

ทีมวิทยากรอบรม

โครงการหลักสูตรอบรม NOSQL WITH MONGODB



ผศ.ดร.อรวรรณ อิ่มสมบัติ
aurawan.i@ku.th



ดร.จิวรรณ เจริญสุข
Jirawan.charo@ku.th



ดร.ชโลธร ชูทอง
chootong.c@ku.th



บทที่ 3. MONGODB AND DATABASE MANAGEMENT WITH PYTHON



KU
KASETSART
UNIVERSITY

บทที่ 3 MONGODB AND DATABASE MANAGEMENT WITH PYTHON

- รู้จักชนิดของข้อมูล ตัวอย่างข้อมูลในการใช้งาน MongoDB
- การเชื่อมต่อ Atlas Cluster ด้วย Python
- เรียนรู้การเชื่อมต่อข้อมูลมาใช้งาน
- การสร้างฐานข้อมูล
- การจัดการฐานข้อมูล
- การเชื่อมโยงข้อมูล
- การทำงานกับข้อมูลที่มีอยู่
- การสร้าง Models of Mongoose





Introduction To Google Colab

(รู้จัก Google Colab)

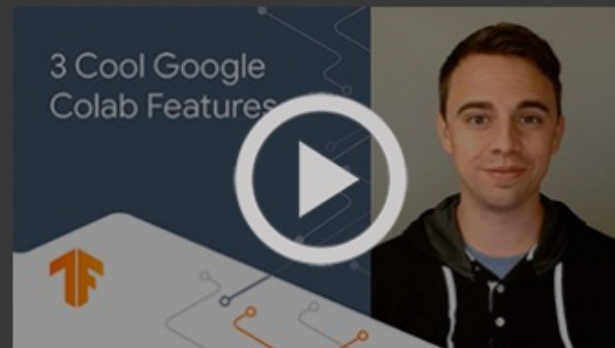
<https://colab.research.google.com>

Welcome to Colab



Welcome to Colab!

If you're already familiar with Colab, check out this video to learn about interactive tables, the executed code history view and the command palette.



What is Colab?

Colab, or 'Colaboratory', allows you to write and execute Python in your browser, with

- Zero configuration required
- Access to GPUs free of charge
- Easy sharing

Whether you're a **student**, a **data scientist** or an **AI researcher**, Colab can make your work easier. Watch [Introduction to Colab](#) to find out more, or just get started below!

Google Collaboratory



<https://colab.research.google.com>

Google search results for "google colab". The search bar shows "google colab" and the results list includes "colab.research.google.com" and "Google Colab".

ผลการค้นหาประมาณ 7,310,000 รายการ (0.41 วินาที)

colab.research.google.com ▾ [เปลี่ยนหน้านี้](#)

Google Colab

Colab notebooks allow you to combine executable code and rich text in a single document along with images, HTML, LaTeX and more. When you create your ...

Colab

Colaboratory หรือที่เรียกสั้นๆ ว่า "Colab" ช่วยให้เขียนและเรียกใช้ ...

Python Data Science

... notebook version of the Python Data Science Handbook by ...

Introduction to Colab and Python

Introduction to Colab and Python - Run in Google Colab ...

Colaboratory

Colaboratory, or "Colab" for short is a product from Google ...

Google Colab Notebook

Enter a search string to filter the list of notebooks shown below. Title ...

Google Drive

Google Drive. You can access files in Drive in a number of ways ...

ผลการค้นหาเพิ่มเติมจาก google.com >

Welcome To Colaboratory

File Edit View Insert Runtime Tools Help

+ Code + Text Copy to Drive

Connect Editing

Table of contents Code snippets Files

EXAMPLES RECENT GOOGLE DRIVE GITHUB UPLOAD

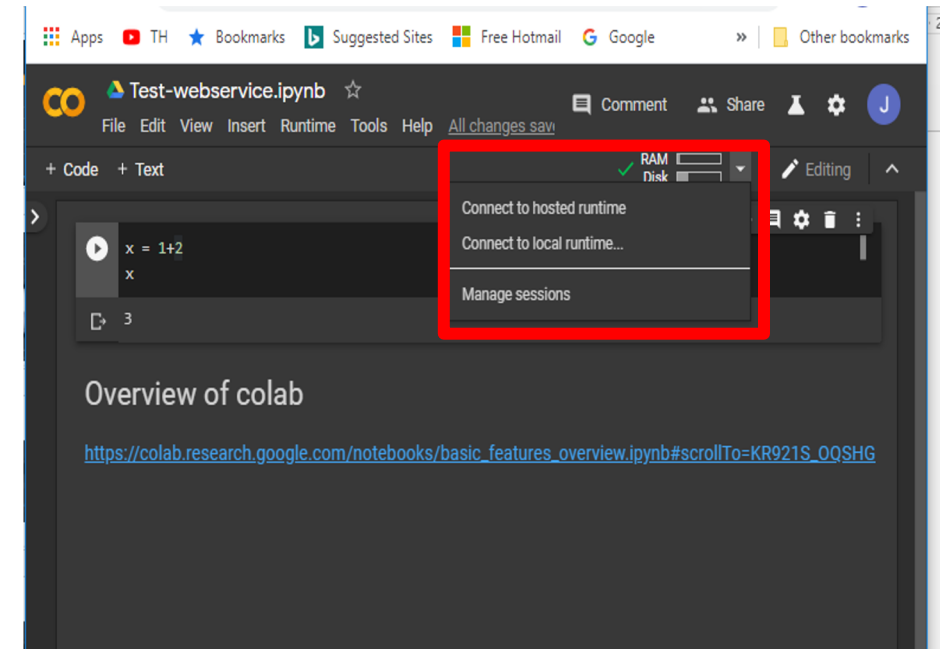
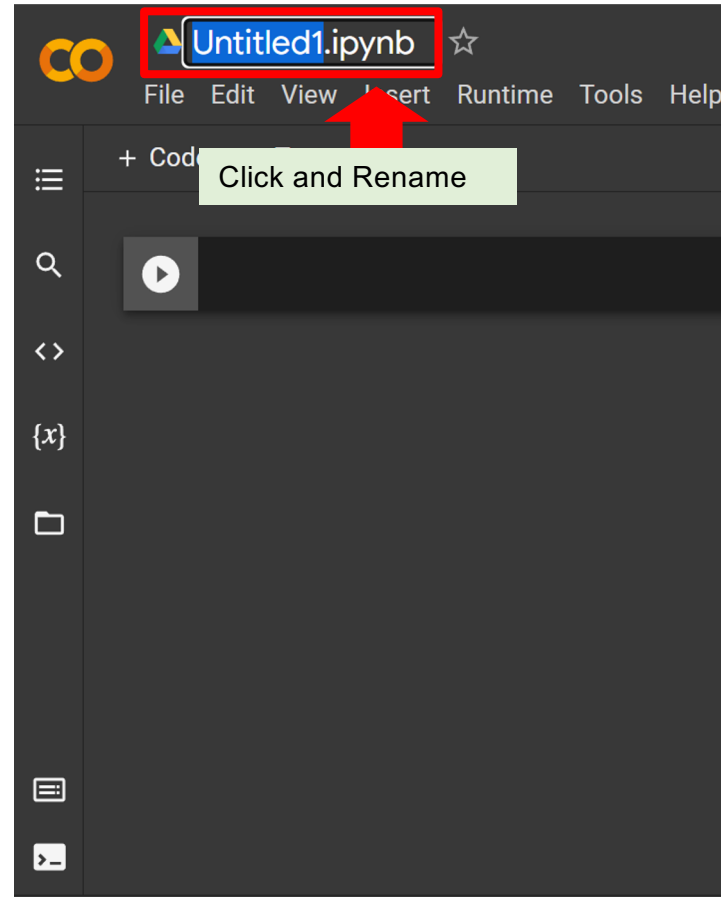
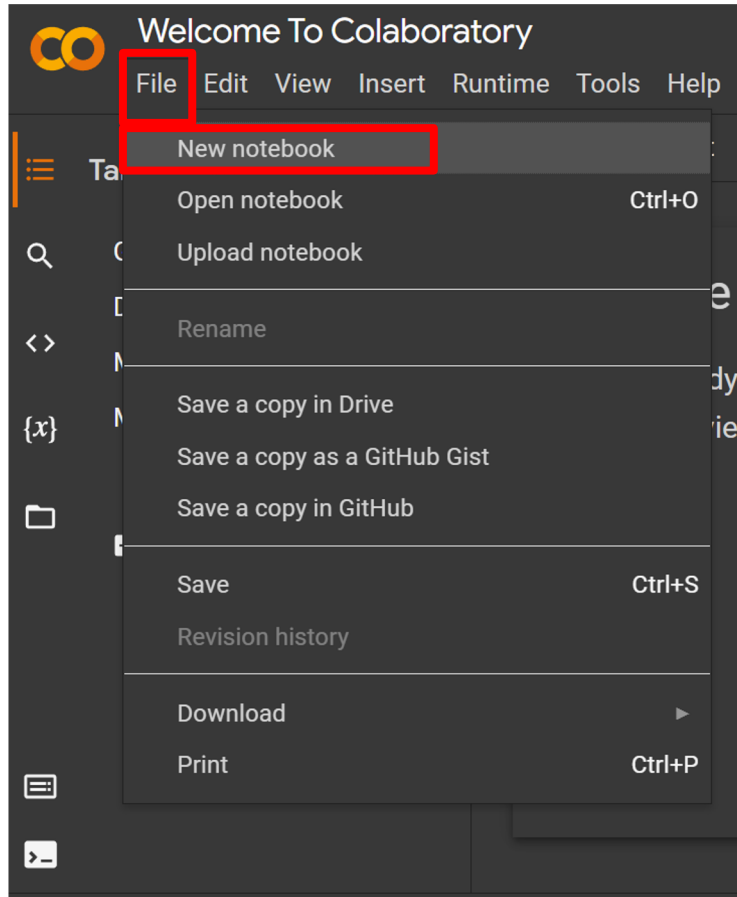
Filter notebooks

Title	First opened	Last opened	
Welcome To Colaboratory	Dec 23, 2018	0 minutes ago	
nltk_preprocess.ipynb	3 hours ago	3 hours ago	
1) Simple Thai text preprocessing.ipynb	6 days ago	3 hours ago	

New notebook & Rename



Select Resource



Insert Code cell/ Text cell



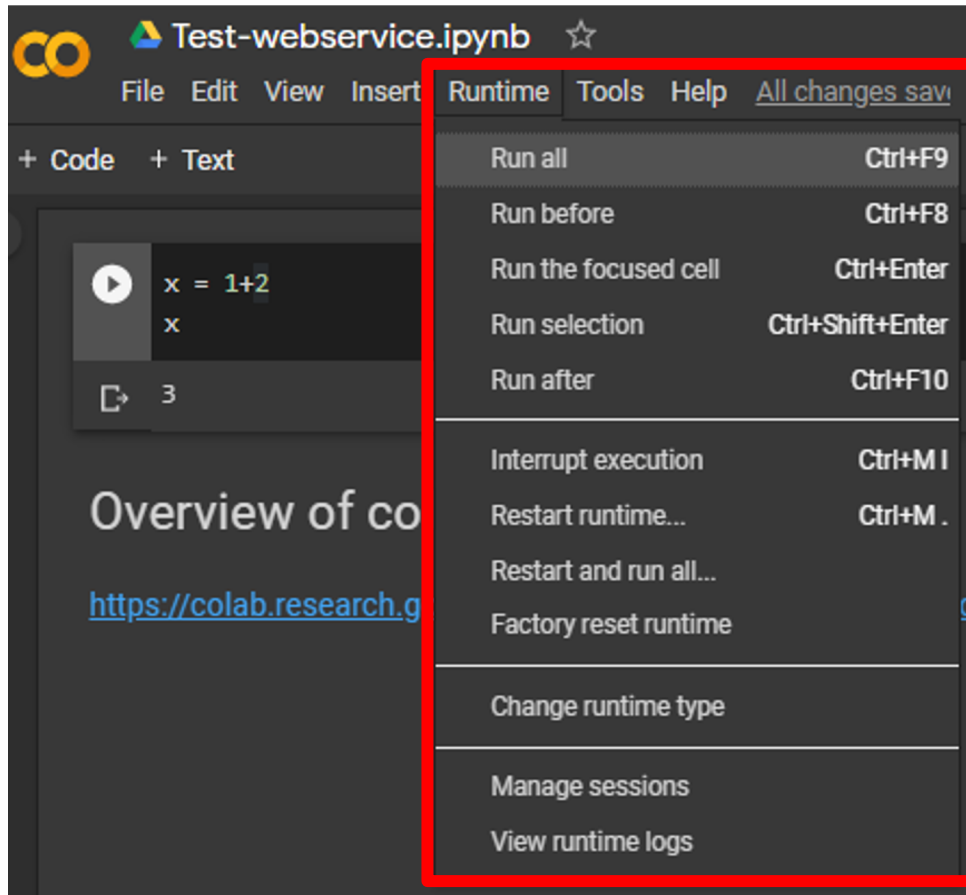
The image displays three overlapping screenshots of a Jupyter Notebook interface, demonstrating how to insert different types of cells. Each screenshot shows a notebook titled 'Test-webservice.ipynb' with a menu bar (File, Edit, View, Insert, Runtime, Tools, Help) and a toolbar with options like Comment, Share, RAM, and Disk usage.

- Top Screenshot:** The '+ Code' button in the toolbar is highlighted with a red box.
- Middle Screenshot:** The '+ Text' button in the toolbar is highlighted with a red box.
- Bottom Screenshot:** The 'Manage cell' toolbar is highlighted with a red box, showing icons for running, moving up/down, linking, commenting, settings, and deleting a cell.

Each screenshot also shows a code cell containing the following Python code:

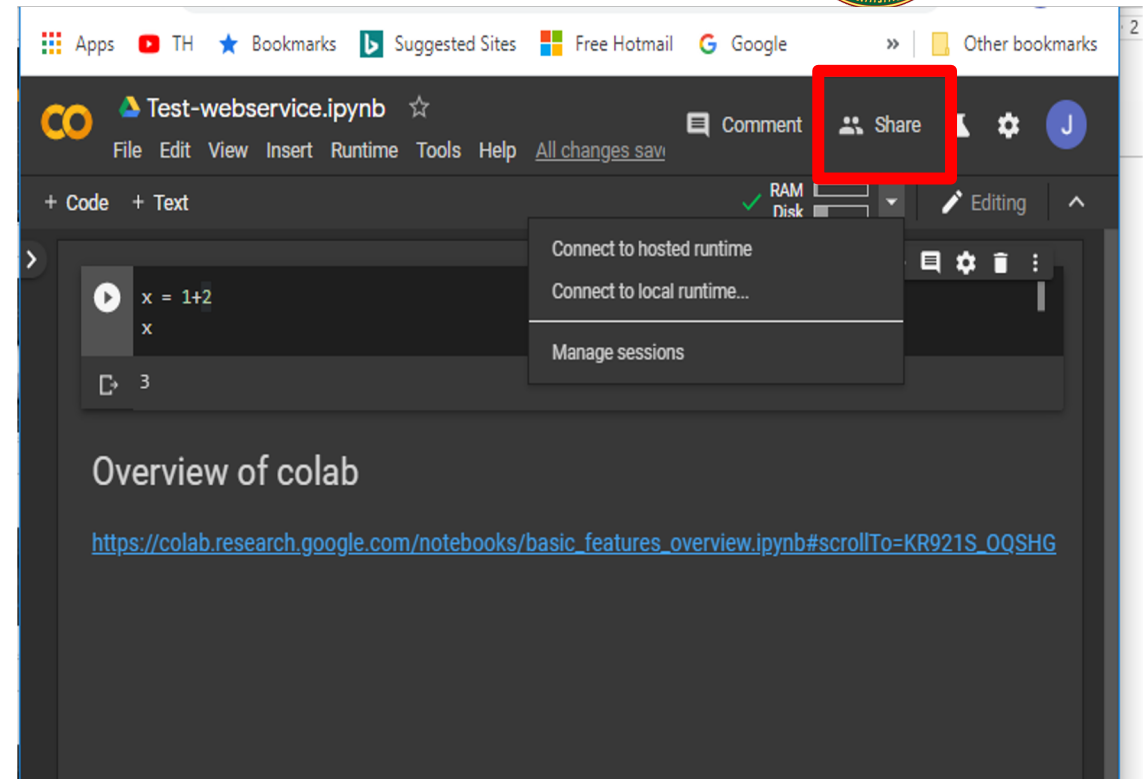
```
[3] x = 1+2
x
```


Run



The screenshot shows the 'Runtime' menu in Google Colab, which is highlighted with a red border. The menu items and their keyboard shortcuts are as follows:

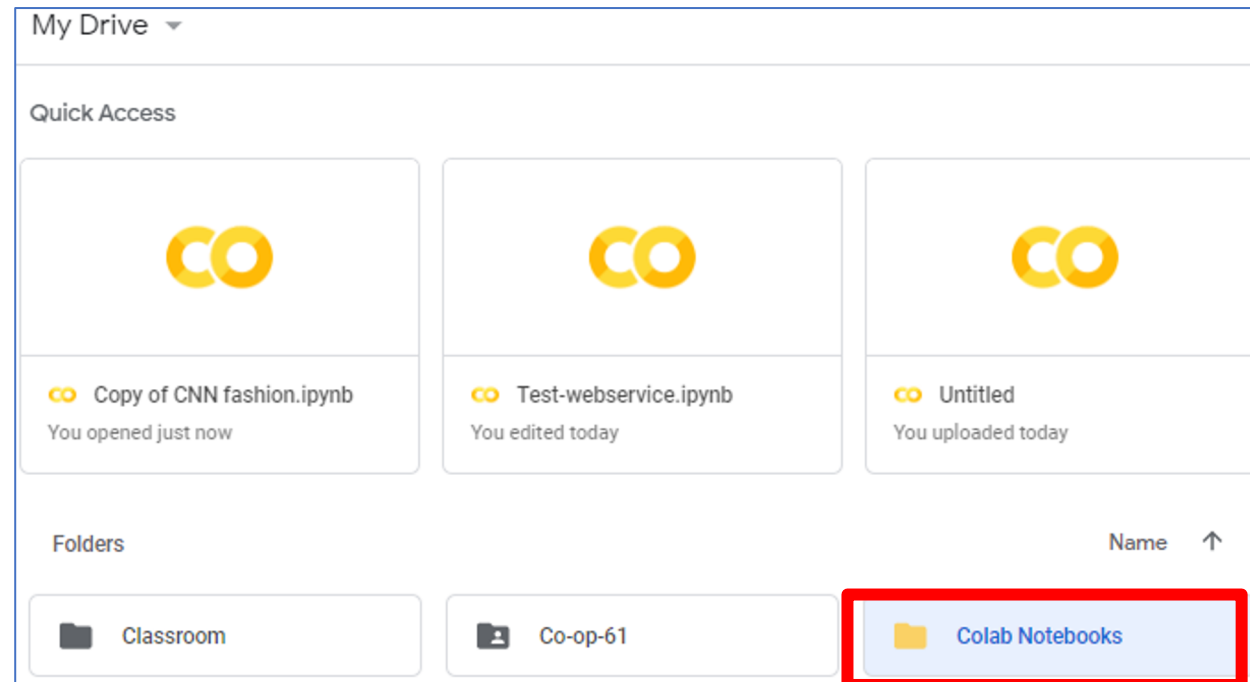
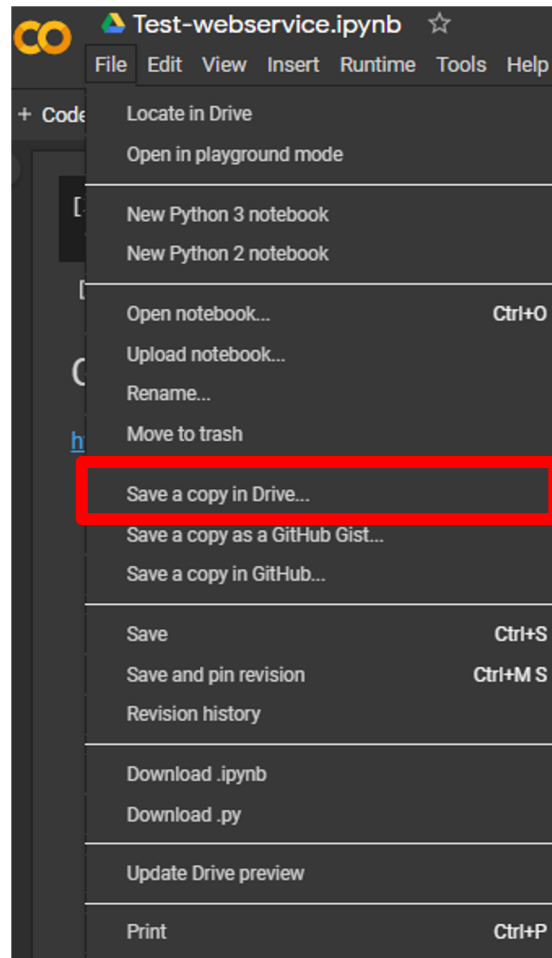
Action	Shortcut
Run all	Ctrl+F9
Run before	Ctrl+F8
Run the focused cell	Ctrl+Enter
Run selection	Ctrl+Shift+Enter
Run after	Ctrl+F10
Interrupt execution	Ctrl+M I
Restart runtime...	Ctrl+M .
Restart and run all...	
Factory reset runtime	
Change runtime type	
Manage sessions	
View runtime logs	



The screenshot shows the 'Share' button in Google Colab, which is highlighted with a red border. The button is located in the top right corner of the interface, next to the 'Comment' button. A dropdown menu is visible below the 'Share' button, containing the following options:

- Connect to hosted runtime
- Connect to local runtime...
- Manage sessions

Save file (.ipynb)





Python Structure

(โครงสร้างของภาษา PYTHON)

Keywords

- คำสงวนไว้ในการเขียนโปรแกรมภาษา python ซึ่งไม่สามารถใช้คำเหล่านี้ในการตั้งชื่อตัวแปร ชื่อฟังก์ชัน คลาส ได้แก่

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

Literals

- literals คือ ข้อมูลที่เป็นค่าคงที่ ตัวอักษร เครื่องหมาย ในโปรแกรมภาษา python
- สามารถแบ่งประเภทของข้อมูลได้ในแบบต่างๆ เช่น Integer, Floats, number boolean และ String

```
a = 9 #กำหนดให้ a มีค่าเท่ากับ 9
b = 22.2 #กำหนดให้ b มีค่าเท่ากับ 22.2
c = True #กำหนดให้ c เป็นรูปแบบ boolean มีค่าเป็น True
d = "Hello" #กำหนดให้ d เป็นรูปแบบ String มีค่าเป็น Hello
e = 'Python'
```

```
f = 3+4
g = f-5
h = 'Hello' + ' MongoDB'
print(f, '\n', g, '\n', h)
```

Expressions



- Expression คือการทำงานร่วมกันระหว่างค่าตั้งแต่หนึ่งไปจนถึงหลายค่า โดยค่าเหล่านี้จะมีตัวดำเนินการสำหรับควบคุมการทำงาน

```
# Boolean expressions
print(a == 4)
print(a == 5)
print(a == 4 and b == 5)
print(a == 4 and b == 8)
```

```
# Non-boolean expressions
print(a + b)
print(a + 2)
print(a * b)
print(((a * a) + (b * b)) / 2)
print("NoSQL " + "MongoDB")
```

Module

- โมดูล(Module) คือไฟล์ .py ที่วางอยู่ใต้ package directory
- ใช้คำสั่ง import โมดูลเข้ามาก่อนจึงจะสามารถใช้งาน ฟังก์ชัน หรือคลาสที่อยู่ในโมดูลได้

```
import numpy as np
arr = np.array([1, 2, 3, 4, 5])
print(arr)
```

```
import pymongo
pymongo.version
```

ชนิดของข้อมูล ตัวอย่างข้อมูลในการใช้งาน MongoDB

- JSON (JavaScript Object Notation) เป็นรูปแบบการแลกเปลี่ยนหรือรับส่งข้อมูล
- XML มีโครงสร้างที่ซับซ้อนและมีขนาดใหญ่ JSON ขนาดเบากว่าและมนุษย์อ่านได้เข้าใจ

(รูปแบบการแลกเปลี่ยนข้อมูลในอดีต)

XML

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <endereco>
3   <cep>31270901</cep>
4   <city>Belo Horizonte</city>
5   <neighborhood>Pampulha</neighborhood>
6   <service>correios</service>
7   <state>MG</state>
8   <street>Av. Presidente Antônio Carlos, 6627</street>
9 </endereco>
```

(ปัจจุบัน)

JSON

```
1 {
2   "endereco": {
3     "cep": "31270901",
4     "city": "Belo Horizonte",
5     "neighborhood": "Pampulha",
6     "service": "correios",
7     "state": "MG",
8     "street": "Av. Presidente Antônio Carlos, 6627"
9   }
10 }
```


ชนิดของข้อมูล ตัวอย่างข้อมูลในการใช้งาน MongoDB

- JSON รองรับการทำงานชนิดข้อมูลพื้นฐานทั้งหมด
 - Number: จำนวนเต็มศูนย์ เต็มบวก เต็มลบ
 - String: ข้อความ (“ ”)
 - Boolean: True or False
 - Null: ค่าว่าง
 - Array: ใช้สัญลักษณ์ [] เช่น [val1, val2]
 - Object: ชุดข้อมูลที่เป็นคู่ key-value ใช้สัญลักษณ์ {key1:value1, key2:value2}



ตัวอย่าง Object หรือ Associate Array

```
{  
  "_id": 1,  
  "schId": 10012001,  
  "schName": "สตรีวิทยา",  
  "subdistrict": "บวรนิเวศ",  
  "district": "เขตพระนคร",  
  "province": "กรุงเทพมหานคร",  
  "postCode": 10200,  
  "areaCode": 1000100001,  
  "areaName": "สพม.กรุงเทพมหานคร เขต 1",  
  "minClassLevel": "ม.1",  
  "maxClassLevel": "ม.6",  
  "latitude": 13.757678,  
  "longitude": 100.50182  
},
```

Field: value
Field: value

```
{  
  "student": {  
    "name": "John",  
    "class": "Intermediate",  
    "address": {  
      "street": "2293 Example Street",  
      "City": "Chicago",  
      "State": "IL"  
    }  
  }  
}
```

Object Data

JSON to BJON

- BSON จะมีขนาดเบาและรับส่งข้อมูลได้เร็วเหมือนกับ JSON โดย MongoDB สามารถเข้าถึงข้อมูลหรือ Object ใน BSON เพื่อสร้าง Index ในการทำระบบสอบถามข้อมูล (Query) โดย BSON จะทำงานซ่อนอยู่เบื้อง ส่วนการแสดงผลกับผู้ใช้จะเป็น JSON

	JSON	BSON
Encoding	UTF-8 String	Binary
Data Support	String, Boolean, Number, Array	String, Boolean, Number (Integer, Float, Long, Decimal128...), Array, Date, Raw Binary
Readability	Human and Machine	Machine Only

PyMongo (<https://pymongo.readthedocs.io/en/stable/>)



- PyMongo จะรวบรวม เครื่องมือ สำหรับทำงานกับ [MongoDB](#)
- การติดตั้ง PyMongo โดยใช้คำสั่ง pip install

```
!python3 -m pip install pymongo
```

- แบบระบุ version

```
!python3 -m pip install pymongo==4.2.0
```

- Update version

```
!python3 -m pip install --upgrade pymongo
```

Pymongo[srv]



- SRV เป็นรูปแบบการกำหนดการทำงานเป็น Single hostname เพื่อแก้ปัญหา Multiple hostname
- ตัวดำเนินการ Driver (Driver Conduct) สามารถหาชื่อ hostname จริงๆได้

```
#! python -m pip install pymongo[srv]==3.7.2  
! python -m pip install pymongo[srv]  
! python -m pip install dnspython
```

การเชื่อมต่อ Atlas Cluster ด้วย Python

- ทำการเรียกใช้งาน pymongo

```
import pymongo  
pymongo.version
```

- สร้าง Client เพื่อจะใช้ในการเชื่อมต่อ Atlas

```
from pymongo import MongoClient
```

```
uri =  
"mongodb+srv://mongo_admin:1234@cluster0.7dqitvp.mongodb.  
net/?retryWrites=true&w=majority"  
client = MongoClient(uri)
```

การเชื่อมต่อ Atlas Cluster ด้วย Python



- ทำการเปลี่ยน username และ Password

```
uri =  
"mongodb+srv://<username>:<password>@cluster  
0.7dqitvp.mongodb.net/?retryWrites=true&w=ma  
jority"  
client = MongoClient(uri)
```

คำสั่งพื้นฐานในการจัดการ Document ด้วย python



- แสดง Database ที่มีทั้งหมด (list_database_name())

```
client.list_database_names()
```

- คำสั่งในการสร้าง Database และ Collection ที่ต้องการ

```
#เข้าถึง collection ที่ต้องการ  
db = client["obec"]  
collection = db["area"]
```


คำสั่งพื้นฐานในการจัดการ Document ด้วย python



- สร้างข้อมูลเข้า document. นำข้อมูลเข้าด้วย insert_one()

```
mydict = {"id": 1,  
"area_code": 101,  
"area_name_full": "สำนักงานเขตพื้นที่การศึกษาประถมศึกษากรุงเทพมหานคร",  
"area_name_short": "สพป.กรุงเทพมหานคร",  
"province_code": 10,  
"province": "กรุงเทพมหานคร"}  
  
x = collection.insert_one(mydict)
```

คำสั่งพื้นฐานในการจัดการ Document ด้วย python



- แสดง Collection ทั้งหมดที่มี ด้วย `list_collections()`

```
list (db.list_collections())
```

- แสดงข้อมูลใน Collection (`find()`)

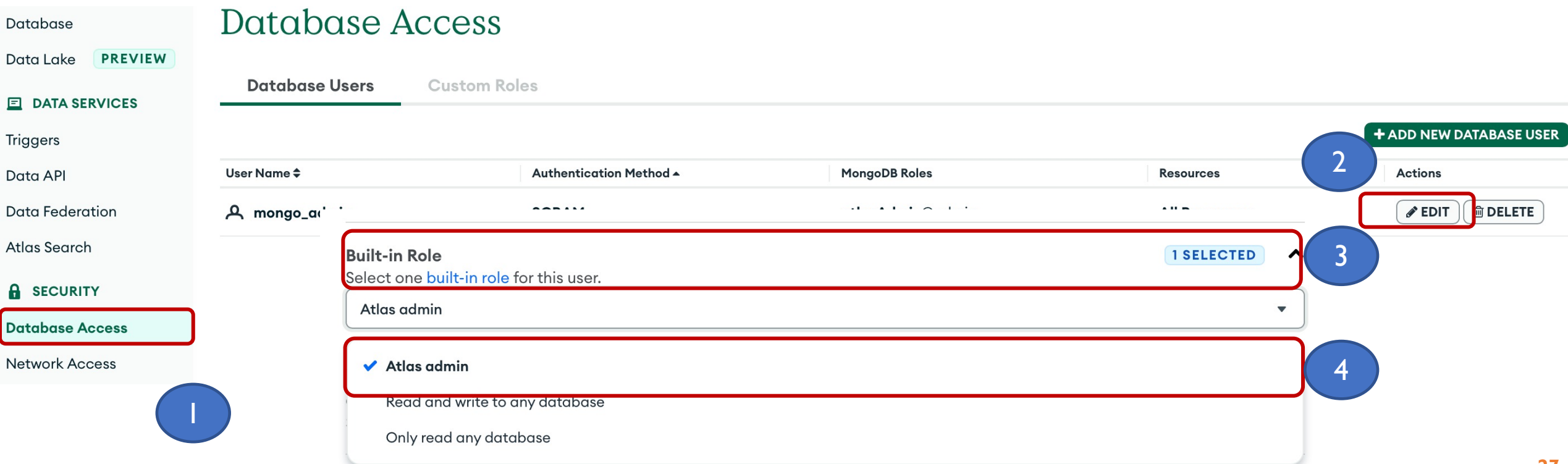
```
list (collection.find())
```

- ลบ database (`drop_database(ชื่อ database)`)

```
client.drop_database('area')
```

กรณีที่ไม่สามารถลบ Document ได้ ทำดังนี้

- ไปที่ Mongo DB Atlas เพื่อกำหนดสิทธิ์ user



The screenshot shows the MongoDB Atlas interface for configuring database access. The left sidebar contains navigation options: Database, Data Lake (PREVIEW), DATA SERVICES, Triggers, Data API, Data Federation, Atlas Search, SECURITY, Database Access (highlighted with a red box and labeled 1), and Network Access. The main content area is titled 'Database Access' and has two tabs: 'Database Users' (active) and 'Custom Roles'. A table lists database users with columns for User Name, Authentication Method, MongoDB Roles, Resources, and Actions. A user named 'mongo_admin' is shown with a red box around the 'EDIT' and 'DELETE' buttons (labeled 2). A dropdown menu for 'Built-in Role' is open, showing 'Atlas admin' selected (labeled 3). Below the dropdown, a list of roles is shown, with 'Atlas admin' selected (labeled 4). The 'Atlas admin' role has the description 'Read and write to any database'. A '+ ADD NEW DATABASE USER' button is visible in the top right.

User Name	Authentication Method	MongoDB Roles	Resources	Actions
mongo_admin	EDIT DELETE

Built-in Role
Select one built-in role for this user. 1 SELECTED

- Atlas admin**
Read and write to any database
- Only read any database

การทำงานกับข้อมูล

- ในการทำงานกับข้อมูลภายใน Collection เราสามารถใช้คำสั่งพื้นฐานดังนี้
- การนำข้อมูลเข้า ด้วยคำสั่ง `insert_one()`

```
id: 2
areacode: "100002"
area_name_full: "Insert text here"
area_name_short: "Insert text here"
province_code: 12
province_name: กรุงเทพมหานคร
```

รับข้อมูลผ่าน **form** เก็บลงตัวแปร

การทำงานกับข้อมูล



ข้อมูลที่ได้รับเข้ามา

```
data = {'id' : id,  
'area_code':areacode,  
'area_name_full': area_name_full,  
'area_name_short': area_name_short,  
'province_code': province_code,  
'province': province_name  
}
```

เพิ่มข้อมูลเข้า
document

```
insert_result = collection.insert_one(data)
```

การทำงานกับข้อมูล



- ตรวจสอบการ `insert` ข้อมูลสำเร็จหรือไม่

```
insert_result.acknowledged
```

- ตรวจสอบ `document id` ว่ามีการนำข้อมูลเข้าเรียบร้อยแล้วหรือไม่

```
insert_result.inserted_id
```

การทำงานกับข้อมูล



- อ่าน document ด้วยคำสั่ง find()

- อ่าน document ทั้งหมด

```
list ( collection.find() )
```

- อ่าน document ตามเงื่อนไข

```
list ( collection.find( { 'area_name_short' : 'สพม นน 1' } ) )
```

การทำงานกับข้อมูล



- การเปลี่ยนแปลงข้อมูลใน document ด้วย update_one()

```
update_result = collection.update_one(  
    {'id' : 2},  
    {'$set': {'area_name_short' : 'BKK'}}  
)
```


การลบ Document

```
#ลบ document
data_delete = { "area_code": "1003" }
collection.delete_one(data_delete)
print('ลบข้อมูลเรียบร้อยแล้ว')
```

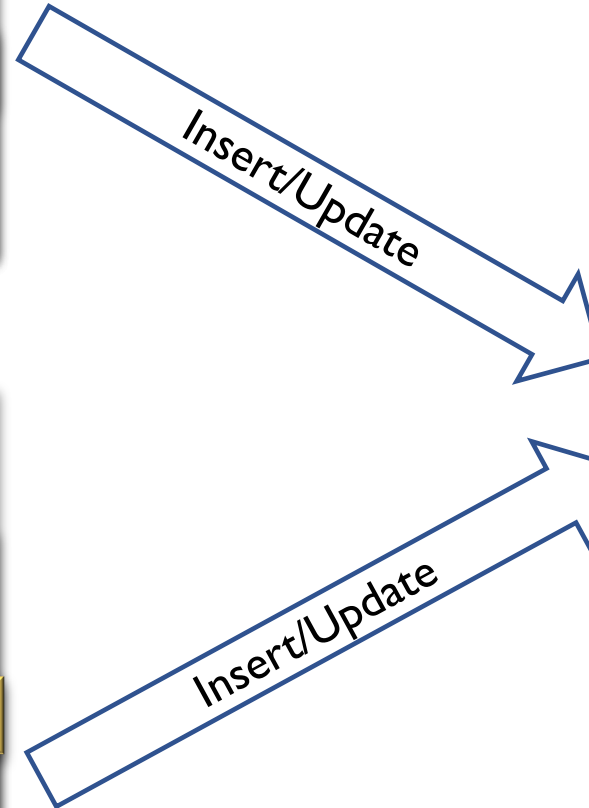
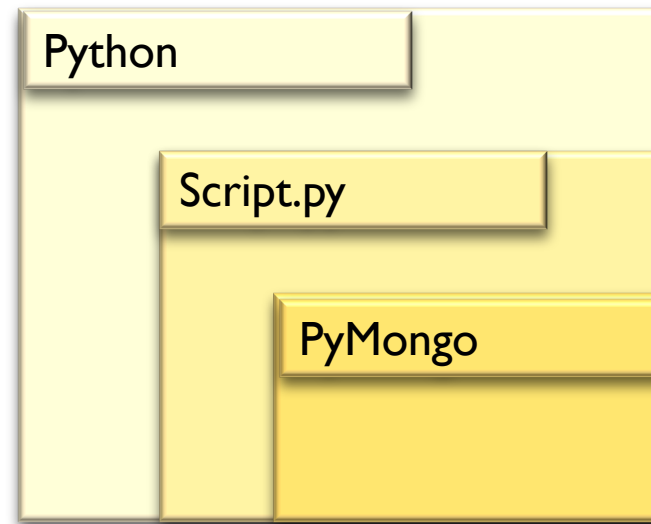
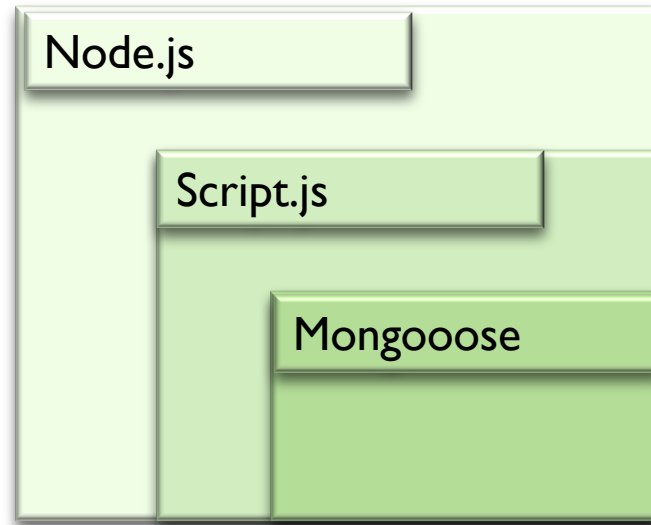
การลบ Collection

```
#ลบ collection
collection_name = db["area"]
collection_name.drop()
print('ลบตาราง/collection area เรียบร้อยแล้ว')
```



การสร้าง MODELS OF MONGOOSE

Mongoose



Mongoose (Schema)

- ในตัว schema ของ mongoose สามารถกำหนด method เอาไว้ใช้งาน ในตอนที่สร้างเป็น model ออกมาได้
- เริ่มต้น ควรทำการสร้าง schema (โครงสร้างข้อมูล) ก่อน

```
import mongoose from 'mongoose';
const { Schema } = mongoose;

const blogSchema = new Schema({
  title: String, // String is shorthand for {type: String}
  author: String,
  body: String,
  comments: [{ body: String, date: Date }],
  date: { type: Date, default: Date.now },
  hidden: Boolean,
  meta: {
    votes: Number,
    favs: Number
  }
});
```

Mongoose (สร้าง model)

- ทำการ convert 'blogSchema' ไปเป็น model (ต้นแบบ)

```
const Blog = mongoose.model('Blog', blogSchema);
```

- Mongoose จะทำการเพิ่ม `_id` ให้อัตโนมัติ

```
const doc = new Blog();  
doc._id instanceof mongoose.Types.ObjectId; // true
```

- Save model

```
Blog.save((err) => {  
  if (err) return handleError(err);  
})
```

Mongoose (สร้าง method)

- ตัวอย่างการสร้าง method ในการดึงข้อมูล “getById” “getAllBook”

```
const getById = async id => {  
  return await Blog.findById(id).lean();  
};  
  
const getAllBook = async () => {  
  return await Blog.find()  
    .populate("title", "-author")  
    //มีการอ้างอิงถึงข้อมูลโดยใช้ คำสั่ง populate  
    //โดย parameter ค่าแรกเป็นฟิลด์ใน model ที่เราต้องการอ้างอิงถึง  
    //และค่าที่สองก็คือค่าที่ต้องการ select ถ้าไม่ต้องการค่าไหนใส่ -  
    .lean();  
};
```

Mongoose (สร้าง method)

- การสร้าง user method

```
blogSchema.methods.showComment = function sComment() {  
  const msg = this.comments  
  console.log(msg);  
};  
  
const blog1 = new Blog({ name: 'blog1' });  
blog1.showComment();
```

Reference



- https://developer.mozilla.org/en-US/docs/Learn/Server-side/Express_Nodejs/mongoose#overview
- <https://nextflow.in.th/2021/mongoose-schema-methods-typescript-thai/>

THANK YOU FOR YOUR ATTENTION

