



กิจกรรมค่ายสู่ฝันวันนักวิทย์

MQTT Tic-Tac-Toe

ภาควิชาวิทยาการคอมพิวเตอร์

คณะวิทยาศาสตร์

มหาวิทยาลัยศรีนครินทรวิโรฒ

30 ตุลาคม 2564



การกำหนดกลุ่มของคำสั่ง (Indentation)

- Indentation คือการเว้นย่อหน้า ที่ส่วนต้นในแต่ละแถวของคำสั่ง
- ในภาษาไพทอนใช้การเว้นย่อหน้านี้ช่วยในการแบ่งกลุ่มของคำสั่ง (block of code)

```
1  for i in range(1,10):
2      print(i)
3      func1()
4  indent
5  ←→
6
7
8
9
```

กลุ่มคำสั่งเดียวกันจะต้องมี
การเว้นย่อหน้าเท่ากัน

- หากเว้นย่อหน้าไม่เท่ากันจะเกิด indentation Error
- ในการย่อหน้าต้องใช้อักขระแบบเดียวกัน
 - 4 spaces ไม่เท่ากับ 1 Tab

```
1  L = list("python")
2  print(L[0])
3  print(L[1])
4  print(L[2])
5  print(L[3])
6  print(L[4])
7  print(L[5])
```

File "<ipython-input-4-be89818fb65a>", line 3
print(L[1])
^
IndentationError: unexpected indent



อักษรตัวเล็กและตัวใหญ่ค่าไม่เท่ากันในไพทอน(case sensitive)

- ต้องให้ความระมัดระวังเรื่องการใช้ตัวอักษรเล็กหรือใหญ่
- การเรียกชื่อตัวแปร การเรียกชื่อฟังก์ชัน ที่ไม่เหมือนกันถือว่าเป็นคนละตัวกัน

```
1 var1 = 10
2 print(var1)
3 print(Var1)
4 print(vaR1)
```

var1, Var1, vaR1 ถือว่าเป็นคนละตัวกัน

10

NameError

Traceback (most recent call last)

<ipython-input-5-2873241e0afa> in <module>

```
1 var1 = 10
2 print(var1)
----> 3 print(Var1)
4 print(vaR1)
```

NameError: name 'Var1' is not defined



คอมเมนต์ (Comments)

- คอมเมนต์คือส่วนของโค้ดที่ไพทอนจะไม่สนใจ
- คอมเมนต์ถูกใช้ในการอธิบายโค้ดเพื่อเพิ่มความเข้าใจ
- ในบางครั้งเราสามารถใช้เพื่อยกเลิกการทำงานของคำสั่ง
- บรรทัดที่เป็นคอมเมนต์จะขึ้นต้นด้วย #

```
1  #This is a comment
2  print("Hello, Python1")
3
4  print("Hello, Python2") #This is a comment
5
6  #print("Hello, Python3")
```

Hello, Python1
Hello, Python2



ตัวแปร (Variable)

- ตัวแปรถูกสร้างขึ้นเพื่อเอาไว้เก็บข้อมูล เพื่อการประมวลผลต่างๆ
- ตัวแปรจะถูกสร้างขึ้นทันทีที่มีการกำหนดค่าให้กับมันในครั้งแรก

```
1 x = 5
2 y = 0.5
3 z = "Python"
4 print(type(x))
5 print(type(y))
6 print(type(z))
```

```
<class 'int'>
<class 'float'>
<class 'str'>
```

ตัวแปรจะมีชนิดตามข้อมูลที่มันเก็บ
x เป็นชนิดเลขจำนวนเต็ม (integer)
y เป็นชนิดเลขทศนิยม (floating point)
z เป็นชนิดข้อความ (string)



โครงสร้างการเขียนโปรแกรม

ส่วน A

```
1 import random
2 import time
3 import json
4 from enum import IntEnum
```

สำหรับอ้างอิงโมดูลที่ต้องการใช้งาน

ส่วน B

```
5
6 department = "Computer Science"
7 university = "SWU"
```

กลุ่มตัวแปรที่ประกาศไว้ใช้งาน ตัวแปรในส่วนนี้สามารถเข้าถึงได้จากทุกที่ในโปรแกรม (global variable)

ส่วน C

```
8
9 def show10student():
10     print("---print from function---")
11     for i in range(0,10):
12         std_id = "id6500"+str(random.randint(0, 1000))
13         student = std_id+":"+department+":"+university
14         print(student)
```

ประกาศฟังก์ชันต่างๆไว้ใช้งาน

ส่วน D

```
15
16 print("Start program")
17 show10student()
18 print("End program")
19
```

จุดเริ่มต้นกลุ่มคำสั่งหลักของโปรแกรม

ผลลัพธ์

Start program

---print from function---

id6500114:Computer Science:SWU

id6500117:Computer Science:SWU

id6500890:Computer Science:SWU

id6500261:Computer Science:SWU

id6500440:Computer Science:SWU

id6500336:Computer Science:SWU

id6500118:Computer Science:SWU

id6500796:Computer Science:SWU

id6500741:Computer Science:SWU

id6500276:Computer Science:SWU

End program



การตรวจสอบเงื่อนไข

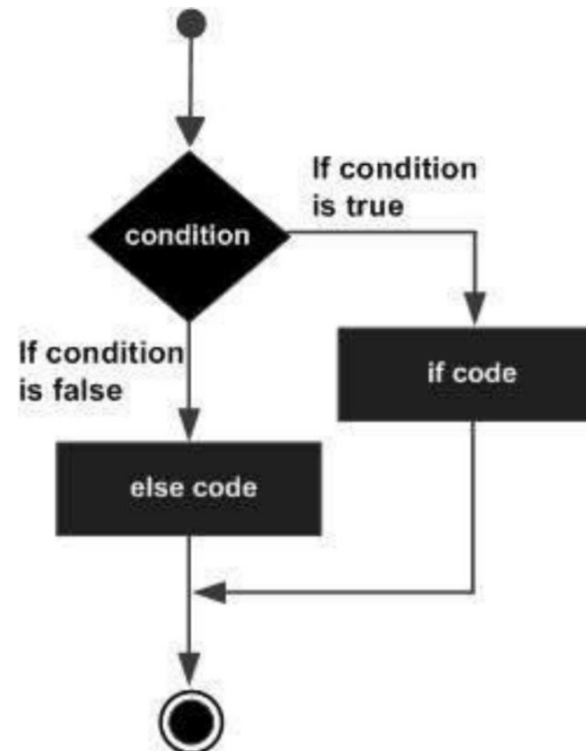
- คำสั่ง if (if statement) : หากเงื่อนไขเป็นจริงจะทำงานตามชุดคำสั่งในส่วนของ if หากไม่จริงจะทำชุดคำสั่งในส่วนของ else

If เงื่อนไข :

{ _____ ชุดคำสั่ง
_____ ทำเมื่อตรงกับเงื่อนไข

else :

{ _____ ชุดคำสั่ง
_____ ทำกรณีที่ไม่ตรงกับเงื่อนไข



ตารางสรุปการใช้เครื่องหมายโอเปอเรเตอร์เปรียบเทียบ

โอเปอเรเตอร์	ความหมาย	ตัวอย่าง
=	เท่ากับ	A == 10
>	มากกว่า	A > 3
>=	มากกว่าหรือเท่ากับ	A >= 3
<	น้อยกว่า	A < 2
<=	น้อยกว่าหรือเท่ากับ	A <= 2
!=	ไม่เท่ากับ	A != 10
not	ตรวจสอบค่าตรงกันข้ามระหว่าง True กับ False	Not A

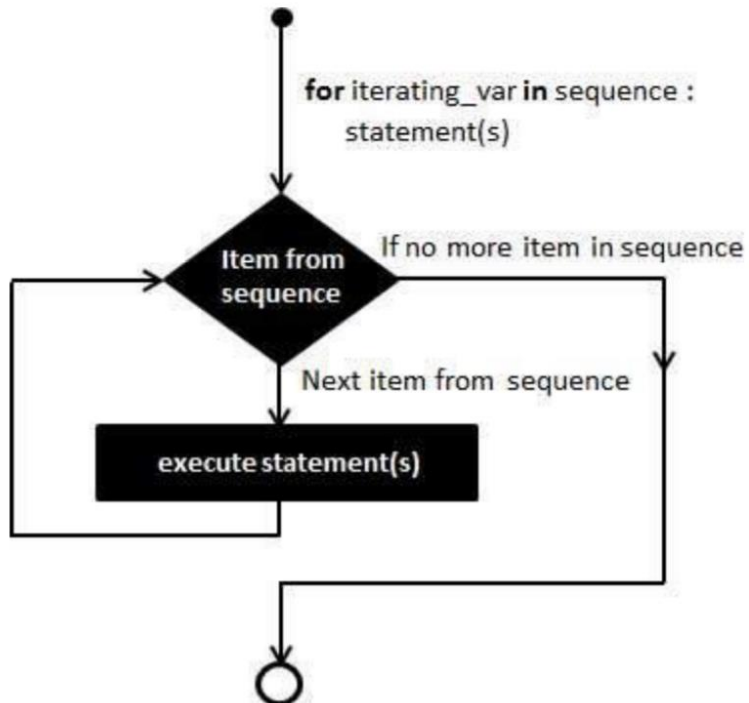


คำสั่งการวนซ้ำ (Repetition, Loop)

- กำหนดให้ทำกลุ่มคำสั่งย่อยตามจำนวนรอบที่กำหนด หรือในขณะที่เงื่อนไขเป็นจริง
- กลุ่มคำสั่งการวนซ้ำประกอบด้วยคำสั่ง **for** และคำสั่ง **while**

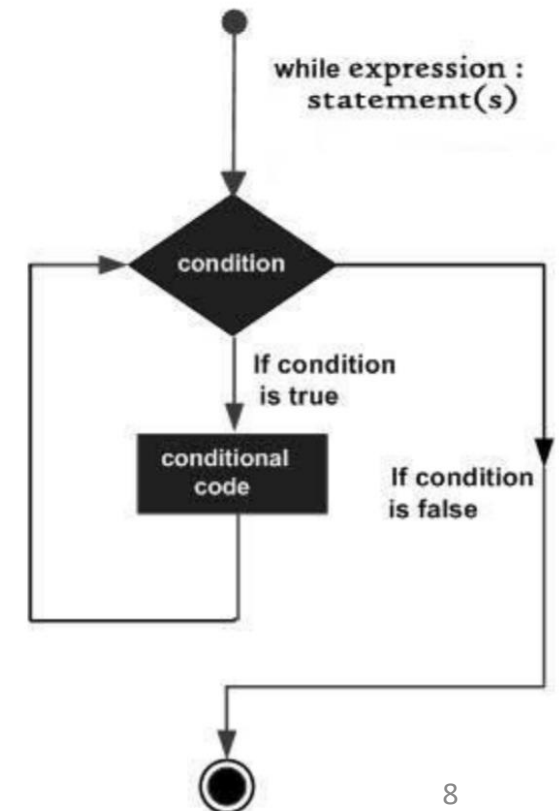
1	for i in [1,2,3,4]:
2	print(i)

1
2
3
4



1	i = 1
2	while i < 5:
3	print(i)
4	i=i+1

1
2
3
4





คำสั่งวนทำซ้ำแบบไม่มีวันสิ้นสุด และการหยุดการวนทำซ้ำ

- การวนซ้ำแบบไม่มีวันสิ้นสุด (infinite loop)
- สามารถหยุดการวนทำซ้ำได้ด้วยคำสั่ง **break**

```
1 while True:
2     d = input()
3     if d == 'exit':
4         print("Bye Bye")
5         break
```

hello
exit
Bye Bye

```
1 while 1:
2     d = input()
3     if d == 'exit':
4         print("Bye Bye")
5         break
```

hello
exit
Bye Bye



ฟังก์ชัน

- อาจเรียกว่า subprogram หรือ subroutine
- เป็นการแยกส่วนสิ่งที่ซ้ำๆ กัน หรือเข้าใจยาก ออกมาจากโปรแกรมหลัก
- ฟังก์ชันควรมีหน้าที่การทำงานชัดเจน เช่น ฟังก์ชันหาค่าเฉลี่ย

องค์ประกอบของฟังก์ชัน

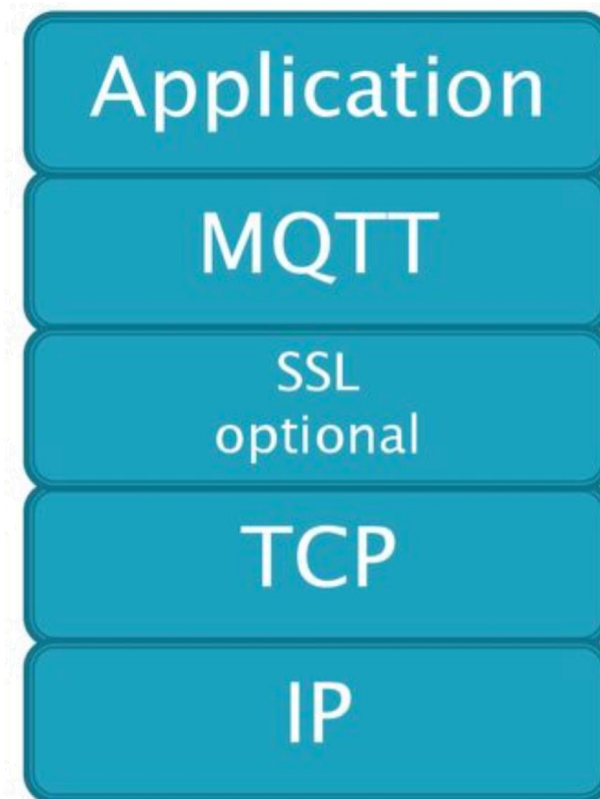
- เริ่มต้นด้วย def เสมอ
- ชื่อฟังก์ชัน มีข้อกำหนดเหมือนการตั้งชื่อตัวแปร
- ค่าที่รับเข้ามา หรือ "พารามิเตอร์" (ไม่จำเป็นต้องมีก็ได้)
- การคืนค่าจากฟังก์ชันด้วยคำสั่ง return (ไม่จำเป็นต้องมีก็ได้)

```
1  def average4(w,x,y,z):  
2      s = w+x+y+z  
3      return s/4
```



MQTT Protocol

- MQTT เป็น Protocol ที่นิยมใช้ในงาน IoT
- ทำงานอยู่บน TCP/IP เหมือนกับ HTTP
- เนื่องจาก data packet ที่รับส่งมีขนาดเล็ก
- ลดการใช้งาน bandwidth – Small header
- ประหยัดพลังงานของอุปกรณ์ IoT โดยเฉพาะอุปกรณ์ที่ใช้แบตเตอรี่เป็นแหล่งพลังงาน
- เป็นการสื่อสารแบบ Asynchronous

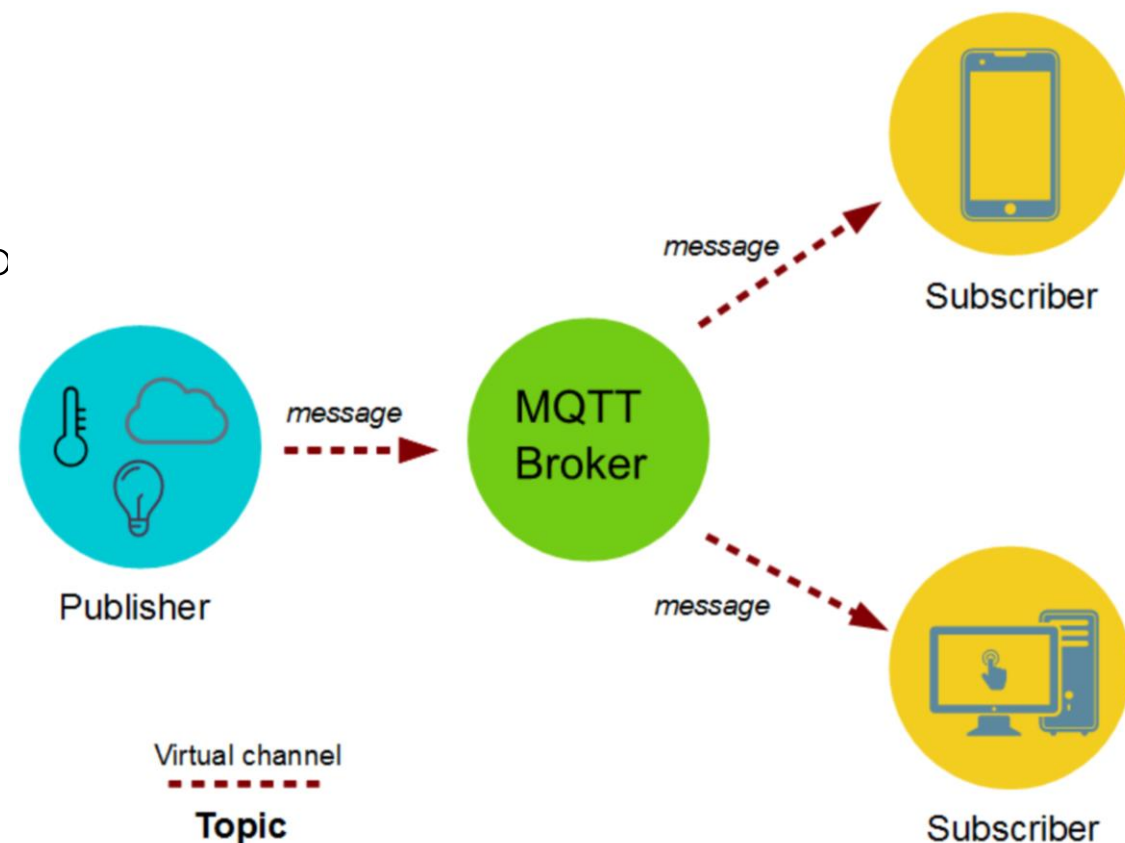


TCP/IP Port: 1883

When running over SSL, TCP/IP port 8883

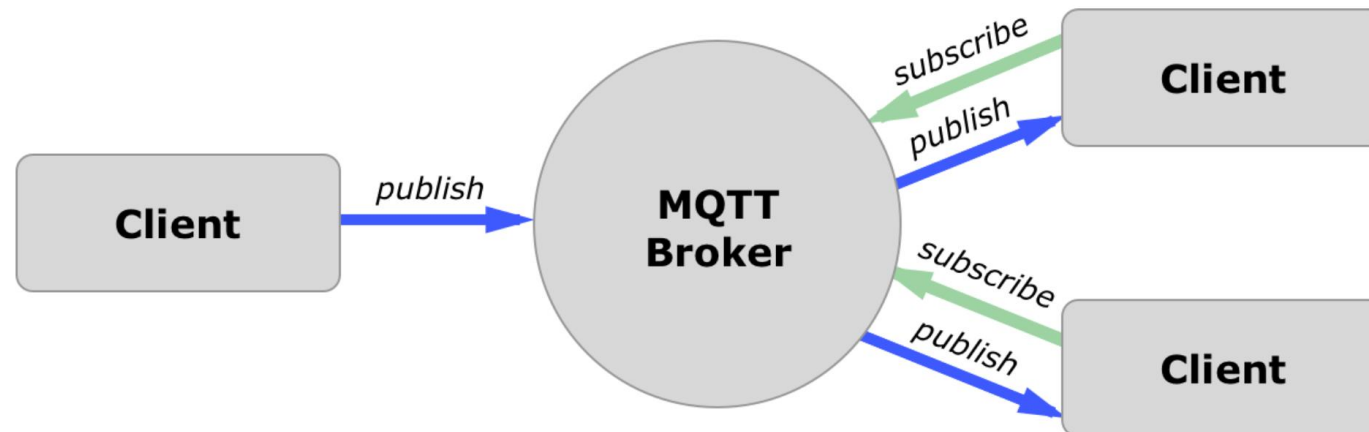
MQTT: Components

- **Broker**, which is the server that handles the data transmission between the clients.
- **A topic**, which is the place a device want to put o retrieve a message to/from.
- **The message**, which is the data that a device receives “when subscribing” from a topic or send “when publishing” to a topic.
- **Publish**, is the process a device does to send its message to the broker.
- **Subscribe**, where a device does to retrieve a message from the broker.



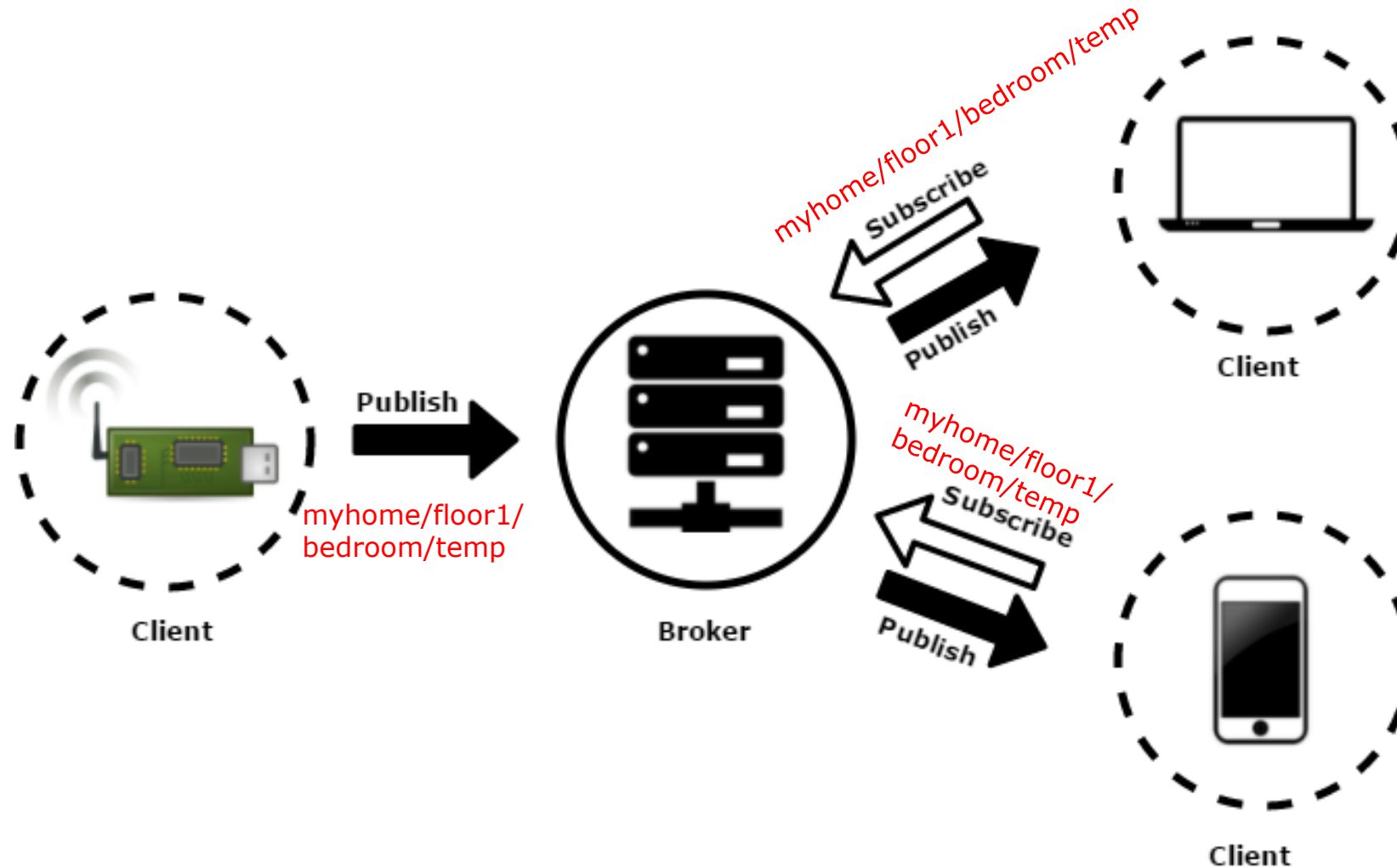
How MQTT works

- MQTT is based on clients and a server.
- MQTT server is called a broker and the clients are simply the connected devices.
 - When a device (a client) wants to send data to the broker, we call this operation a “publish”.
 - When a device (a client) wants to receive data from the broker, we call this operation a “subscribe”.
- Clients are publishing and subscribing to topics. So, the broker here is the one that handles the publishing/subscribing actions to the target topics.





MQTT Routing – Topic based





Topics

- MQTT Topics are structured in a hierarchy similar to folders and files in a file system using the forward slash (/) as a delimiter.
- Allow to create a user friendly and self descriptive naming structures

- Topic names are:

- Case sensitive
- use UTF-8 strings.
- Must consist of at least one character to be valid.

Topic level separator

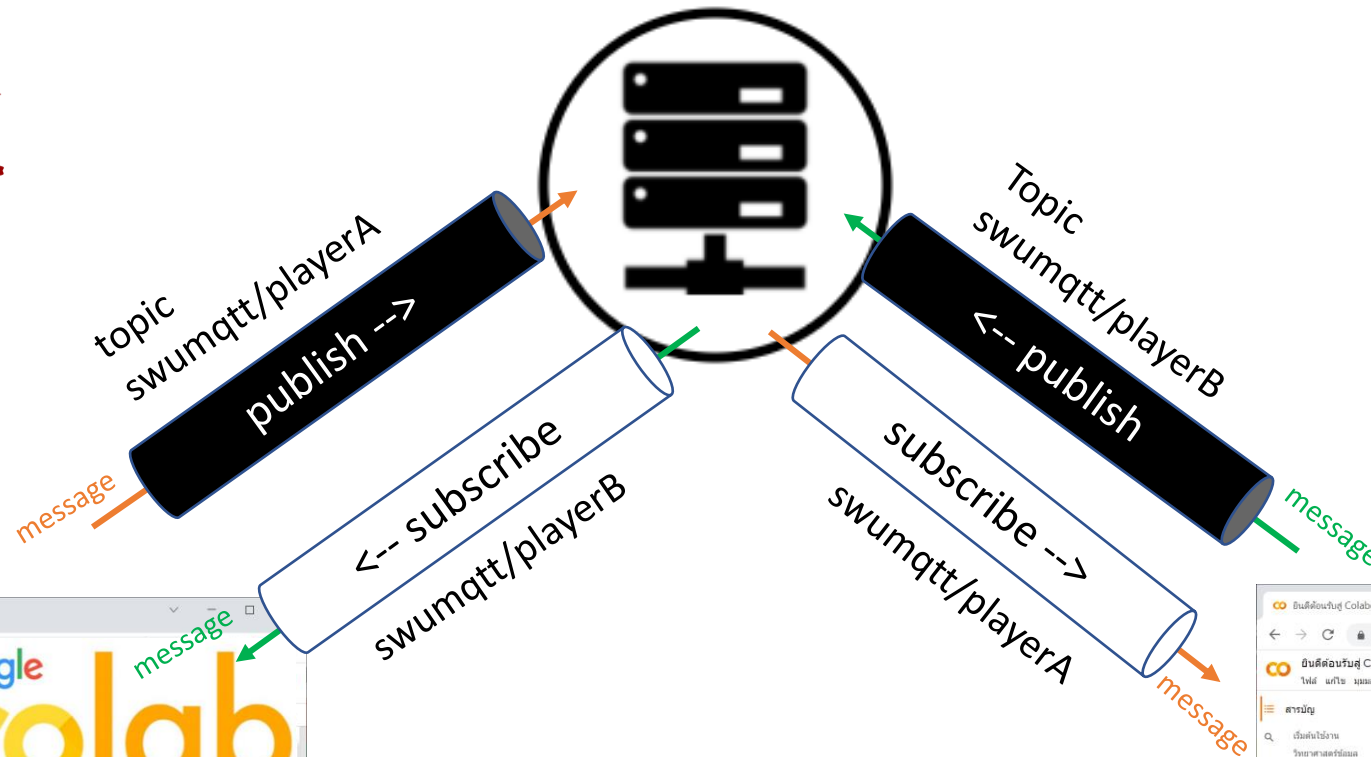
swumqtt/group1/player1

Topic level Topic level

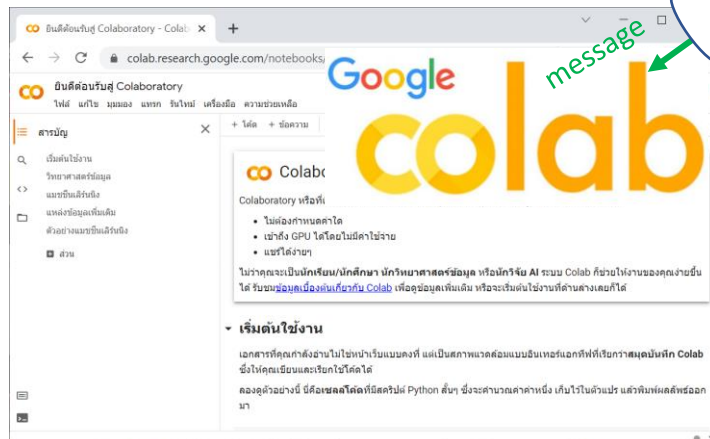


MQTT Tic-Tac-Toe

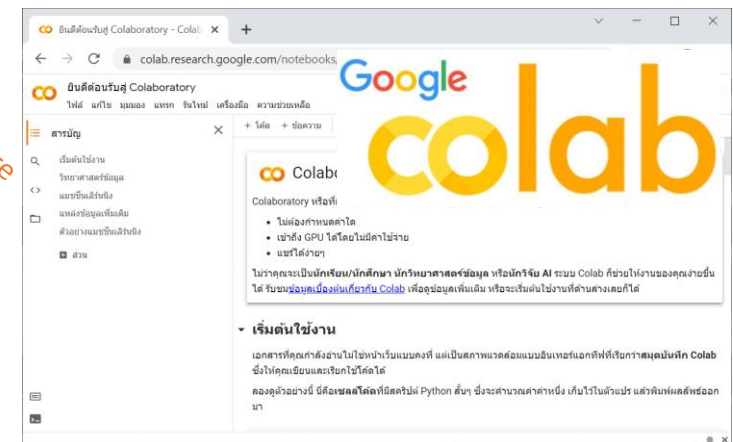
Broker: mqtt.eclipseprojects.io port: 1883



Tic-Tac-Toe player A



Tic-Tac-Toe player B





MQTT-Message

- ข้อมูลใน mqtt message มักอยู่ในรูปแบบ JSON
- JSON (JavaScript Object Notation) เป็นรูปแบบที่ใช้ในการแลกเปลี่ยนรับส่งข้อมูล
- เข้าใจง่าย, ขนาดเล็ก, สิ้นเปลือง Network Bandwidth น้อย
- ง่ายต่อเครื่องจักร (Machines, Computer) ในการวิเคราะห์(Parse) หรือสร้าง (Generate)
- โครงสร้างอยู่ในรูปแบบ **Key: Value**
- รูปแบบในการเขียน JSON
 - JSON Object: คล้ายกับข้อมูล 1 เรคอร์ด สามารถมีฟิลด์เดียว หรือหลายฟิลด์ก็ได้
 - JSON Array: เป็นข้อมูลหลายเรคอร์ด โครงสร้างคล้าย Array 2 มิติ



JSON Syntax

- **JSON objects** start the object with “{” and end in with “}”
 - Members (properties), use pairs of “key : value”

ข้อมูล Key:Value อยู่ภายใน “{ Key:Value , Key:Value, ... ,Key:Value}”

Ex.

```
{ "title" : "เครื่องทำน้ำอุ่น" }  
  
{ "id" : 0,  
  "title" : "เครื่องทำน้ำอุ่น",  
  "Status" : "On" }
```

- **JSON arrays** put the arrays between “[]”
 - Elements put the values directly separated by commas

ข้อมูลเก็บอยู่ใน “[value, value, ... ,value]”

Ex.

```
[ { "id" : 0, "title" : "เครื่องทำน้ำอุ่น", "Status" : "On" },  
  { "id" : 1, "title" : "เครื่องกรองอากาศ", "Status" : "Off" },  
  { "id" : 2, "title" : "ไฟส่องสว่างชั้น1", "Status" : "On" } ]
```



อ้างอิง

- Python ๑๐๑ หนังสือสอนเขียนโปรแกรมภาษา Python
<https://www.cp.eng.chula.ac.th/books/python101/>