心理與神經資訊學
(Psychoinformatics & Neuroinformatics)

課號: Psy5261

識別碼: 227U9340

教室:彷彿在雲端

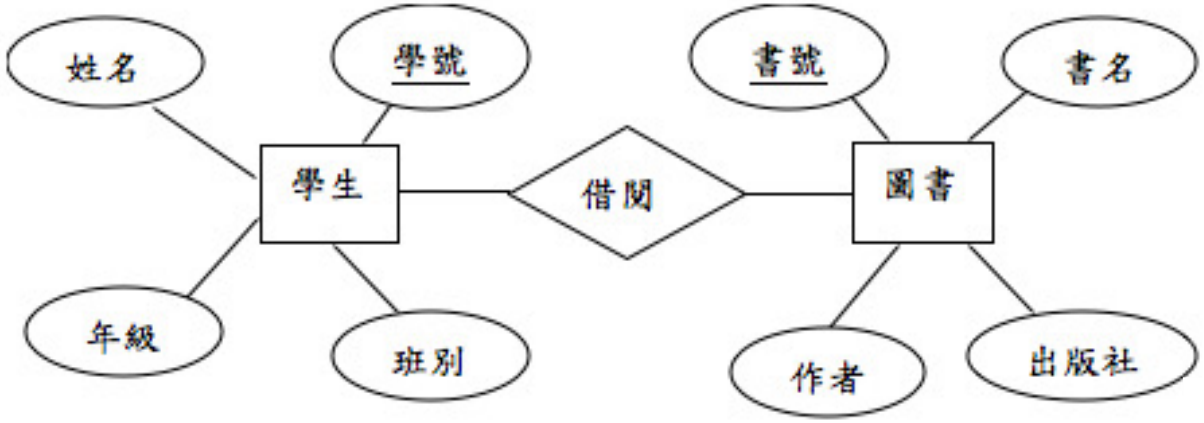
時間: 一789

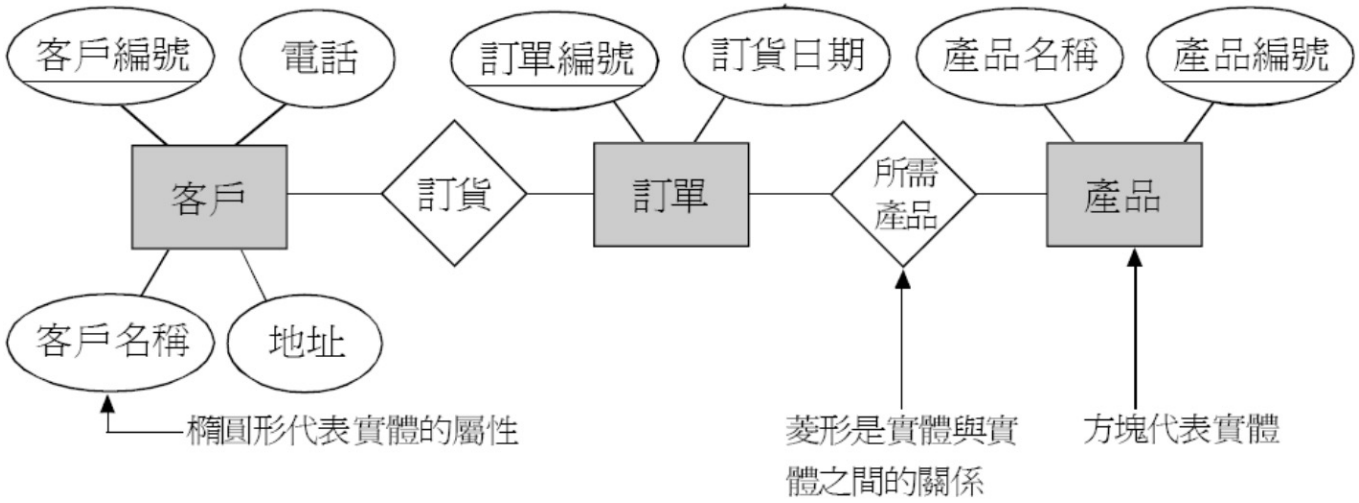

XAMPP裡有MySQL喔
!!

SQL & NoSQL資料庫

關聯式資料庫 (Relational DB)

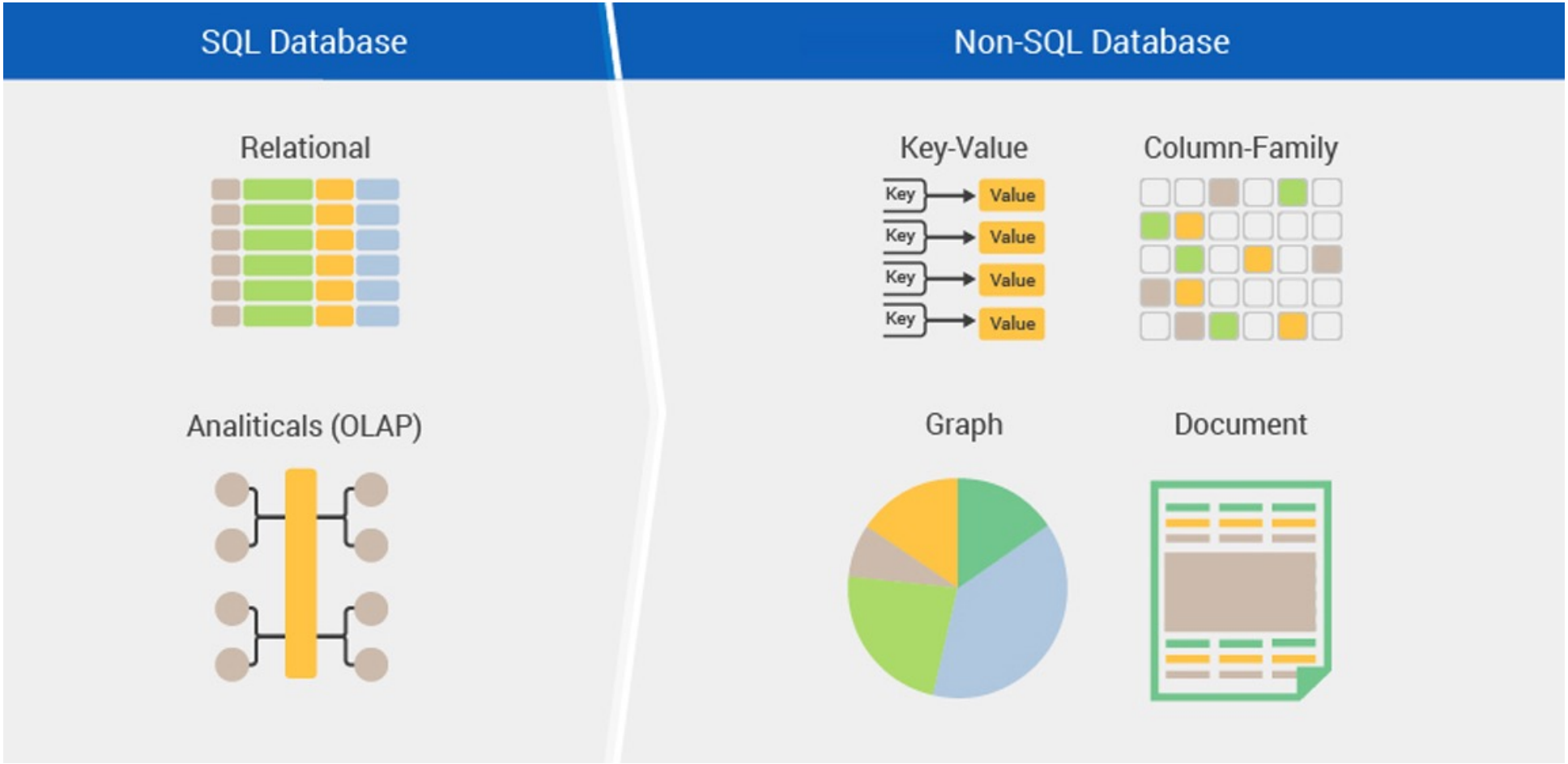
設計/規劃時會畫Entity-Relationship Diagram





SQL vs. NoSQL

前者比較直觀好用但處理大/非結構化數據常需後者



但Facebook & Twitter主要還是用MySQL

SQL語法(1/3)

SQL CHEAT SHEET http://www.sqltutorial.org

QUERYING DATA FROM A TABLE

```
SELECT c1, c2 FROM t;
```
Query data in columns c1, c2 from a table

```
SELECT * FROM t;
```
Query all rows and columns from a table

```
SELECT c1, c2 FROM t
WHERE condition;
```
Query data and filter rows with a condition

```
SELECT DISTINCT c1 FROM t
WHERE condition;
```
Query distinct rows from a table

```
SELECT c1, c2 FROM t
ORDER BY c1 ASC [DESC];
```
Sort the result set in ascending or descending order

```
SELECT c1, c2 FROM t
ORDER BY c1
LIMIT n OFFSET offset;
```
Skip *offset* of rows and return the next n rows

```
SELECT c1, aggregate(c2)
FROM t
GROUP BY c1;
```
Group rows using an aggregate function

```
SELECT c1, aggregate(c2)
FROM t
GROUP BY c1
HAVING condition;
```
Filter groups using HAVING clause

QUERYING FROM MULTIPLE TABLES

```
SELECT c1, c2
FROM t1
INNER JOIN t2 ON condition;
```
Inner join t1 and t2

```
SELECT c1, c2
FROM t1
LEFT JOIN t2 ON condition;
```
Left join t1 and t1

```
SELECT c1, c2
FROM t1
RIGHT JOIN t2 ON condition;
```
Right join t1 and t2

```
SELECT c1, c2
FROM t1
FULL OUTER JOIN t2 ON condition;
```
Perform full outer join

```
SELECT c1, c2
FROM t1
CROSS JOIN t2;
```
Produce a Cartesian product of rows in tables

```
SELECT c1, c2
FROM t1, t2;
```
Another way to perform cross join

```
SELECT c1, c2
FROM t1 A
INNER JOIN t2 B ON condition;
```
Join t1 to itself using INNER JOIN clause

USING SQL OPERATORS

```
SELECT c1, c2 FROM t1
UNION [ALL]
SELECT c1, c2 FROM t2;
```
Combine rows from two queries

```
SELECT c1, c2 FROM t1
INTERSECT
SELECT c1, c2 FROM t2;
```
Return the intersection of two queries

```
SELECT c1, c2 FROM t1
MINUS
SELECT c1, c2 FROM t2;
```
Subtract a result set from another result set

```
SELECT c1, c2 FROM t1
WHERE c1 [NOT] LIKE pattern;
```
Query rows using pattern matching %, _

```
SELECT c1, c2 FROM t
WHERE c1 [NOT] IN value_list;
```
Query rows in a list

```
SELECT c1, c2 FROM t
WHERE c1 BETWEEN low AND high;
```
Query rows between two values

```
SELECT c1, c2 FROM t
WHERE c1 IS [NOT] NULL;
```
Check if values in a table is NULL or not

SQL語法(2/3)

SQL CHEAT SHEET http://www.sqltutorial.org

MANAGING TABLES

```
CREATE TABLE t (
    id INT PRIMARY KEY,
    name VARCHAR NOT NULL,
    price INT DEFAULT 0
);
```
Create a new table with three columns

```
DROP TABLE t;
```
Delete the table from the database

```
ALTER TABLE t ADD column;
```
Add a new column to the table

```
ALTER TABLE t DROP COLUMN c;
```
Drop column c from the table

```
ALTER TABLE t ADD constraint;
```
Add a constraint

```
ALTER TABLE t DROP constraint;
```
Drop a constraint

```
ALTER TABLE t1 RENAME TO t2;
```
Rename a table from t1 to t2

```
ALTER TABLE t1 RENAME c1 TO c2;
```
Rename column c1 to c2

```
TRUNCATE TABLE t;
```
Remove all data in a table

USING SQL CONSTRAINTS

```
CREATE TABLE t(
   c1 INT, c2 INT, c3 VARCHAR,
   PRIMARY KEY (c1,c2)
);
```
Set c1 and c2 as a primary key

```
CREATE TABLE t1(
   c1 INT PRIMARY KEY,
   c2 INT,
   FOREIGN KEY (c2) REFERENCES t2(c2)
);
```
Set c2 column as a foreign key

```
CREATE TABLE t(
   c1 INT, c1 INT,
   UNIQUE(c2,c3)
);
```
Make the values in c1 and c2 unique

```
CREATE TABLE t(
  c1 INT, c2 INT,
  CHECK(c1> 0 AND c1 >= c2)
);
```
Ensure c1 > 0 and values in c1 >= c2

```
CREATE TABLE t(
    c1 INT PRIMARY KEY,
    c2 VARCHAR NOT NULL
);
```
Set values in c2 column not NULL

MODIFYING DATA

```
INSERT INTO t(column_list)
VALUES(value_list);
```
Insert one row into a table

```
INSERT INTO t(column_list)
VALUES (value_list),
       (value_list), ....;
```
Insert multiple rows into a table

```
INSERT INTO t1(column_list)
SELECT column_list
FROM t2;
```
Insert rows from t2 into t1

```
UPDATE t
SET c1 = new_value;
```
Update new value in the column c1 for all rows

```
UPDATE t
SET c1 = new_value,
    c2 = new_value
WHERE condition;
```
Update values in the column c1, c2 that match the condition

```
DELETE FROM t;
```
Delete all data in a table

```
DELETE FROM t
WHERE condition;
```
Delete subset of rows in a table

SQL語法(3/3)

SQL CHEAT SHEET http://www.sqltutorial.org

MANAGING VIEWS

```
CREATE VIEW v(c1,c2)
AS
SELECT c1, c2
FROM t;
```
Create a new view that consists of c1 and c2

```
CREATE VIEW v(c1,c2)
AS
SELECT c1, c2
FROM t;
WITH [CASCADED | LOCAL] CHECK OPTION;
```
Create a new view with check option

```
CREATE RECURSIVE VIEW v
AS
select-statement  -- anchor part
UNION [ALL]
select-statement;  -- recursive part
```
Create a recursive view

```
CREATE TEMPORARY VIEW v
AS
SELECT c1, c2
FROM t;
```
Create a temporary view

```
DROP VIEW view_name;
```
Delete a view

MANAGING INDEXES

```
CREATE INDEX idx_name
ON t(c1,c2);
```
Create an index on c1 and c2 of the table t

```
CREATE UNIQUE INDEX idx_name
ON t(c3,c4);
```
Create a unique index on c3, c4 of the table t

```
DROP INDEX idx_name;
```
Drop an index

SQL AGGREGATE FUNCTIONS

AVG returns the average of a list

COUNT returns the number of elements of a list

SUM returns the total of a list

MAX returns the maximum value in a list

MIN returns the minimum value in a list

MANAGING TRIGGERS

```
CREATE OR MODIFY TRIGGER trigger_name
WHEN EVENT
ON table_name TRIGGER_TYPE
EXECUTE stored_procedure;
```
Create or modify a trigger

WHEN
- **BEFORE** – invoke before the event occurs
- **AFTER** – invoke after the event occurs

EVENT
- **INSERT** – invoke for INSERT
- **UPDATE** – invoke for UPDATE
- **DELETE** – invoke for DELETE

TRIGGER_TYPE
- **FOR EACH ROW**
- **FOR EACH STATEMENT**

```
CREATE TRIGGER before_insert_person
BEFORE INSERT
ON person FOR EACH ROW
EXECUTE stored_procedure;
```
Create a trigger invoked before a new row is inserted into the person table

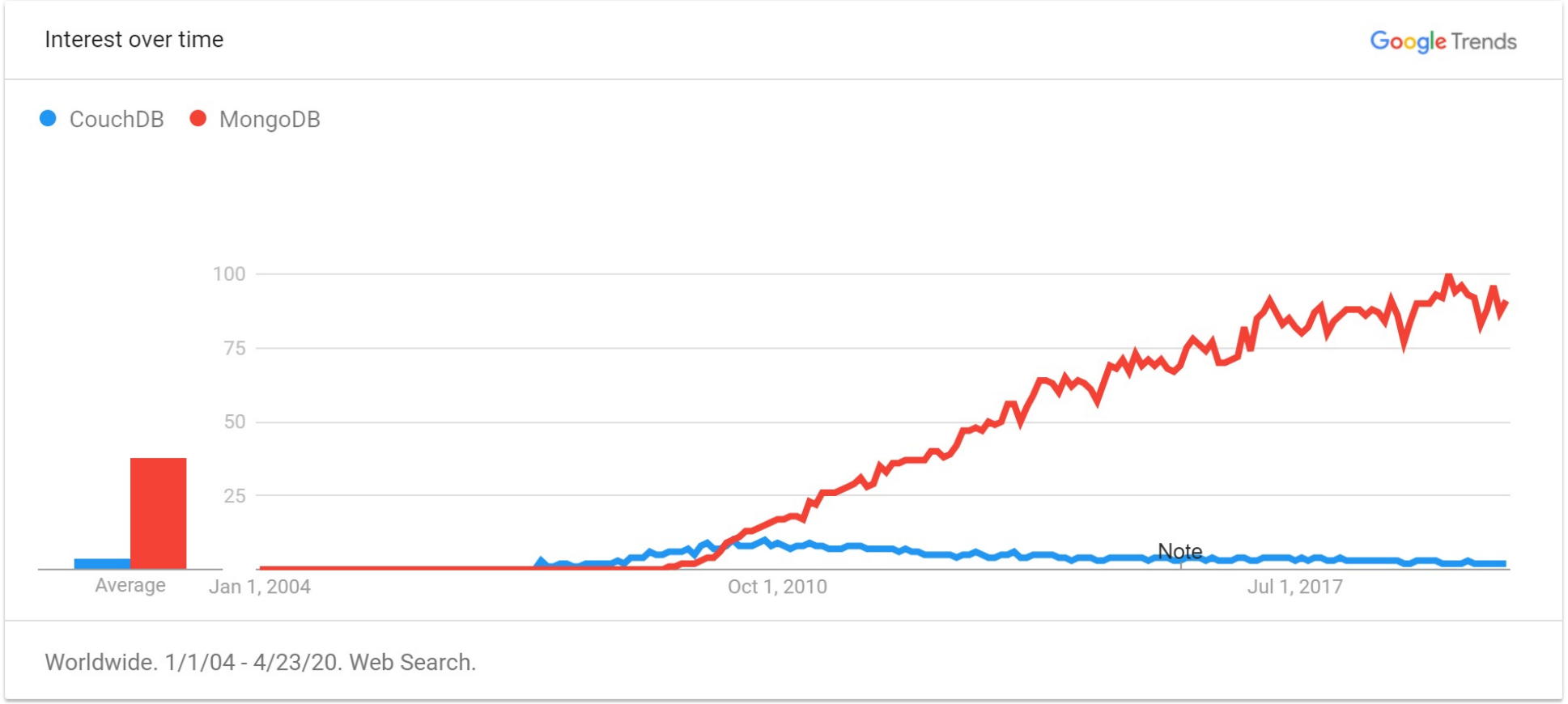
```
DROP TRIGGER trigger_name;
```
Delete a specific trigger

SQLite

整個資料庫是一個容易備份的檔案

Document-based NoSQL

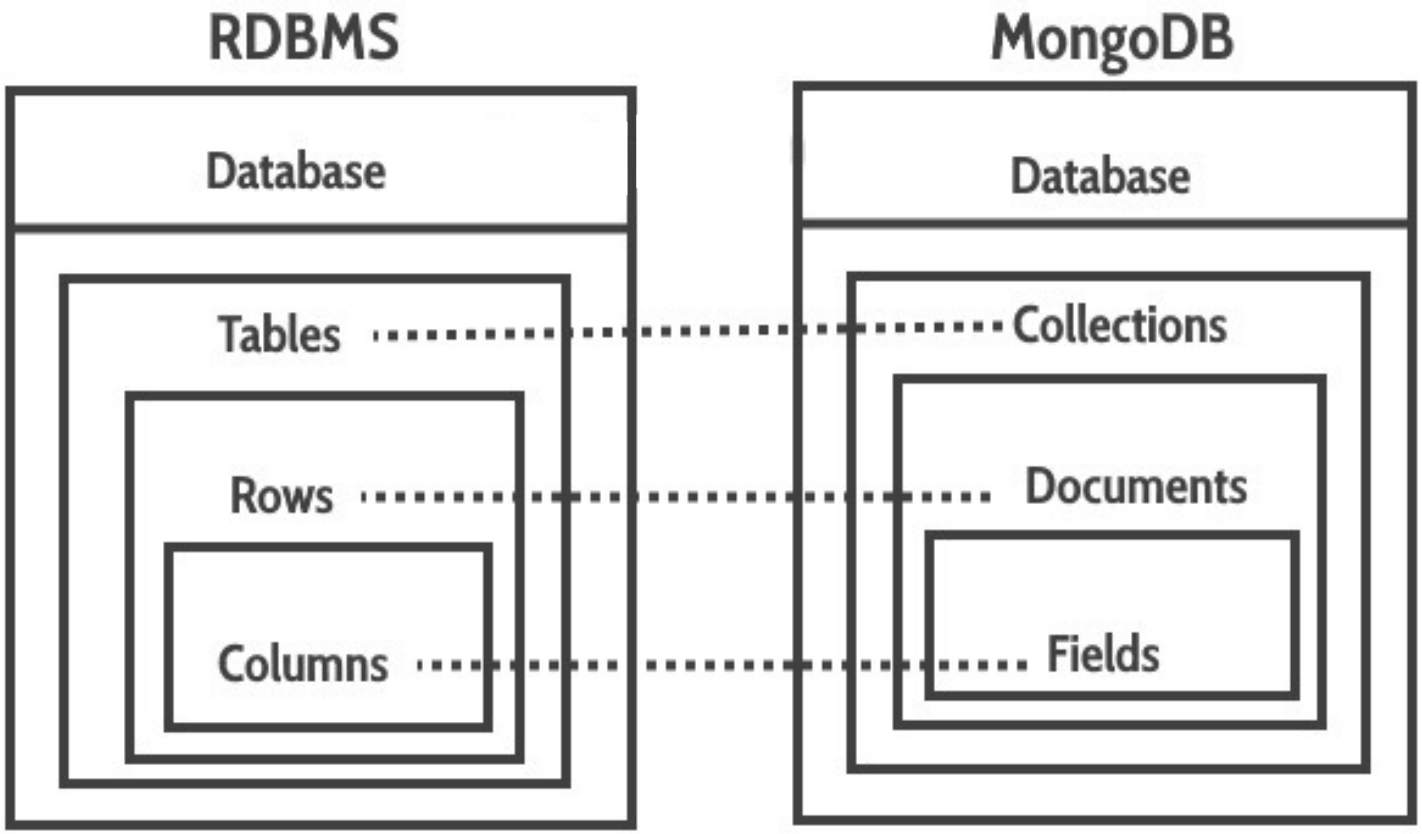
比較著名的有CouchDB & MongoDB



CouchDB 的資料操作主要用(難用的) map-reduce

MongoDB (1/3)

一些概念和SQL大致上有一對一的對應



MongoDB (2/3)

Document裡面是一群key-value pairs

Relational Database

Student_Id	Student_Name	Age	College
1001	Chaitanya	30	Beginnersbook
1002	Steve	29	Beginnersbook
1003	Negan	28	Beginnersbook

MongoDB

```
{
 "_id": ObjectId("....."),
 "Student_Id": 1001,
 "Student_Name": "Chaitanya",
 "Age": 30,
 "College": "Beginnersbook"
}
{
 "_id": ObjectId("....."),
 "Student_Id": 1002,
 "Student_Name": "Steve",
 "Age": 29,
 "College": "Beginnersbook"
}
{
 "_id": ObjectId("....."),
 "Student_Id": 1003,
 "Student_Name": "Negan",
 "Age": 28,
 "College": "Beginnersbook"
}
```

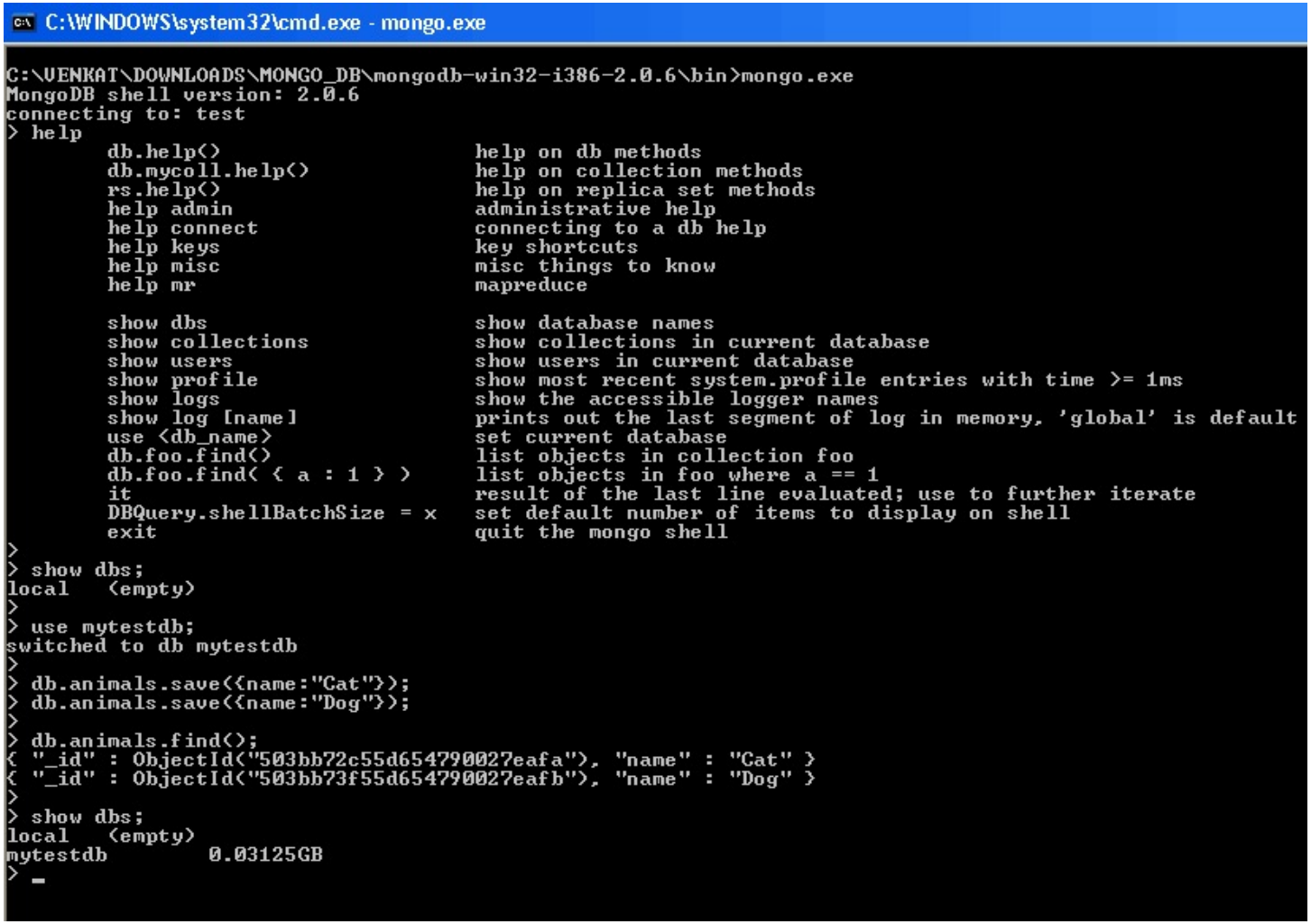
MongoDB (3/3)

有類SQL的語法和各種好用的資料處理語法

MySQL	MongoDB
INSERT	
INSERT INTO account (`A/c number`, `first name`, `last name`) VALUES ('12345746352', 'Mark', 'Jacobs');	db.account.insert({ A/c number: "12345746352", first name: "Mark", last name: "Jacobs" });
UPDATE	
UPDATE account SET contact number = 9426227364 WHERE A/c number = '12345746352'	db.account.update({ A/c number: '12345746352' }, { $set: {contact number: 9426227364 } });
DELETE	
DELETE FROM account WHERE e-mail address = 'jv1994@gmail.com';	db.account.remove({ "E-mail address": "jv1994@gmail.com" });

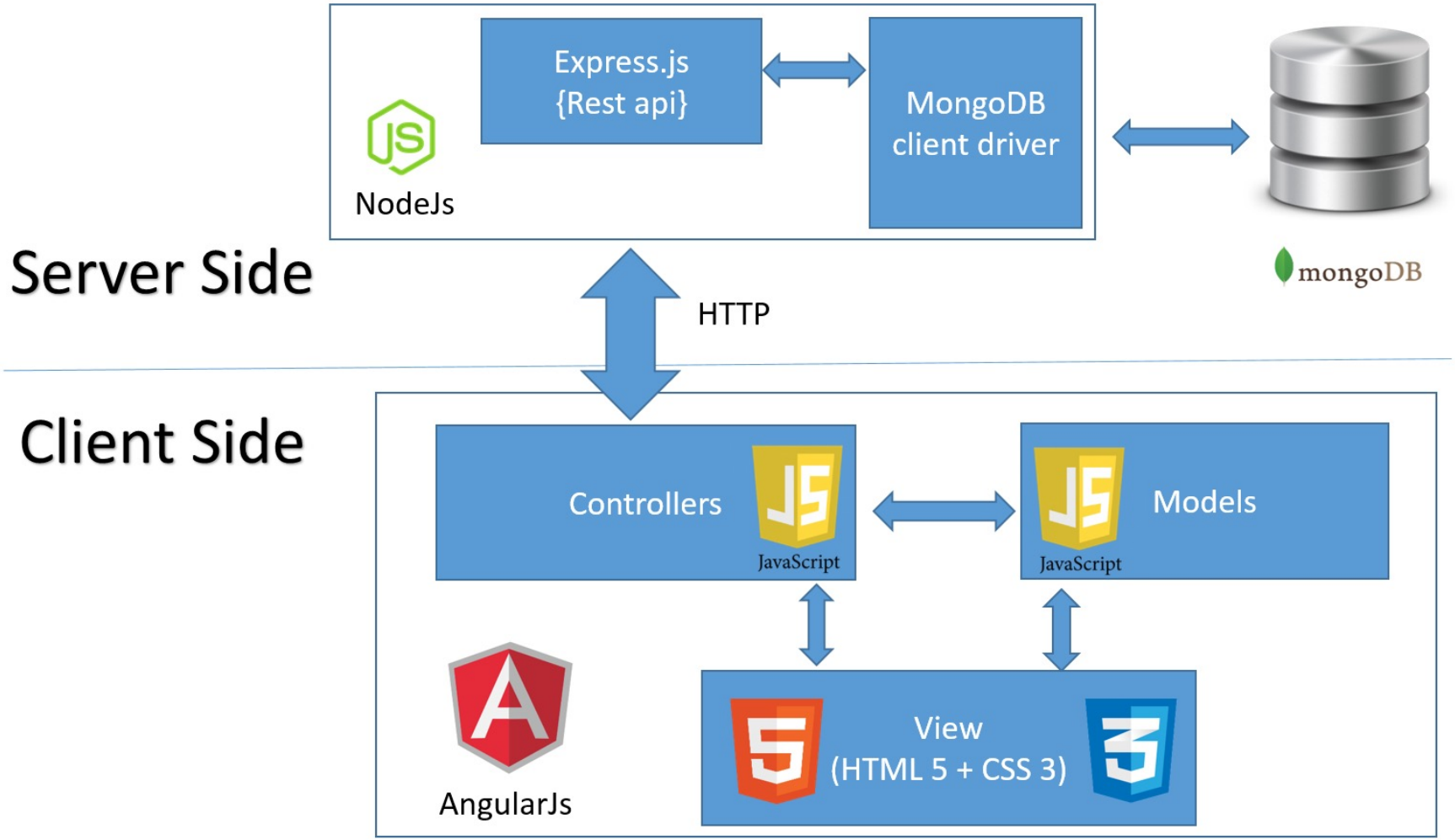
Mongo Shell

裡面可用部分Javascript語法



Node.js+Mongo (1/2)

兩個哥倆好



Node.js+Mongo (2/2)

就是寫JavaScript

Example

Insert a document in the "customers" collection:

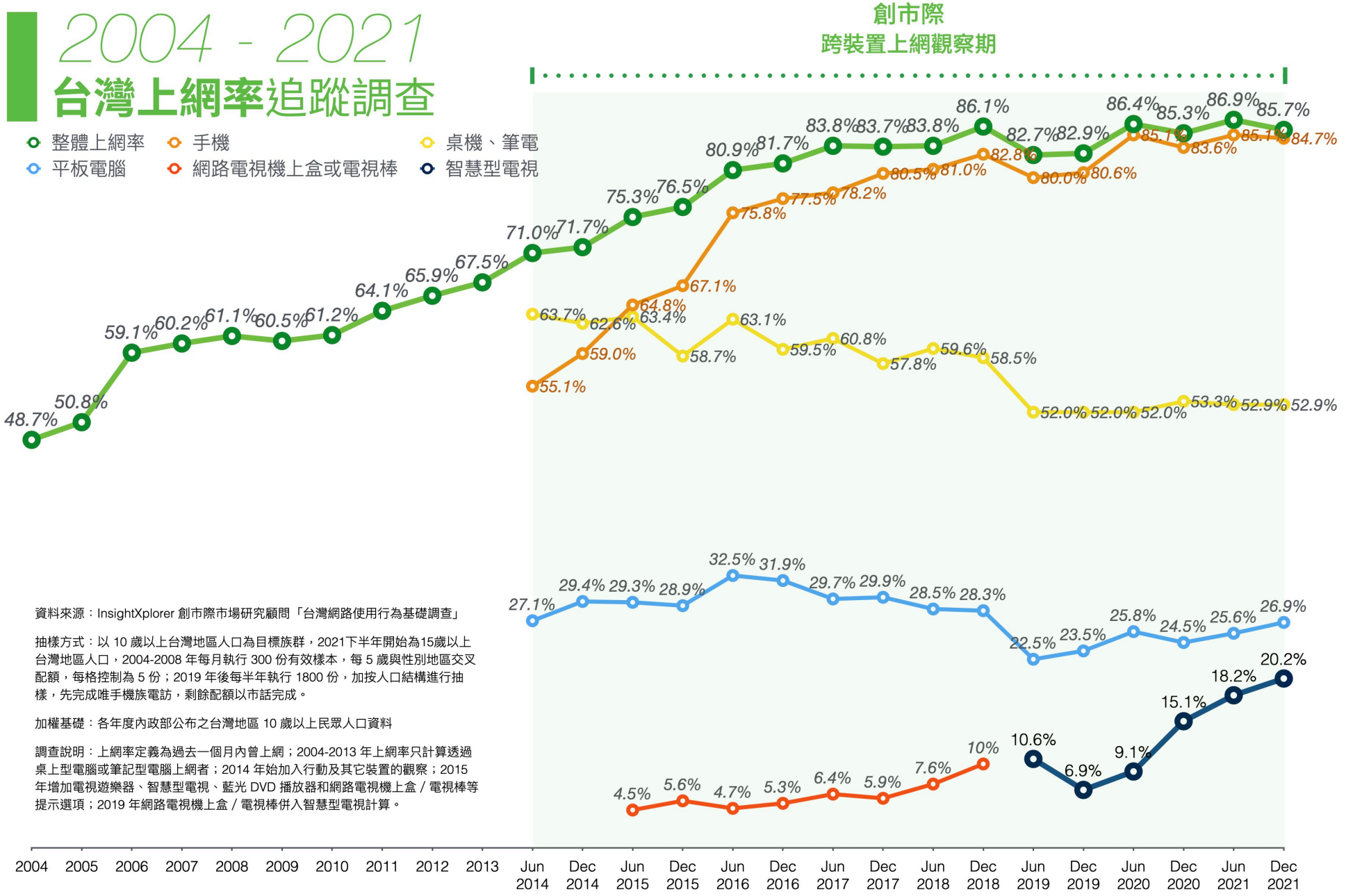
```
var MongoClient = require('mongodb').MongoClient;
var url = "mongodb://localhost:27017/";

MongoClient.connect(url, function(err, db) {
  if (err) throw err;
  var dbo = db.db("mydb");
  var myobj = { name: "Company Inc", address: "Highway 37" };
  dbo.collection("customers").insertOne(myobj, function(err, res) {
    if (err) throw err;
    console.log("1 document inserted");
    db.close();
  });
});
```

Run example »

手機研究與技術

台灣上網率 & 智慧型手機普及率



資料來源：InsightXplorer 創市際市場研究顧問「台灣網路使用行為基礎調查」

抽樣方式：以 10 歲以上台灣地區人口為目標族群，2021下半年開始為15歲以上台灣地區人口，2004-2008 年每月執行 300 份有效樣本，每 5 歲與性別地區交叉配額，每格控制為 5 份；2019 年後每半年執行 1800 份，加按人口結構進行抽樣，先完成唯手機族電訪，剩餘配額以市話完成。

加權基礎：各年度內政部公布之台灣地區 10 歲以上民眾人口資料

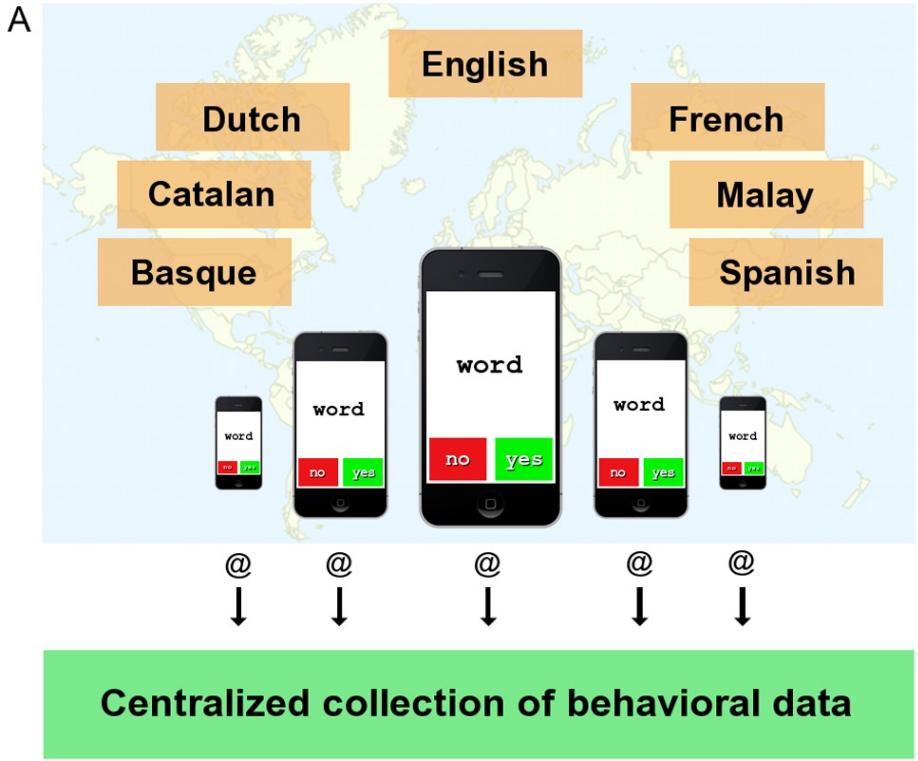
調查說明：上網率定義為過去一個月內曾上網；2004-2013 年上網率只計算透過桌上型電腦或筆記型電腦上網者；2014 年始加入行動及其它裝置的觀察；2015 年增加電視遊樂器、智慧型電視、藍光 DVD 播放器和網路電視機上盒 / 電視棒等提示選項；2019 年網路電視機上盒 / 電視棒併入智慧型電視計算。

Responsive Design

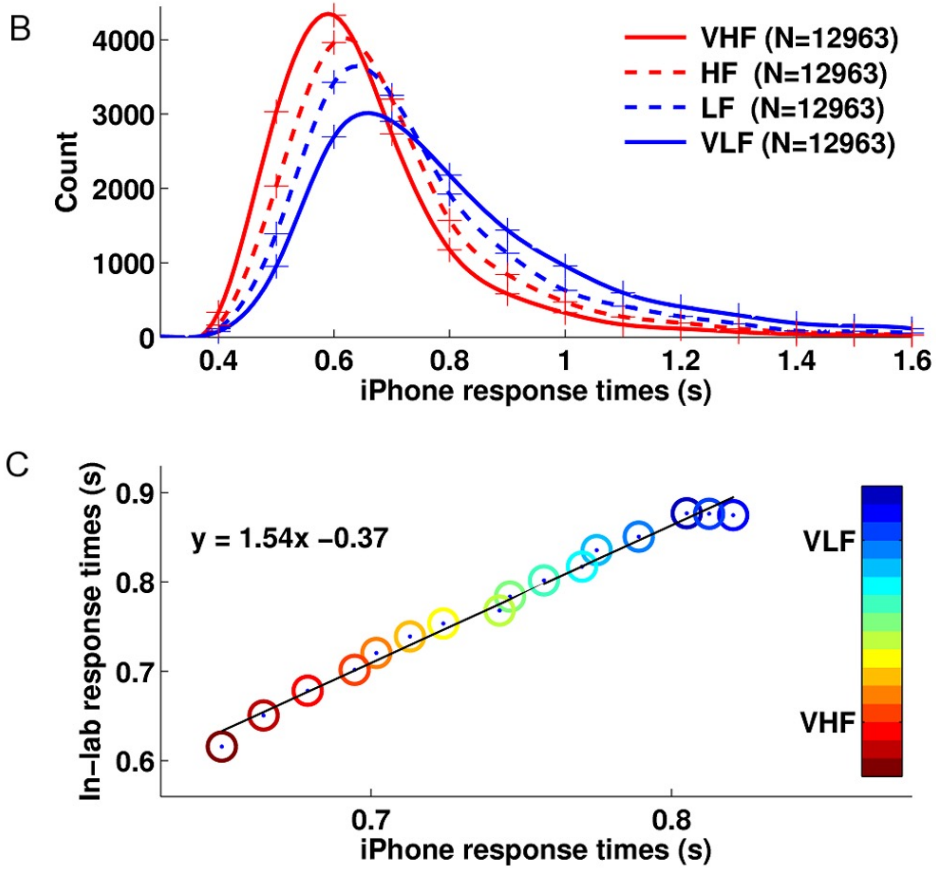
上網裝置太多，HTML & CSS要主動適應螢幕大小

手機研究的資料可靠性

手機研究可重製認知研究如Lexical Decision Task



Dufau et al., 2011



這是另一個例子

Experience Sampling Method

在70年代左右由Csikszentmihalyi與Larson所發展

“你現在在幹嘛?”
“你現在的情緒為何?”

ESM的信度和效度竟然都不錯

- ~80%的配合填答率
- 中到高程度的折半信度
- 經驗抽樣結果和人格特質相關

簡單的ESM實作

可利用現成的開發引擎甚至是Line/Telegram即可

movisensXS PRICING SIGN IN

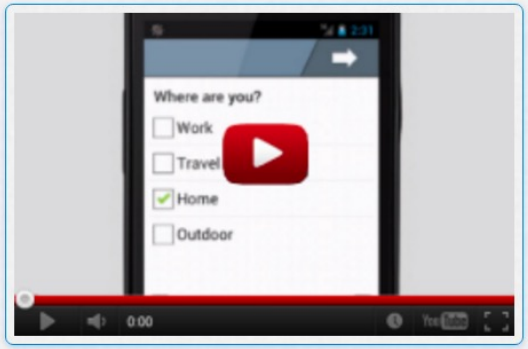
movisensXS

eXperience Sampling for Android!

movisensXS is the next generation research tool for ambulatory assessment. Ambulatory assessment refers to the use of computer-assisted methodology for self-reports, behavior records, or physiological measurements, while the participant undergoes normal daily activities. This approach includes the experience sampling method (ESM) a.k.a. ecological momentary assessment (EMA).

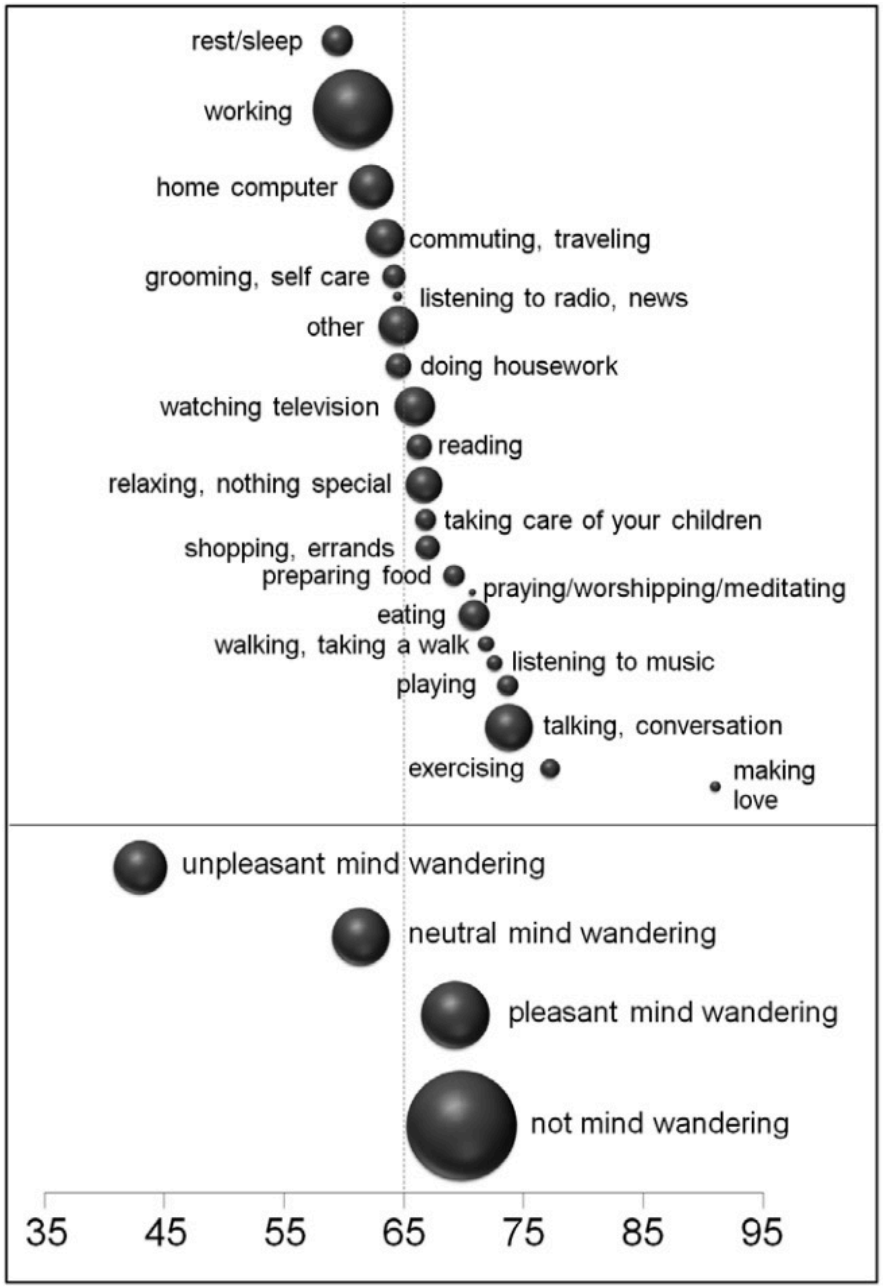
Launch your free study today!

Start now »



Demo: Launch your study in 2 minutes!

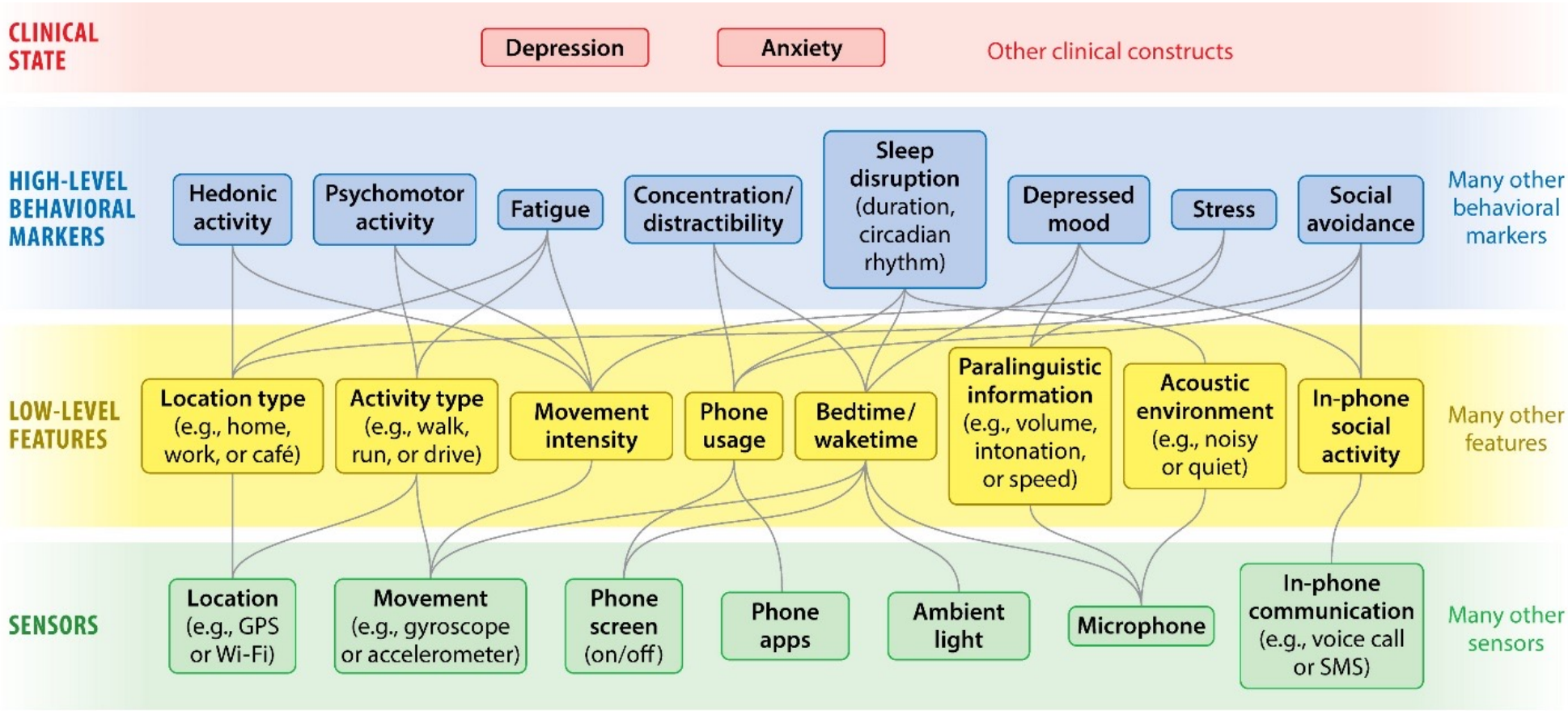
ESM範例:做什麼比較快樂?



Killingsworth & Dilbert, 2010, *Science*

Digital Phenotyping

不用去勞煩受試者回答問題



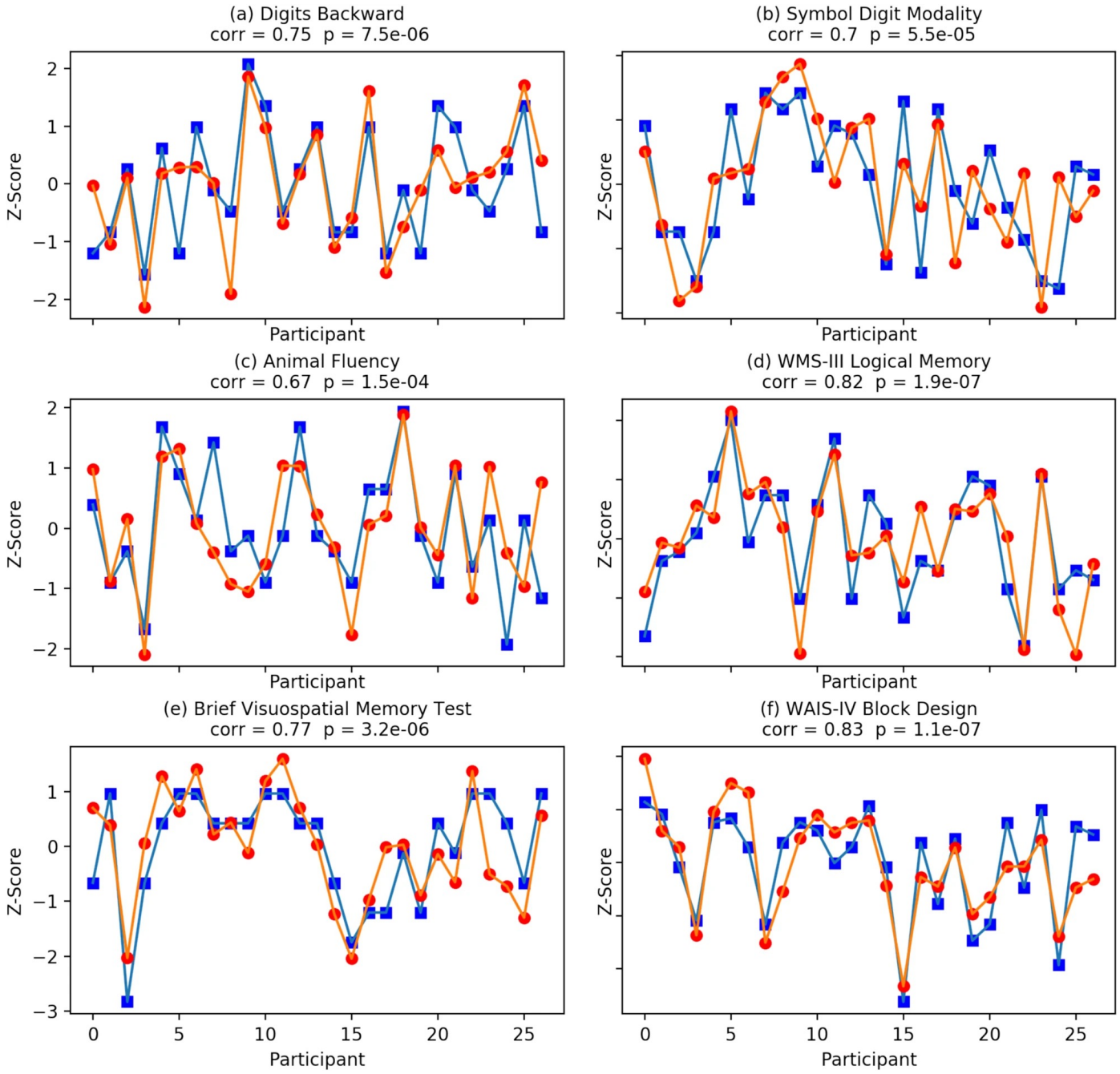
Mohr et al., 2017

DP範例1: 女生比男生多話？

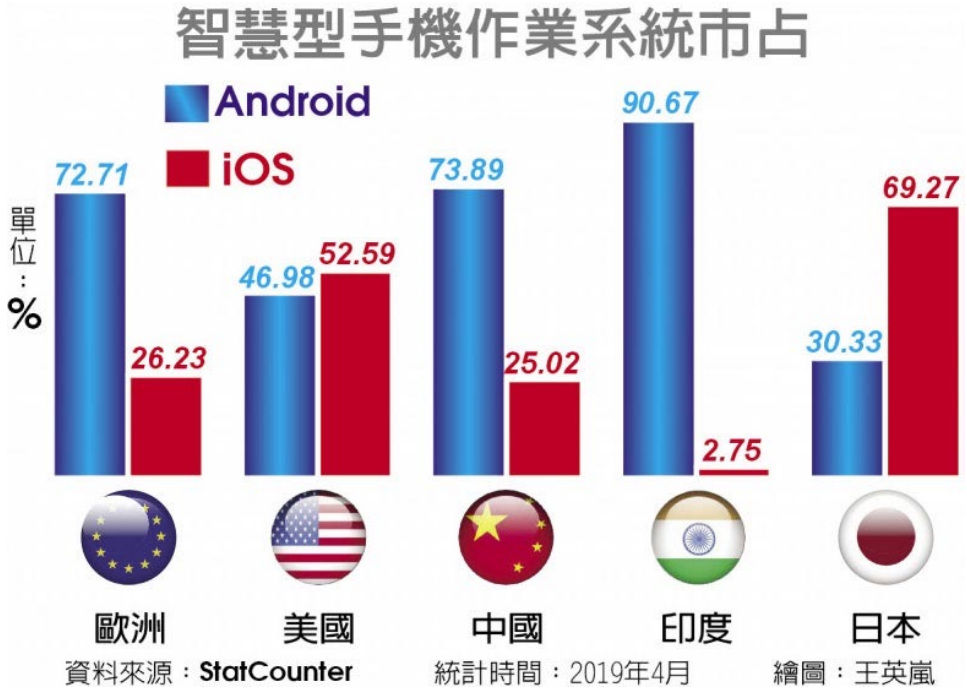
Sample	Year	Location	Duration	Age range (years)	Sample size (*N*)		Estimated average number (SD) of words spoken per day	
					Women	Men	Women	Men
1	2004	USA	7 days	18–29	56	56	18,443 (7460)	16,576 (7871)
2	2003	USA	4 days	17–23	42	37	14,297 (6441)	14,060 (9065)
3	2003	Mexico	4 days	17–25	31	20	14,704 (6215)	15,022 (7864)
4	2001	USA	2 days	17–22	47	49	16,177 (7520)	16,569 (9108)
5	2001	USA	10 days	18–26	7	4	15,761 (8985)	24,051 (10,211)
6	1998	USA	4 days	17–23	27	20	16,496 (7914)	12,867 (8343)
					Weighted average		16,215 (7301)	15,669 (8633)

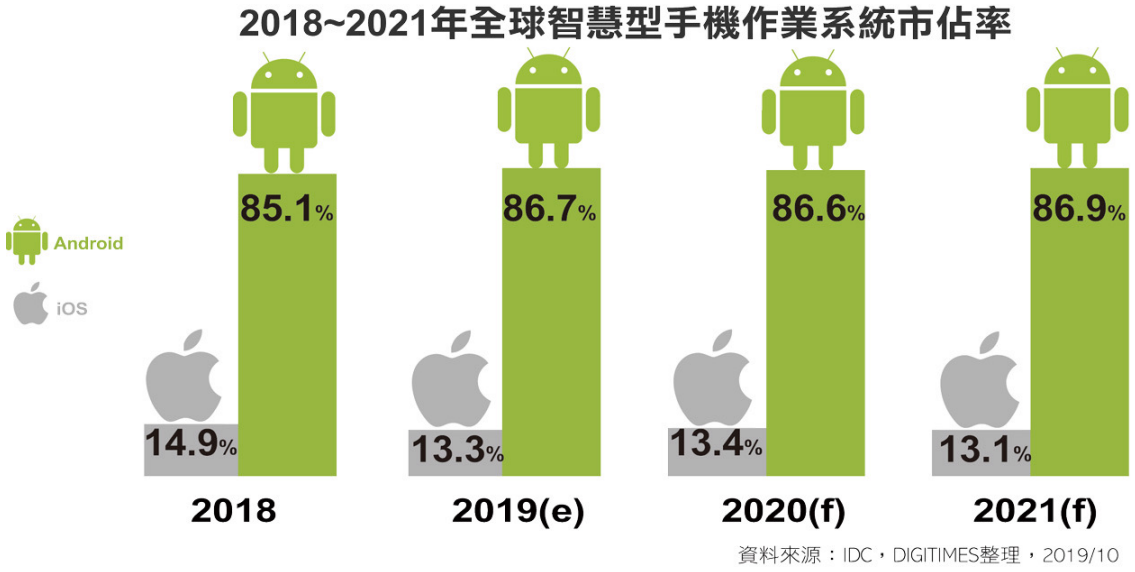
Mehl et al., 2007, *Science*

DP範例2: 評估一個人的認知能力

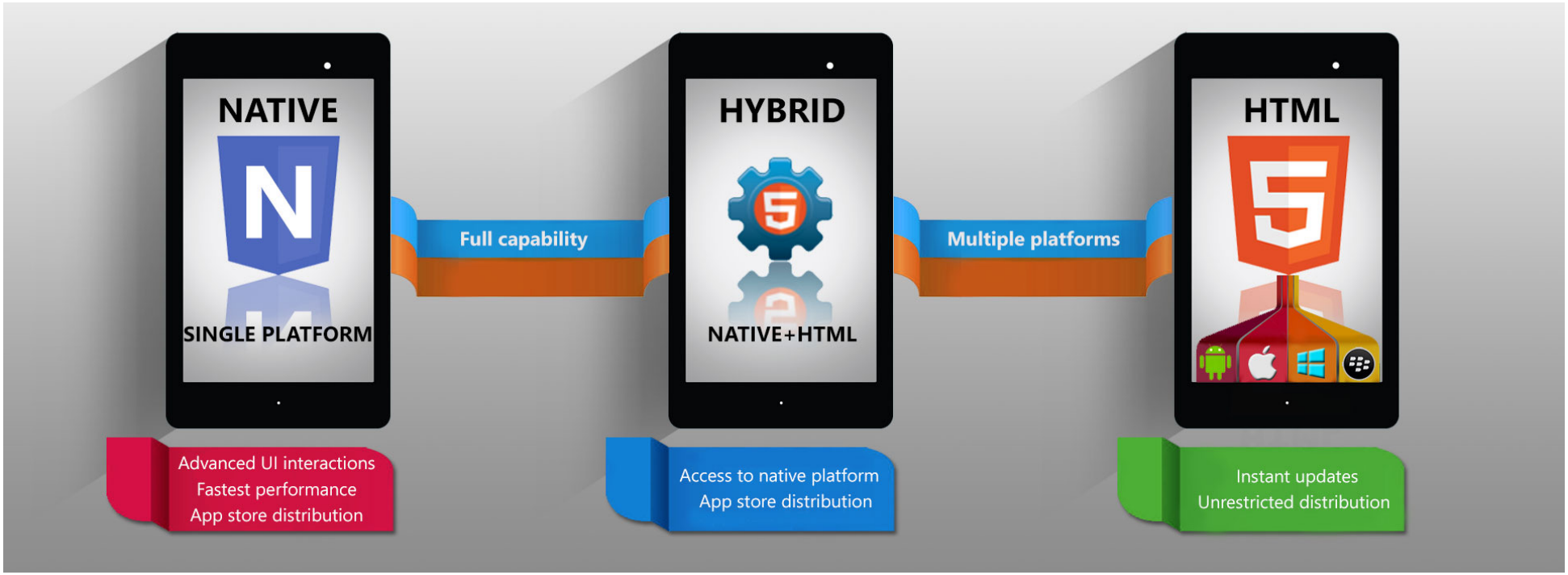


Android vs. iOS





手機程式的分類

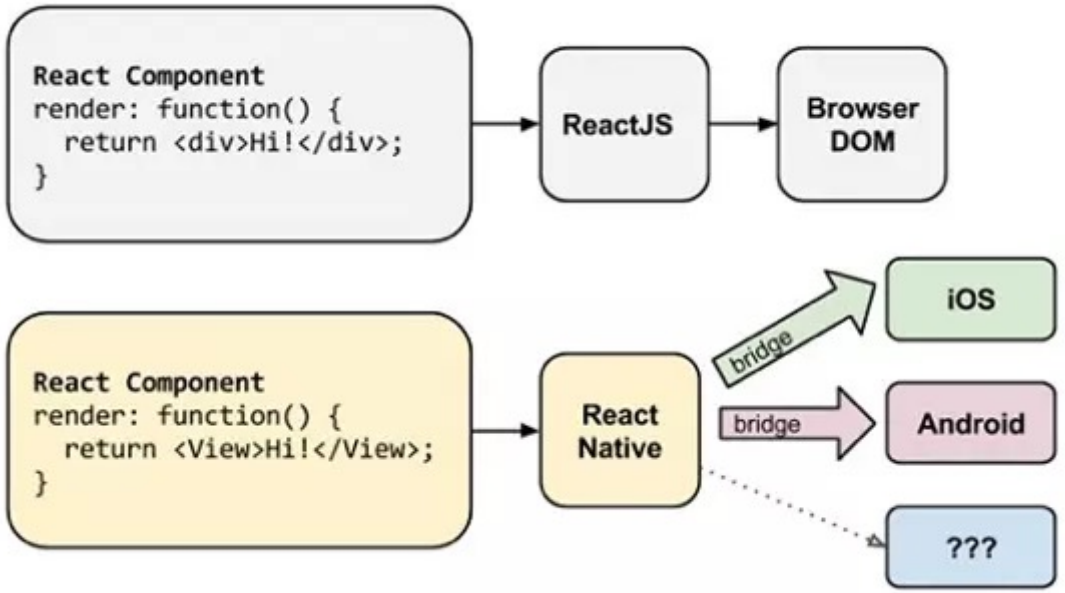


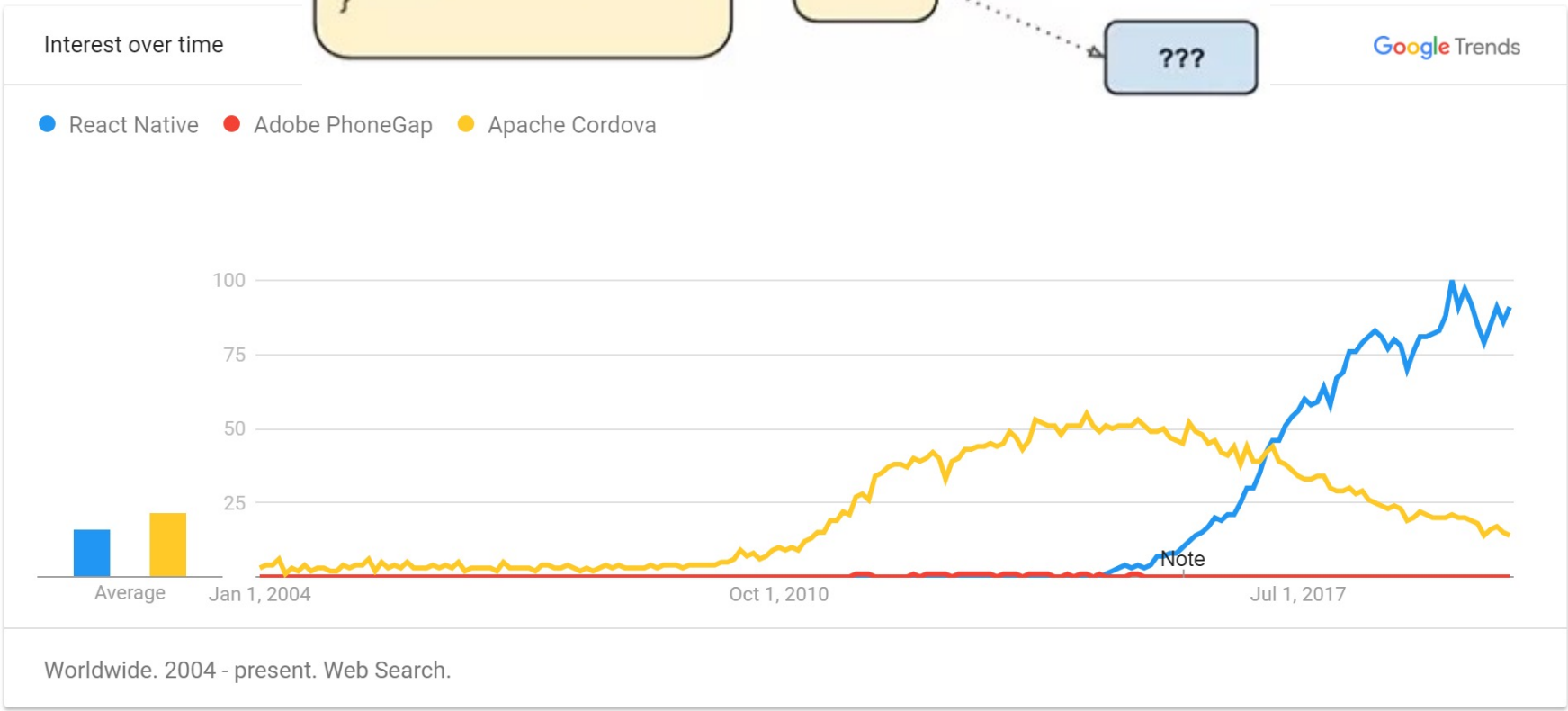
Native App: Android SDK (JAVA) + Xcode (Objective C)

Web App: HTML5 + CSS3 + Javascript/jQuery

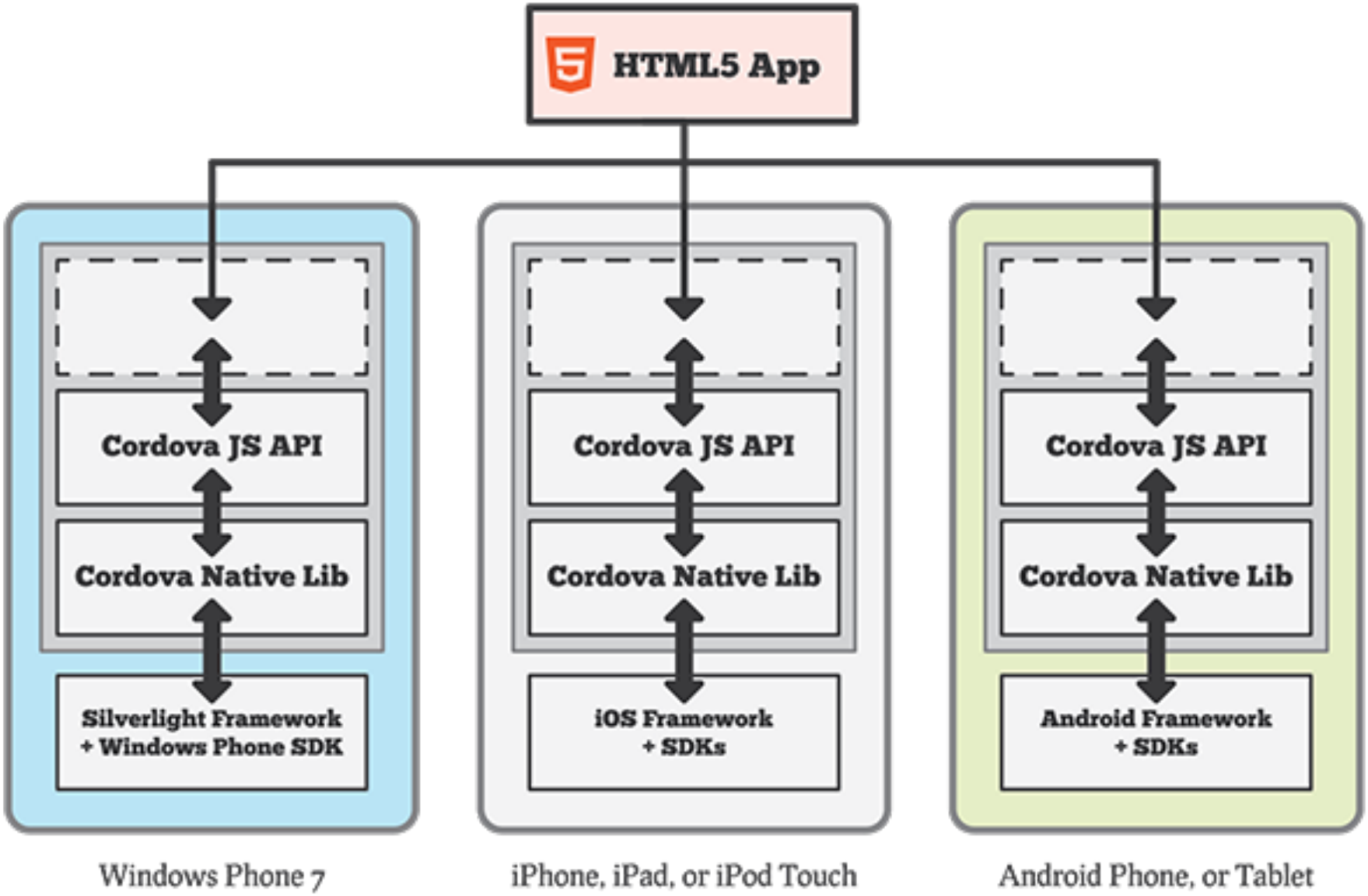
Hybrid App: Web App + PhoneGap/Cordova (Javascript)

Native App新選擇: React Native





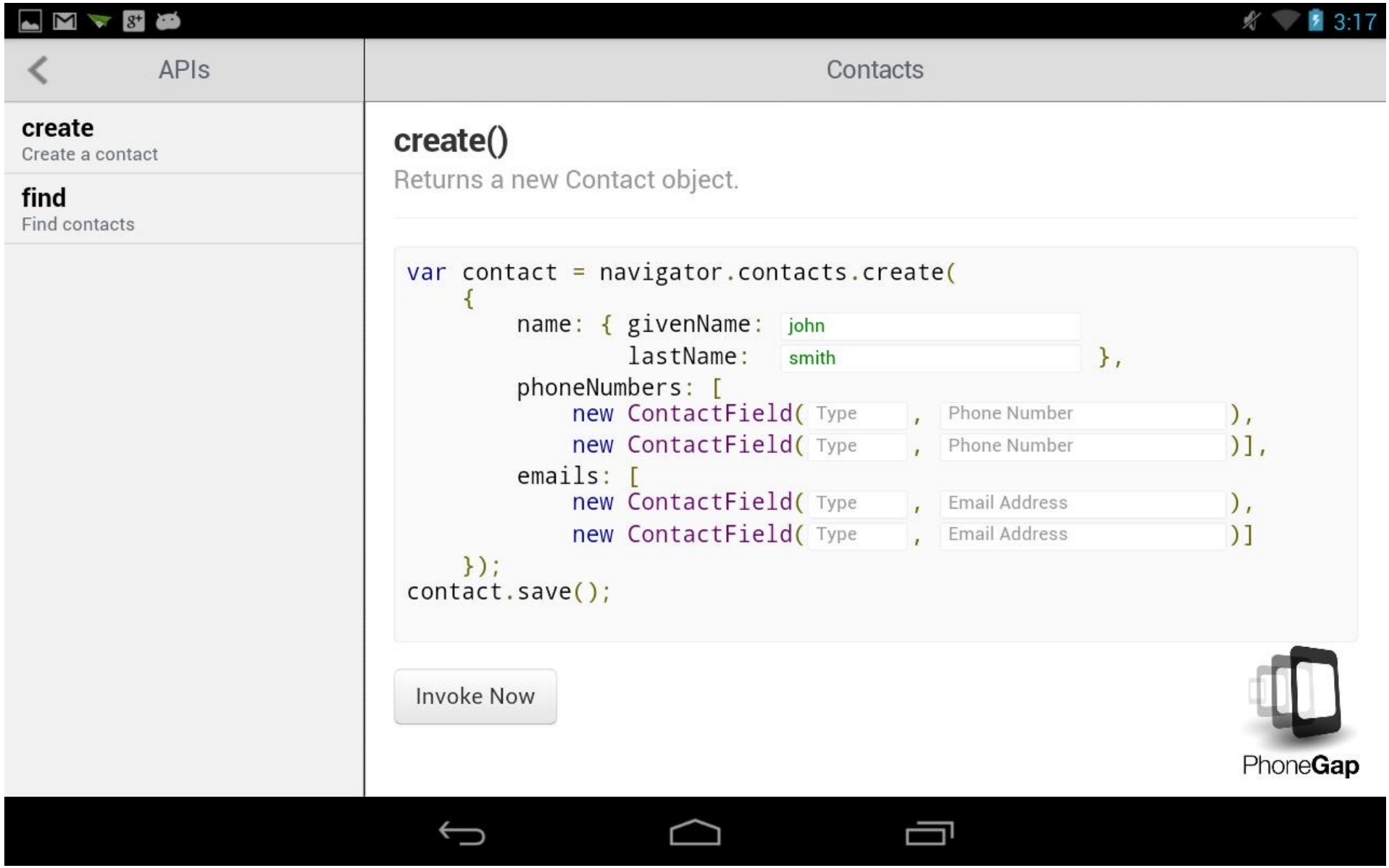
PhoneGap/Cordova Framework



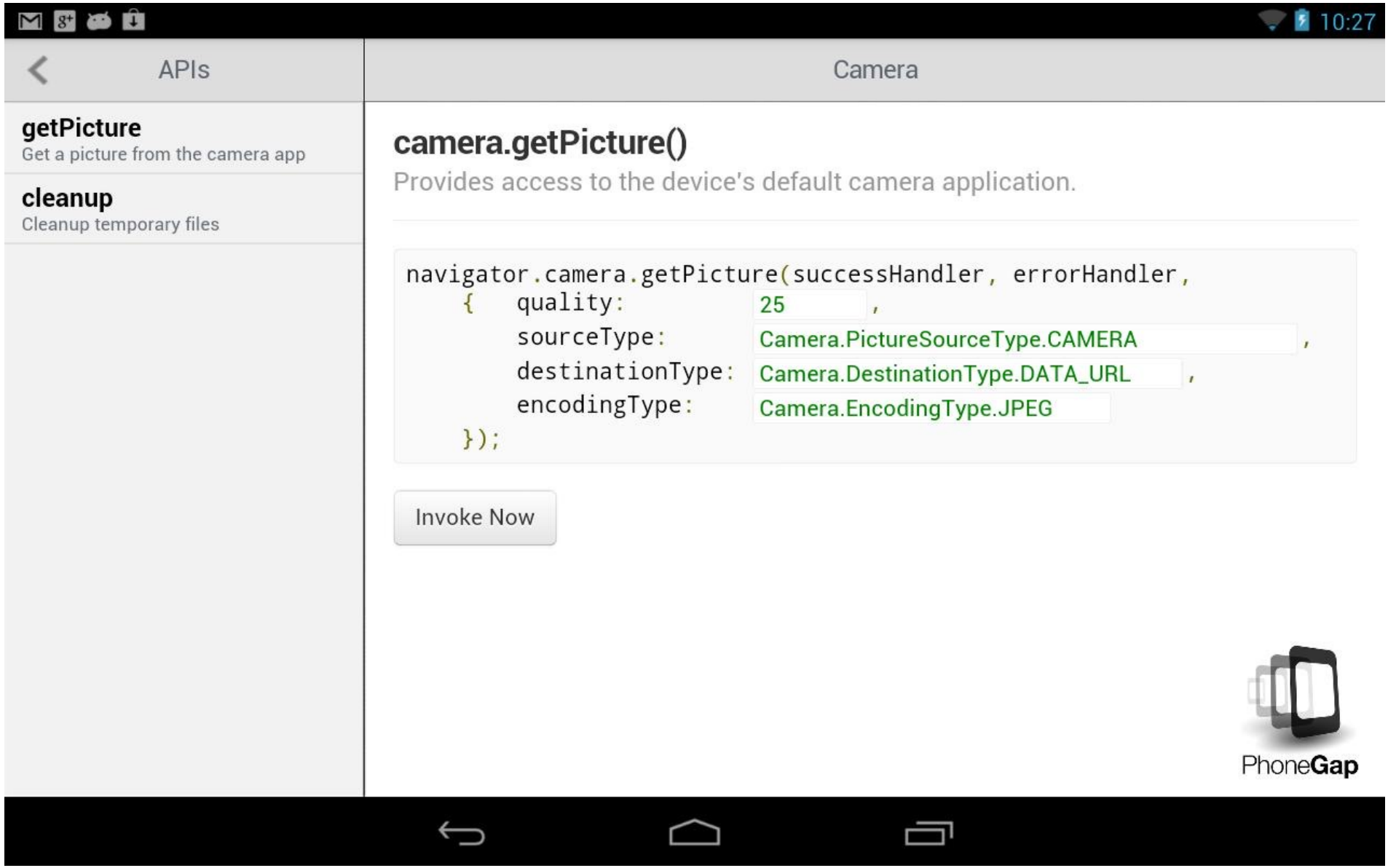
PhoneGap/Cordova APIs

	iOS iPhone / iPhone 3G	iOS iPhone 3GS and newer	Android	OS 4.6-4.7	OS 5.x	OS 6.0+	WebOS	WP7	Symbian	Bada
ACCELEROMETER	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓
CAMERA	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓
COMPASS	✗	✓	✓	✗	✗	✗	✗	✓	✗	✓
CONTACTS	✓	✓	✓	✗	✓	✓	✗	✓	✓	✓
FILE	✓	✓	✓	✗	✓	✓	✗	✓	✗	✗
GEOLOCATION	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
MEDIA	✓	✓	✓	✗	✗	✗	✗	✓	✗	✗
NETWORK	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
NOTIFICATION (ALERT)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
NOTIFICATION (SOUND)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
NOTIFICATION (VIBRATION)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
STORAGE	✓	✓	✓	✗	✓	✓	✓	✓	✓	✗

PhoneGap/Cordova Example 1



PhoneGap/Cordova Example 2



本週作業

學習前後端傳輸資料

1. 修改09_Codes中的colors程式,使其submit後能將作答結果送到自己開發，能秀出平均正確率與反應時間給受試者的scores.php (請使用txt, csv, or sqlite當database)。

----------以下是不計分的參考功能: ----------

2. 把Facebook與Google Login的功能加入到colors中，使受試者分數能以具名的方式儲存在後端。

Game Over