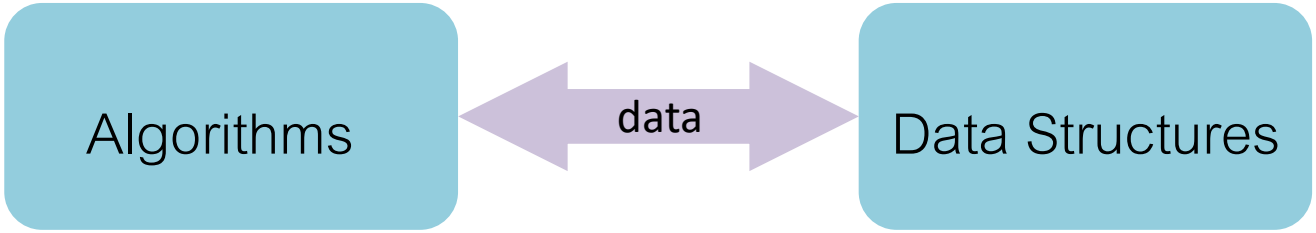
บทที่ 1
โครงสร้างข้อมูลเบื้องต้น

สุนทรี คุ้มไพโรจน์

การเขียนโปรแกรมภาษาคอมพิวเตอร์

พื้นฐานที่สำคัญคือ

- อัลกอริทึม(Algorithms)
- โครงสร้างข้อมูล (Data Structures)



อัลกอริทึม

- ลำดับขั้นตอนวิธีในการทำงานของโปรแกรมเพื่อแก้ปัญหาใดปัญหาหนึ่ง
- การปฏิบัติตามขั้นตอนที่ถูกต้อง ช่วยแก้ปัญหาหรือประมวลผลให้สำเร็จได้ตามต้องการ

A data structure

“An organization of information,
usually in memory,
for better algorithm efficiency.”

Credit: Black, Paul E. (15 December 2004). "data structure".
In Pieterse, Vreda; Black, Paul E. (eds.).
Dictionary of Algorithms and Data Structures [online].
National Institute of Standards and Technology.

โครงสร้างข้อมูล

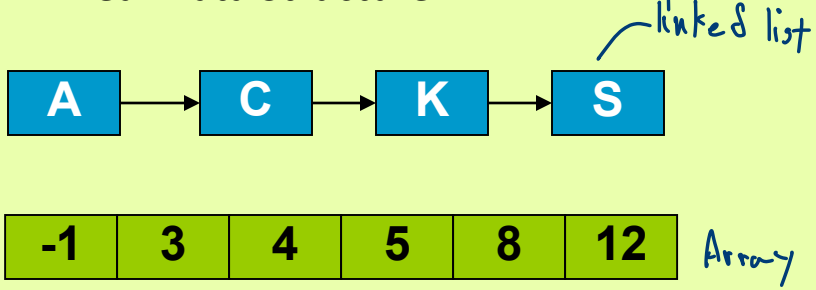
- การจัดเก็บข้อมูลเป็นกลุ่มที่สัมพันธ์กันเข้าไว้ด้วยกัน
- อาจจะเป็นการรวมระหว่างข้อมูลประเภทเดียวกัน ต่างประเภทกัน หรือต่างโครงสร้างข้อมูลกันก็ได้
  - เพื่อให้สะดวกในการเรียกใช้
  - เพื่อประสิทธิภาพของการใช้งาน

ประเภทของโครงสร้างข้อมูล

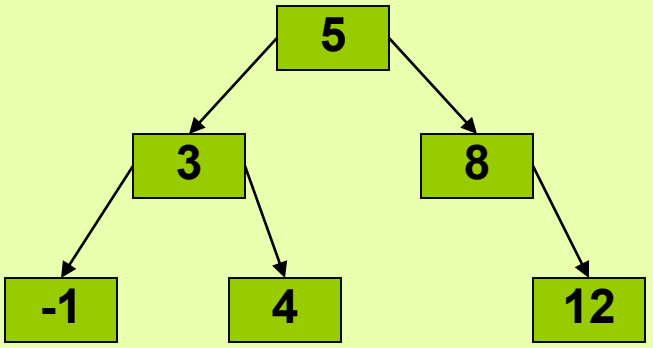
- Basic Data Type(ประเภทข้อมูลพื้นฐาน)
  - ชนิดข้อมูลเชิงเดี่ยว เช่น Char, integer, float, Boolean
  - ชนิดข้อมูลเป็นกลุ่ม
    - กลุ่มข้อมูลที่มีลำดับ เช่น List, String, Tuples
    - กลุ่มข้อมูลที่ไม่มีลำดับ เช่น Sets, Dictionaries

- Abstract Data Type(ประเภทข้อมูลนามธรรมหรือประเภทข้อมูลที่สร้างขึ้น)

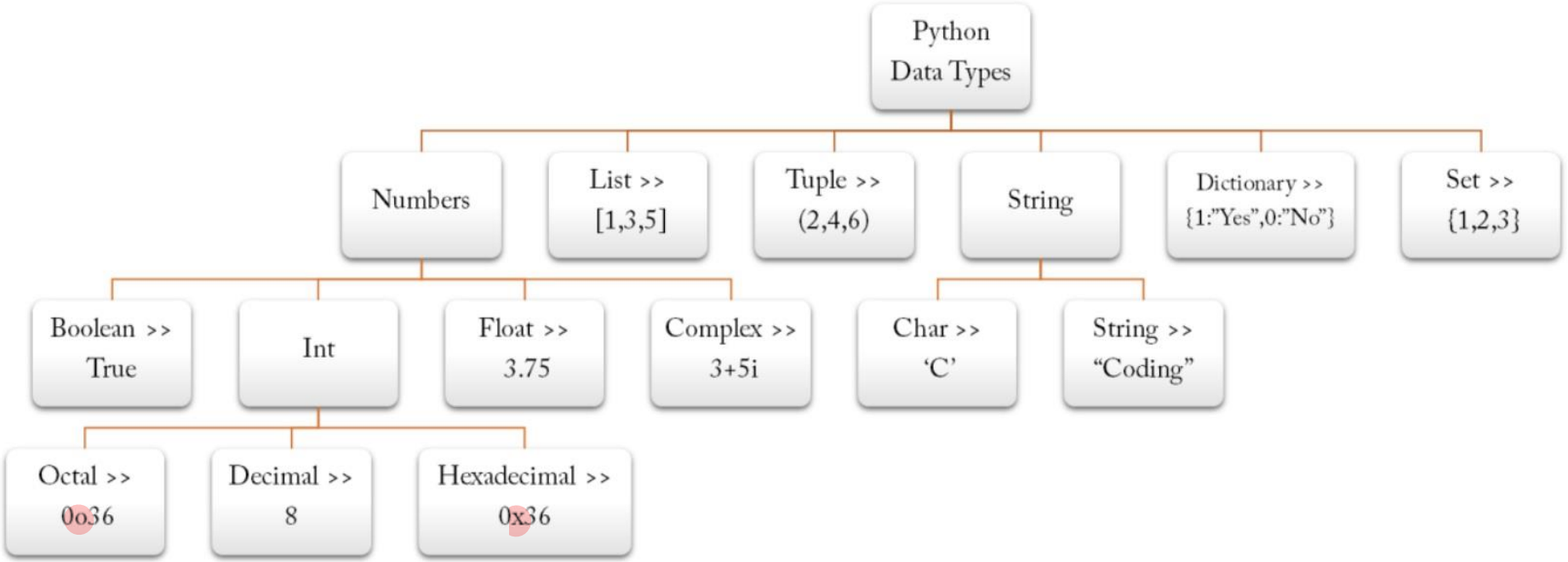
- Linear Data Structure



- Non-linear Data Structure

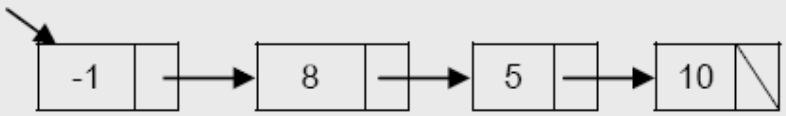


Data Type ในภาษา Python

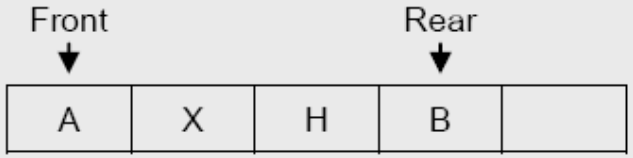


ที่มา https://ajpaeng.wordpress.com/2020/01/24/datatypes-in-python/

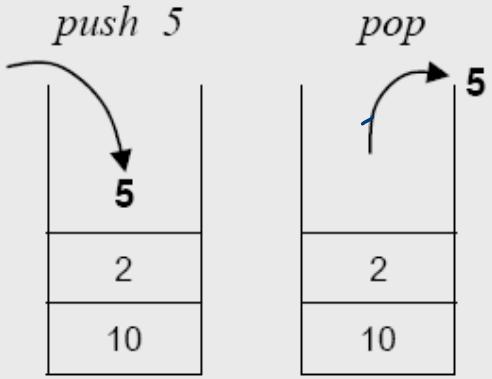
Abstract Data Types



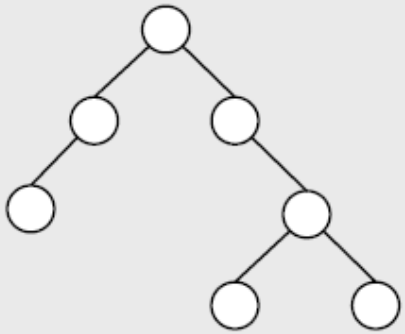
Linked list



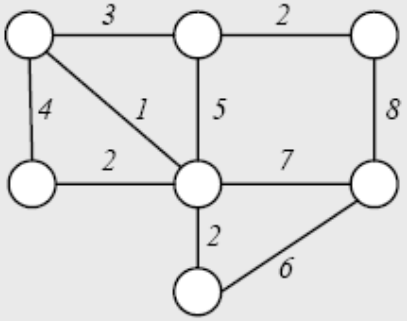
Queue



Stack



Tree



Graph

ที่มา: https://sites.google.com/site/pmtcitajmon/home/hnwy-thi-1-rucak-kab-khorngsrang-khx-mul-laea-xal-kx-ri-thum

ประโยชน์ของโครงสร้างข้อมูลในการเขียนอัลกอริทึม

- เพื่อนำโครงสร้างข้อมูลที่มีอยู่แล้ว มาประยุกต์แก้ปัญหา
- เพื่อเพิ่มประสิทธิภาพในการเขียนโปรแกรม
- เพื่อให้สามารถเลือกใช้โครงสร้างข้อมูลและอัลกอริทึมที่เหมาะสมในงานนั้นๆ

Basic Data Type

เลขจำนวนเต็ม (Integer)

- เซตของจำนวนเต็ม

  { ... , -3 , -2 , -1 , 0 , 1 , 2 , 3 , ... }

- เลขจำนวนเต็มสามารถมีค่าเท่าไรก็ได้ไม่จำกัด
- การกำหนดค่า สามารถทำได้โดยตั้งชื่อตัวแปรและกำหนดค่าให้ เช่น

  c = 16

Case sensitive!

ตัวอย่างคำสั่งในภาษา python ในการใช้งาน integer

- input("Input a number: ")
  - แสดงข้อความในข้อมูลขาเข้าออกหน้าจอ แล้วให้ผู้ใช้ป้อนข้อมูล
- print("Hello world.")
  - แสดงข้อความในข้อมูลขาเข้าออกหน้าจอ

พิมพ์สองบรรทัดนี้ลงบนไฟล์ test.py แล้วสั่ง Run (F5)

ตัวอย่าง

```
inName = input("Please enter your name : ")
print("Hello", inName)
```

ตัวอย่างคำสั่งในภาษา python ในการใช้งาน integer

- int(A)
  - แปลงข้อมูลขาเข้าให้เป็นเลขจำนวนเต็ม
  - เช่น A = int(3.142) จะได้ค่า A เป็นเท่าใด
- สามารถให้ผลของฟังก์ชันไปเป็นข้อมูลขาเข้าของ function อีกตัวหนึ่งได้ โดยที่ฟังก์ชันตัวในสุดจะถูกเรียกก่อน แล้วนำผลไปใส่ในฟังก์ชันตัวนอก

  เช่น A = int(input("Enter a number:"))
  - จะเรียก input("Enter a number:") ก่อน ให้ผู้ใช้ป้อนข้อมูล แล้วนำข้อมูลนั้นไปแปลงเป็นเลขจำนวนเต็ม

เลขจำนวนจริง(float)

- เป็นเลขทศนิยม floating point

  เช่น 15.20, -0.5

- การกำหนดค่า สามารถทำได้โดยตั้งชื่อตัวแปรและกำหนดค่าให้เช่นเดียวกับ integer เช่น

  i = 5.0

- การใช้คำสั่งอื่นๆ ก็เช่นเดียวกับตัวแปร integer เช่น

```
height = float(input('Plese input your height'))
```

ตัวอย่างโปรแกรม

```
1  import math
2  print("Hello World")
3  a = float(input())
4  b = float(input())
5  C = float(input())
6  CR = C*math.pi/180
7  area = 1/2*a*b*math.sin(CR)
8  print("area=",area," sq cm")
```

ข้อมูล Boolean

บูลีน (Boolean) คือค่า จริง , เท็จ ในภาษาไพทอน ใช้คำว่า True False

ในการเปรียบเทียบ จะได้ผลลัพธ์เป็นบูลีน เช่น

การเปรียบเทียบ 4 > 1 เป็นจริง ผลที่ได้คือบูลีน True

6 > 7 เป็นเท็จ ผลคือได้บูลีน False

เมื่อเขียนโปรแกรม เราจะใช้ผลการเปรียบเทียบ เพื่อตรวจสอบเงื่อนไขในการทำงาน

- ตัวอย่างการประกาศตัวแปรบูลีน

```
a = True
b = False
print(a or b) #True or False ==> True
print(a and b) #True and False ==> False
print(not (a or b)) #False

val1 = 5 == 10 #False
val2 = 10 > 5 #True
print((5 >= 1) and (5 <= 10)) #True and True ==> True
```