IP位置: 可以追蹤你的裝置在哪裡，跟家裡的地址、門牌一樣的概念

IP Address 之前由4個bytes組成(IPv4)> 有42億9000萬個可能性組成

但馬上就被用完了

IP Address 現在由16個bytes組成(IPv6)> 有天文數字個可能性組成

DNS(Domian Name System): IP位置與網址的轉換

Ipv4 : 168.1.0.1 (四個數字介於0~255之間)

DNS : 把上面不好記住的數字，轉換為網址

https://www.youtube.com/

Port: 接口/通訊阜

User打的網址透過DNS轉換成IP位置後，不是只有168.1.0.1 四個數字

後面還會加掛一個Port (Ex: 168.1.0.1:80) 這個80就是根據user的需求，給予你要的東西

Ex: port number

80 HTTP

20 FTP

53 DNS

443 HTTPS

* **MongoDB**

MongoDB儲存資料的方式為BSON

BSON特色是他可接受的資料型態較多、而且只有電腦看得懂

要跟mongoDB 互動的話要先進到他的shell (類似服務台概念)

CMD > cd到C:\Program Files\MongoDB\Server\4.2\bin 這個資料夾，輸入mongo就可以進入shell

基本 shell指令

show dbs : 列出目前現有的資料庫

db : 顯示目前所在資料庫

use + {dbname} : 移動到不同的資料庫 ，若原本沒有這個資料庫存在時, mongoDB會自動新增

show collections: 列出目前db中的資料表

cls: 清空畫面上程式碼

ctrl+c : 跳出shell回到CMD

mongo CRUD指令 :

新增 Insertion

https://www.mongodb.com/docs/manual/reference/insert-methods/

修改 Update

https://www.mongodb.com/docs/manual/tutorial/update-documents/

刪除 Delete

https://www.mongodb.com/docs/manual/tutorial/remove-documents/

查詢 Find

https://www.mongodb.com/docs/manual/tutorial/query-documents/

其他詳細CRUD指令可以看這裡:

https://www.mongodb.com/docs/manual/crud/

**\* mongo CRUD指令 :**

**>新增 Insertion :**

>db.{collectionnames}.insertOne({object});

//在collection內新增一個object【原本沒有的話會自動新增】

Ex:

> db.dog.insertOne({name:"mix",age:"5"};

{

"acknowledged" : true, //這邊顯示acknowledged true代表這個collection已被接受

"insertedId" : ObjectId("6346f4a73e7d310d4e8c031f")

}

>db.dog.insertMany([{object1},{object2}])

//在collection內新增複數個object

Ex:

> db.dog.insertMany([{name:"doggy",age:"3"},{name:"happy",age:"6"}]);

{

"acknowledged" : true,

"insertedIds" : [

ObjectId("6346f8253e7d310d4e8c0320"),

ObjectId("6346f8253e7d310d4e8c0321")

]

}

**>查詢 Find:**

> db.{collectionnames}.find()

//列出collection內容

Ex:

> db.dog.find()

{ "\_id" : ObjectId("6346f4a73e7d310d4e8c031f"), "name" : "mix", "age" : "5" }

//這邊回傳的\_id是剛剛insertOne()時候mongoDB自己生成的

**\* mongo CRUD指令 :**

>修改 **Update:**

**原本collection內物件長這樣:**

**{ "\_id" : ObjectId("6346f8253e7d310d4e8c0320"), "name" : "mix", "age" : "3" }**

**現在我要改他的名字從mix改為max**

//在collection內更新一個object

>db. {collectionnames}.updateOne(

{原本的property},

{ $set: {要改的property}, $currentDate:{lastModified:true} }

//$set和$currentDate後面的冒號不能忘記

)

紅色部分是{filter}，讓mongoDB了解你要改哪個object(只會找到第一個符合條件的object)

藍色部分是要修改的東西，你可以一口氣改object中複數個property

綠色部分是讓mongoDB順便紀錄修改時間

**Ex:**

db.dog.updateOne({name:"mix"},{$set:{name:"max",age:2},$currentDate:{lastModified:true}})

{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }

**原本collection內物件長這樣:**

> db.dog.find()

**{ "\_id" : ObjectId("6346f4a73e7d310d4e8c031f"), "name" : "max", "age" : 2, "lastModified" : ISODate("2022-10-13T05:33:21.655Z"), "mood" : "happy" }**

**{ "\_id" : ObjectId("6346f8253e7d310d4e8c0320"), "name" : "doggy", "age" : "3", "lastModified" : ISODate("2022-10-13T05:34:35.794Z"), "mood" : "happy" }**

**{ "\_id" : ObjectId("6346f8253e7d310d4e8c0321"), "name" : "happy", "age" : "6", "lastModified" : ISODate("2022-10-13T05:34:43.077Z"), "mood" : "happy" }**

**現在我要改所有物件的age改為10**

//在collection內更新複數個object

>db. {collectionnames}.updateMany(

{原本的property},

{ $set: {要改的property}, $currentDate:{lastModified:true} }

)

Ex: db.dog.updateMany({"mood":"happy"},{$set:{"age":10},$currentDate:{lastModified:true}})

{ "acknowledged" : true, "matchedCount" : 3, "modifiedCount" : 3 }

**\*mongo CRUD指令 :**

>刪除 **Delete:**

//在collection內刪除一個object

>db. {collectionnames}.deleteOne({原本的property })

紅色部分是{filter}，讓mongoDB了解你要刪哪個object(只會刪除第一個符合條件的object)

//在collection內刪除複數個object

>db. {collectionnames}.deleteMany{{原本的property }}

紅色部分是{filter}，讓mongoDB了解你要刪哪個object(刪除所有符合條件的object)