4121201 วิทยาศาสตร์ข้อมูลเบื้องต้นและการวิเคราะห์ข้อมูล

Chapter 4 Python Programming For Data Science part1

Python

- Python เป็นภาษาที่ถูกพัฒนาขึ้นโดย Guido von Rossum ในปี ค.ศ. 1990 และปัจจุบันถูกดูแลโดย Python Software Foundation (PSF)
- Official Site ของ python คือ http://www.python.org

Python

Python เป็นทั้ง Cross Platform และมี Open Source License หมายความว่า

- ✓ถ้าพัฒนาโปรแกรมด้วย Python สามารถที่จะเอาไปทำงานบน Operating System ได้หลากหลาย ไม่ว่าจะเป็น MS Windows, Linux, หรือ OS/X เป็นต้น - - > (Cross Platform)
- ✓มีอิสระในการแก้ไข Library ที่ Python ให้มา รวมถึงการนำ Software ที่พัฒนาขึ้นจาก Python ไปทำประโยชน์ทางธุรกิจได้อย่างเต็มที่โดย ไม่เสียเงิน - - > (Open Source License)

ข้อดี Python

- ✓ไวยากรณ์ (Syntax) ของภาษานั้นอ่านและทำความเข้าใจได้ง่าย มีความตรงไปตรงมา คล้ายคลึงกันกับภาษาอังกฤษ
- ✓สามารถนำไปประยุกต์ใช้งานได้หลากหลายด้าน ไม่ว่าจะเป็น Web Development, Data Science, AI & Machine Learning, Games, Web Scraping เป็นต้น
- ✓มี community ขนาดใหญ่ เนื่องมาจากความนิยมของภาษา ทำให้การเรียนหรือค้นหาข้อมูลเกี่ยวกับ Python มีแหล่งเรียนรู้เยอะ
- ✓Python เป็น open source ซอฟต์แวร์ เราสามารถโหลดใช้งานได้ฟรี
- ✓มีไลบรารี่ต่าง ๆ มากมายให้เลือกใช้งาน

ข้อเสีย Python

- ✓ในด้านความเร็ว ถือว่าเป็นภาษาที่มีความเร็วในการประมวลผลช้ากว่า C, C++ เป็นต้น เพราะว่า Python จัดการหน่วยความจำให้อัตโนมัติ เช่น กำหนดตัวแปร ก็ไม่ต้องกำหนด Type ของตัวแปรเองเลย

Cr. https://devhub.in.th/blog/what-is-python-programming-used-for

Python

ตัวอย่างที่ 1: ตัวอย่างโปรแกรมยอดฮิต “Hello world !”

สำหรับ Java

```
public class
{
    public static void main(String[]args)
    {
        System.out.println("Hello, world!");
    }
}
```

สำหรับ Python

```
print ("Hello, world!")
```

Google Colab

- เป็นโครงการที่ Google ทำงานร่วมกับทีม Jupyter มาตั้งแต่ 2014 แต่เพิ่งเปิดให้ใช้สาธารณะ
- ข้อดี คือ clolab ไม่ต้องลงอะไรเพิ่มเติม มีความสามารถเหมือน Jupyter Notebook แทบทุกอย่าง ดีกว่าก็คือ ไม่ต้องโหลดมาลงบนเครื่อง เขียนเสร็จ ไฟล์ก็ save อยู่บน Google Drive ของตัวเองอัตโนมัติ และมี GPU ให้ใช้ฟรี
- ข้อเสีย คือ อาจจะยังมีอาการช้าบ้างเวลาสั่งรัน code

jupyter spectrogram (autosaved)

File Edit View Insert Cell Kernel Help | Python 3

Markdown CellToolbar

Simple spectral analysis

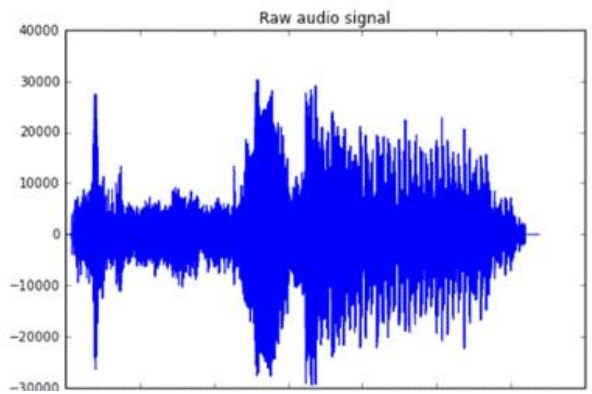
An illustration of the Discrete Fourier Transform

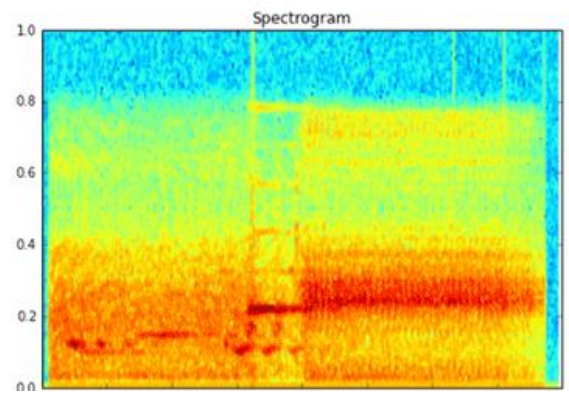
$$X_k = \sum_{n=0}^{N-1} x_n exp^{\frac{-2\pi i}{N} kn} \quad k = 0, \ldots, N-1$$

In [2]:
```
from scipy.io import wavfile
rate, x = wavfile.read('test_mono.wav')
```

And we can easily view it's spectral structure using matplotlib's builtin specgram routine:

In [5]:
```
fig, (ax1, ax2) = plt.subplots(1,2,figsize(16,5))
ax1.plot(x); ax1.set_title('Raw audio signal')
ax2.specgram(x); ax2.set_title('Spectrogram');
```





GPU คืออะไร

- GPU ย่อมาจาก Graphics Processing Unit ถูกสร้างมาเพื่อประมวลผลด้านกราฟิกเป็นหลัก

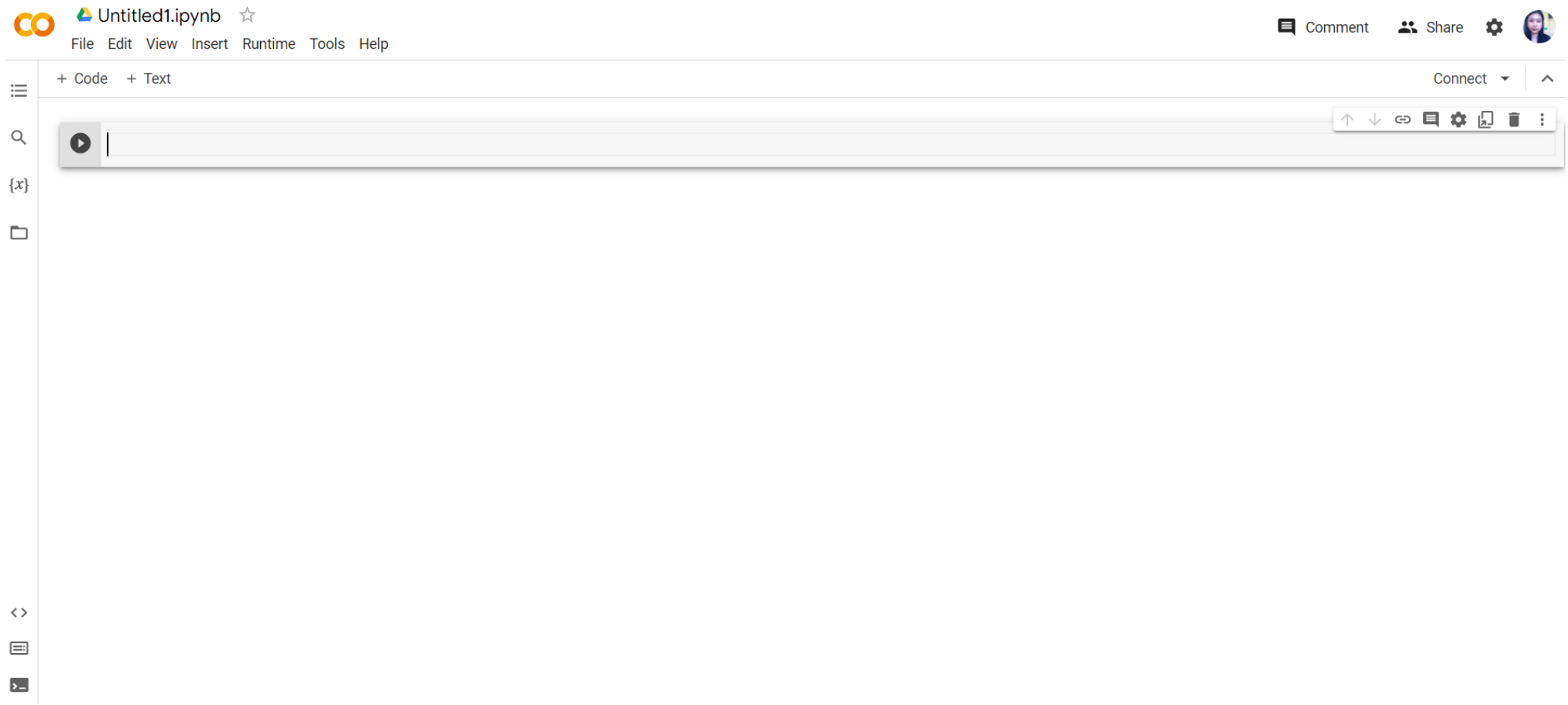
- แม้ว่า CPU จะสามารถคำนวณงานต่างๆ ได้อย่างซับซ้อน แต่ลักษณะการทำงานของมันจะเป็นแบบ Linear (ทำงานแบบเชิงเส้น ประมวลผลไปตามลำดับ) การจะให้ CPU มาเรนเดอร์กราฟิกต่าง ๆ จริง ๆ ก็ทำได้ แต่ว่าจะทำให้ CPU ต้องแบกภาระหนักมากเกินไป (และทำได้ไม่ดี) GPU จึงถูกพัฒนาขึ้นมาเพื่อลดภาระในจุดนี้

- เนื่องจากมันทำหน้าที่เฉพาะด้านที่ชัดเจน สถาปัตยกรรมของมันได้จึงถูกออกแบบมาให้ประมวลผลแบบ Parallel (ขนาน) รับข้อมูลมาทีเดียว และประมวลผลลัพธ์ออกมาพร้อม ๆ กัน

เครื่องมืออื่น ๆ ในการเขียน Python

- PyCharm
- VScode
- Spyder
- Sublime
- Jupeter Notebook

สร้าง Colab ของตนเอง



กิจกรรม 1 : พิมพ์ข้อความออกหน้าจอ

คำสั่ง print

คำสั่งพิมพ์ข้อความในภาษา Python คือ print

เช่น print('Hello')

ก็จะแสดงข้อความว่า Hello ออกที่หน้าจอ

แบบฝึกหัด 1.1

คราวนี้เรามาลองพิมพ์สตริง 'สวัสดีจ๊ะ' ออกที่หน้าจอ ลองเติมข้อความดังกล่าว ลงในเครื่องหมายคำพูดเดี่ยวดู

```
print('สวัสดีจ๊ะ')
```

ลองเปลี่ยนสตริงดูตามใจชอบ แล้วลองรันดู

หมายเหตุ (comment)

บางครั้งเราอยากใส่หมายเหตุลงไปในโค้ดเพื่อเป็นบันทึกช่วยจำ เราจะใช้สัญลักษณ์ # (แฮช หรือ hash)
นำหน้าหมายเหตุ โดยเมื่อไรก็ตามที่ใช้เครื่องหมาย # สิ่งที่ตามหลังเครื่องหมายนี้จะเป็นหมายเหตุทั้งหมด เช่น

```
print('Hello')       #แสดงข้อความ Hello ออกที่หน้าจอ
```

หมายเหตุ (comment)

หากเราต้องการหมายเหตุทั้งบรรทัด ให้เราใส่เครื่องหมาย `#` ที่ด้านหน้าสุด เช่น

```
#แสดงข้อความ Hello ออกที่หน้าจอ
print('Hello')
```

หมายเหตุ (comment)

เวลาคอมพิวเตอร์ทำงาน มันจะกระโดดข้ามส่วนที่เป็นหมายเหตุไป

เช่น ในโค้ดด้านล่างนี้ คอมพิวเตอร์จะกระโดดข้ามบรรทัดที่ 3 ไป เพราะถูกมองเป็นหมายเหตุ

```
print('Please')
print('Mr.')
# print('and Ms.')
print('Postman')
print('Look and See')
```

แบบฝึกหัด 1.2

- ลองใส่เครื่องหมาย # หน้าบรรทัดต่างๆ ในโค้ด แล้วสังเกตความเปลี่ยนแปลงของผลลัพธ์

```
print('Hello')
print('World')
print('My')
print('name')
print('is')
print('Peter')
```

หมายเหตุ (comment)

หากเราต้องการหมายเหตุทุกบรรทัด ให้กดปุ่ม ctrl+/

รับค่า ด้วย input()

แบบฝึกหัด 1.3

ลองพิมพ์

```
age = input("กรุณากรอกอายุ : ")
print("คุณอายุ ", age , "ปี")
```

กรุณากรอกอายุ :

คุณอายุ 44 ปี

กิจกรรม 2: การคำนวณทางคณิตศาสตร์

- เราสามารถทำการคำนวณทางคณิตศาสตร์ได้ โดยใช้เครื่องหมายบวก +, ลบ -, คูณ *, และหาร /กับตัวเลขได้
  ข้อสังเกต: เราจะใช้เครื่องหมายดอกจัน * แทนการคูณ
- เช่น ถ้าเราสั่งให้คอมพิวเตอร์คำนวณค่าของนิพจน์ทางคณิตศาสตร์ 5×2+10 เราจะได้ผลลัพธ์ดังนี้

```
5 * 2 + 10
```

```
20
```

- ถ้าเราสั่งให้คอมพิวเตอร์คำนวณค่าของนิพจน์ $10 + \frac{5}{2}$ เราจะได้ผลลัพธ์ดังนี้

  10 + 5 / 2

  ↳ 12.5

- จะสังเกตได้ว่า คอมพิวเตอร์คำนวณการหาร 5 / 2 และใส่ค่าทศนิยมให้โดยอัตโนมัติ

แบบฝึกหัด 2.1

เรามาลองคำนวณค่าของนิพจน์ทางคณิตศาสตร์ต่อไปนี้กันดูบ้าง

$$5 \times 6 + 7 \times 8 - \frac{2}{5}$$

↳85.6

วงเล็บในนิพจน์ทางคณิตศาสตร์

วงเล็บในนิพจน์ทางคณิตศาสตร์

ทั้งนี้ เราสามารถใช้เครื่องหมายวงเล็บ ( ) ได้เหมือนกับนิพจน์ทางคณิตศาสตร์ปกติ

เช่น เราสามารถคำนวณค่าของนิพจน์ $4\times3+(1+2)/4$

ได้ด้วยคำสั่งนี้

```
4 * 3 + (1 + 2) / 4
```

↳ 12.75

วงเล็บในนิพจน์ทางคณิตศาสตร์

- ข้อแตกต่างคือ ในกรณีที่มีวงเล็บซ้อนวงเล็บ เราจะใช้เครื่องหมายวงเล็บ ( ) เสมอ **ห้ามใช้** เครื่องหมายก้ามปู [ ] และเครื่องหมายปีกกา { } โดยเด็ดขาด
- เช่น ถ้าเราจะคำนวณค่าของนิพจน์ ([1/{2+3}]×4+5)/6 เราจะต้องเปลี่ยนก้ามปูและปีกกากลับมาเป็นวงเล็บก่อนเสมอ

ลองเขียนคำสั่งหาค่า ([1/{2+3}]×4+5)/6 ดู จะได้เป็น

((1 / (2 + 3)) * 4 + 5) / 6

↳0.9666666666666667

Tip* สามารถแปลงทศนิยมให้เป็นจำนวนเต็ม หรือกำหนดตำแหน่งจำนวนทศนิยมได้ ตามรูปแบบ round(ค่าที่ต้องการกำหนด, ตำแหน่งจำนวนทศนิยมที่ต้องการ)

วงเล็บในนิพจน์ทางคณิตศาสตร์

หัวใจ!! : เครื่องหมายวงเล็บมีหน้าที่เปลี่ยนลำดับการทำงาน โดยจะเริ่มทำในวงเล็บก่อนเสมอ โดยลำดับการทำงานจะเริ่มจาก (1) ทำในวงเล็บ (2) คูณและหาร (3) บวกและลบ

วงเล็บในนิพจน์ทางคณิตศาสตร์

เช่น ลองสังเกตผลลัพธ์ของนิพจน์ทั้งสองนี้ >> (1+2)×3 กับ 1+(2×3)

(1 + 2) * 3

↳ 9

1 + (2 * 3)

↳ 7

ทั้งนี้ขอให้ระมัดระวังการใช้เครื่องหมายคูณ * และหาร / ด้วย เพราะถ้าหากไม่ระบุวงเล็บให้ชัดเจน ลำดับการคำนวณก็อาจจะไม่เป็นไปตามที่ต้องการได้

แบบฝึกหัด 2.2

มาลองคำนวณค่าของนิพจน์ทางคณิตศาสตร์ต่อไปนี้กัน

$$3 \times (4 + 1 - 2 \times 5)$$

↳ -15

เครื่องหมายยกกำลังในภาษา Python จะใช้เครื่องหมาย ** (ดอกจันเขียนติดกัน 2 ดวง)
เช่น ถ้าเราจะคำนวณค่าของนิพจน์ 2^5 เราจะใช้คำสั่งว่า

```
2 ** 5
```

```
32
```

เมื่อเรานำ *เครื่องหมายยกกำลังมาใส่ในนิพจน์ทางคณิตศาสตร์* ลำดับการทำงานจะเปลี่ยนไปตามนี้

a) ทำในวงเล็บก่อน
b) ทำเครื่องหมายยกกำลัง
c) ทำเครื่องหมายคูณและหาร
d) ทำเครื่องหมายบวกและลบ

```
2 * (5**3)
```

↳250

```
2 * 5**3
```

↳250

ดังนั้นเพื่อความปลอดภัย ***เราจึงนิยมเขียนเครื่องหมายยกกำลัง ** ติดกับตัวเลข*** เพื่อให้เป็นที่รู้กันว่า **ถ้าหากไม่มีวงเล็บ เราจะเริ่มทำจากเครื่องหมายยกกำลังก่อนเสมอ** เช่น เราจะเขียนนิพจน์ 2×5^3 ด้วยคำสั่ง 2 * 5**3 เป็นต้น

นอกจากนี้เรายังสามารถคำนวณค่าราก (root) ได้โดยการใช้สูตร:

$$\sqrt[r]{a} = a^{1/r}$$

เช่น เราสามารถคำนวณค่า $\sqrt[3]{8}$ (รากที่ 3 ของ 8) ได้โดยคำสั่ง 8**(1/3)

ลองพิมพ์

8**(1/3)

2.0

แบบฝึกหัด 2.3

เรามาลองคำนวณค่าของนิพจน์ดังต่อไปนี้กัน

$$\sqrt[2]{(7+3\times 4)}$$

↳ 4.358898943540674

กิจกรรม 3 : ตัวแปร

- ตัวแปรเปรียบเสมือนที่ทดเลข เพื่อที่จะได้นำผลลัพธ์จากการคำนวณค่าของนิพจน์ที่ทดไว้ ไปใช้ต่อได้ในอนาคต
- เนื่องจากกระดาษในคอมพิวเตอร์มีขนาดใหญ่มาก เราจึงต้องจองพื้นที่สำหรับทดเลขและตั้งชื่อเอาไว้ เวลาจะหยิบสิ่งที่ทดไว้กลับมาใช้ใหม่ จะได้ไม่หลงที่นั่นเอง

คำสงวนใน Python

- คำสงวน คือคำที่ถูกภาษาไพธอนใช้เพื่อสร้างไวยากรณ์ ดังนั้นผู้เขียนโปรแกรมห้ามนำไปใช้ในการสร้าง หรือประกาศเป็นตัวแปรโดยเด็ดขาด เพราะจะทำให้เกิดข้อผิดพลาด คือ SyntaxError: invalidsyntax

คำสงวนใน Python มีดังต่อไปนี้คือ

คำสงวนในภาษา Python (Reserved Words)

and	as	assert	break	continue	class
def	del	elif	else	except	false
finally	for	from	global	if	import
in	is	lambda	none	not	nonlocal
or	pass	return	raise	true	try
while	with	yield			

คำต่อไปนี้แม้ว่า Python ไม่ได้ห้ามไว้แต่ก็ไม่ควรใช้เพราะไปตรงกับชื่อของฟังก์ชันใน Python คือ

- data, float, Int, numeric, Oxphys, array, close, int, input, open, range, type, write, zeros

คำต่อไปนี้ก็ควรหลีกเลี่ยงด้วย ถ้ามีการนำเข้า (import) ไลบรารี่ math มาใช้งาน คือ

- acos, asin, atan, cos, e, exp, fabs, floor, log, log10, pi, sin, sqrt, tan

Cr.https://pythonlanguage64.wordpress.com/%E0%B8%84%E0%B8%B3%E0%B8%AA%E0%B8%87%E0%B8%A7%E0%B8%99/

เช่น สามารถทดค่าของ 2^5 เก็บไว้ในตัวแปร a และทดค่าของ $\sqrt[2]{5}$ เก็บไว้ในตัวแปร b เพื่อใช้ต่อในการคำนวณต่อ ๆ ไปได้

ซึ่งในตัวอย่างนี้เราจะคำนวณค่าของ $a^3 + b^3$ กัน

```
a = 2**5              # ทดค่าไว้ในตัวแปร a
b = 5**(1/2)          # ทดค่าไว้ในตัวแปร b
```

จากนั้นนำมาเข้าตามสูตรด้านบน ได้ดังนี้

```
a**3 + b**3
```

↳ 32779.1803398875

สามารถพิมพ์ค่าตัวแปรออกหน้าจอได้ด้วยคำสั่ง print เช่นเดียวกับการแสดงข้อความ เช่น

```
a = 2**5
b = 5**(1/2)
print(a)
print(b)
```

```
32
2.23606797749979
```

แต่บางครั้งการตั้งชื่อตัวแปรด้วยตัวอักษรเดี่ยว ๆ ก็จำยาก เดี๋ยวมาอ่านทีหลังก็อาจจะลืม เลยนิยมตั้งชื่อตัวแปรให้เป็นคำที่อ่านแล้วจำได้ง่ายแทน เช่น width (ความกว้าง), height (ความสูง), และ area (พื้นที่)
ดังตัวอย่างด้านล่างนี้

```
width = 40
height = 30
area = width * height
print(area)
```

```
1200
```

หากชื่อตัวแปรประกอบด้วยหลายคำ ให้ใช้เครื่องหมาย _ (ขีดเส้นใต้ หรือ underscore) คั่นระหว่างคำ เพื่อให้อ่านง่ายขึ้น เช่น area_of_triangle พื้นที่สามเหลี่ยม เป็นต้น

```
width = 40

height = 30

area_of_triangle = 1/2 * width * height

print(area_of_triangle)
```

600.0

นอกจากนี้คำสั่ง print ยังสามารถพิมพ์ค่าออกหน้าจอได้หลาย ๆ ค่าพร้อมกัน เช่น

```
width = 40
height = 30
area_of_triangle = 1/2 * width * height

print('width =', width, 'and height =', height)
print('area = 1/2*width * height =', area_of_triangle)
```

```
width = 40 and height = 30
    area = 1/2*width * height = 600.0
```

แบบฝึกหัด 3.1

1.ทดลองกำหนดค่าในตัวแปร a, b, และ c ในโค้ดด้านล่าง แล้วสังเกตความเปลี่ยนแปลงของผลลัพธ์

```
a = 10
b = 20
c = 30
average = (a + b + c) / 3
print(average)
```

```
20.0
```

2.จงเขียนโค้ดลงในช่องว่างให้สมบูรณ์ เพื่อคำนวณค่าเฉลี่ยฮาร์โมนิก (harmonic mean) ของตัวแปร a, b, และ c โดยค่าเฉลี่ยฮาร์โมนิก มีสูตรดังนี้

$$hmean(a,b,c) = \frac{3}{\frac{1}{a}+\frac{1}{b}+\frac{1}{c}}$$

a = 10

b = 20

c = 30

hmean = ...

print(hmean)

↳16.363636363636363

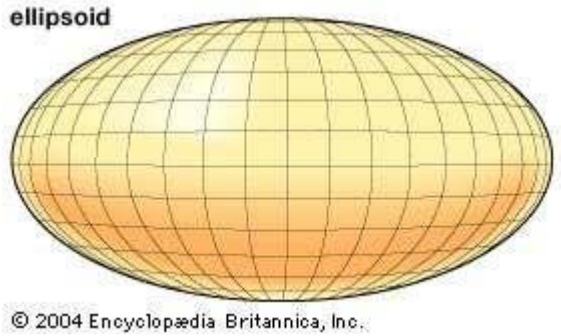
หมายเหตุ* ค่าเฉลี่ยฮาร์โมนิก นิยมใช้สำหรับการหาอัตราเฉลี่ยหรือใช้สำหรับเฉลี่ยข้อมูล เช่น ระยะทางต่อชั่วโมง งานต่อหน่วยเวลา เป็นต้น

3.เขียนโค้ดเพื่อคำนวณปริมาตรของทรงรี (ellipsoid) จากตัวแปรต่อไปนี้

ความกว้างเก็บอยู่ในตัวแปร width

ความยาวเก็บอยู่ในตัวแปร length

ความลึกเก็บอยู่ในตัวแปร depth



สูตรของปริมาตรทรงรี มีดังนี้

$$volume = \frac{4}{3} \times \pi \times width \times length \times depth$$

pi = 3.1415926 #ค่า π (ไพ)

กำหนดความกว้างในตัวแปร width ให้เท่ากับ 20

กำหนดความยาวในตัวแปร length ให้เท่ากับ 30

กำหนดความลึกในตัวแปร depth ให้เท่ากับ 40

คำนวณปริมาตรของทรงรี แล้วเก็บในตัวแปร volume และพิมพ์ค่าในตัวแปร volume

กิจกรรม 4: Data Type

ชนิดของข้อมูล

int: 1 2 3 4 5 6 7 8 9 #เลขจำนวนเต็ม

float: 2.4 3.5 6.9 10.2 #เลขทศนิยม

string: 'hello world ' ' food ' ' dog’ #ข้อความ ตัวอักษร

Boolean : True False #ค่าความจริง

การกำหนดชนิดข้อมูล

สามารถกำหนดชนิดข้อมูล โดยระบุค่าที่เป็นประเภทข้อมูลนั้น ๆ ให้กับตัวแปร

```
a = 5    #กำหนดตัวแปร a เป็นเลขจำนวนเต็ม
b = 1.2 #กำหนดตัวแปร b เป็นเลขทศนิยม
c = 'Python' #กำหนดตัวแปร c เป็นข้อความ
d = True #กำหนดตัวแปร c เป็น True
```

String

String จะอยู่ในเครื่องหมายคำพูด “ ” หรือ ' ' ได้ทั้งสองรูปแบบ

word1 = “Python”
word2 = “Hello World”

print(word1)

↳ Python

print(word2)

↳ Hello World

เราสามารถเช็ค data type ด้วย ฟังก์ชัน type()

เช่น

```
a = “Python”
type(a)
```
↳ str

```
b =  123
type(b)
```
↳ int

นับความยาวของ string นั้น ๆ ด้วยคำสั่ง len()

เช่น

```
word1 = 'Python'
len(word1)
```


```
word2 = 'Hello World'
len(word2)
```

แบบฝึกหัด 4.1

1.สร้างตัวแปร 3 ตัวขึ้นมา เป็นแบบ String, int, float อย่างละ 1

2.print ทั้ง 3 ตัวแปร

3.print ประเภทข้อมูลของทั้ง 3 ตัวแปร

4.หาความยาวของตัวแปรที่เก็บข้อมูล String

Reference

- ธนารักษ์ ธีระมั่นคง และคณะ. (2562). **การโปรแกรมภาษา Python (ไพธอน) เบื้องต้น.** กรุงเทพฯ: สมาคมปัญญาประดิษฐ์ประเทศไทย.
- กษิดิศ สตางค์มงคล (2563). **Python for Non-Programmer.** กรุงเทพฯ: Datarockie.
- กฤษฎา เฉลิมสุข (2563). **Introduction to Python for Data Science.** กรุงเทพฯ: The Self Made Serial Entrepreneur.
- http://www.ayarafun.com