



**KSU AI Excellence**  
Excellence Center of Applied Artificial Intelligence



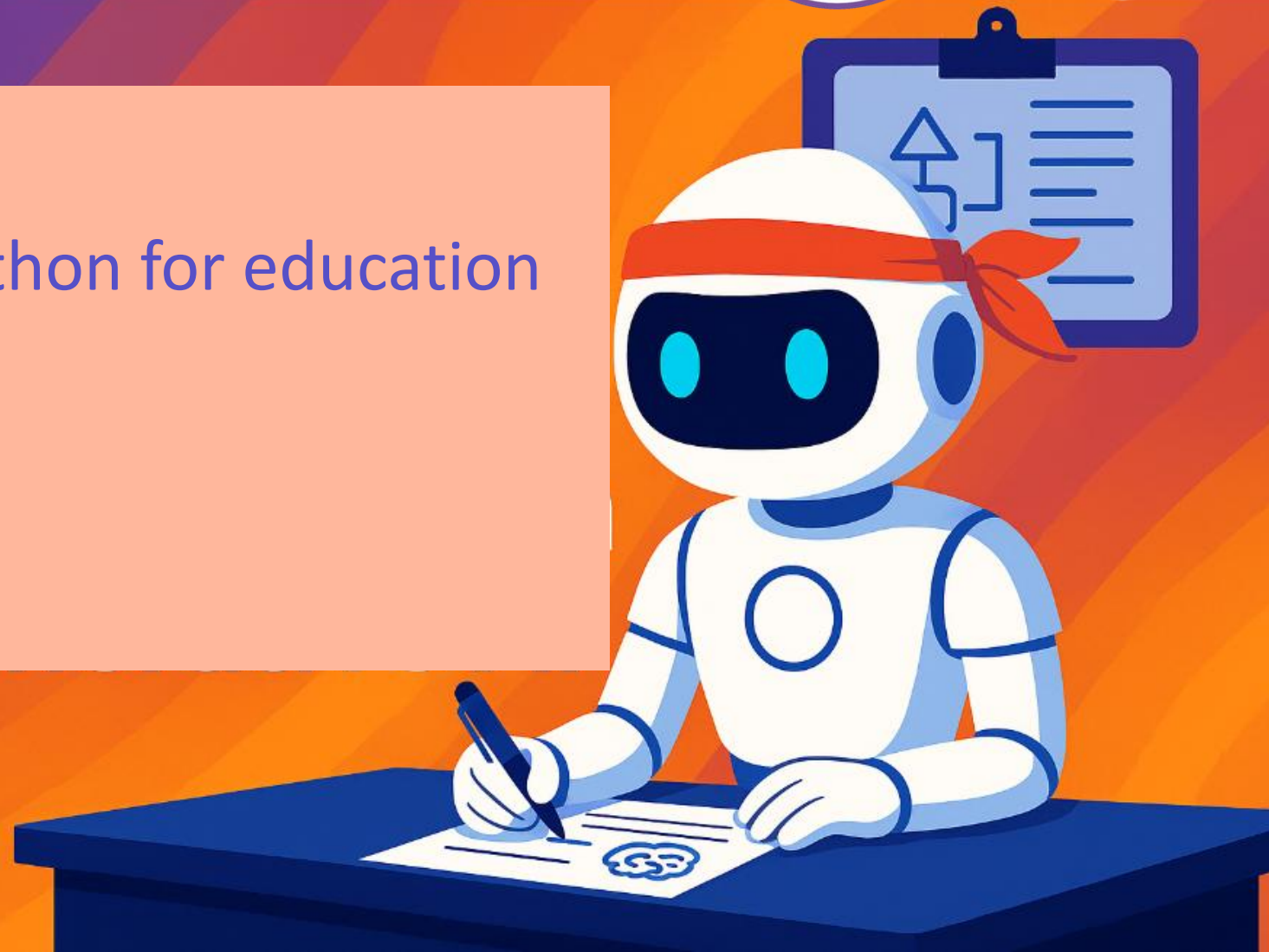
วิทยาการ

อาจารย์ธนกร ญาณกาย  
อาจารย์ภูริ จันทิมา

สาขาวิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์และ  
เทคโนโลยีอุตสาหกรรม

Python for education



# PYTHON SYNTAX

คณิตศาสตร์เบื้องต้น + - \* / \*\* %

```
print (100+50)
```

```
print (100*50)
```

```
print (100/50)
```

```
print (50**2)
```

```
print (53**5)
```

# PYTHON SYNTAX

คำสั่งที่ใช้ในการแสดงข้อมูล

```
print("Hello World")
```

```
print("Student")
```

```
print(100)
```

```
print(50)
```

- การคอมเมนต์ใช้ #

# PYTHON SYNTAX

คำสั่งที่ใช้ในการแสดงข้อมูล

```
print ("ตัวเลข 100 บวกกับ 50 จะได้")
```

```
print ("ผลลัพธ์คือ", 100+50)
```

- การคอมเมนต์ใช้ #

# PYTHON SYNTAX

สร้างสิ่งที่ใช้เก็บข้อมูลของเรา นั่นคือ ตัวแปร การประกาศตัวแปร ทำดังนี้

```
my_name="Thanakorn yarnguy"
```

```
number1=30
```

```
number2=25
```

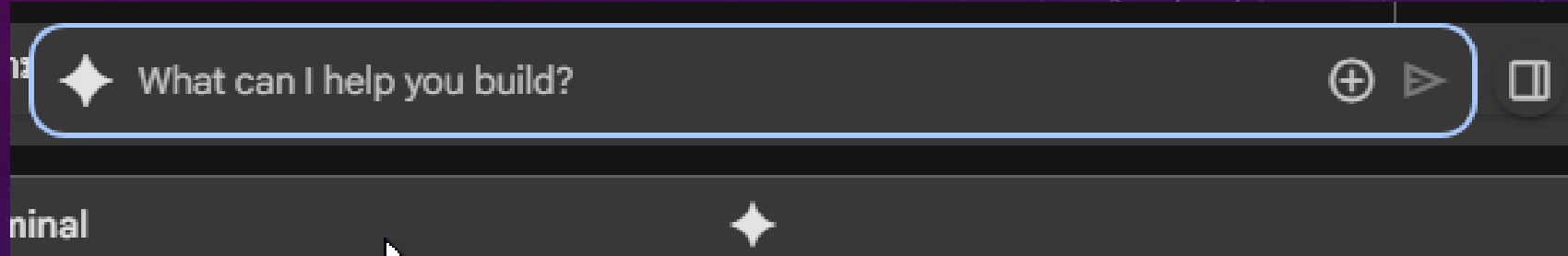
```
print ("ฉันชื่อ",my_name)
```

```
print ("เลขตัวที่ 1 คือ",number1)
```

```
print ("เลขตัวที่ 2 คือ",number2)
```



# GEMINI PROMPT



- แสดงข้อความ ว่า คนที่ 1 มีน้ำหนักเท่ากับ *number1* คนที่ 2 มีน้ำหนักเท่ากับ *number2* หน่วยเป็นกิโลกรัม

```
print(f"คนที่ 1 มีน้ำหนักเท่ากับ {number1} กิโลกรัม คนที่ 2 มีน้ำหนักเท่ากับ {number2} กิโลกรัม")
```

คนที่ 1 มีน้ำหนักเท่ากับ 30 กิโลกรัม คนที่ 2 มีน้ำหนักเท่ากับ 25 กิโลกรัม

# GEMINI PROMPT

- #ถ้าคนที่ 1 น้ำหนักมากกว่าคนที่ 2 ให้แสดงข้อความว่า คนที่ 1 น้ำหนักมากกว่าคนที่ 2 ถ้าไม่ใช่ ให้แสดงข้อความกลับกัน

```
If number1 > number2:  
    print ("คนที่ 1 น้ำหนักมากกว่าคนที่ 2")  
else:  
    print ("คนที่ 2 น้ำหนักมากกว่าคนที่ 1")
```

# PYTHON SYNTAX

- List คือ รายการที่เก็บข้อมูลได้มากกว่า **1** ค่า

```
mylist = ["apple", "banana", "cherry"]
```

- Index

```
print(mylist)  
print(mylist[0])  
print(mylist[1])
```



# GEMINI PROMPT

- สร้าง *List* ของรายชื่อเพื่อน จำนวน 15 คน เป็นภาษาอังกฤษ
- สร้าง *list* สำหรับ เก็บค่าคะแนน จำนวน 15 ค่า
- ให้แสดงข้อความชื่อเพื่อนและคะแนนที่ได้ เรียงตามลำดับใน *list*

```
➡ Alice: 85  
Bob: 90  
Charlie: 78  
David: 92  
Eve: 88  
Frank: 76  
Grace: 95  
Heidi: 89  
Ivan: 81  
Jack: 65  
Kate: 74  
Liam: 69  
Mia: 59  
Noah: 81  
Olivia: 61
```

# GEMINI PROMPT

- สร้าง **List** ของรายชื่อเพื่อน จำนวน 15 คน  
เป็นภาษาอังกฤษ
- สร้าง **list** สำหรับ เก็บค่าคะแนน จำนวน 15 ค่า
- ให้แสดงข้อความชื่อเพื่อนและคะแนนที่ได้  
เรียงตามลำดับใน **list**

```
friends = [  
    "Alice", "Bob", "Charlie", "David", "Eve",  
    "Frank", "Grace", "Heidi", "Ivan", "Judy",  
    "Kevin", "Liam", "Mia", "Noah", "Olivia"  
]  
  
scores = [  
    85, 92, 78, 65, 95,  
    70, 88, 72, 80, 90,  
    60, 83, 75, 89, 91  
]  
  
print("--- Friend Scores ---")  
for i in range(len(friends)):  
    print(f"{friends[i]}: {scores[i]} points")
```

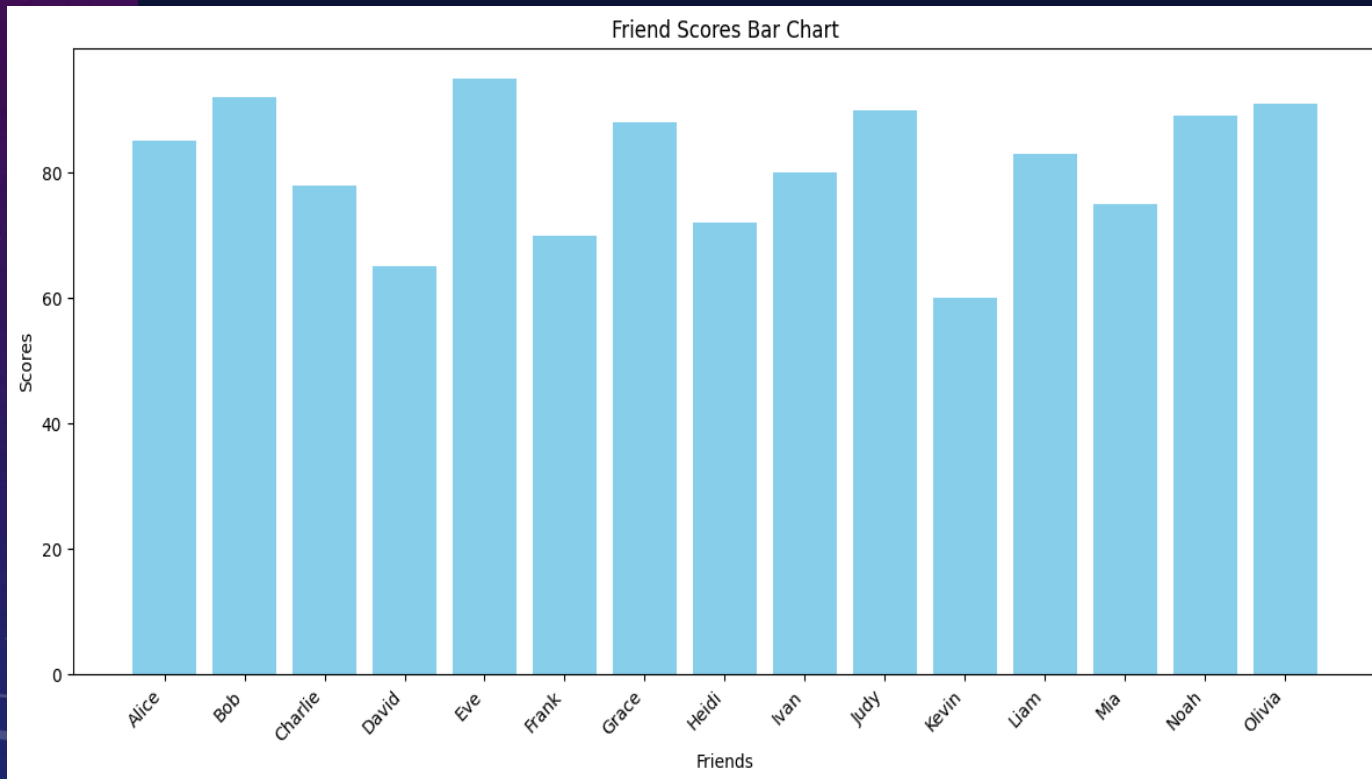
# GEMINI PROMPT

- ให้แสดงชื่อเพื่อนและคะแนนของ คนที่ได้คะแนนมากที่สุด น้อยที่สุด และคะแนนเฉลี่ย แสดงข้อความเป็นภาษาไทย

➡ เพื่อนที่ได้คะแนนมากที่สุดคือ Grace ได้คะแนน 95  
เพื่อนที่ได้คะแนนน้อยที่สุดคือ Mia ได้คะแนน 59  
คะแนนเฉลี่ยคือ 78.87

# GEMINI PROMPT

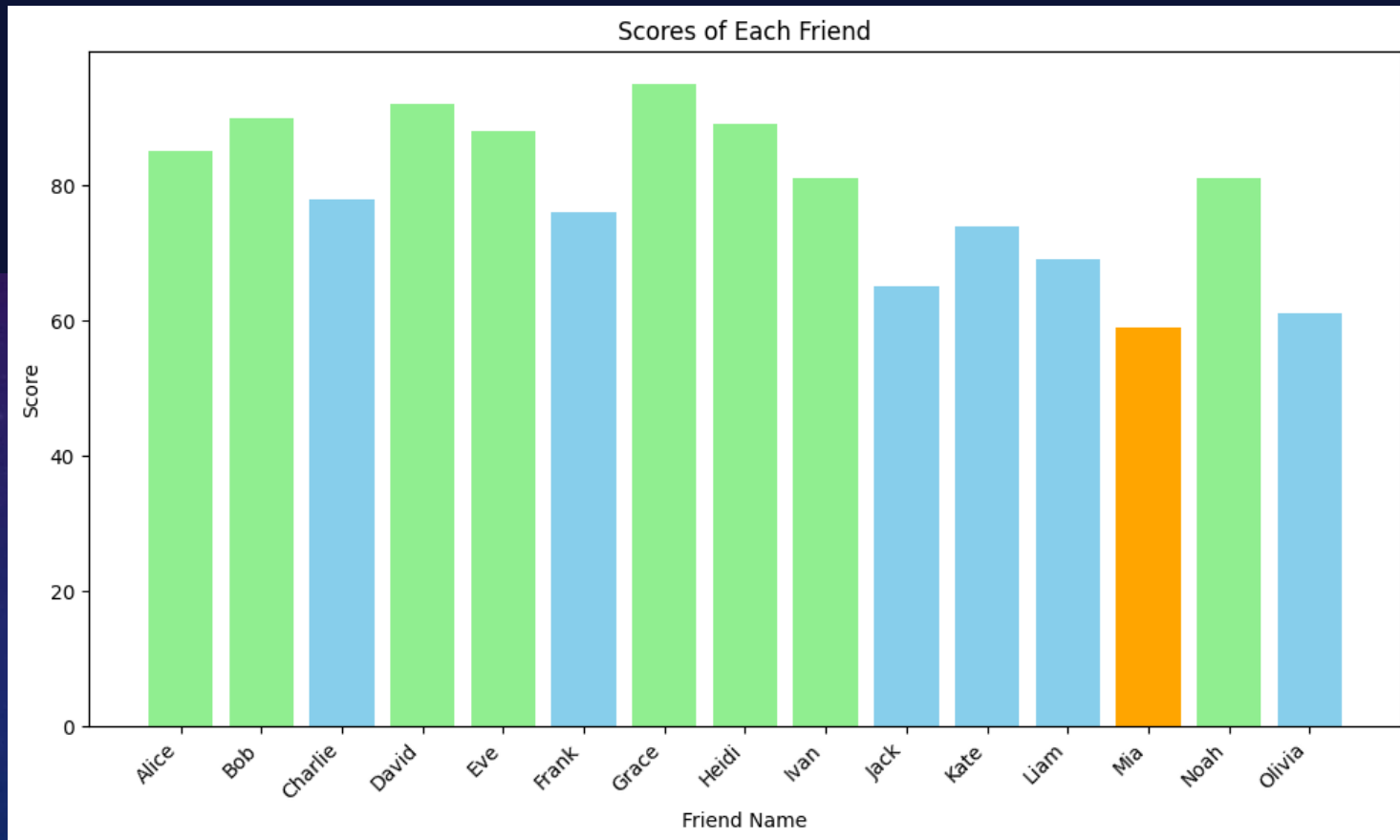
- ช่วยสร้าง กราฟแท่งของคะแนน ชื่อแทนเป็นภาษาอังกฤษ



```
import matplotlib.pyplot as plt
plt.figure(figsize=(12, 6))
plt.bar(friends, scores,
color='skyblue')
plt.xlabel('Friends')
plt.ylabel('Scores')
plt.title('Friend Scores Bar
Chart')
plt.xticks(rotation=45,
ha='right') # plt.tight_layout()
plt.show()
```

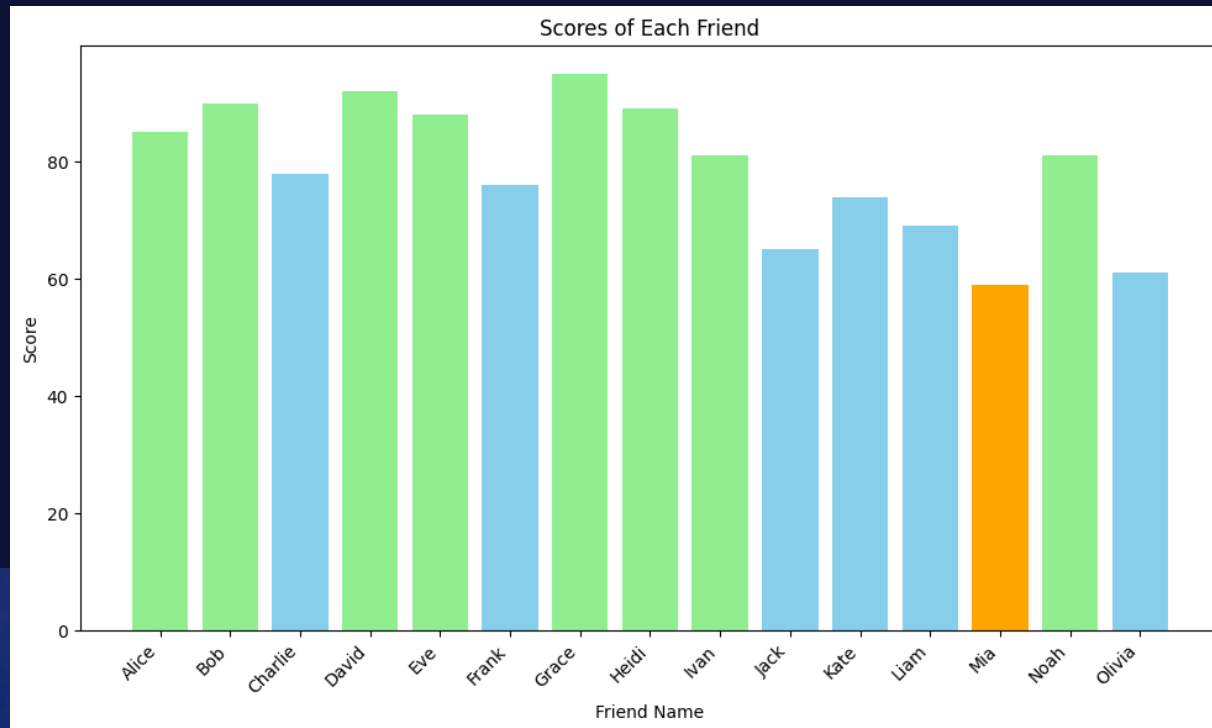
# GEMINI PROMPT

- ช่วยสร้าง กราฟแท่งของคะแนน ชื่อแทนเป็นภาษาอังกฤษ



# GEMINI PROMPT

- ปรับแก้กราฟแท่งถ้าคะแนน ต่ำกว่า 60 ให้เป็นสีส้ม มากกว่า 80 ให้เป็นสีเขียว นอกจากนั้นสีฟ้าอ่อน

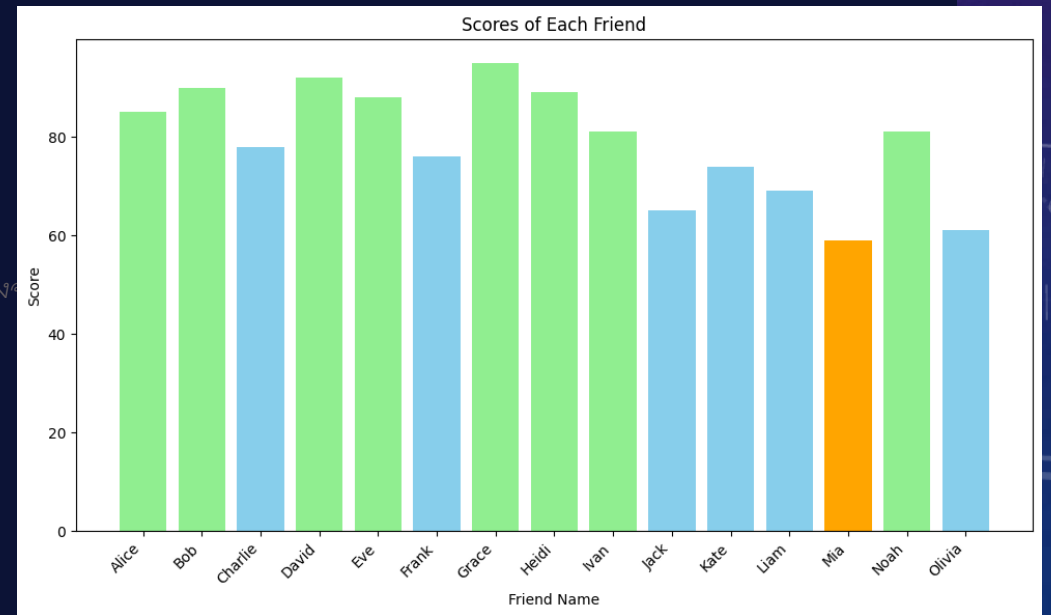




# GEMINI PROMPT

```
import matplotlib.pyplot as plt
import numpy as np

plt.figure(figsize=(10, 6))
colors = ['skyblue'] * len(scores) # เริ่มต้นด้วยสีฟ้าอ่อนทั้งหมด
for i, score in enumerate(scores):
    if score < 60:
        colors[i] = 'orange'
    elif score > 80:
        colors[i] = 'lightgreen' # ใช้ lightgreen เพื่อให้แตกต่าง
plt.bar(friends, scores, color=colors)
plt.xlabel("Friend Name")
plt.ylabel("Score")
plt.title("Scores of Each Friend")
plt.xticks(rotation=45, ha='right')
plt.tight_layout()
plt.show()
```

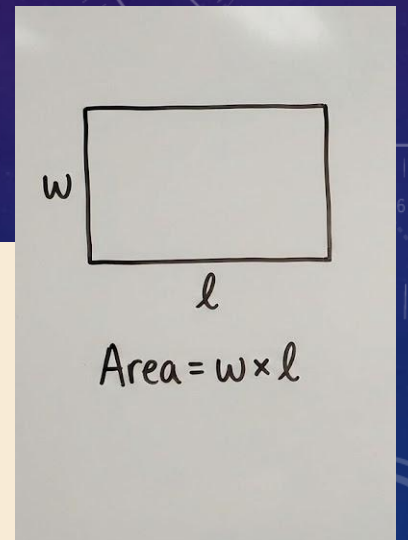


# PYTHON-คำนวณพื้นที่อย่างง่าย

การหาพื้นที่ต้องใช้สูตรทางคณิตศาสตร์ที่แตกต่างกันไปตามรูปทรง ใน Python เราสามารถใช้โมดูล **math** เพื่อดึงค่า พาย (pi) มาใช้ในการคำนวณพื้นที่วงกลมได้

คำนวณพื้นที่สี่เหลี่ยมผืนผ้า (กว้าง X ยาว)

```
width = float(input("ป้อนความกว้าง: "))
length = float(input("ป้อนความยาว: "))
#คำนวณพื้นที่
area = width * length
#แสดงผลลัพธ์
print("พื้นที่สี่เหลี่ยมผืนผ้าคือ:", area)
```

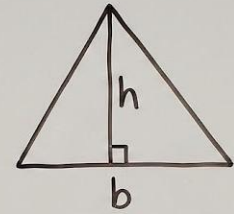


# PYTHON-คำนวณพื้นที่อย่างง่าย

คำนวณพื้นที่สามเหลี่ยม (กว้าง X ยาว)

“โค้ด คำนวณพื้นที่สามเหลี่ยม โดยรับ input”

```
import math
# รับค่าฐานและความสูงจากผู้ใช้
base = float(input("ป้อนความยาวฐานของสามเหลี่ยม: "))
height = float(input("ป้อนความสูงของสามเหลี่ยม: "))
# คำนวณพื้นที่สามเหลี่ยม
area = 0.5 * base * height
# แสดงผลลัพธ์
print("พื้นที่สามเหลี่ยมคือ:", area)
```

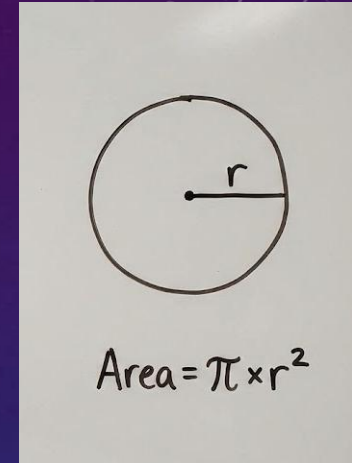


$$\text{Area} = \frac{1}{2} \times b \times h$$

# PYTHON-คำนวณพื้นที่อย่างง่าย

คำนวณพื้นที่วงกลม

“ได้ด คำนวณพื้นที่วงกลม โดยรับ input”

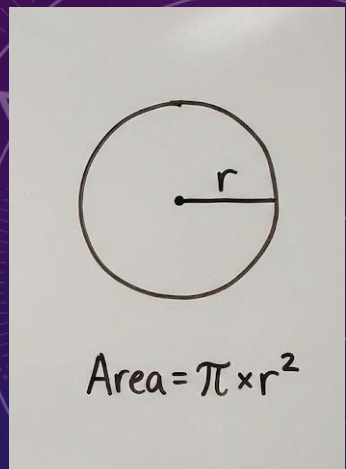
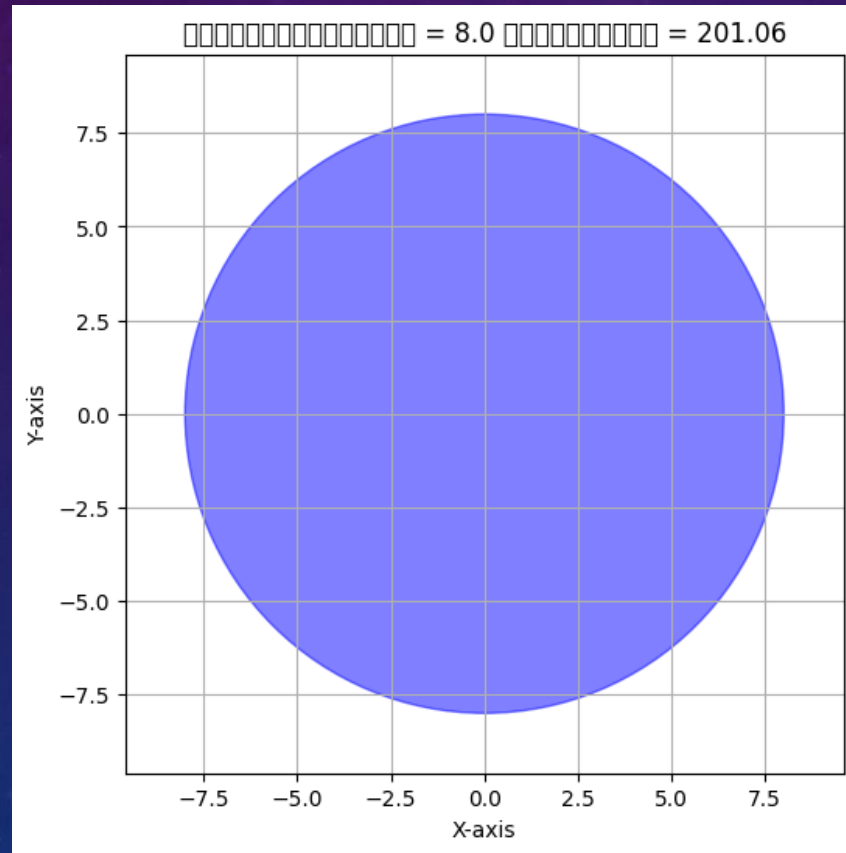


```
import math
radius = float(input("ป้อนความยาวรัศมีของวงกลม: "))
# คำนวณพื้นที่(pi * r^2)
area = math.pi * (radius ** 2)
print("พื้นที่วงกลมคือ:", area)
```

# PYTHON-คำนวณพื้นที่อย่างง่าย

คำนวณพื้นที่วงกลม

“ได้ คำนวณพื้นที่วงกลม โดยรับ **input** และแสดงผลลัพธ์ภาพด้วยกราฟ”





# PYTHON-คำนวณพื้นที่อย่างง่าย

## คำนวณพื้นที่วงกลม

“โค้ด คำนวณพื้นที่วงกลม โดยรับ

input และแสดงผลลัพธ์ภาพด้วย

”  
กราฟ

```
import math
import matplotlib.pyplot as plt
# รับค่ารัศมีจากผู้ใช้
radius = float(input("ป้อนความยาวรัศมีของวงกลม: "))
# คำนวณพื้นที่(pi * r^2)
area = math.pi * (radius ** 2)
# แสดงผลลัพธ์
print(f"พื้นที่วงกลมคือ:{area:.2f}")
# สร้างกราฟวงกลม
fig, ax = plt.subplots(figsize=(6, 6))
circle = plt.Circle((0, 0), radius, color='blue',
alpha=0.5)
ax.add_artist(circle)
# ตั้งค่าแกน
ax.set_xlim([-radius * 1.2, radius * 1.2])
ax.set_ylim([-radius * 1.2, radius * 1.2])
ax.set_aspect('equal', adjustable='box')

plt.title(f'วงกลมที่มีรัศมี={radius} และพื้นที่={area:.2f}')
plt.xlabel('X-axis')
plt.ylabel('Y-axis')
plt.grid(True)
plt.show()
```



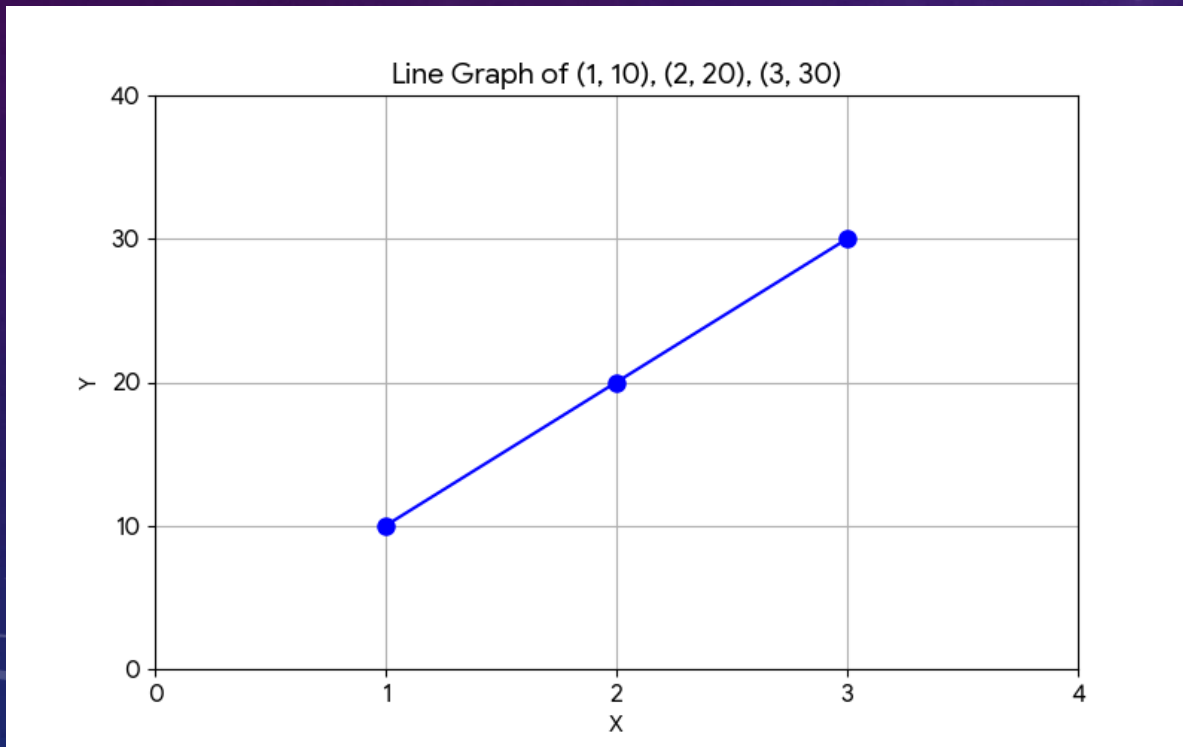
# PYTHON-ความสัมพันธ์เชิงเส้น

- ความสัมพันธ์เชิงเส้น คือ ความสัมพันธ์ระหว่างตัวแปรสองตัว (มักจะใช้ **x** และ **y**) ที่เมื่อ **x** เปลี่ยนแปลงไปอย่างสม่ำเสมอ **y** ก็จะเปลี่ยนแปลงตามในอัตราส่วนที่เท่าๆ กันเสมอ
- ตัวอย่างในชีวิตประจำวัน การซื้อสมุดราคาเล่มละ 10 บาท
- ซื้อ 1 เล่ม จ่าย 10 บาท
- ซื้อ 2 เล่ม จ่าย 20 บาท
- ซื้อ 3 เล่ม จ่าย 30 บาท

จำนวนสมุด หรือ ค่า <b>x</b> (เล่ม)	ราคา หรือ ค่า <b>y</b> (บาท)
1	10
2	20
3	30

# PYTHON-ความล้มพันธ์เชิงเส้น

- เขียนแบบคู่อันดับ (**Ordered Pairs**) จับคู่ค่า  $x$  และ  $y$  ในวงเล็บ  $(x, y)$  จากตัวอย่างจะได้
- $(1, 10), (2, 20), (3, 30)$



จำนวนสมุด หรือ ค่า $x$ (เล่ม)	ราคา หรือ ค่า $y$ (บาท)
1	10
2	20
3	30

# PYTHON-ความสัมพันธ์เชิงเส้น

## สมการเชิงเส้น (Linear Equation)

เมื่อเราเห็นรูปแบบที่ชัดเจน เราสามารถเขียนสรุปเป็นสมการคณิตศาสตร์ได้ รูปแบบสมการทั่วไปของความสัมพันธ์เชิงเส้นคือ  $y = ax + b$

จำนวนสมุด หรือ ค่า $x$ (เล่ม)	ราคา หรือ ค่า $y$ (บาท)
1	10
2	20
3	30

จากตัวอย่างสมุดเล่มละ 10 บาท  
จะเขียนเป็นสมการได้ง่ายๆ ว่า  $y = 10x$

“วาดกราฟ เส้น สำหรับ  $y=10x$  โดยที่  $x$  มีค่า 1,2,3”

# PYTHON-ความล้มพันธ์เชิงเส้น

```
import matplotlib.pyplot as plt

import numpy as np

# Define a range of x values
x = [1, 2, 3]

# Calculate y values using the equation y = x (element-wise multiplication)
y = [10* val for val in x]

# Create the plot
plt.figure(figsize=(8, 6))

plt.plot(x, y, label='y = x', color='blue')

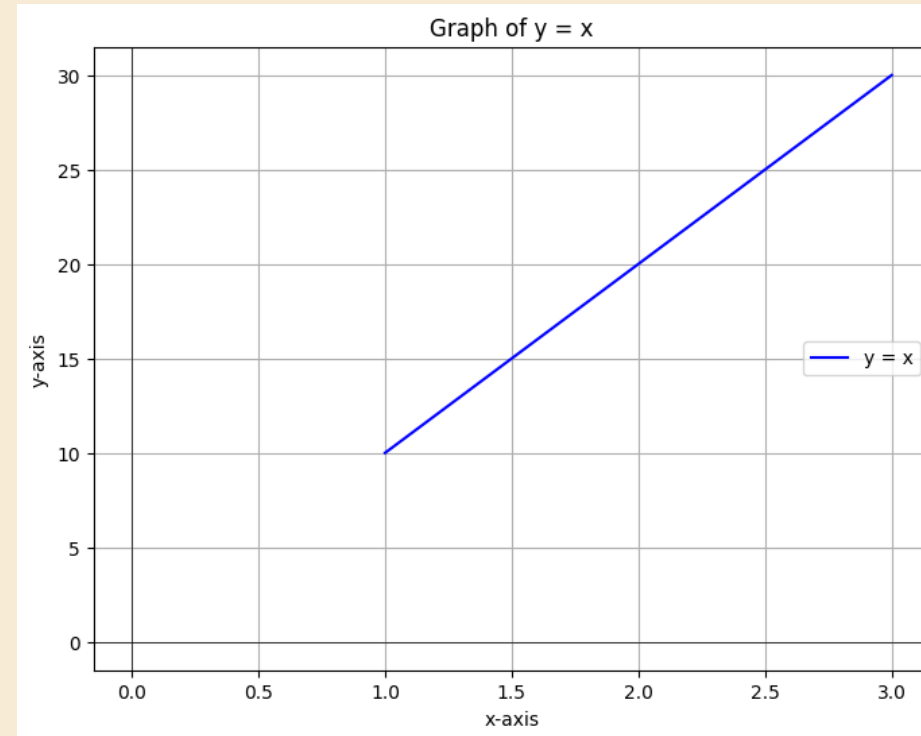
# Add labels and title
plt.title('Graph of y = x')
plt.xlabel('x-axis')
plt.ylabel('y-axis')

plt.grid(True)

plt.axhline(0, color='black',linewidth=0.5) # Add x-axis line
plt.axvline(0, color='black',linewidth=0.5) # Add y-axis line

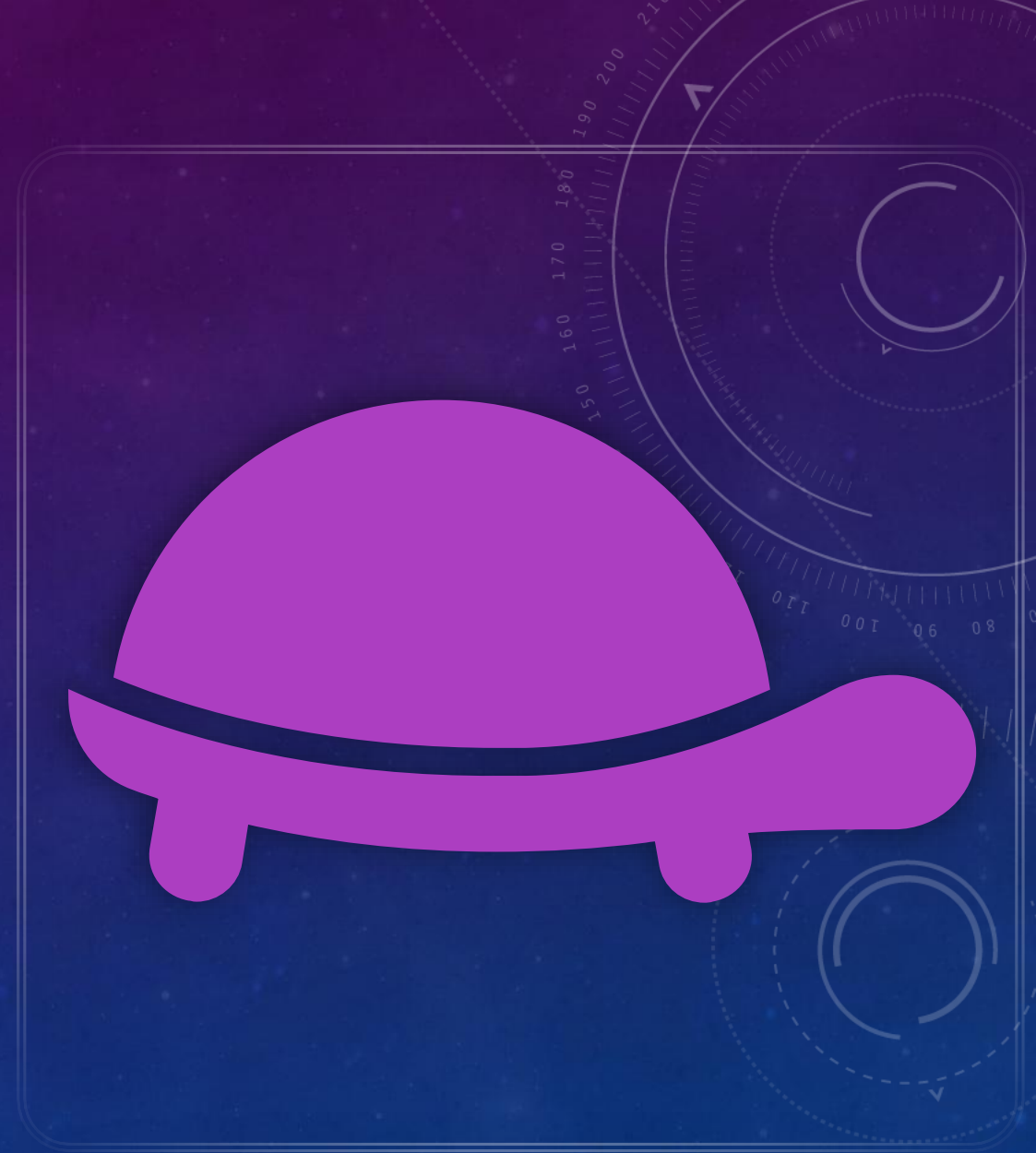
plt.legend()

plt.show()
```



# PYTHON-TURTLE

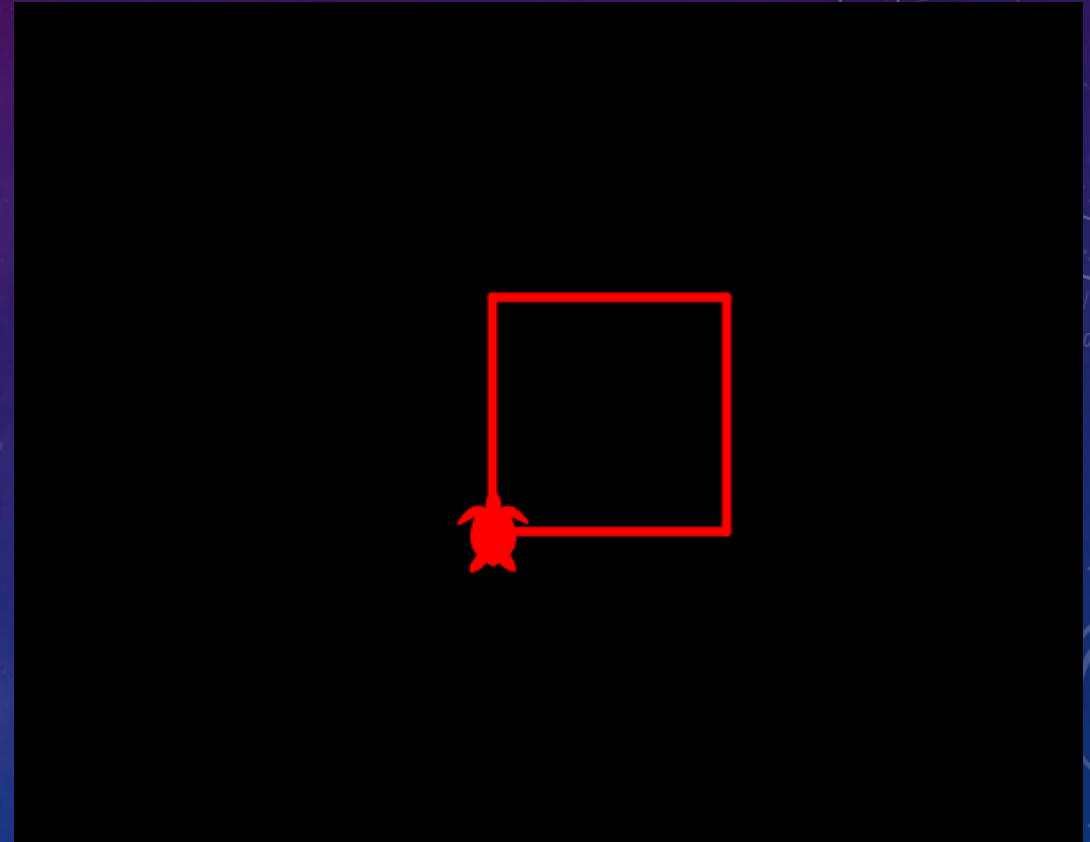
- ติดตั้งเพื่อใช้งาน เต่า ไพธอน
- **`!pip3 install ColabTurtle`**





# PYTHON-TURTLE

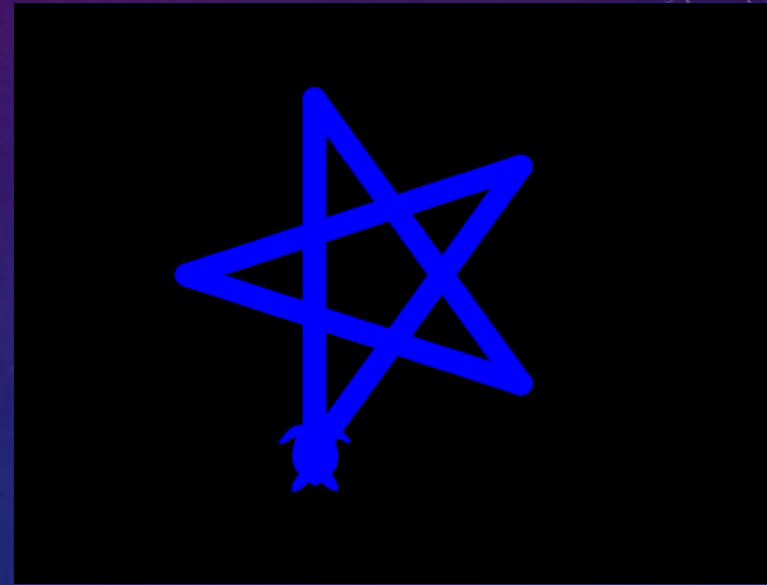
```
from ColabTurtle.Turtle import *  
initializeTurtle()  
color("red")  
forward(100)  
right(90)  
forward(100)  
right(90)  
forward(100)  
right(90)  
forward(100)  
right(90)
```





# PYTHON-TURTLE

```
initializeTurtle()  
color("blue") # กำหนดสีเต่าเป็นสีน้ำเงิน  
pensize(10) # กำหนดความหนาของเส้น  
speed(5) # กำหนดความเร็วในการวาด  
for _ in range(5):  
    forward(150)  
    right(144)
```

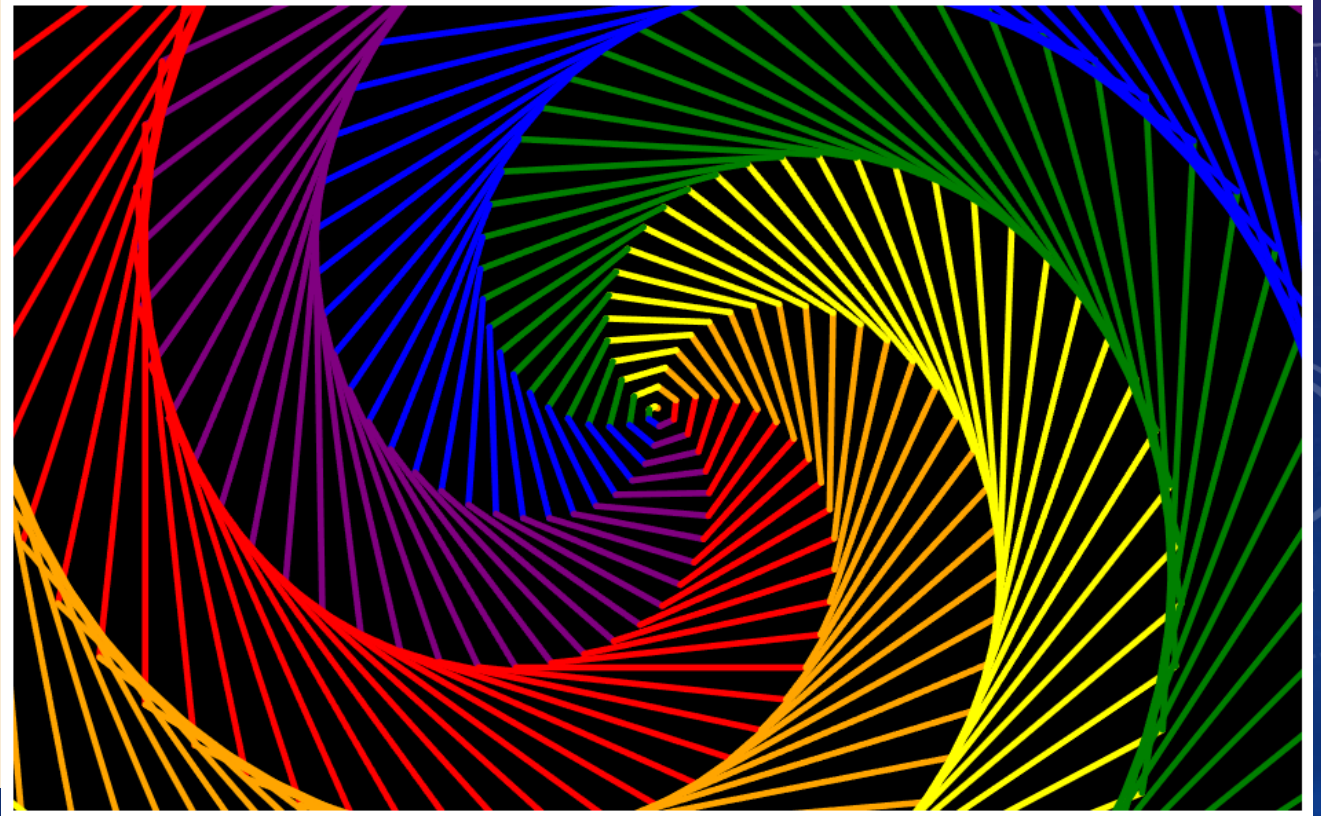


# PYTHON-TURTLE

```
initializeTurtle()  
speed(10)  
colors = ["red", "orange", "yellow", "green", "blue", "purple"]  
for i in range(250):  
    pencolor(colors[i % 6])  
    forward(i * 2)  
    left(59)
```

# PYTHON-TURTLE

```
initializeTurtle()  
speed(10)  
colors = ["red", "orange", "yellow", "green", "blue", "purple"]  
for i in range(250):  
    pencolor(colors[i % 6])  
    forward(i * 2)  
    left(59)
```



# PYTHON-TURTLE

```
initializeTurtle()
speed(10)
pensize(8)
colors = ["red", "orange", "yellow", "green"]
for i in range(250):
    pencolor(colors[i % 4]) # สลับสีไปเรื่อยๆ ทีละสี 4 สี
    forward(i * 2)
    left(121)
```

**\*\***การกำหนดการแสดงผลสี ปรับที่ colors และ l%6  
**\*\***การกำหนดรูปทรง ปรับที่ left(121)

มุม (องศา)รูปทรงที่ได้	มุม (องศา)รูปทรงที่ได้
<b>121</b>	สามเหลี่ยมปิดเกิลียว (Triangle)
<b>91</b>	สี่เหลี่ยมปิดเกิลียว (Square)
<b>73</b>	ห้าเหลี่ยมปิดเกิลียว (Pentagon)
<b>61</b>	หกเหลี่ยมปิดเกิลียว (Hexagon)
<b>144</b>	รูปดาว (Star)