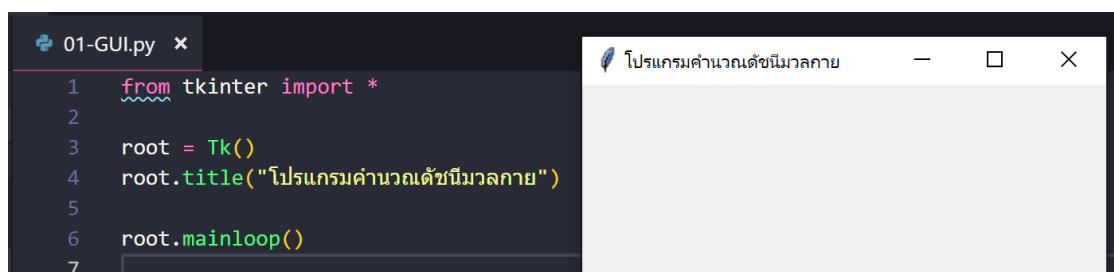
root = Tk()
root.title("โปรแกรมคำนวณดัชนีมวลกาย")

root.mainloop()
```

8. ทดลองเรียกหน้าต่างโปรแกรมอีกครั้ง โดยการคลิกเมาส์ขวา

แล้วคลิกเลือกคำสั่ง Run Python File in Terminal

จากนั้นให้คลิกลากขยายหน้าต่างโปรแกรมให้กว้างขึ้น จะพบกับชื่อโปรแกรมตามที่ระบุ



Workshop ที่ 2 การใส่รูปลงในหน้าจอโปรแกรม

คุณสามารถใส่รูปภาพลงในโปรแกรมของคุณได้โดยใช้คำสั่ง PhotoImage โดยนามสกุลไฟล์รูปภาพที่รองรับและนิยมใช้จะเป็นไฟล์ .png ในตัวอย่างนี้ ให้ Copy ไฟล์รูป bmi-mahidol.png กับ bmi-scale.png ไปเก็บไว้ในโฟลเดอร์ C:\BMI ก่อนที่จะเรียกใช้คำสั่ง PhotoImage

```
from tkinter import *

root = Tk()
root.title("โปรแกรมคำนวณดัชนีมวลกาย")

photo = PhotoImage(file="bmi-mahidol.png")
Label(root, image=photo).pack()

photo2 = PhotoImage(file="bmi-scale.png")
Label(root, image=photo2).pack()

root.mainloop()
```

1. สร้างตัวแปรชื่อ photo แล้วเรียกใช้คำสั่ง PhotoImage เพื่อดึงรูปภาพมาใช้งาน จากนั้นระบุว่าไฟล์รูปที่เรียกใช้งานชื่อ bmi-mahidol.png

```
photo = PhotoImage(file="bmi-mahidol.png")
```

* หากคุณเก็บไฟล์รูปไว้ใน Directory เดียวกันกับไฟล์โปรแกรม

ก็ไม่จำเป็นต้องอ้างอิง Directory Path ของรูปใน file=" "

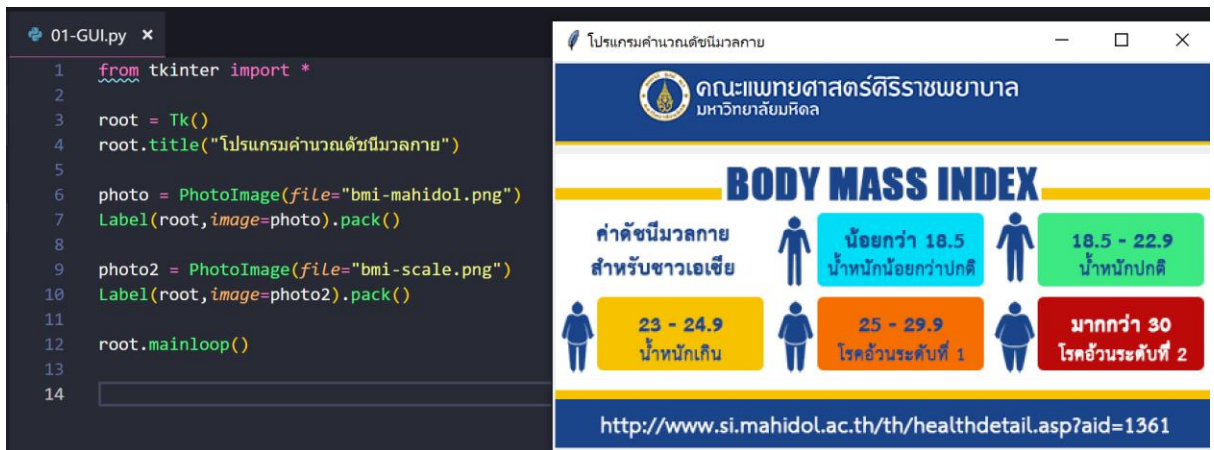
2. เรียกใช้คำสั่ง Label เพื่อสร้าง widget ชนิด Label ขึ้นมา โดยให้แสดงผล Label นี้ในหน้าต่าง root โดยให้แสดงรูปภาพที่รับค่าไฟล์มาจากตัวแปรที่ชื่อว่า photo จากนั้นให้ระบุรูปแบบ geometry ในการแสดงผล เป็นแบบ pack หรือแบบกล่องข้อมูล แสดงกล่องต่าง ๆ เรียงจากด้านบนลงมาด้านล่าง

```
Label(root, image=photo).pack()
```

3. ใส่รูปที่สองลงในโปรแกรม โดยตั้งชื่อตัวแปรว่า photo2 และเรียกใช้งานรูป bmi-scale.png

```
photo2 = PhotoImage(file="bmi-scale.png")
Label(root, image=photo2).pack()
```

4. ทดลอง Run Python File in Terminal คุณจะพบกับหน้าต่างโปรแกรมที่มีรูปภาพแสดงขึ้นมา



Workshop ที่ 3 การใส่ข้อความลงในหน้าจอโปรแกรม

คุณสามารถใส่ข้อความต่าง ๆ ลงไปในหน้าต่างโปรแกรมได้ตามต้องการ โดยใช้งานผ่านคำสั่ง Label ซึ่ง เป็นคำสั่งพื้นฐานที่ใช้งานง่าย นอกจากนี้คุณยังสามารถกำหนดคุณสมบัติต่าง ๆ ให้กับตัวอักษรได้ด้วย

```
from tkinter import *

root = Tk()
root.title("โปรแกรมคำนวณดัชนีมวลกาย")

root.option_add("*Font", "Tahoma 16")

photo = PhotoImage(file="bmi-mahidol.png")
Label(root, image=photo).pack()

Label(root, text="น้ำหนักตัว หน่วยเป็น กิโลกรัม").pack()

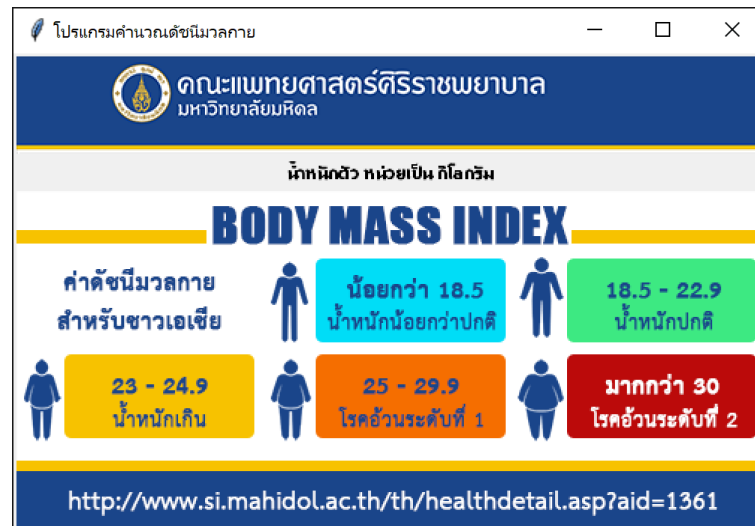
photo2 = PhotoImage(file="bmi-scale.png")
Label(root, image=photo2).pack()

root.mainloop()
```

1. ให้พิมพ์คำสั่ง Label แทรกลงไประหว่างคำสั่งในการแทรกรูปที่ 1 กับคำสั่งในการแทรกรูปที่ 2 เพื่อสร้าง Widget ชนิด Label ขึ้นมา จากนั้นกำหนดให้แสดงผลในหน้าต่าง root โดยใน Label ให้แสดงข้อความคำว่า “น้ำหนักตัว หน่วยเป็นกิโลกรัม”
จากนั้นให้ระบุรูปแบบ geometry ในการแสดงผล เป็นแบบ pack หรือแบบกล่องข้อมูลแสดงกล่องต่าง ๆ เรียงจากด้านบนลงมาด้านล่าง

```
Label(root, text="น้ำหนักตัว หน่วยเป็น กิโลกรัม").pack()
```

2. ทดลอง Run Python File in Terminal คุณจะพบกับหน้าต่างโปรแกรมที่มีข้อความแสดงขึ้นมา



3. ในการเพิ่มข้อความลงไปโปรแกรม ค่าขนาดตัวอักษรเริ่มต้น จะค่อนข้างเล็กทำให้อ่านข้อความได้ลำบาก ให้กำหนดค่าใหม่ให้ Font ทั้งหมดในโปรแกรมเป็นชนิด Tahoma ขนาด 16

```
root.option_add("*Font", "Tahoma 16")
```

4. ทดลอง Run Python File in Terminal คุณจะพบกับหน้าต่างโปรแกรมที่มีข้อความแสดงขึ้นมา



Workshop ที่ 4 การสร้างกล่องรับค่าข้อมูลด้วยการพิมพ์ลงในหน้าจอโปรแกรม

ในการรับค่าข้อมูลด้วยการพิมพ์ ให้ใช้คำสั่ง Entry เพื่อสร้าง Widget ในการรับค่าข้อมูลด้วยการพิมพ์ลงในโปรแกรมของคุณ

```
from tkinter import *

root = Tk()
root.title("โปรแกรมคำนวณดัชนีมวลกาย")
root.option_add("*Font", "Tahoma 16")

weigh_kg = DoubleVar()
height_cm = DoubleVar()

photo = PhotoImage(file="bmi-mahidol.png")
Label(root, image=photo).pack()

Label(root, text="น้ำหนักตัว หน่วยเป็น กิโลกรัม").pack()
ent_kg = Entry(root, textvariable = weigh_kg).pack()

Label(root, text="ส่วนสูง หน่วยเป็น เซนติเมตร").pack()
ent_cm = Entry(root, textvariable = height_cm).pack()

photo2 = PhotoImage(file="bmi-scale.png")
Label(root, image=photo2).pack()

root.mainloop()
```

1. สร้างตัวแปรขึ้นมาใหม่ระหว่างบรรทัดคำสั่ง root.option_add กับคำสั่งในการแทรกรูปที่ 1 ในโปรแกรม ตั้งชื่อว่า weigh_kg กำหนดให้เป็นตัวแปรชนิดที่เป็นค่าจำนวนจริง เพื่อใช้เก็บค่าน้ำหนักที่ป้อนเข้ามา

```
weigh_kg = DoubleVar()
```

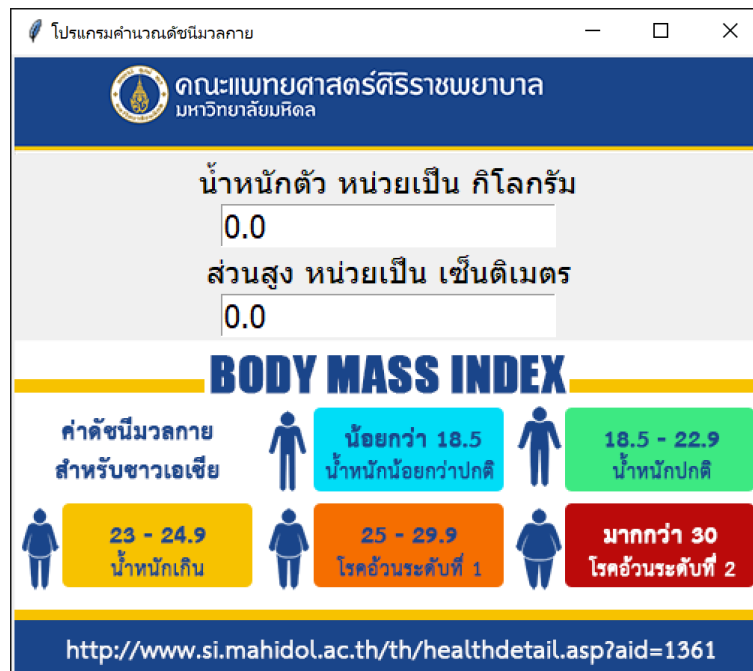
2. สร้างตัวแปรขึ้นมาใหม่ระหว่าง Label “น้ำหนักตัว หน่วยเป็น กิโลกรัม” กับคำสั่งในการแทรกรูปที่ 2 ในโปรแกรม ตั้งชื่อว่า ent_kg (ใช้รับค่าแบบ Entry ในหน่วยกิโลกรัม) ใช้คำสั่ง Entry เพื่อสร้าง Widget เพื่อรับค่าข้อมูล โดยให้ Entry แสดงผลในหน้าต่าง root รับค่าแล้วเอาไปเก็บในและตัวแปรที่ชื่อว่า weigh_kg จากนั้นให้ระบุรูปแบบ geometry ในการแสดงผล เป็นแบบ pack หรือแบบกล่องข้อมูล แสดงกล่องต่าง ๆ เรียงจากด้านบนลงมาด้านล่าง

```
ent_kg = Entry(root, textvariable = weigh_kg).pack()
```

3. ทดลอง Run Python File in Terminal คุณจะพบกับหน้าต่างโปรแกรมที่มีช่องให้พิมพ์ข้อความแสดงขึ้นมา



4. ให้สร้าง Label และ Entry สำหรับรับค่า **ส่วนสูง หน่วยเป็น เซนติเมตร** ขึ้นมาอีก 1 ชุด
5. ทดลอง Run Python File in Terminal คุณจะพบกับหน้าต่างโปรแกรมที่มีช่องให้พิมพ์ข้อความแสดงขึ้นมาทั้งสองช่อง



Workshop ที่ 5 การสร้างปุ่มลงในหน้าจอโปรแกรม

ในการสร้างปุ่มลงไปบนหน้าต่างโปรแกรม จะใช้คำสั่ง Button ในการสร้างปุ่มในรูปแบบต่าง ๆ ซึ่งปุ่มที่สร้างขึ้น สามารถกำหนดรูปแบบให้แสดงผลเป็นปุ่มตัวอักษรธรรมดา หรือปุ่มที่กำหนดสีเส้นของข้อความและสีพื้นหลังได้ หรือแม้กระทั่งการสร้างปุ่มด้วยรูปภาพก็สามารถทำได้เช่นกัน

```
from tkinter import *

root = Tk()
root.title("โปรแกรมคำนวณดัชนีมวลกาย")
root.option_add("*Font", "Tahoma 16")

weigh_kg = DoubleVar()
height_cm = DoubleVar()

photo = PhotoImage(file="bmi-mahidol.png")
Label(root, image=photo).pack()

Label(root, text="น้ำหนักตัว หน่วยเป็น กิโลกรัม").pack()
ent_kg = Entry(root, textvariable = weigh_kg).pack()

Label(root, text="ส่วนสูง หน่วยเป็น เซนติเมตร").pack()
ent_cm = Entry(root, textvariable = height_cm).pack()

btn1 = Button(root , text = " คำนวณค่าBMI ").pack()

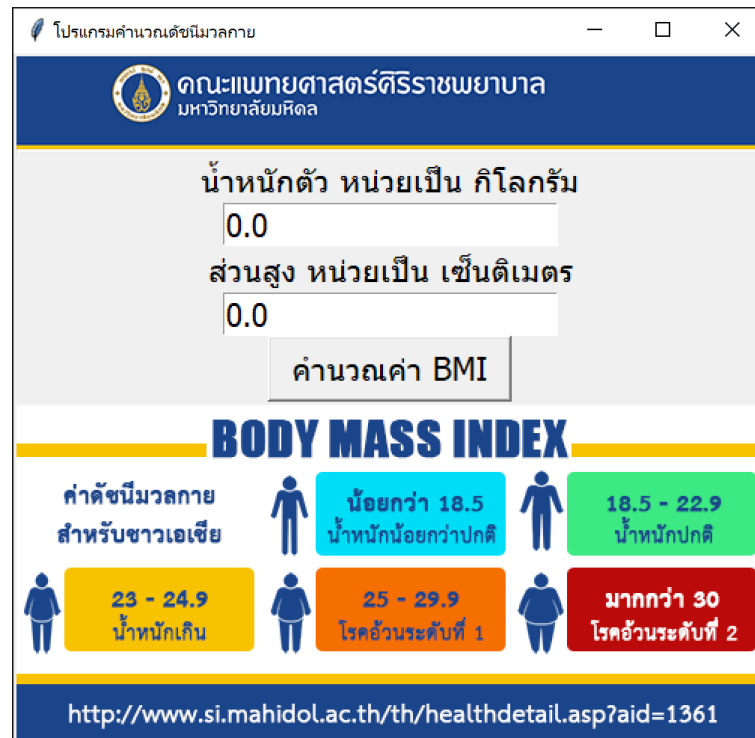
photo2 = PhotoImage(file="bmi-scale.png")
Label(root, image=photo2).pack()

root.mainloop()
```

1. สร้างตัวแปรขึ้นมาใหม่ระหว่างช่องรับค่าข้อมูลส่วนสูงกับคำสั่งในการแทรกที่ 2 ในโปรแกรม ตั้งชื่อตัวแปรว่า btn1 จากนั้นเรียกใช้คำสั่ง Button เพื่อสร้างปุ่มขึ้นมาในหน้าต่าง root โดยให้แสดงข้อความบนปุ่มว่า “ คำนวณค่า BMI ” จากนั้นให้ระบุรูปแบบ geometry ในการแสดงผล เป็นแบบ pack หรือแบบกล่องข้อมูล แสดงกล่องต่าง ๆ เรียงจากด้านบนลงมาด้านล่าง

```
btn1 = Button(root , text = " คำนวณค่าBMI ").pack()
```

2. ทดลอง Run Python File in Terminal คุณจะพบกับหน้าต่างโปรแกรมที่มีปุ่มเพิ่มขึ้นมา



Workshop ที่ 6 การสร้างข้อความแสดงผลจากการคำนวณลงในหน้าจอโปรแกรม

การแสดงผลจากการคำนวณลงในหน้าจอโปรแกรมนั้นสามารถทำได้หลายรูปแบบ ในตัวอย่างนี้จะแสดงผลเป็นข้อความธรรมดา ผ่าน Widget ที่ชื่อว่า Label

```
from tkinter import *

root = Tk()
root.title("โปรแกรมคำนวณดัชนีมวลกาย")
root.option_add("*Font", "Tahoma 16")

weigh_kg = DoubleVar()
height_cm = DoubleVar()
bmi_val = StringVar()

photo = PhotoImage(file="bmi-mahidol.png")
Label(root, image=photo).pack()

Label(root, text="น้ำหนักตัว หน่วยเป็น กิโลกรัม").pack()
ent_kg = Entry(root, textvariable=weigh_kg).pack()

Label(root, text="ส่วนสูง หน่วยเป็น เซนติเมตร").pack()
ent_cm = Entry(root, textvariable=height_cm).pack()
```

```
btn1 = Button(root , text = " คำนวณค่า BMI ").pack()

Label(root,text="ค่าดัชนีมวลกาย เท่ากับ").pack()
Label(root,textvariable = bmi_val).pack()

photo2 = PhotoImage(file="bmi-scale.png")
Label(root,image=photo2).pack()

root.mainloop()
```

1. สร้างตัวแปรขึ้นมาใหม่ให้อยู่ใต้บรรทัดตัวแปร weigh_kg, hight_cm โดยตั้งชื่อตัวแปรว่า bmi_val เพื่อเอาไว้ใช้เก็บค่า BMI ที่ได้จากการคำนวณ จากนั้นให้กำหนดค่าตัวแปรนี้เป็นชนิด StringVar() ใช้เก็บค่าตัวแปรเป็นตัวอักษร เพื่อให้ง่ายในการกำหนดค่าในการแสดงผลตัวเลข

```
bmi_val = StringVar()
```

2. ใช้คำสั่ง Label ระหว่างปุ่ม คำนวณค่า BMI กับคำสั่งในการแทรกกรูปีที่ 2 ในโปรแกรมเพื่อสร้าง Widget สำหรับแสดงข้อความลงในหน้าต่าง root โดยแสดงข้อความคำว่า “ค่าดัชนีมวลกาย เท่ากับ” จากนั้นให้ระบุรูปแบบ geometry ในการแสดงผล เป็นแบบ pack

```
Label(root,text="ค่าดัชนีมวลกาย เท่ากับ").pack()
```

3. ใช้คำสั่ง Label เพื่อสร้าง Widget ลงในหน้าต่าง root สำหรับแสดงข้อความที่รับค่ามาจากตัวแปร bmi_val จากนั้นให้ระบุรูปแบบ geometry ในการแสดงผล เป็นแบบ pack

```
Label(root,textvariable = bmi_val).pack()
```

4. ทดลอง Run Python File in Terminal คุณจะพบกับหน้าต่างโปรแกรมที่มีข้อความผลลัพธ์เพิ่มขึ้น



Workshop ที่ 7 การเขียนโปรแกรมเพื่อการรับค่าจาก Entry นำไปคำนวณและแสดงผล

หลังจากที่ได้เขียนโปรแกรมเพื่อวาง Widget ต่าง ๆ ลงไปในหน้าต่างโปรแกรมเรียบร้อยแล้ว ส่วนสำคัญต่อมาคือเรื่องการเขียนโปรแกรมให้สามารถรับค่าจากการป้อนข้อมูลแล้วนำไปคำนวณประมวลผลและนำผลลัพธ์ที่ได้ออกมาแสดง ซึ่งในขั้นตอนนี้จะมีการเขียน function ใหม่เพื่อใช้ในการคำนวณ และเขียนคำสั่งเพิ่มเติมลงในคำสั่งที่มีอยู่เดิมอีกเล็กน้อยเพื่อให้โปรแกรมทำงานและแสดงผลได้อย่างสมบูรณ์

1. สร้าง Function ในการคำนวณขึ้นมาใหม่ โดยตั้งชื่อว่า on_click

```
def on_click():  
    height_m = height_cm.get()/100  
    height_m_sq = height_m**2  
    bmi = weigh_kg.get() / height_m_sq  
    bmi_val.set(f'{bmi:.2f} bmi.')
```

2. สร้างตัวแปรขึ้นมาใหม่ชื่อว่า height_m โดยให้ height_m มีค่าเท่ากับ ค่าที่ได้จากการดึงค่าตัวแปร height_cm ในช่องรับข้อมูลความสูงที่มีหน่วยเป็นเซนติเมตรนำมาหารด้วย 100 เพื่อแปลงค่าจากหน่วยเซนติเมตรให้เป็นหน่วยเมตร

```
height_m = height_cm.get()/100
```

3. สร้างตัวแปรขึ้นมาใหม่ชื่อว่า height_m_sq โดยให้ height_m_sq มีค่าเท่ากับ ค่าที่ได้จากตัวแปร height_m นำมายกกำลังสอง

```
height_m_sq = height_m**2
```

4. สร้างตัวแปรขึ้นมาใหม่ชื่อว่า bmi โดยให้ bmi มีค่าเท่ากับ ค่าที่ได้จากการดึงค่าตัวแปร weigh_kg ในช่องรับข้อมูลน้ำหนักที่มีหน่วยเป็นกิโลกรัม นำมาหารด้วยค่าตัวแปร height_m_sq

```
bmi = weigh_kg.get() / height_m_sq
```

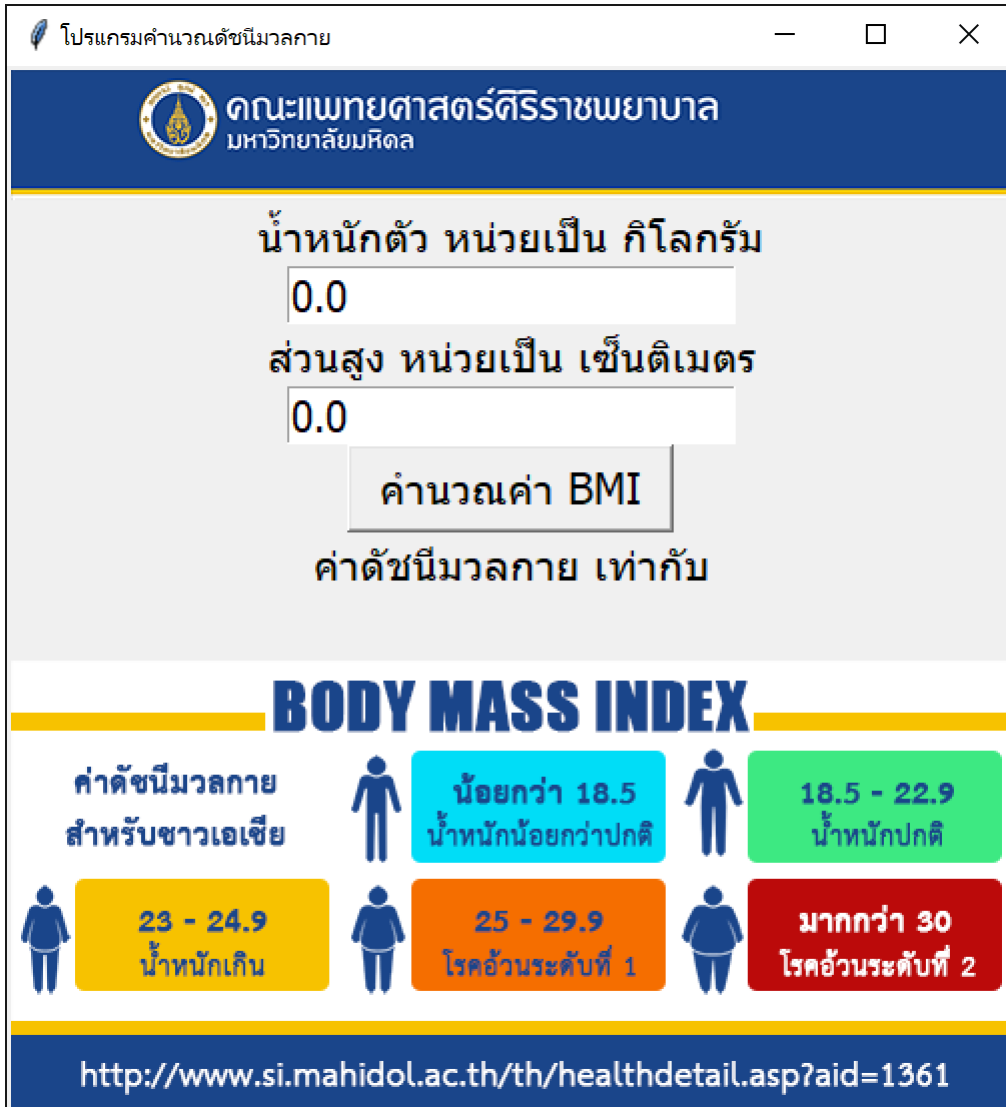
5. สร้างตัวแปรขึ้นมาใหม่ชื่อว่า bmi_val โดยกำหนดให้ bmi_val มีค่าเท่ากับค่าของตัวแปร bmi และเพื่อให้ง่ายต่อการนำค่าตัวเลขที่ได้ไปแสดงผล ให้เพิ่มคุณสมบัติ f เพื่อกำหนด format ของค่าตัวแปรเป็น .2f เพื่อให้โปรแกรมแสดงผลตัวเลขโดยมีจุดทศนิยมเพียงแค่ 2 ตำแหน่งเท่านั้น จากนั้นพิมพ์ข้อความ BMI ต่อท้ายเพื่อนำไปแสดงผลหลังค่าที่ได้จากการคำนวณ

```
bmi_val.set(f'{bmi:.2f} BMI')
```

6. ให้ทำการแก้ไขคำสั่งในบรรทัดของตัวแปร btn1 ที่เคยสร้างไว้ โดยให้เพิ่มคำสั่ง command=on_click ลงไปเพื่อให้โปรแกรมนำค่าที่ได้จากการกรอกข้อมูลน้ำหนักและส่วนสูงไปประมวลผลในฟังก์ชันที่ชื่อ on_click ที่เพิ่งสร้างไว้ แล้วนำค่าที่ได้กลับมาแสดง

```
btn1 = Button(root,command=on_click,text = " คำนวณค่า BMI ").pack()
```

7. ทดลอง Run Python File in Terminal คุณจะพบกับหน้าต่างโปรแกรมที่สามารถกรอกข้อมูลแล้วนำค่าที่ได้ไปประมวลผล แล้วนำผลลัพธ์ที่ได้กลับมาแสดงเป็นข้อความ



หน้าต่างโปรแกรมที่สามารถคำนวณดัชนีมวลกายได้

Workshop ที่ 8 การกำหนดคุณสมบัติเพิ่มเติมเพื่อปรับแต่งหน้าต่าง Widget ในโปรแกรม

หลังจากที่ได้เขียนโปรแกรมให้ทำงานได้เรียบร้อยแล้ว ส่วนที่เหลือคือการกำหนดคุณสมบัติเพิ่มเติมเพื่อปรับแต่งหน้าต่างของ Widget ให้แสดงผลได้เรียบร้อยและสวยงาม โดยมีตัวอย่างการปรับแต่งดังนี้

1. การเปลี่ยนสีตัวอักษรใน Widget

ให้เพิ่มคำสั่ง `fg="` ชื่อสี หรือ รหัสค่าสีฐานสิบหก " ลงไปแล้วลองเรียกใช้งานโปรแกรม

```
Label(root, text="น้ำหนักตัว หน่วยเป็น กิโลกรัม", fg='red').pack()
```

ชื่อสี : <http://www.science.smith.edu/dftwiki/images/3/3d/TkInterColorCharts.png>

2. การกำหนดสีพื้นหลังให้กับ Widget

ให้เพิ่มคำสั่ง `bg=` ชื่อสี หรือ รหัสค่าสีฐานสิบหก ” ลงไปแล้วลองเรียกใช้งานโปรแกรม

```
Label(root, text="ส่วนสูง หน่วยเป็น เซนติเมตร", bg='deep sky blue').pack()
```

3. การกำหนดความกว้างของ Widget

ให้เพิ่มคำสั่ง `width=`ค่าความกว้างเป็นจำนวนตัวอักษร เช่น 23

```
ent_kg = Entry(root, textvariable = weigh_kg, width = 23).pack()
```

4. การกำหนดค่าเส้นขอบให้กับ Widget

ให้เพิ่มคำสั่ง `borderwidth=`เลขขนาดของเส้นขอบ โดยค่าเริ่มต้นที่ 0 แล้วเพิ่มขึ้นไปตามต้องการ

```
btn1 = Button(root, command=on_click, text = " คำนวณค่า BMI ",  
borderwidth=0).pack()
```

5. การกำหนดระยะห่างระหว่าง Widget

ให้เพิ่มคำสั่ง `padx` และ `pady` พร้อมค่าพารามิเตอร์ลงไปใน option ของ pack ดังนี้

```
ent_kg = Entry(root, textvariable = weigh_kg, width = 23).pack(padx = 10  
, pady = 10)
```

6. การกำหนดให้ตัวอักษรขึ้นบรรทัดใหม่ใน Widget

ให้เพิ่มคำสั่ง `\n` เข้าไปด้านหน้าข้อความที่ต้องการให้ขึ้นบรรทัดใหม่

```
Label(root, text="\nค่าดัชนีมวลกาย เท่ากับ").pack()
```

แหล่งเรียนรู้เพิ่มเติม

คุณสามารถศึกษารายละเอียดอื่น ๆ เพิ่มเติมเกี่ยวกับการใช้งาน tkinter ได้ที่

<https://docs.python.org/3/library/tk.html>

หรือค้นหาข้อมูลด้วยคำว่า Python tkinter

Coding โปรแกรมคำนวณดัชนีมวลกาย

ให้สร้างโฟลเดอร์ชื่อ BMI ใน Drive C: เพื่อใช้จัดเก็บไฟล์โปรแกรม จากนั้นให้สร้างไฟล์โปรแกรมขึ้นมาใหม่ ตั้งชื่อว่าไฟล์ว่า BMI.py และให้นำไฟล์รูป bmi-mahidol.png และ ไฟล์ bmi-scale.png มาเก็บไว้ในโฟลเดอร์ BMI เช่นกัน จากนั้นให้ทดลองใช้ Code ดังตัวอย่างด้านล่างนี้เขียนโปรแกรม

```
from tkinter import *

def on_click():
    height_m = height_cm.get()/100
    height_m_sq = height_m**2
    bmi = weigh_kg.get() / height_m_sq
    bmi_val.set(f'{bmi:.2f} BMI')

root = Tk()
root.title("โปรแกรมคำนวณดัชนีมวลกาย")
root.option_add("*Font", "Tahoma 16")

weigh_kg = DoubleVar()
height_cm = DoubleVar()
bmi_val = StringVar()

photo = PhotoImage(file="bmi-mahidol.png")
Label(root, image=photo).pack()

Label(root, text="น้ำหนักตัว หน่วยเป็น กิโลกรัม", fg='red').pack()
ent_kg = Entry(root, textvariable = weigh_kg, width = 23).pack(pady = 10)

Label(root, text="ส่วนสูง หน่วยเป็น เซนติเมตร", fg='navy').pack()
ent_cm = Entry(root, textvariable = height_cm, width = 23).pack(pady = 10)

btn1 = Button(root, command=on_click, text = " คำนวณค่า BMI ", width= 21,
borderwidth=1, bg='deep sky blue').pack()

Label(root, text="\nค่าดัชนีมวลกาย เท่ากับ").pack()
Label(root, textvariable = bmi_val).pack(pady = 20)

photo2 = PhotoImage(file="bmi-scale.png")
Label(root, image=photo2).pack()

root.mainloop()
```

หมายเหตุ : จะต้องมียไฟล์รูปภาพที่ชื่อ bmi-mahidol.png และ bmi-scale.png บันทึกไว้ใน Directory เดียวกันกับไฟล์ BMI.py ด้วย โปรแกรมจึงจะแสดงผลได้ถูกต้อง

โปรแกรมคำนวณดัชนีมวลกาย

 คณะแพทยศาสตร์ศิริราชพยาบาล
มหาวิทยาลัยมหิดล

น้ำหนักตัว หน่วยเป็น กิโลกรัม

ส่วนสูง หน่วยเป็น เซนติเมตร

คำนวณค่า BMI

ค่าดัชนีมวลกาย เท่ากับ

BODY MASS INDEX

ค่าดัชนีมวลกาย สำหรับชาวเอเชีย		น้อยกว่า 18.5 น้ำหนักน้อยกว่าปกติ		18.5 - 22.9 น้ำหนักปกติ	
	23 - 24.9 น้ำหนักเกิน		25 - 29.9 โรคอ้วนระดับที่ 1		มากกว่า 30 โรคอ้วนระดับที่ 2

<http://www.si.mahidol.ac.th/th/healthdetail.asp?aid=1361>

หน้าตาโปรแกรมคำนวณดัชนีมวลกาย





โจทย์ในการพัฒนาโปรแกรมเพิ่มเติม

ให้เพิ่มเงื่อนไขในการแสดงผลค่า BMI ดังต่อไปนี้

หากค่า BMI น้อยกว่า 18.5	ให้แสดงสีพื้นหลังข้อความเป็นสี 00ddf7
หากค่า BMI 18.5 – 22.90	ให้แสดงสีพื้นหลังข้อความเป็นสี 3de982
หากค่า BMI 23 – 24.90	ให้แสดงสีพื้นหลังข้อความเป็นสี f7c200
หากค่า BMI 25 – 29.90	ให้แสดงสีพื้นหลังข้อความเป็นสี f56e00
หากค่า BMI มากกว่า 30 ขึ้นไป	ให้แสดงสีพื้นหลังข้อความเป็นสี bb0a0a