

# Web Programming

## 網際網路程式設計

Darby Chang

張天豪

# Time to headhunt/job-hop 挖角/跳槽時間

本學期的一大重頭戲

# 挖角/跳槽原則

- 仿照真實世界
  - 老闆登廣告，員工投履歷，過程不公開
- 挖角跟跳槽理論上都令人愉快，不受限，但被挖角及被裁員則相反，因此
  - 除非老闆願意(例如倒閉)，一家公司最多離開25%的員工
  - 除非該員工願意，否則不得裁員，但可以提醒該員工若不努力，會影響成績(必要時由老師介入)
- 最佳配對
  - 老闆跟員工互相排名，由配對分數較高者優先異動
  - 不排名視為不想加入該公司/不想聘用該員工
  - 例如，A公司有兩位員工  $x$  及  $y$  希望跳到  $X$  及  $Y$  公司，其中  $x$  將  $X$  排在第一位且  $X$  亦將  $x$  排在第一位，但  $y$  將  $Y$  排在第二位且  $Y$  亦將  $y$  排在第二位，因為A公司只能有一位離職，所以  $x$  成功離職， $y$  只好留下

# 挖角/跳槽特殊規則

- 例如有員工完全無法忍受現在的公司，但也不想撿現成的跳槽到已經成功的公司，想要自己重新組隊
- 或是有老闆想要主動挖角，雖然有同學沒有投履歷到自己公司，但仍然想要找他面談
- 有任何特殊理由，都請找老師幫忙協調，真的無法協調，只要有決心且願意承擔後果(例如重新組隊收到的員工品質)，在公平的原則下，最後會盡量尊重個人意願

日期	動作	日期	動作
11/21(一)	老闆訪談	11/28(一)	期中成果展，老師報告勞資意願現況
11/22(二)		11/29(二)	開放排名
11/23(三)	老闆繳交廣告文案，包含公司現況、人才需求以及一個圖示	11/30(三)	最後自我推薦、搶人時間，請務必填寫排名，未填寫者視為不想改變
11/24(四)	公告徵才訊息	12/01(四)	
11/25(五)	投履歷、面談，自由媒合時間	12/02(五)	關閉排名
11/26(六)	(歡迎借場地面試)	12/03(六)	
11/27(日)		12/04(日)	
		12/05(一)	宣佈結果，重組完成

有特殊需求的同學，請盡早與我聯絡，例如要重新組隊的同學，可能也要在 11/23 給我資料，才能幫忙徵才，要主動出擊的老闆，可以盡早跟老師要該員工的聯絡資料。最後，如果你有想法，盡早行動才是上策，不用乖乖照上面的時程走

# Midterm exhibition

## 期中成果展

另一個重要工作

# 期中成果展

- 每組 10 分鐘
- 投影片應包含設計理念，目前的成果，每位成員的工作、貢獻度，未來的目標，預期進度(時程表)，需要的人力(請考慮班上人才的比例)
  - 簡單的說，可以想成是挖角/跳槽曝光的好機會
- 實際操作網站



不論網站製作得如何，上台就是個專業的團隊，  
避免扭扭捏捏，拿出氣勢！連自己都覺得自己的  
東西不好，怎麼推銷給別人？

網站可以不好，demo不可以不好





# Ajax

Asynchronous JavaScript XML  
非同步的JavaScript與XML技術

# Ajax

- Ajax是多項技術的綜合應用
  - 運用HTML+CSS來表達資訊
  - 運用Javascript操作DOM物件執行動態效果
  - 運用XML進行資料交換
  - 運用XMLHttpRequest與網頁伺服器進行非同步資料交換
  - 運用Javascript技術實現
- Ajax不是指一種單一的技術，而是有機地利用了一系列相關的技術

# 有創意的利用現有技術

- 傳統的Web應用允許使用者端填寫表單，當送出表單時就向Web伺服器發送一個請求。伺服器接收並處理傳來的表單，然後送回一個新的網頁，但這個做法浪費了許多頻寬，因為在前後兩個頁面中的大部分HTML碼往往是相同的
  - 簡單的說，就是只更新頁面的一小部分
- 以前的人都有這些技術，例如用Javascript只更新一塊區域的內容(例如跑馬燈)，或是用XML來傳資料(但不是網頁)
  - 其實只要用Javascript依據這些資料來改某個區域的內容，就達到這個效果了 ← 創意的重要性

# 殺手級應用

- 傳統網頁郵件軟體
  - <http://mail.ncku.edu.tw/>
- 現代網頁郵件軟體
  - <http://gmail.com/>
  - Ajax 需要 Javascript，所以好的網站會提供非 Ajax 版(不需仰賴 Javascript)
- 其他 Google 軟體
  - <http://calendar.google.com/>
  - <http://docs.google.com/>
  - 俗稱 rich-content 軟體，因為一次只更新部分區域，所以介面非常豐富也沒有關係，不用每個動作都重繪整個頁面
  - 有些內容太豐富了，非 Ajax 版不知道該怎麼做

# 常見需要Ajax的設計元件

- 自動完成(auto complete)
  - [Ajax Autocomplete for jQuery](#)
  - [FCBKcomplete](#)
  - [ajax autocomplete - Google 搜尋](#)
- 工具提示(tooltip)
  - [40+ Tooltips Scripts With AJAX, JavaScript & CSS](#)
  - [tooltip - Google 搜尋](#)
- 上傳
  - <http://valums.com/ajax-upload/>
  - [upload - Google 搜尋](#)
- 人是很慢的，先給人一點東西看(或是讓人輸入)，中間的間隔就很夠機器做事了，這時不偷偷處理資料太可惜

# 一個簡易的Ajax範例

## ■ 設計想法

- 在頁面中可以填寫資料，按個鈕送出該資料
- 送出後不等伺服器回應，控制權立刻回到瀏覽器 (Javascript) 手上 ← 這種行為稱為非同步 (asynchronous)
- 改變內容顯示loading，會讓使用者感到安心
- 伺服器真的運算完畢後，顯示回傳的內容

# HTML 重點

- 表單的設計最好依照非Ajax的慣例，方便製作非Ajax版本
- 通常用來顯示回傳資料的DOM元件會設定特別的id屬性，方便Javascript操作



# Javascript 重點

- 用來啟動Ajax的事件
- 取得表單資料
- 呼叫Ajax
  - 設定傳給伺服器的資料
  - 指定CGI程式
  - <http://api.jquery.com/category/ajax/>
- 呼叫完Ajax後產生了兩個非同步的流程
  - 傳資料給伺服器後，顯示訊息讓使用者安心
  - 伺服器運算結束後，顯示運算結果



# Today's assignment

## 今天的任務

# 讓網站更流暢

- 將既有透過CGI的功能改為Ajax版本，使得送出表單資料後，不用重新產生整個頁面，而是只更新部分區域，如果配上適合的訊息，使用者會覺得整個操作都是即時的
- 參考資料
  - <http://api.jquery.com/category/ajax/>
  - <http://www.ajaxload.info/>
  - <http://www.preloaders.net/>
  - [5 Ways to Make Ajax Calls with jQuery](#)
- 記得寄快照給助教，首頁(或寄信給老師)要說明怎麼驗收改版重點
- 老師會在 12/4星期日 23:59 之後到各組的首頁去看

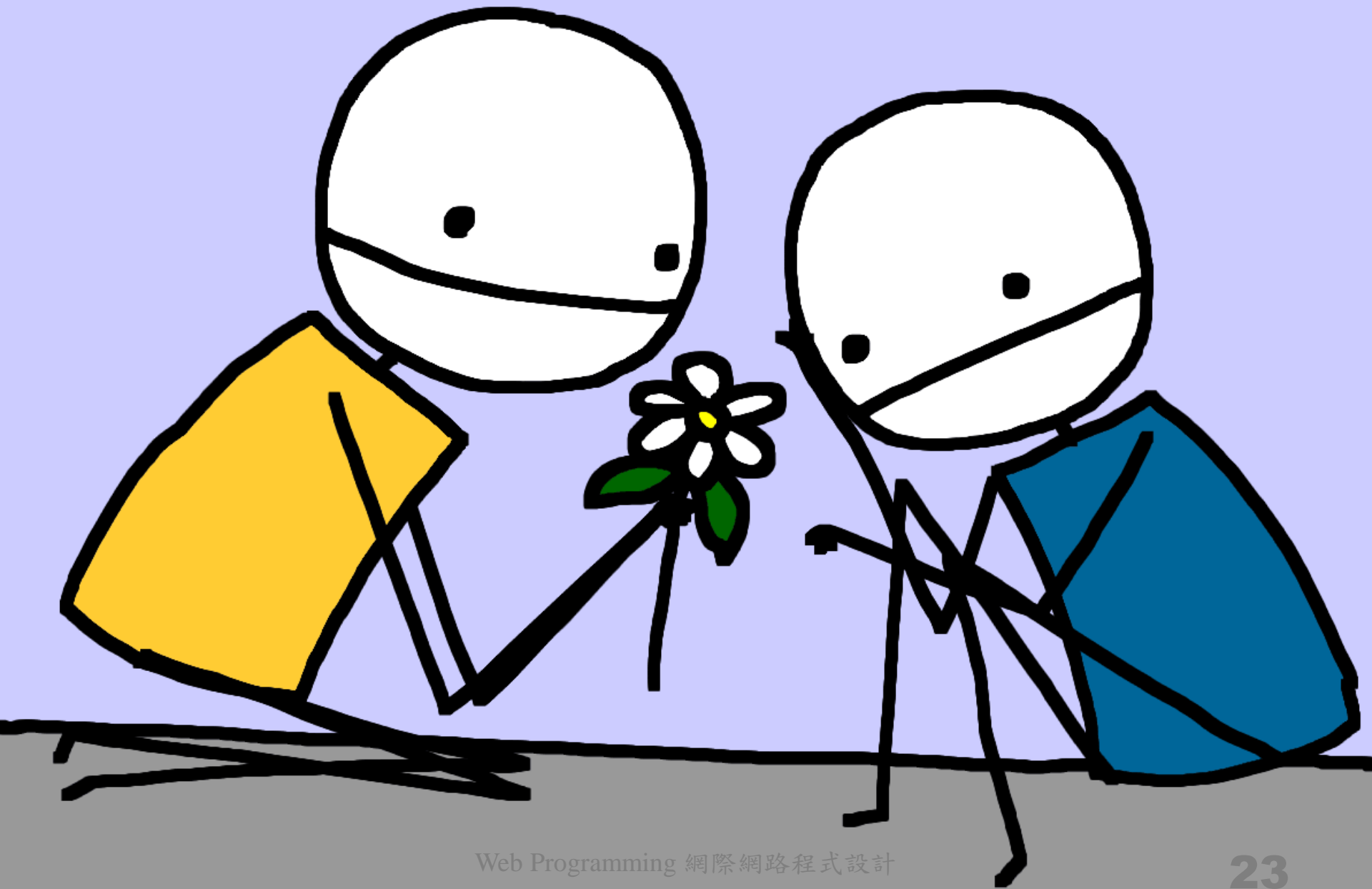
# Midterm exhibition

## 期中成果展

千萬別忘了這件事，佔學期總成績最少 30%

# Appendix

## 附錄



# 其實真的沒那麼難，要做的就是

流程	技術	範例
有地方讓使用者輸入/動作	HTML表單	<code>&lt;input type="text" /&gt;</code> <code>&lt;button&gt;讚&lt;/button&gt;</code>
能夠取得/偵測使用者的輸入/動作	事件	<code>jQuery("[type=text]").val();</code> <code>jQuery("button").click();</code>
將資料送至伺服器	HTTP request	<code>jQuery.ajax();</code>
伺服器有個程式處理這些資料	CGI	<code>my \$cgi = new CGI;</code> <code>\$cgi-&gt;param();</code>
將結果回傳至瀏覽器	CGI	<code>print;</code>
接收CGI結果	事件	<code>jQuery.ajax(success:function(data){</code> <code>});</code>
將結果呈現給使用者	HTML	<code>jQuery().html()</code>



# 還來得及，你們做得到的

- Facebook有什麼功能是你做不到的嗎？
- 他們一組才三個人
- 剩下還有一個多月，加上實際上機，你們的作品還大有可為
- 要有氣勢！