



Programming Foundation Live Class

Tags	Live Class	R
Class		
Finished Yet?	<input checked="" type="checkbox"/>	
Knowledge	Live Classes	

Live Class

PDF:

[https://s3-us-west-2.amazonaws.com/secure.notion-static.com/ce0cba69-03fb-40b7-af29-58f3570a8631/Programming_Foundation_\(Updated\).pdf](https://s3-us-west-2.amazonaws.com/secure.notion-static.com/ce0cba69-03fb-40b7-af29-58f3570a8631/Programming_Foundation_(Updated).pdf)

Code from Teacher Toy:

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/a43b90ea-2209-4202-a526-c68c15299c3c/R_Teacher_Toy_Code.r

-R เป็นภาษาที่เกิดมาเพื่องานสถิติและคณิตศาสตร์โดยเฉพาะ

-R เขียนได้ใน [posit.cloud](#) และ

<https://colab.research.google.com/#create=true&language=r> (รวมไปถึง

<https://datalore.jetbrains.com/>)

*2 เว็บหลังใช้ในการทำกราฟบ้าน เพื่อให้แฮร์กราฟบ้านได้

-Notebook จะมี Text Cell และ Code Cell ใช้พิมพ์ Markdown และ Code ตามลำดับ

-R เป็น Fast Data Crunching Language ทำงานกับข้อมูลได้อย่างรวดเร็ว

-เราจึงเขียน Code ในหน้าต่าง Script เพราะเขียนใน Console จะเก็บ Code ไว้ไม่ได้

-R เมื่อตอน Calculator ใช้คิดเลขได้ เช่น $5*2$, $8/4$ เป็นต้น

-กด CTRL + L เพื่อเคลียร์หน้าจอ

-ภาษา R กับ Python จะเหมือนกับการทำ Data Analysis เพราะไม่ต้อง Compile ก่อน สามารถ Run Code บรรลุต่อบรรลุได้เลย

-หลัง Save File ตัว R Script ที่เราทำงาน ตัว File จะอยู่ใน Working Directory ซึ่งเราสามารถใช้ `getwd()` ในการหา Working Directory ได้

[Basic Programming Knowledge]

1. Variable

2. Data Types

3. Data Structures

4. Control Flow

5. Function

*แต่ละภาษาจะต่างกันที่ Syntax แต่ทุกภาษาจะมีจุดร่วมที่เหมือนกัน

-Create Variable: `income ← 28000`

-การเขียนโปรแกรมมีข้อดีคือ Reusable (สำคัญมาก)

-ถ้าประกาศค่าใหม่กับตัวแปรเดิม ค่าใหม่จะกับค่าเดิม ถ้าอยากจะได้ตัวแปรใหม่ต้องสร้างตัวแปรใหม่ เช่น:

```
income <- 1000
income <- 2000
print(income)
#income = 2000
```

*ตัวแปรใน R Case Sensitive พิมพ์ใหญ่-พิมพ์เล็กได้ผลต่างกัน

-Data Types:

1. Numeric
2. Character (text)
3. Date
4. Logical (Boolean: TRUE,FALSE)

-เช็คได้ด้วย class() หรือ is.numeric(), is.character(), is.logical() [เป็นตัวเลข/ตัวอักษร/ค่าความจริงหรือไม่] เช่น

```
class(my_age)
is.numeric(my_age)
```

-ใช้ as.Date เปลี่ยน Character (ตัวอักษร) เป็นวันที่ เช่น:

```
Today <- "2023-02-07"
Today <- as.Date(Today)
class(Today)
```

-Data Structures:

1. Vector
2. Matrix
3. List
4. Data Frame

-เราสามารถดึงค่าใน Vector ได้ ถ้าเรารู้ตำแหน่ง เช่น friends[1:2]

-Subsetting หรือ Indexing คือการดึงค่าส่วนหนึ่งจาก Object

*Indexing ใน R เริ่มที่ 1 ไม่ใช่ 0

-ถ้าเราอยากรู้ว่า Function ใน R ทำงานยังไง เราสามารถไปที่ Help เพื่อค้นหาวิธีการใช้งาน Function นั้น ๆ ได้

-Vector เก็บข้อมูลได้แค่ 1 ประเภท

-Matrix = 2D Vector

-Element-Wise Computation (Vectorization): เมื่อเราทำการบวกกลบคุณหารกับ Matrix ตัวเลขจะเข้าไปบวกกลบคุณหารกับทุก ๆ ค่าใน Matrix นั้น ๆ

-ถ้าเรา Matrix ที่ Dimension เท่ากันมาคูณกัน จะจับเลขตำแหน่งเดียวกันมาคูณกัน ถ้าอยากรวบคูณแบบต่อนเรียนมาระยมให้ m1 %*% m2 แทน

```
y <- c(1:6)
dim(y) <- c(3, 2)
y
#Or
matrix(1:6, ncol = 2)
```

-List เก็บข้อมูลได้หลายประเภท (เหมือน Dictionary ใน Python)

-เราสามารถ Subset ตัว List ได้ง่าย ๆ ด้วยการใช้ \$ หรือ []

```
my_playlist$fav_movie
#[1] "Avengers"

my_playlist[1]
#$fav_movie
#[1] "Avengers"

my_playlist[[1]]
#"Avengers"
```

-ใน List แต่ละ List ไม่จำเป็นต้องมี Key = Value เหมือนกัน

-เราสามารถสร้าง Database จาก List หลาย ๆ อันได้ เช่น:

```
customerdb <- list(
  customer1,
  customer2
)
```

-เราสามารถใช้ \$ ช้อนกันได้

-เราสามารถ Update ค่าของตัวแปรได้ใน R ต่างจากบางภาษาที่ประกาศค่าแล้วจะไม่สามารถเปลี่ยนแปลงได้ (Immutable)

-Data Frame เป็นประเภทข้อมูลที่ใช้งานกันมากที่สุดในงาน Data Analytics มีลักษณะเหมือน Table ใน Spreadsheets

-วิธีการสร้าง Data Frame:

```
friends <- c("Kaeya", "Diluc", "Nilou", "Cyno", "Candace")
region <- c("Mondi", "Mond", "Sumeru", "Sumeru", "Sumeru")
meme_lover <- c(TRUE, FALSE, TRUE, TRUE, TRUE)

df <- data.frame(id = 1:5,
                  friend = friends,
                  reg = region,
                  meme_love = meme_lover)
View(df)
```

- ค่า NULL ใน R จะเรียกว่า NA (Not Available) ใช้ is.na() ในการเช็คได้
- df[1:3,] = ดึง row ตั้งแต่ 1 ถึง 3 และดึงทุก column
- Replace value: df[1, 2] ← “New Jeans”
- Subset by Logic: df%age < 30
- cat() = Concat&Print
- OR = | (เช่น cond <- df\$age <= 28 | df\$movie_love)
- write.csv(df, friends.csv) ใช้บันทึกตารางเป็นไฟล์ .csv
- Function: ชุดคำสั่งที่เราสามารถเรียกใช้ซ้ำได้ (Reusable Code)

```
#User-defined function
add_two_nums <- function (v1,v2) {
  return(v1+v2)
}

add_two_nums(10, 20)
#30

#Define function in 1 line
cube <- function (x) x**3

my_power <- function (base, pow) base**pow
my_power(base = 5, pow = 2)
#25
#(base, pow) parameters
#(5,2) arguments
```

*return เป็น optional

-เราสามารถกำหนด default argument ได้ และตัว default argument นั้นผู้ใช้งานสามารถเขียน argument เป็นค่าอื่นได้

-เราสามารถรับ Input จาก User ได้ด้วย readline() เช่น:

```
#Chatbot
greeting_bot <- function() {
  username = readline("What's your name: ")
  print(paste("Hello!", username))

  your_age = readline("How old are you? ")
  print(paste("You are", your_age, "years old."))
}
```

*ໃນ replit ໃຫ້ໃຊ້ readlines() ໃນການຮັບຂໍ້ມູນແກ່ readline()

-Control Flow (If, For, While): ເນື້ອງ Code ຄວບຄຸມພຸດີກຣມຂອງໂປຣແກຣມ

```
#if
score <- 85
if (score > 85) {
  print("Passed")
} else {
  print("Failed")
}

#ifs (elif)
if (score > 85) {
  print("Passed")
} else if (score > 65) {
  print("OK")
} else {
  print("Failed")
}

#while
count <- 0 #counter

while (count < 5) {
  print("GG")
  count <- count+1 #Update counter
}

#for
numbers <- 1:5
numbers ** 2
friends <- c("toy",
            "lisa",
            "jisoo")
for (friend in friends) {
  text = paste("Hi!", toupper(friend))
  print(text)
}

#Easy than using For loop
paste("Hi!", toupper(friends))
```

*ใน R แบบไม่จำเป็นต้องใช้ For Loop

-Regular Expression:

```
state.name  
  
city_start_A <- grep("^A", state.name)  
  
city_end_s <- grep("s$", state.name)  
  
city_contain_new <- grep("new", state.name, ignore.case = TRUE)  
  
state.name[city_contain_new]
```

Conclusion:

ภูรินทร์ Today at 1:33 AM

1. Core Concepts คือ Variable (ประกาศตัวแปร), Data Types (ประเภทของข้อมูล ประกอบด้วย numeric, character, date, logical), Data Structure (โครงสร้างของข้อมูล ประกอบด้วย vector, list, matrix, data frame), Function (ชุดคำสั่งที่เราสามารถสร้างเองและนำมาใช้ซ้ำได้), และ Control Flow (ควบคุมพฤติกรรมของโปรแกรม เช่น if for while) (edited)
2. Data Structure มี Vector (เก็บข้อมูลได้หลายก้อน แต่ทุก ก้อนต้องเป็นข้อมูลประเภทเดียวกัน), Matrix (Vector 2 มิติ), List (เก็บข้อมูลได้หลายประเภท), และ Data Frame (เก็บข้อมูลเป็นตาราง)
3. Subset ทำได้ 3 วิธี คือ Subset by Position (ตำแหน่งของข้อมูล ใช้ []), Subset by Name (ชื่อของ ข้อมูล ใช้ \$), และ Subset by Condition (เงื่อนไข)

Homeworks

1st Homework: ทำ Chatbot รับ 4-5 ข้อความให้ User พิมพ์เข้าระบบ

*หาก print และงผลไม่ได้ผล ให้ลองใช้ cat หรือ message แทน (งานปกติใช้ print หรือ cat)

Google Colaboratory

 <https://colab.research.google.com/drive/1ny5ugssF3d6PlQijq4N-0Wgo0yP1zen2?usp=sharing>



2nd Homework: สร้างโปรแกรมเป่ายิ้งฉุบ

Google Colaboratory

 https://colab.research.google.com/drive/1owDCTjomll1BbwZz5w-UzvhyZTA1-n_s?usp=sharing



Extra Project

-Two input calculator:

Google Colaboratory

 https://colab.research.google.com/drive/1ZCL8QdMwh_QrDiu1rvGFcRs59F2lsnPO?usp=sharing

