

心理與神經資訊學

(Psychoinformatics & Neuroinformatics)

課號：Psy1007

識別碼：20710100

教室：博雅 101

時間：四 234





聽說老師徹夜未眠

?

後端技術 **(PHP & Node.js)**

PHP vs. Django

這兩種後端的方法要學 / 用哪一種？

PHP

- 語法類似 C/Java/Javascript/Perl
- 適合靜態為主動態為輔的網頁
- 易學且使用人口和參考文件都豐富

Django

- Python 語法 (但有很多自己的規定)
- 適合動態為主靜態為輔的網頁
- 難學且使用人口和參考文件都較少

PHP vs. Node.js

這兩種後端的方法要學 / 用哪一種？

PHP

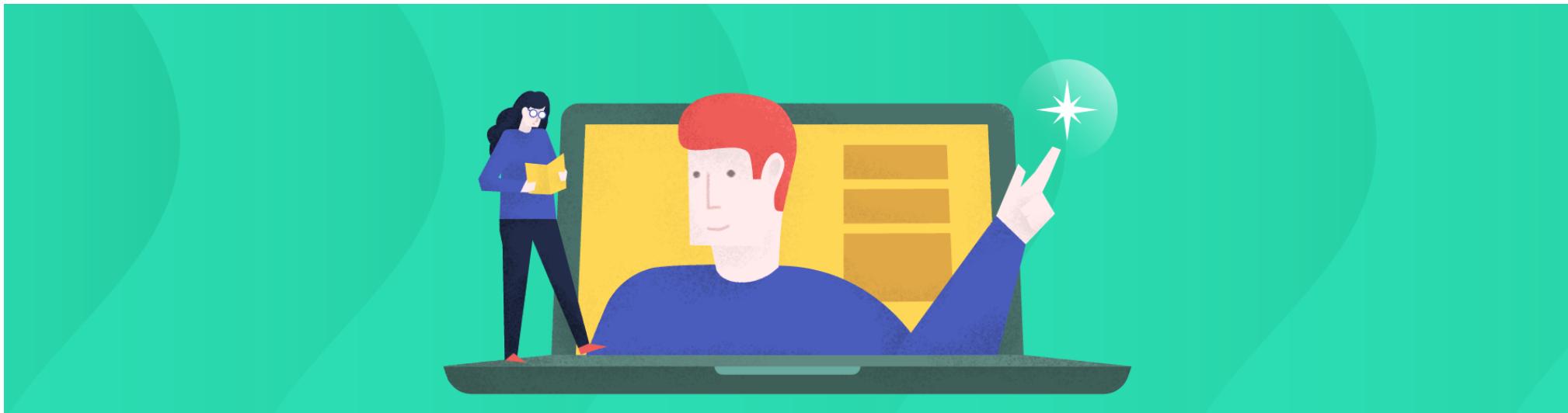
- 語法類似 C/Java/Javascript/Perl
- 效能較差
- 能和 HTML 混搭
- 易學且使用人口和參考文件都豐富

Node.js

- Javascript 語法 (但要處理很多 server events)
- 效能較好 (因 non-blocking I/O)
- 不能和 HTML 混搭
- ~~使用人口較少且學習門檻較高~~

同步化的 PHP vs. 非同步化的 JS

JS 中與 I/O 無關的指令不等待 I/O 的指令 (即非同步化)



當秘書去查取其它資料時 (即 slow I/O),
老闆可以不等待她，先做其它與資料無關的事。

非同步化 / 指令間不互相等待的 JS 工作效率比 PHP 高

非同步化 / 不等待的缺點

若很多 blocking/sequential codes 會變成 callback hell

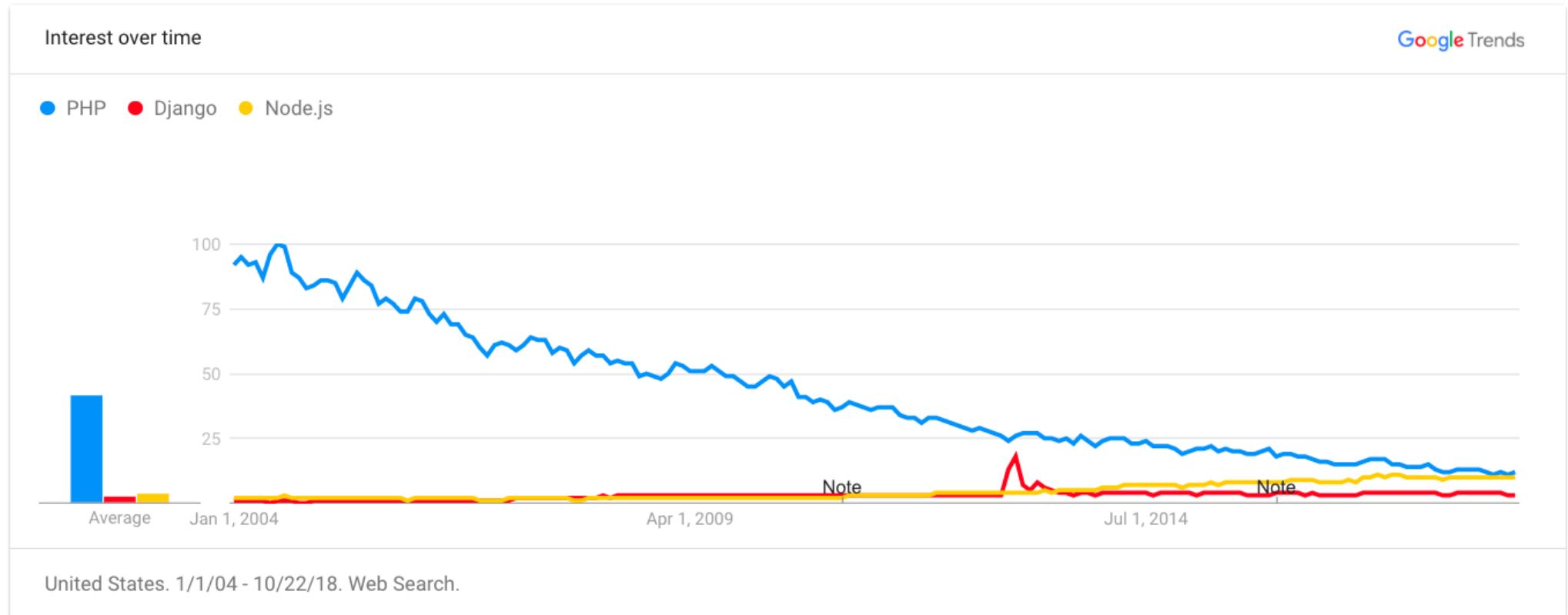
```
1  function hell(win) {
2      // for listener purpose
3      return function() {
4          loadLink(win, REMOTE_SRC+'/assets/css/style.css', function() {
5              loadLink(win, REMOTE_SRC+'/lib/async.js', function() {
6                  loadLink(win, REMOTE_SRC+'/lib/easyXDM.js', function() {
7                      loadLink(win, REMOTE_SRC+'/lib/json2.js', function() {
8                          loadLink(win, REMOTE_SRC+'/lib/underscore.min.js', function() {
9                              loadLink(win, REMOTE_SRC+'/lib/backbone.min.js', function() {
10                             loadLink(win, REMOTE_SRC+'/dev/base_dev.js', function() {
11                                 loadLink(win, REMOTE_SRC+'/assets/js/deps.js', function() {
12                                     loadLink(win, REMOTE_SRC+'/src/' + win.loader_path + '/loader.js', function() {
13                                         async.eachSeries(SCRIPTS, function(src, callback) {
14                                             loadScript(win, BASE_URL+src, callback);
15                                         });
16                                         });
17                                         });
18                                         });
19                                         });
20                                         });
21                                         });
22                                         });
23                                         });
24                                         });
25                                         );
26     }
```



但有將此階層性架構扁平化的各種JS套件

PHP vs. Django vs. Node.js

這三種後端的流行程度為何？



在米國 PHP 和 Node.js 快黃金交叉了

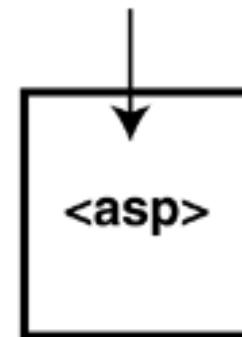
Request 傳送資料方式：Get vs. Post

Using GET

http://www.somedomain.com/register.asp?name=jobe&email=jobe@electrotank.com



比較方便



Using POST

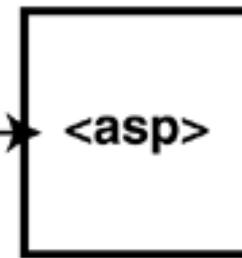
比較安全

http://www.somedomain.com/register.asp

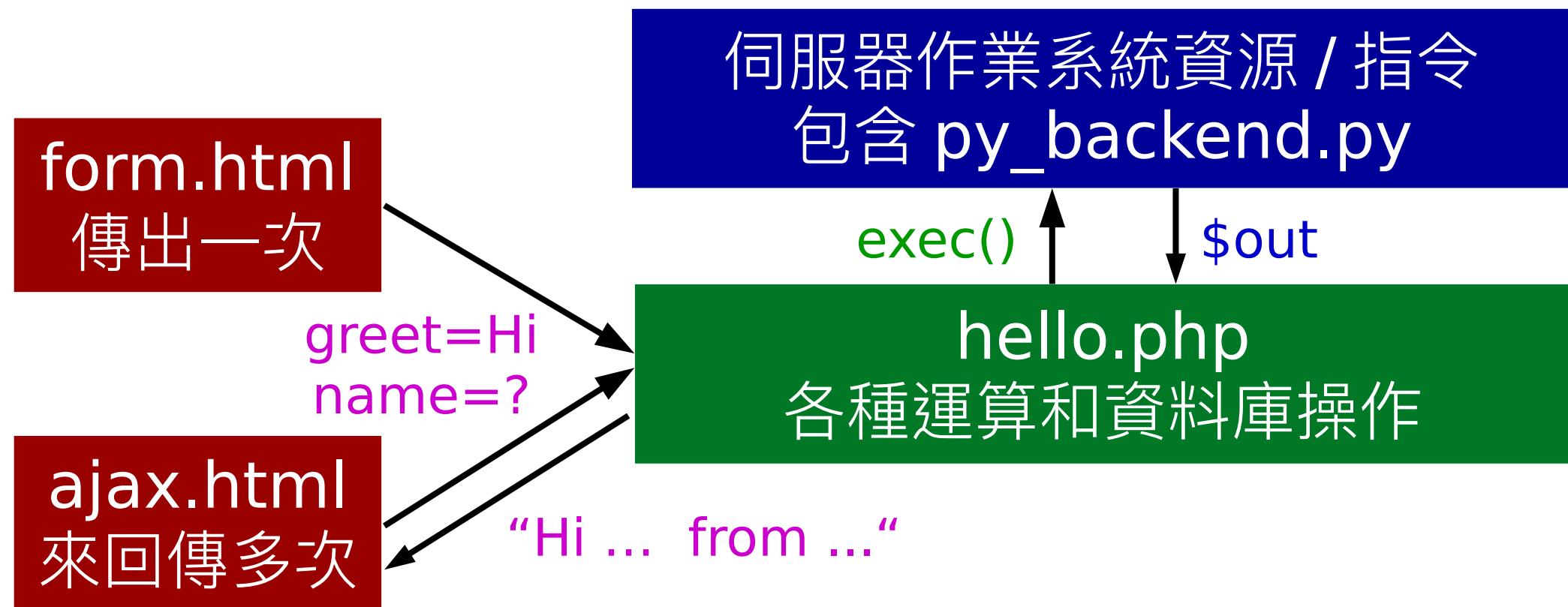


HTTP Request

name=jobe&
email=jobe@
electrotank.com



前端→後端 (1/2)



form.html

```
<form name="input" action="hello.php" method="get">  
<input type="hidden" name="greet" value="Hi">  
Username: <input type="text" name="name">  
<input type="submit" value="Submit">  
</form>
```

前端→後端 (2/2)

hello.php

```
<?php
$NAME=$_GET['name'];
$IP=$_SERVER['REMOTE_ADDR'];
echo $_GET['greet']."' $NAME from $IP <br>";
exec("python py_backend.py $NAME",$out);
exec("echo $NAME >> data.txt"); // append to data.txt
for($i=0;$i<count($out);$i++){
    echo $out[$i].'<br>';
}
?>
```

py_backend.py

```
import sys
print(str(sys.argv))
```

Node.js(+Express) 的版本

Node's req.query.name = PHP's \$_get['name']

hello3.js

```
var express = require('express');
var app = express();
const { execSync } = require('child_process');

app.get('/', function(req, res){
  res.send('name: ' + req.query.name);
  stdout=execSync('python py_backend.py'+
req.query.name).toString();
  console.log(stdout);
});

app.listen(8080);
```

後端→前端

後端產生 JSON 格式的資料
前端用 jQuery 的 `$.getJSON()`

test_json.php: 後端產生:

`{"name": "John", "age": 35}`

test_json.html: 前端取得上述資料後秀出 John35

test_jsonp.php: 後端產生:

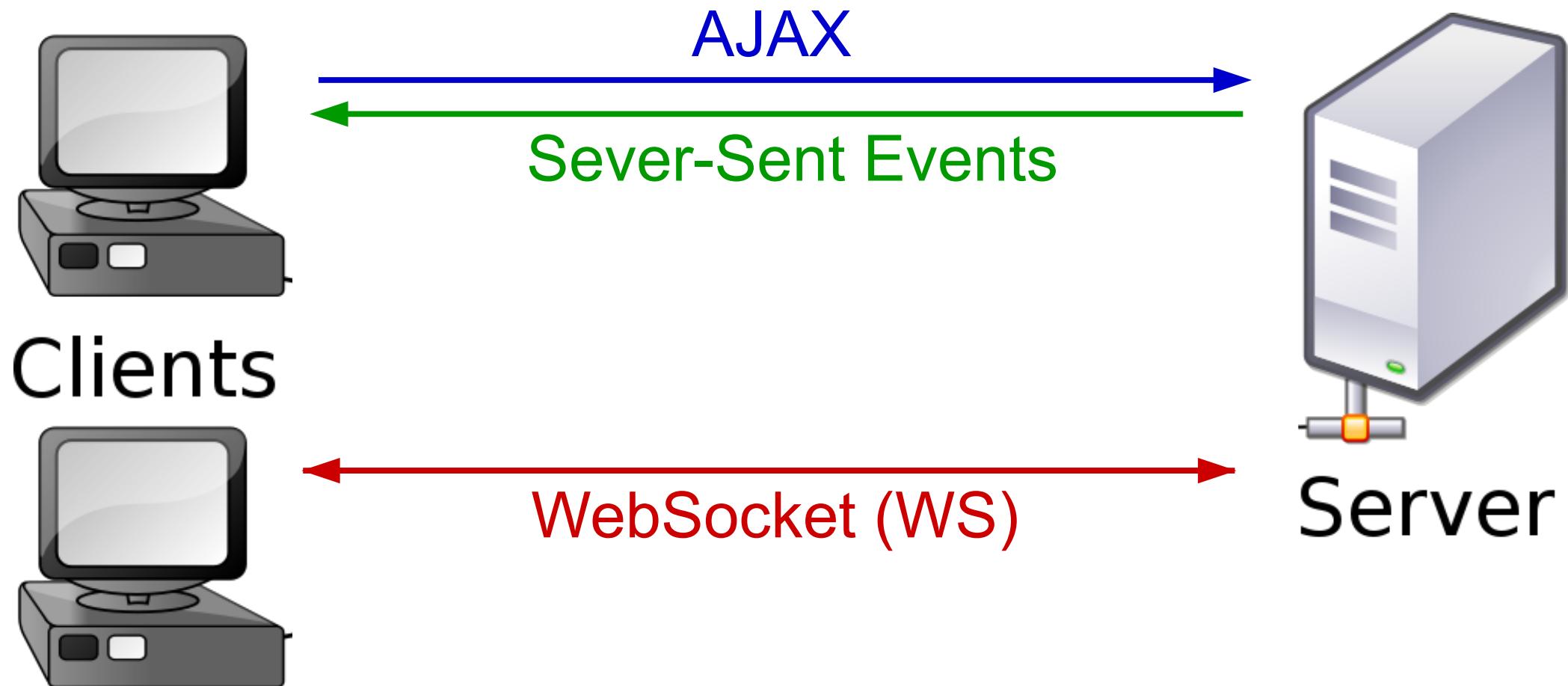
`({"name": "John", "age": 35})`

test_jsonp.html: 前端取得上述資料後秀出 John35

flickr.html: 前端取得 Flickr 四張圖片網址各做成
`` 後加入 `<div id="images"></div>`

Real-time 後端↔前端

AJAX/SSE 並無持久性的前後端連線但 WebSocket 有



前端 HTML5 與後端 Node.js/PHP 皆有實作 WS 協定

再比 PHP vs. Node.js

這兩種後端的方法要學 / 用哪一種？

PHP

- 雖有 ReactPHP: Event-driven, non-blocking I/O
- 但 PHP 核心和周邊各種套件本質上是序列執行
- 適合開發不需要那麼即時的大型網站

Node.js

- JS 和其套件都是 event-driven, non-blocking I/O
- 要序列執行或開發大型網站反而麻煩
- 適合開發 real-time bidirectional 應用 (如遊戲)
- 可 recycle 前端的 JS codes (如表單驗證)

JavaScript 與 PHP 混搭範例

rmet/index.php 用 JS 的 nextPage() 累計次數
並呼叫自己 index.php?i=k 來把圖片換成 k.jpg

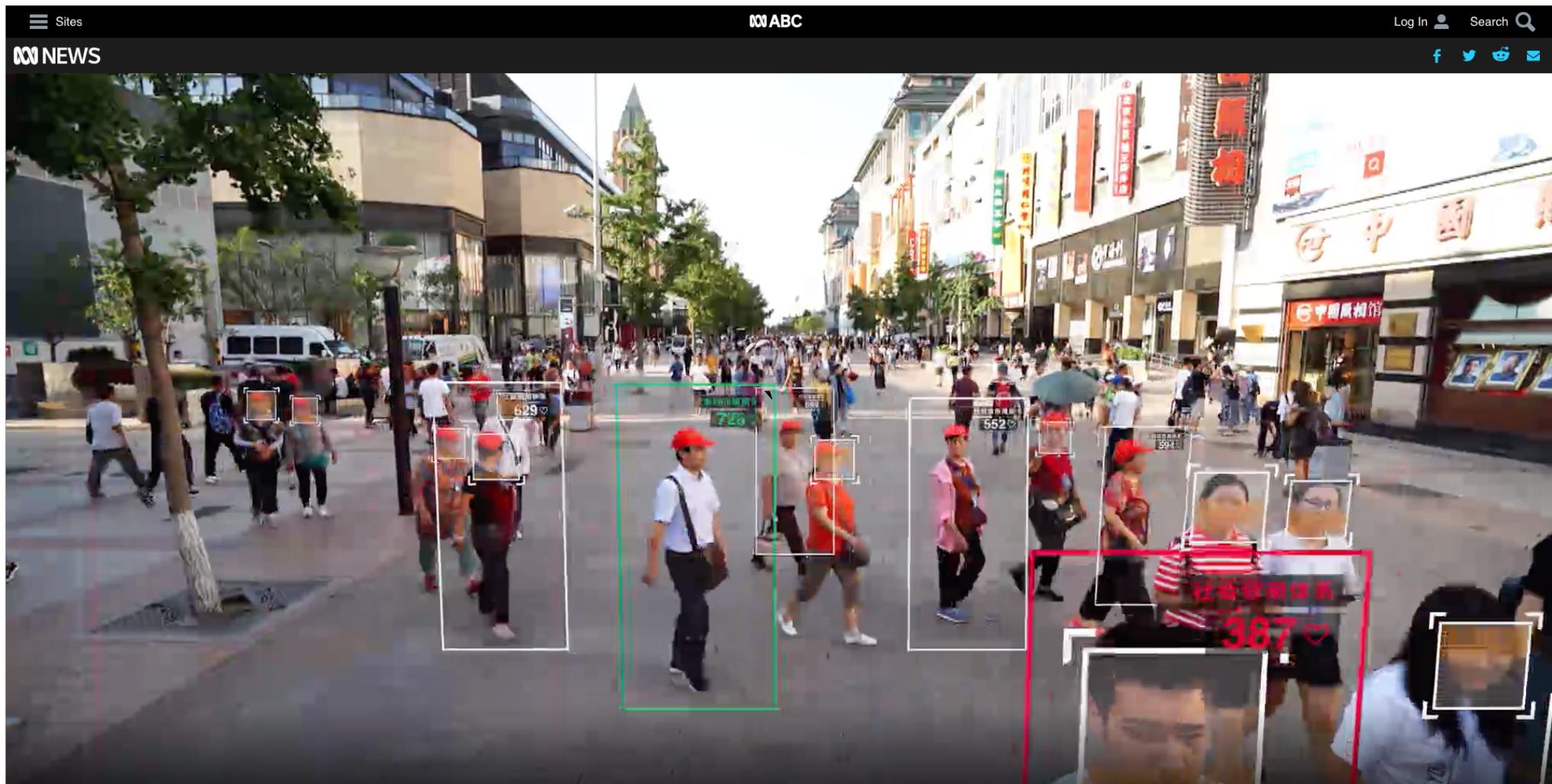
<?=\$a?> 是 <? echo \$a; ?> 的簡寫

\$a=(empty(\$_GET['i'])) ? 1 : \$_GET['i']) 可展開為：

```
If(empty($_GET['i'])){ //if not there
    $a=1;
}
else{
    $a=$_GET['i'];
}
```



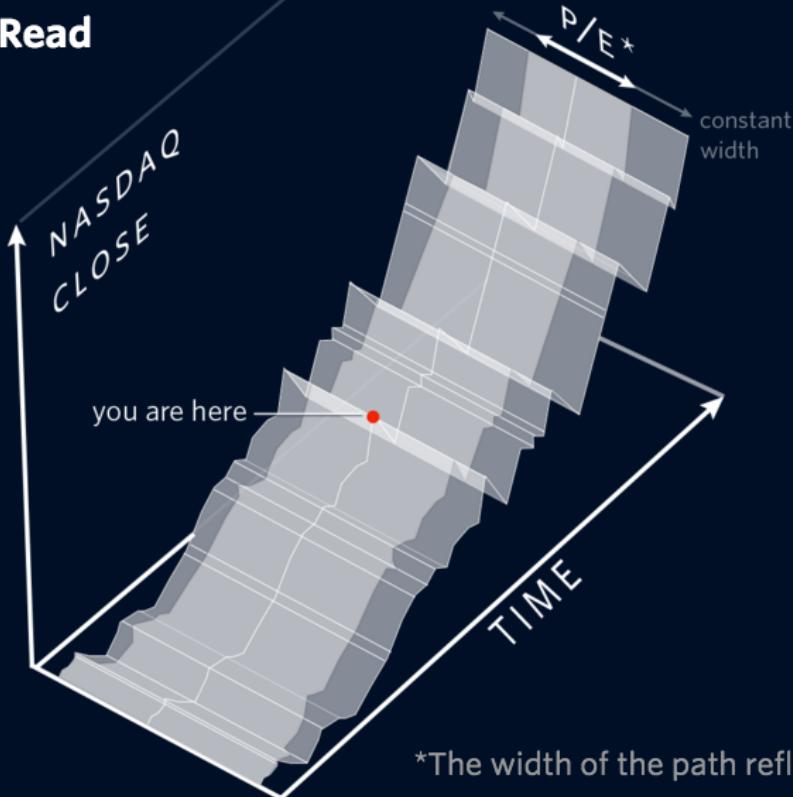
驚人的網頁程式範例 (1/4)



驚人的網頁程式範例 (2/4)

WSJ

How to Read



*The width of the path reflects the index's price/earnings ratio, with a narrower path reflecting higher share prices relative to earnings. A narrower path indicates a "bubble."

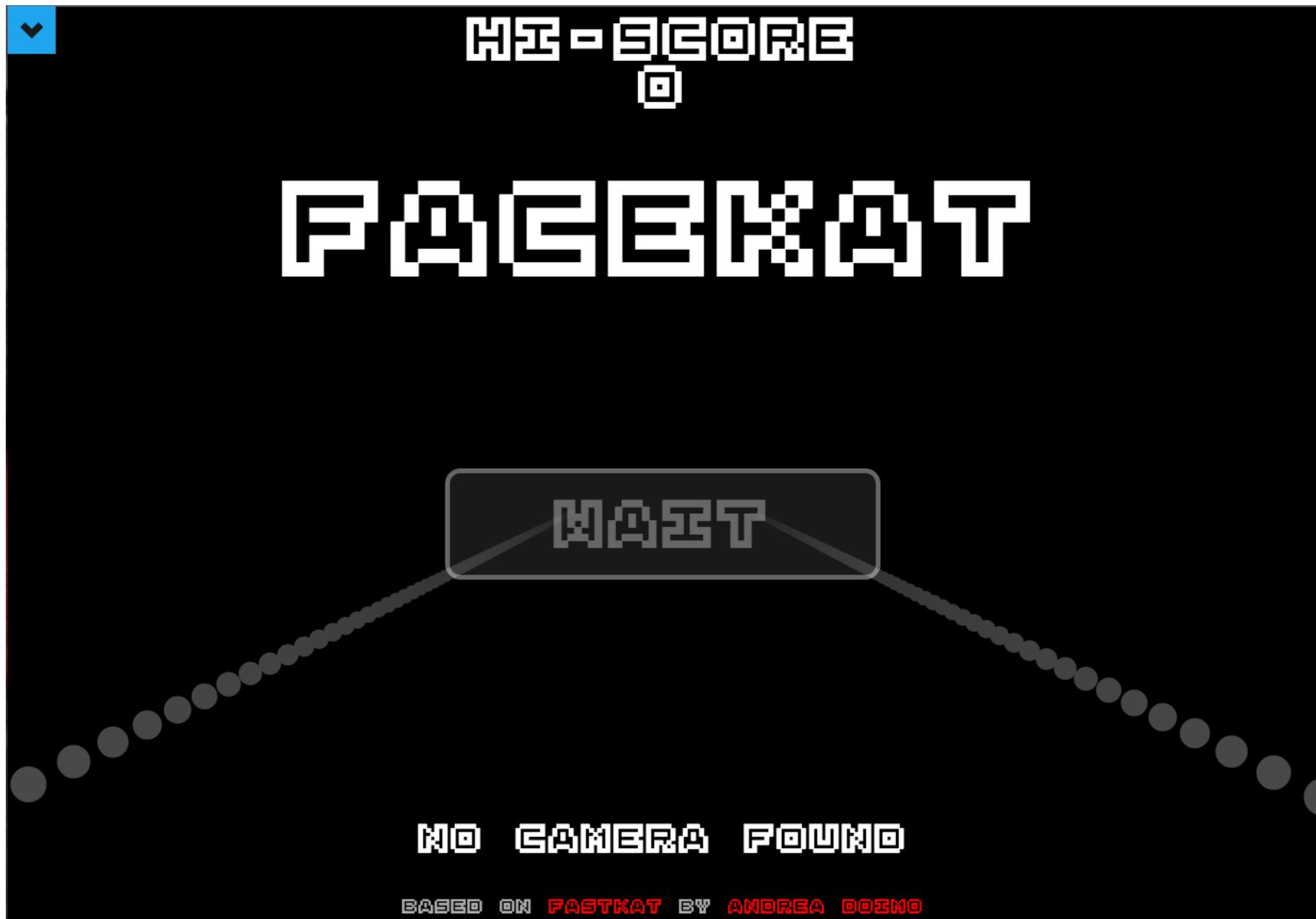
BACK

BEGIN

驚人的網頁程式範例 (3/4)



驚人的網頁程式範例 (4/4)



Game Over

