

# PYTHON

程式設計與網站擷取

Lecturer: 林哲緯

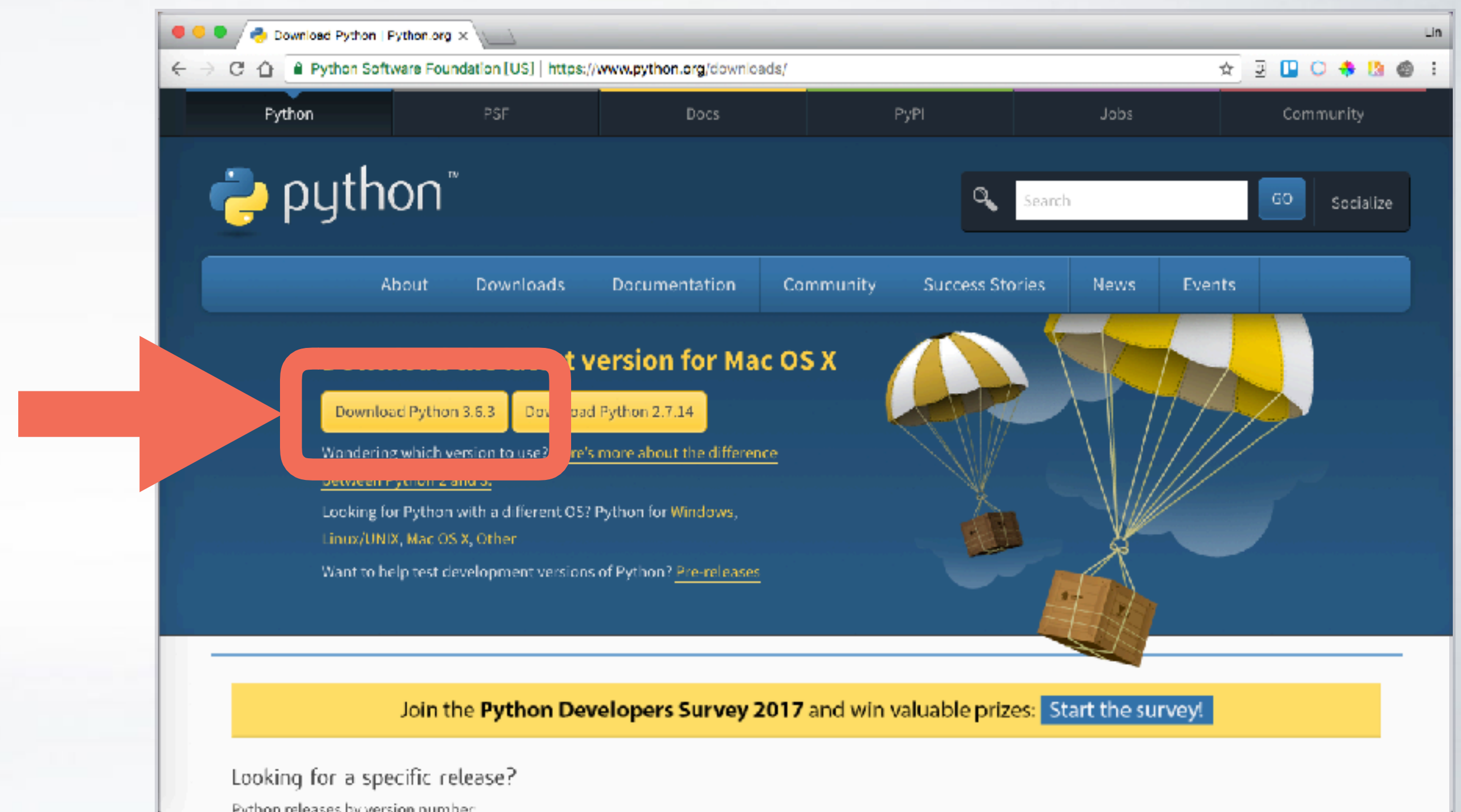


# Welcome to Python!

## 開始安裝 Python

- Windows:
  - ★ python-3.X.X.exe [安裝時注意勾選加入PATH選項]
- Mac OSX:
  - ★ python-3.X.X-macosx...pkg
- Linux:

```
sudo apt-get update  
sudo apt-get install python3
```



[按我前往](#)



# 上手 命令提示字元/終端機/Terminal 小教室

## 1. 打開方式

- Windows:

- ★ 開始->附屬應用程式->命令提示字元

- ★ (或)開始->搜尋->輸入:CMD

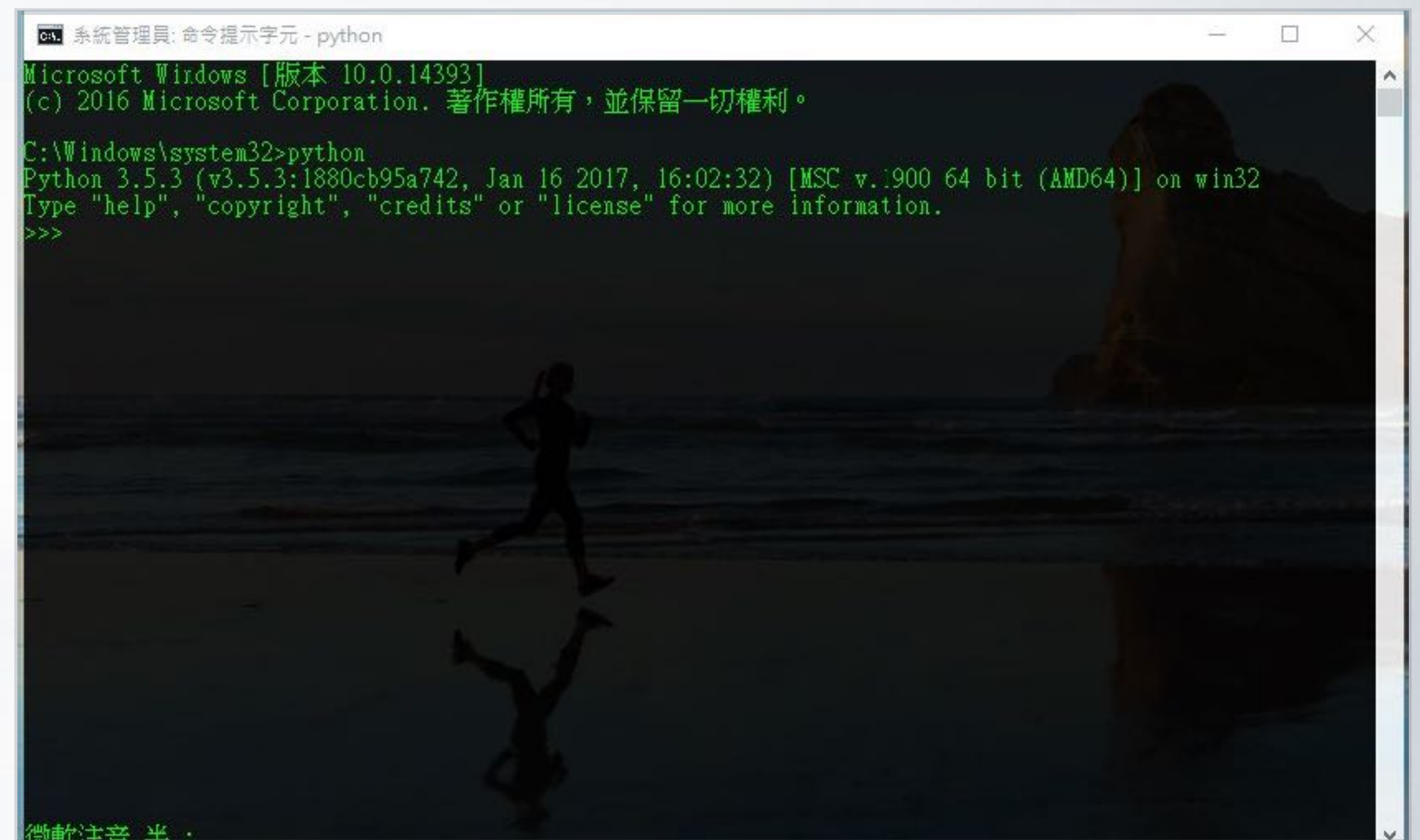
[提示] 可以使用「以管理員身份執行」排除安裝權限問題，  
但須注意與非管理員方式開啟路徑差異

- Mac OSX:

- ★ 應用程式(Applications)->工具程式->終端機

- Linux:

- ★ 搜尋->Terminal



# 上手 命令提示字元/終端機/Terminal 小教室

## 2.常用命令

功能	Windows	Mac/Linux
目錄跳轉	<code>cd /d d:/demo</code>	<code>cd d:/demo</code>
新增資料夾	<code>md d:/demo</code>	<code>mkdir d:/demo</code>
刪除檔案	<code>del d:/demo.txt</code>	<code>rm d:/demo.txt</code>
顯示目錄內檔案	<code>dir</code>	<code>ls</code>
顯示目錄路徑	<code>cd</code>	<code>pwd</code>
清除畫面	<code>cls</code>	<code>clear</code>
移動/重新命名檔案	<code>ren d:/demo.txt d:/test.txt</code>	<code>mv d:/demo.txt d:/test.txt</code>
複製檔案	<code>copy d:/demo.txt d:/test.txt</code>	<code>cp d:/demo.txt d:/test.txt</code>



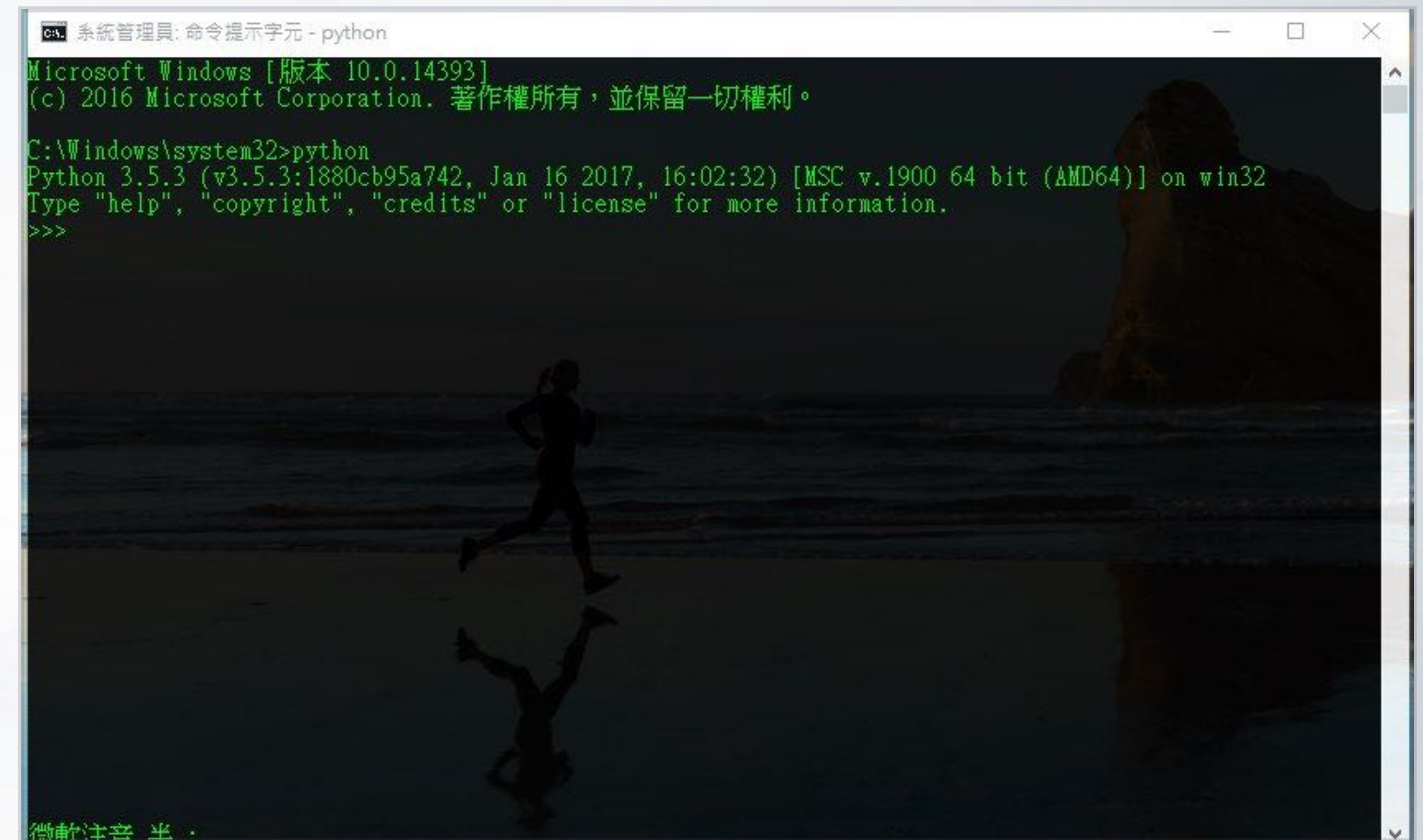
# Welcome to Python!

## 如何確認你已經裝好Python 或本來就已經安裝好了？

打開：

1. 命令提示字元/終端機/Terminal
2. 輸入python/ python3

你已經可以開始寫程式了！



```
系統管理員: 命令提示字元 - python
Microsoft Windows [版本 10.0.14393]
(c) 2016 Microsoft Corporation. 著作權所有，並保留一切權利。
C:\Windows\system32>python
Python 3.5.3 (v3.5.3:1880cb95a742, Jan 16 2017, 16:02:32) [MSC v.1900 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

# Welcome to Python!

```
Python 3.6.1 (default, Mar 23 2017, 16:49:06)  
[GCC 4.2.1 Compatible Apple LLVM 8.0.0 (clang-800.0.42.1)] on darwin  
Type "help", "copyright", "credits" or "license" for more information.  
>>>
```

試著輸入

```
>>> 5+6
```

Enter

```
>>> 5+6  
11
```



# Welcome to Python!

如何退出 Python shell ?

```
>>> 5+6  
11  
>>> 23-32  
-9  
>>> exit()
```

Enter



輸入exit() 或者試試 Ctrl + D

```
>>> 5+6  
11  
>>> 23-32  
-9  
>>> exit()  
JheWeide-MacBook-Pro:~ kevin$
```

## 再次回到Python Shell 並試試以下例子

```
JheWeide-MacBook-Pro:~ kevin$ python  
Python 3.6.1 (default, Mar 23 2017, 16:49:06)  
[GCC 4.2.1 Compatible Apple LLVM 8.0.0 (clang-800.0.42.1)] on darwin  
Type "help", "copyright", "credits" or "license" for more information.  
>>>
```



```
>>> 11*11  
???  
>>> 11**2  
???  
>>> 321%(123+231/3)  
???
```



# Welcome to Python!

安裝完了嗎？不。  
還有一個東西很重要

## 互動式記事本： Jupyter Notebook

```
pip3 install --upgrade pip  
pip3 install jupyter
```



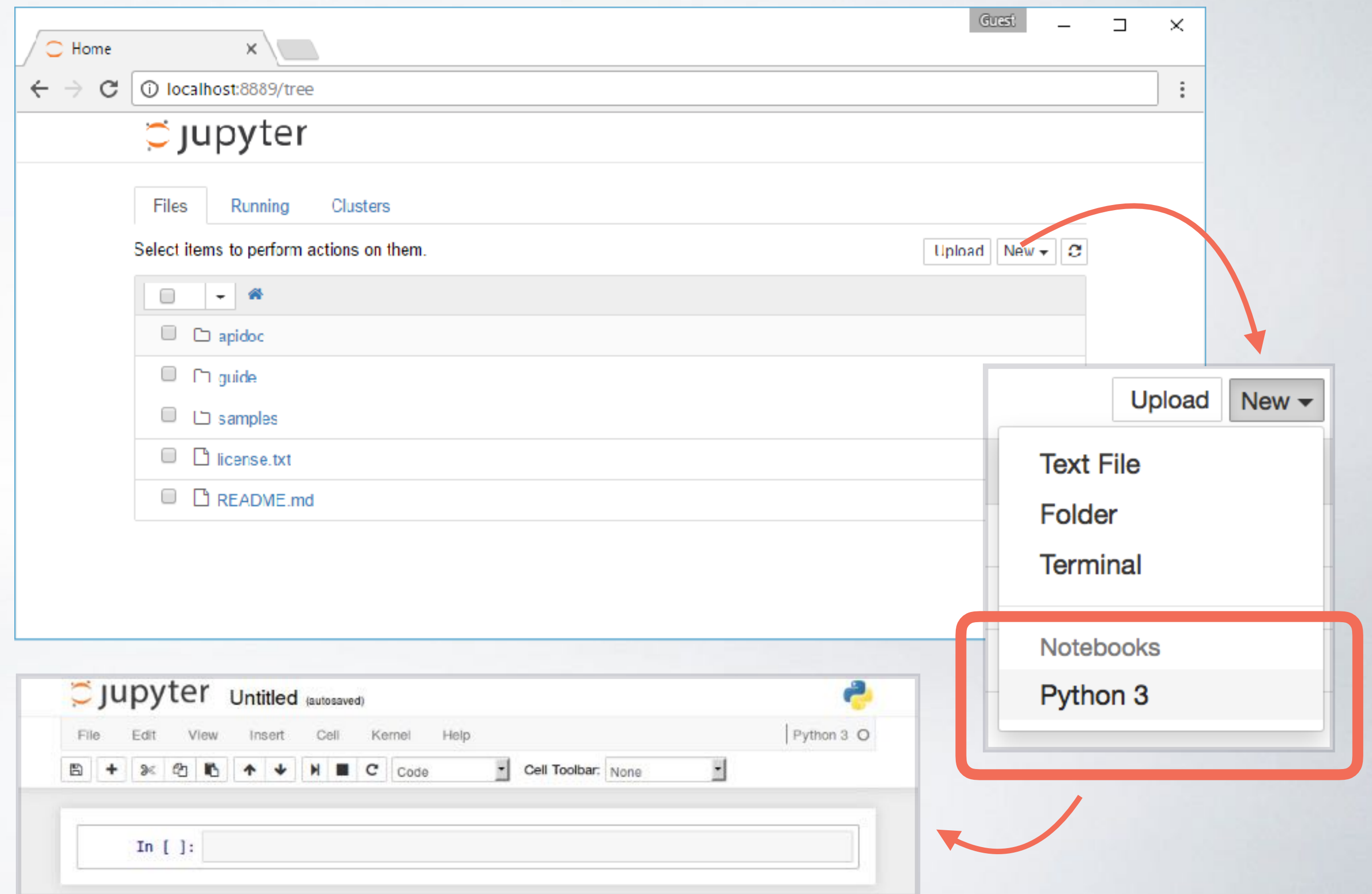
Jupiter>Jup**py**ter



# Welcome to Python!

## 如何啟動 Jupyter Notebook

jupyter notebook







## Web Scrapping 網頁擷取

- 即「網路爬蟲」，意指透過程式獲取網際網路上「有用的」資訊。
- 既生Google，何生「我的蟲」？
- 廣義來說，便是「自動化」地模擬人類在瀏覽器上的操作行為，舉凡：
  - 追蹤網站最新資訊(定時排程)
  - 外掛機器人(重複且快速)
  - 搜尋引擎應用(儲存、分析、查詢)



# 撰寫爬蟲的第一步 用爬蟲的角度來看世界



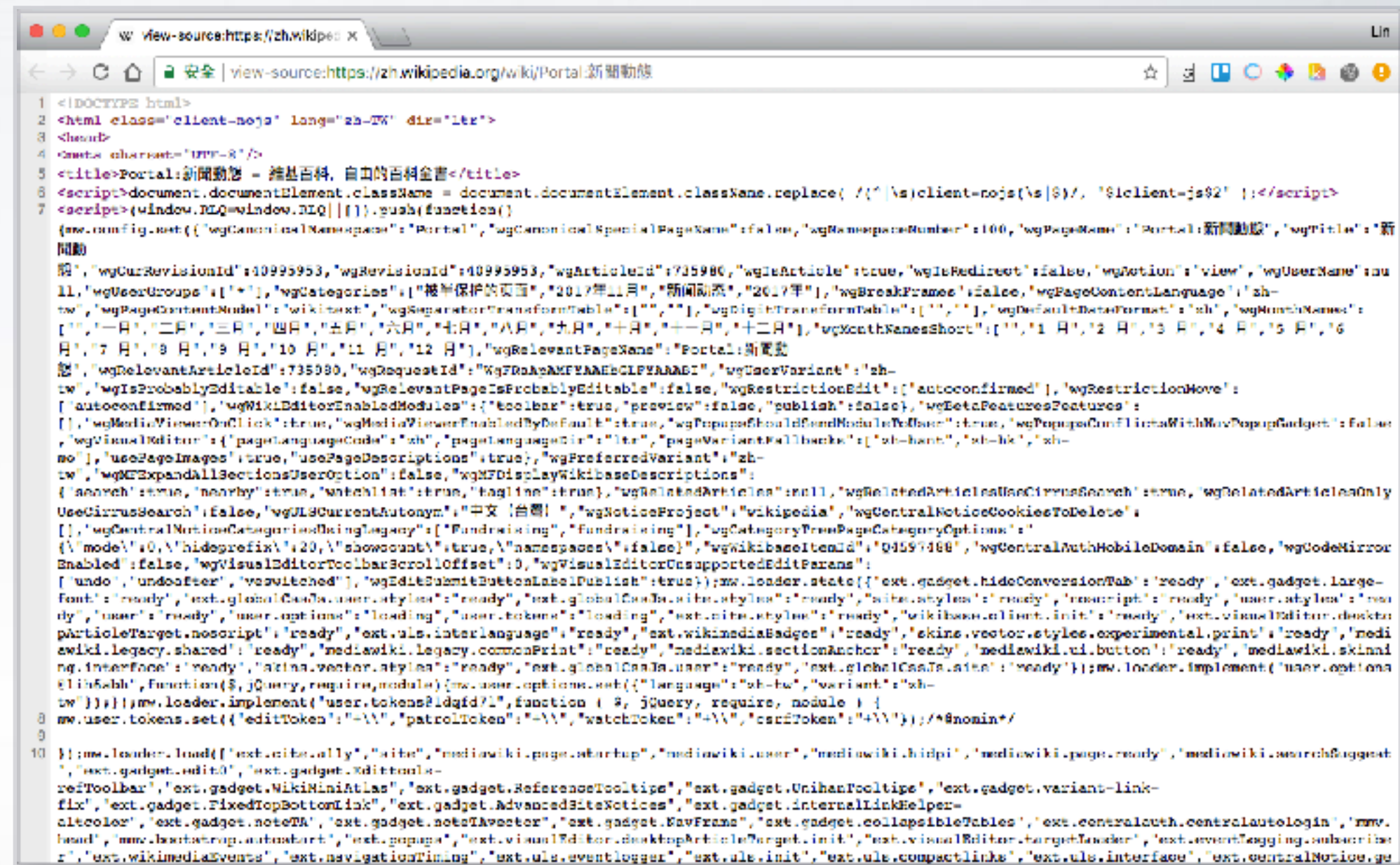
# 這是你所看到的網頁



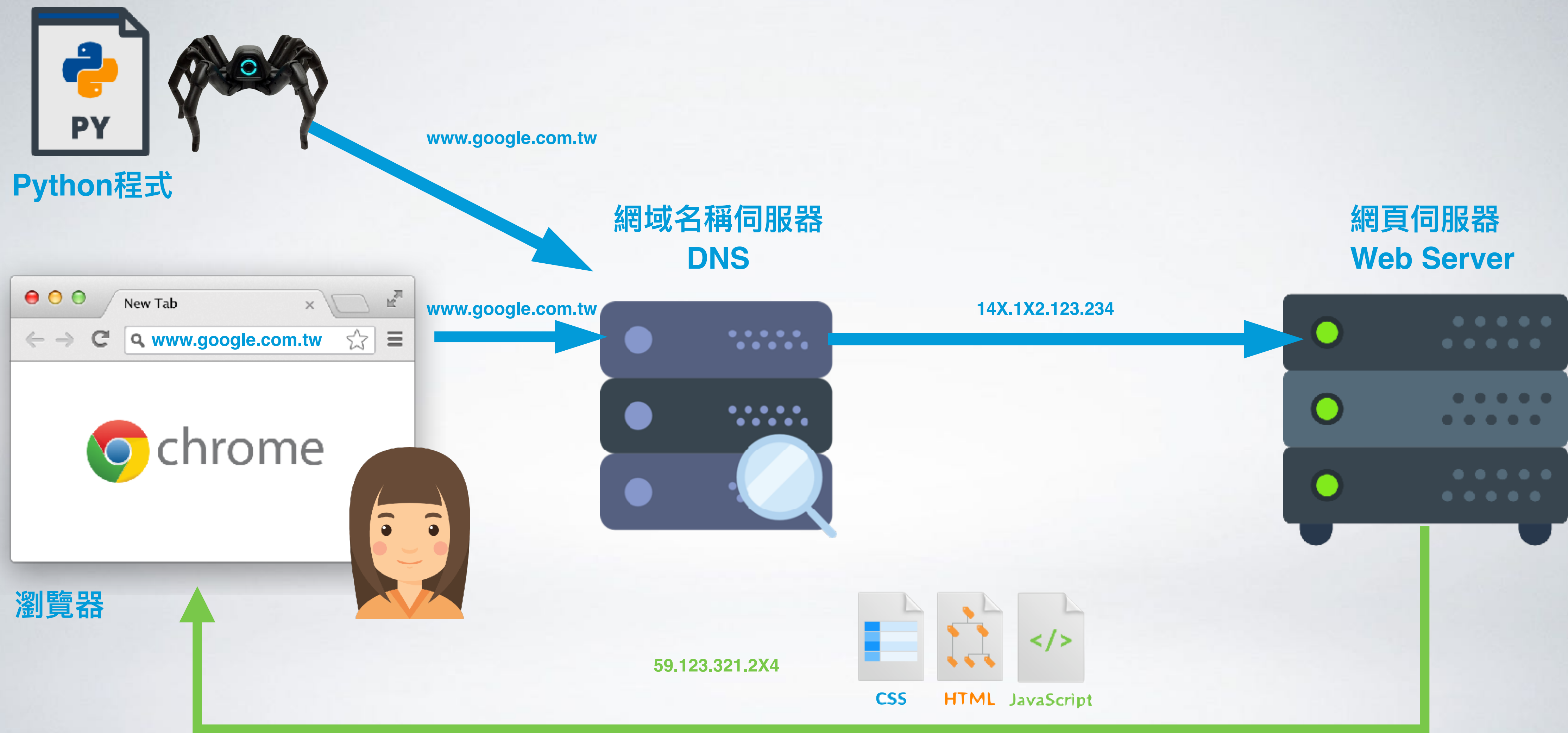
以維基百科-新聞動態為例



# 這是爬蟲所看到的網頁







# URL (Uniform Resource Locator)

## 統一資源定位地址

- URL也就是我們說的網址，是網際網路上各種資源的地址
- 一種從網路上得到資源的簡潔訪問表示法
- URL是爬蟲獲取數據的基本依據

## URL的格式

<http://www.ptt.cc/bbs/Gossiping/index.html>



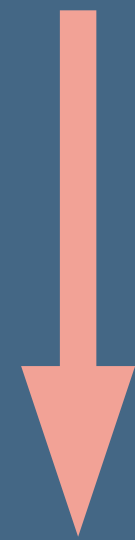
1.協議  
(或稱為  
服務方式)



2.伺服器  
域名



3.目錄/子目錄  
資源位址

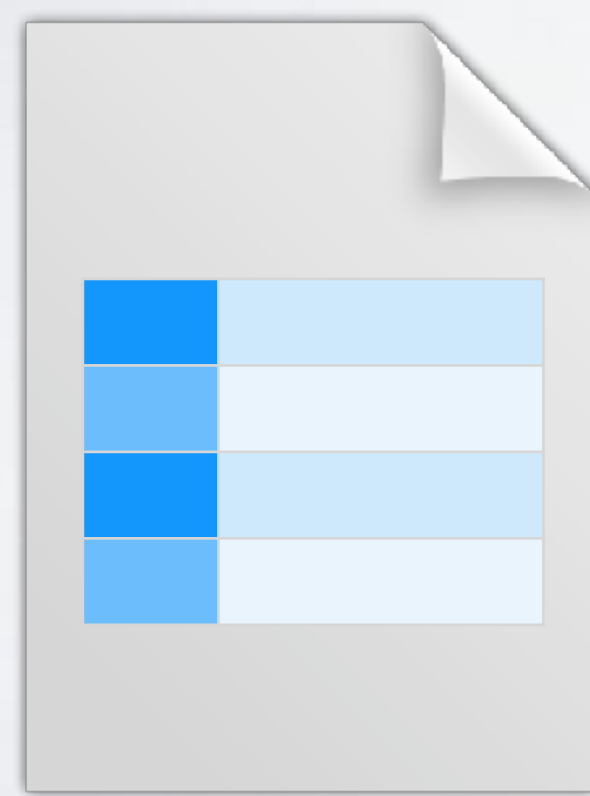


4.檔案名稱

FTP://  
Telnet://  
HTTPS://



# 網頁構成要素



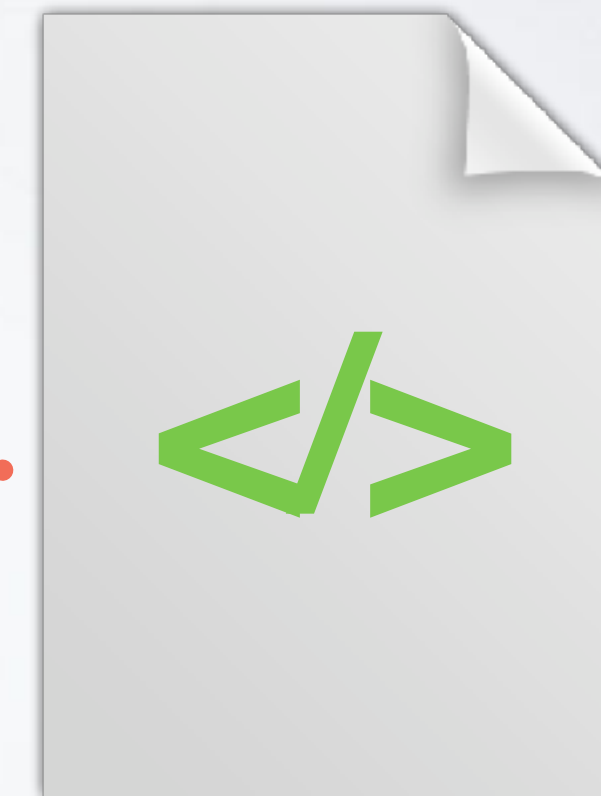
**CSS**

外觀



**HTML**

框架 / 內容



**JavaScript**

互動

jQuery  
React.js  
D3.js  
...

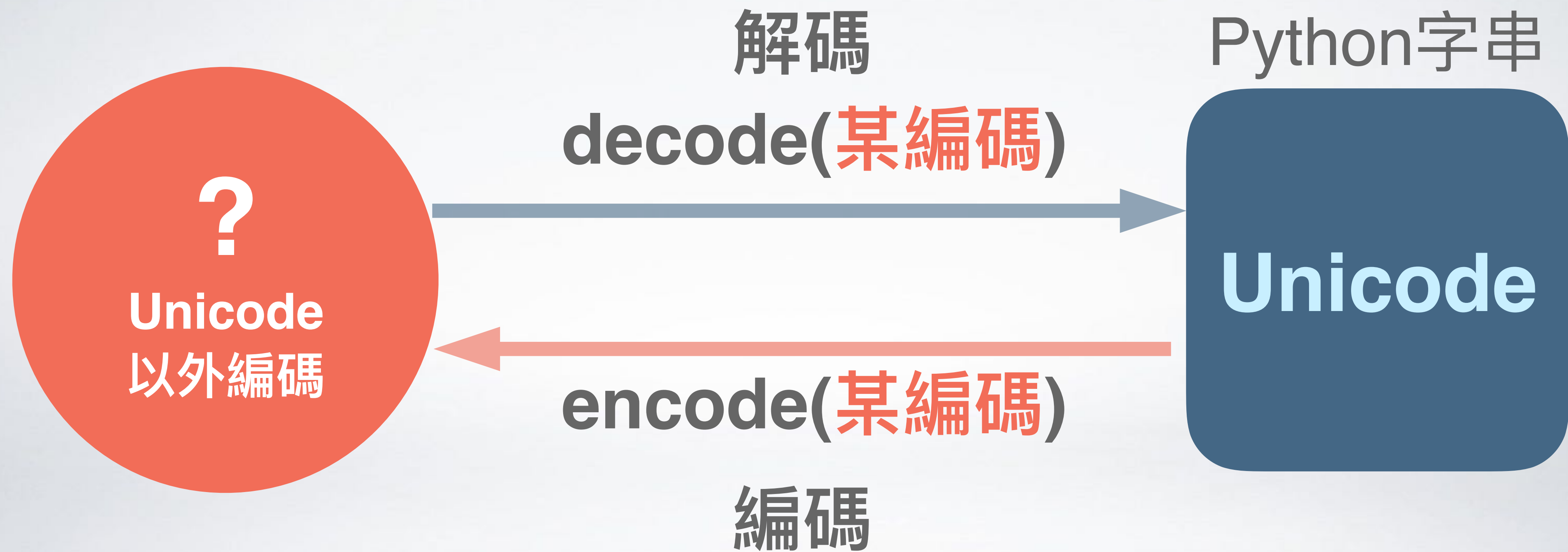
# Unicode 萬國碼

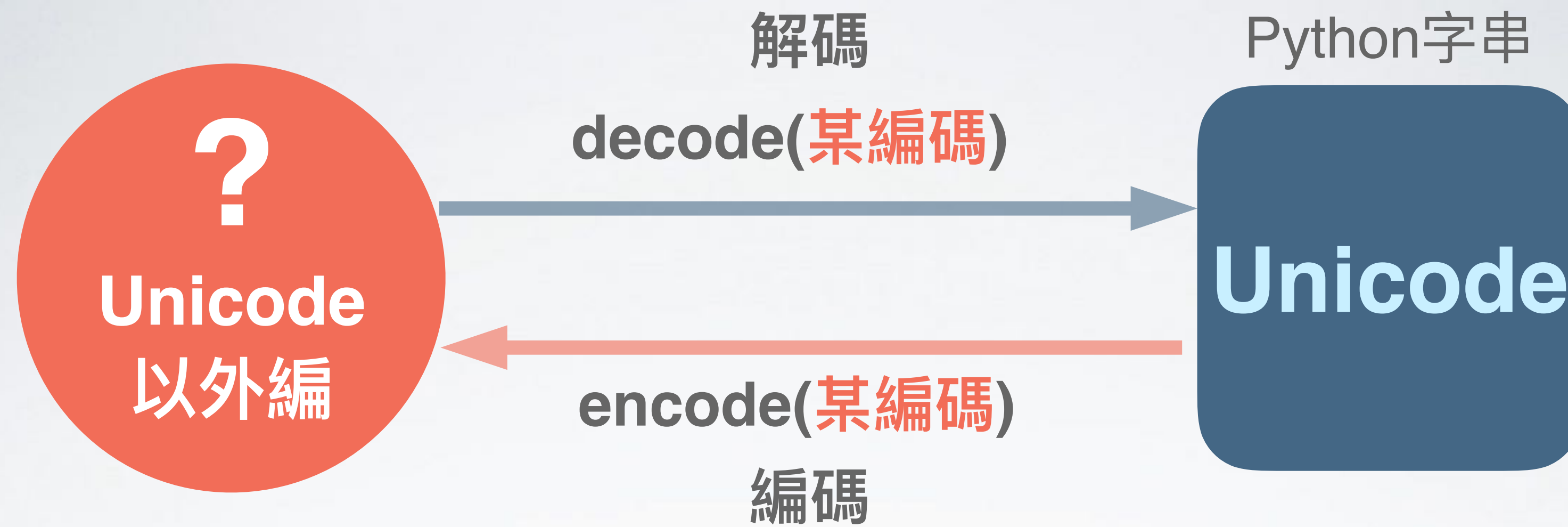
- 定義了全世界所有語言的字元，加上數字與各式領域的符號
- Unicode是為了解決傳統的字元編碼方案的侷限而產生的，可以表示所有的文字
- Python的字串就是Unicode編碼



<https://unicode-table.com/>







## 情境練習題 #1

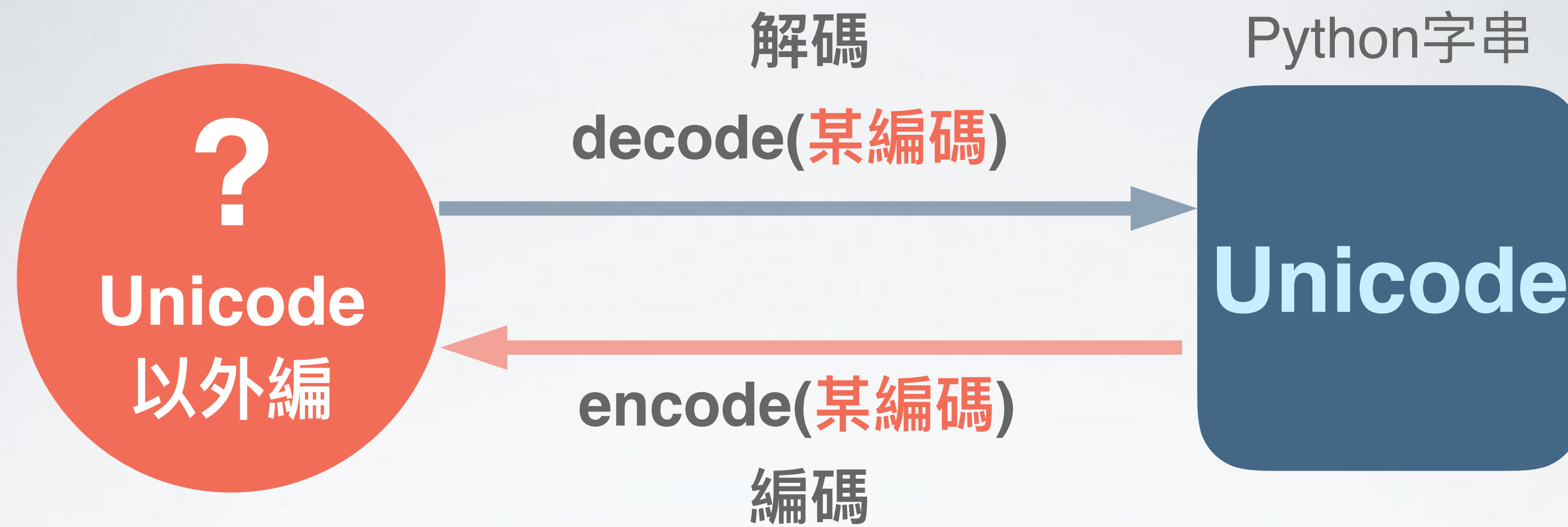
1. 某網站是用big5編碼(str1)，如何在Python中顯示？
2. 如何把big5編碼的str2轉換成utf-8？
3. 如何將Python字串str3輸出成utf-8編碼的文字檔？

str1.???

str2.???

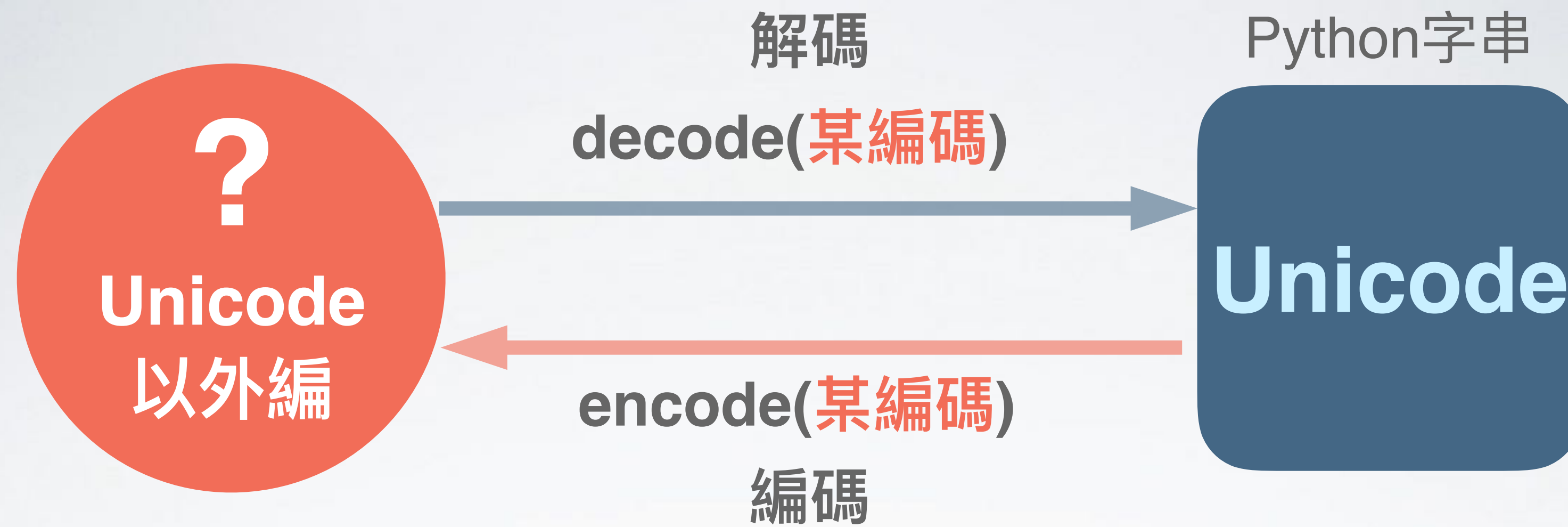
str3.???





## 情境練習題 #1

1. 某網站是用big5編碼(str1)，如何在Python中顯示？ `str1.decode("big5")`
2. 如何把big5編碼的str2轉換成utf-8？ `str2.decode("big5").encode("utf-8")`
3. 如何將Python字串str3輸出成utf-8編碼的某文字檔？ `str3.encode("utf-8")`



## 情境練習題 #2

1. 執行`str4.decode("utf-8")`，`str4`原本是什麼編碼？
2. 執行`str5.encode("utf-8")`，`str5`原本是什麼編碼？
3. 執行`str6.decode("gb2312").encode("utf-8")`，`str6`原本是什麼編碼？



使用Python第三方套件

## Requests來獲取網路資源

- 是 Python 裡大名鼎鼎的一個網路庫套件，其設計哲學是「為人類而設計」，所以他提供的功能更為人性化
- 提供一個比起Urllib更為簡潔優雅的介面能接受多種不同的協議來獲得網路上的資料
- Requests 會自動解碼來自服務器的內容。大多數編碼都能被無縫地解碼成unicode

```
pip3 install requests
```

使用 **requests** 模組 請求頁面

```
import requests  
response = requests.get("https://  
www.ptt.cc/bbs/movie/index.html")
```

```
response.url  
response.text
```

# PyQuery

- “The API is as much as possible the similar to jQuery.”
- Python版的jQuery，完全依照原本jQuery的邏輯去實作的網頁解析器，語法與jQuery幾乎一模一樣

```
pip3 install pyquery
```

```
from pyquery import PyQuery as  
pq
```

```
doc = pq('<html></html>')
```

```
doc = pq(url='http://google.com/')
```

```
doc = pq(filename=檔案路徑)
```

```
doc
```

```
doc.html()
```

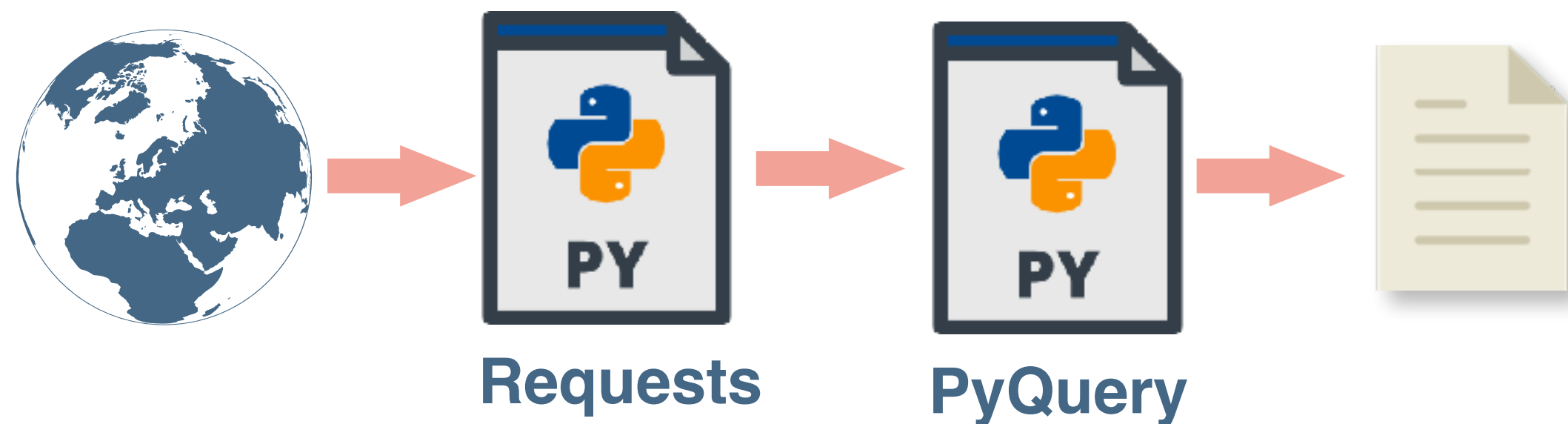
html: 印出當前選中元素的HTML所有內容

text: 印出當前選中元素的純文字內容



# 試試Requests+PyQuery

- Requests負責發送請求，取得並自動編碼網頁內容
- PyQuery負責解析HTML網頁內容



```
import requests
from pyquery import PyQuery as pq
```

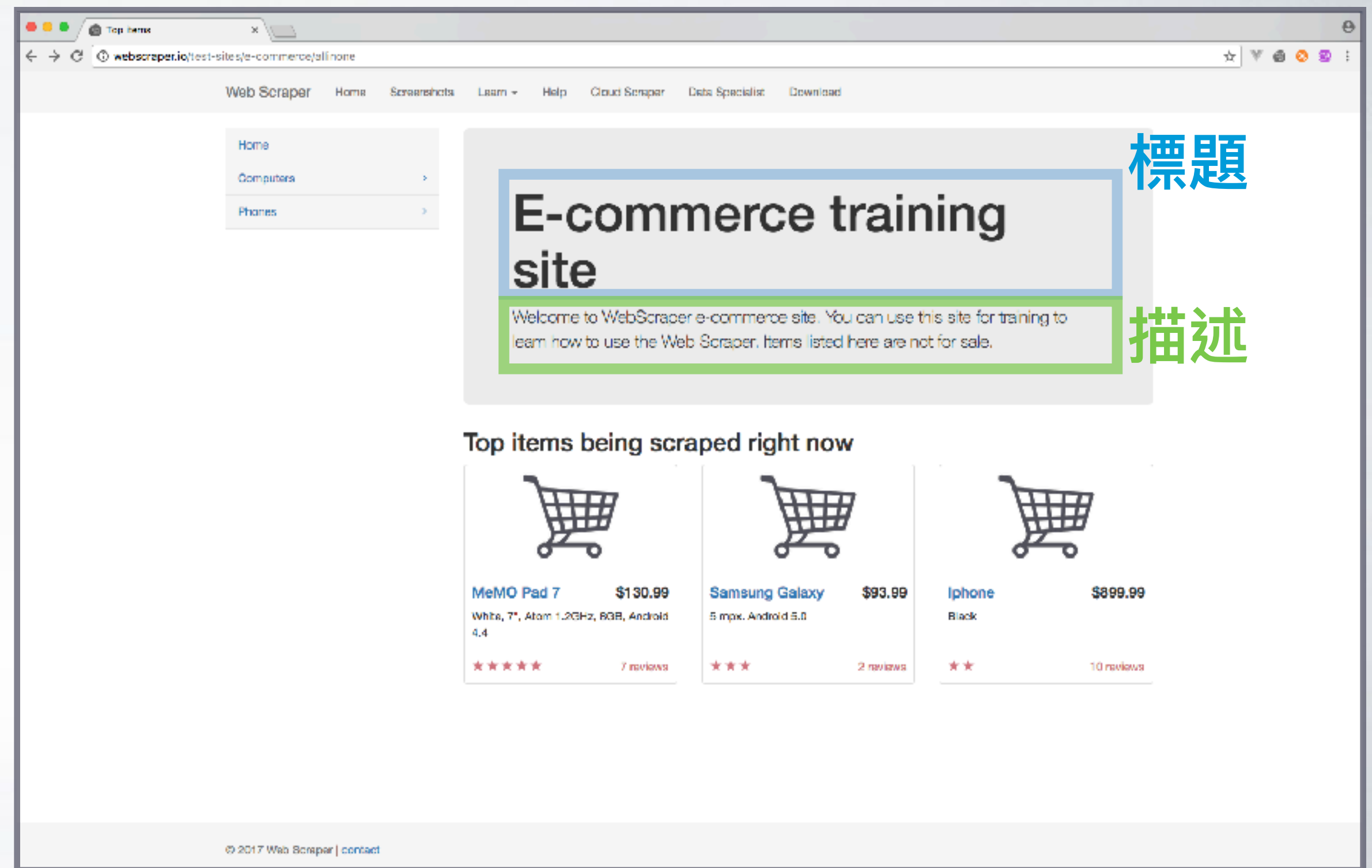
```
response = requests.get("https://www.ptt.cc/bbs/movie/index.html")
doc = pq(response.text)
```

```
params = {"key1": "value1", "key2": "value2"}
response = requests.get("https://www.ptt.cc/bbs/movie/index.html",
    params=params)  —————→ 帶參數
doc = pq(response.text)
response.url
```

# 練功坊#0-攫取

從較簡潔的網頁  
來進行爬蟲練習

- 1.先試著把網頁內容印出來
- 2.印出本頁**標題**與**描述**
- 3.印出本頁3個商品的名稱



[按我連結](#)



# 使用基本選擇器

## 用法：PyQuery物件(選擇器)

選擇器	用法舉例	選取元素舉例
標籤選擇器	<code>doc("h2")</code>	<code>&lt;h2&gt;Top items being scraped right now&lt;/h2&gt;</code>
屬性選擇器	<code>doc('[role="navigation"]')</code>	<code>&lt;div class="navbar navbar-default navbar-fixed-top" role="navigation"&gt;...&lt;/div&gt;</code>
類別(Class)選擇器	<code>doc('.price')</code>	<code>&lt;h4 class="pull-right price"&gt;\$148.99&lt;/h4&gt; &lt;h4 class="pull-right price"&gt;\$489.99&lt;/h4&gt; &lt;h4 class="pull-right price"&gt;\$416.99&lt;/h4&gt;</code>
ID選擇器	<code>doc("#side-menu")</code>	<code>&lt;ul class="nav" id="side-menu"&gt;...&lt;/ul&gt;</code>

用逗號，連接選擇器：`doc('.footer, #side-menu')`

# Class跟id哪裡不同？使用時機？

	差異性	使用時機
Class	Class是可被拿來被重覆使用的，可將同一群組或類別來進行設定	設定CSS樣式
id	ID是唯一性的、不可重覆的，每個元素有不同於他人的id	Javascript或D3.js其它的程式語言，找尋物件

**.circle**  
<div class="circle">....</div>

**#circle**  
<div id="circle">....</div>



# 選擇器 - 說文解字

標籤 選擇器	類別(Class) 選擇器	ID 選擇器	屬性 選擇器
--------	---------------	--------	--------

選DEF	?
選DE	?
選D	?
選ABCDEF	?

```
<a class="tall box">A</a>
<button class="tall">B</button>
<canvas class="tall">C</canvas>
<div class="tall box">D</div>
<div class="short box">E</div>
<div class="short" id="text">F</div>
```

# 選擇器 - 說文解字

標籤 選擇器	類別(Class) 選擇器	ID 選擇器	屬性 選擇器
--------	---------------	--------	--------

選DEF	div
選DE	div.box
選D	div.tall
選ABCDEF	.tall, .short

```
<a class="tall box">A</a>
<button class="tall">B</button>
<canvas class="tall">C</canvas>
<div class="tall box">D</div>
<div class="short box">E</div>
<div class="short" id="text">F</div>
```

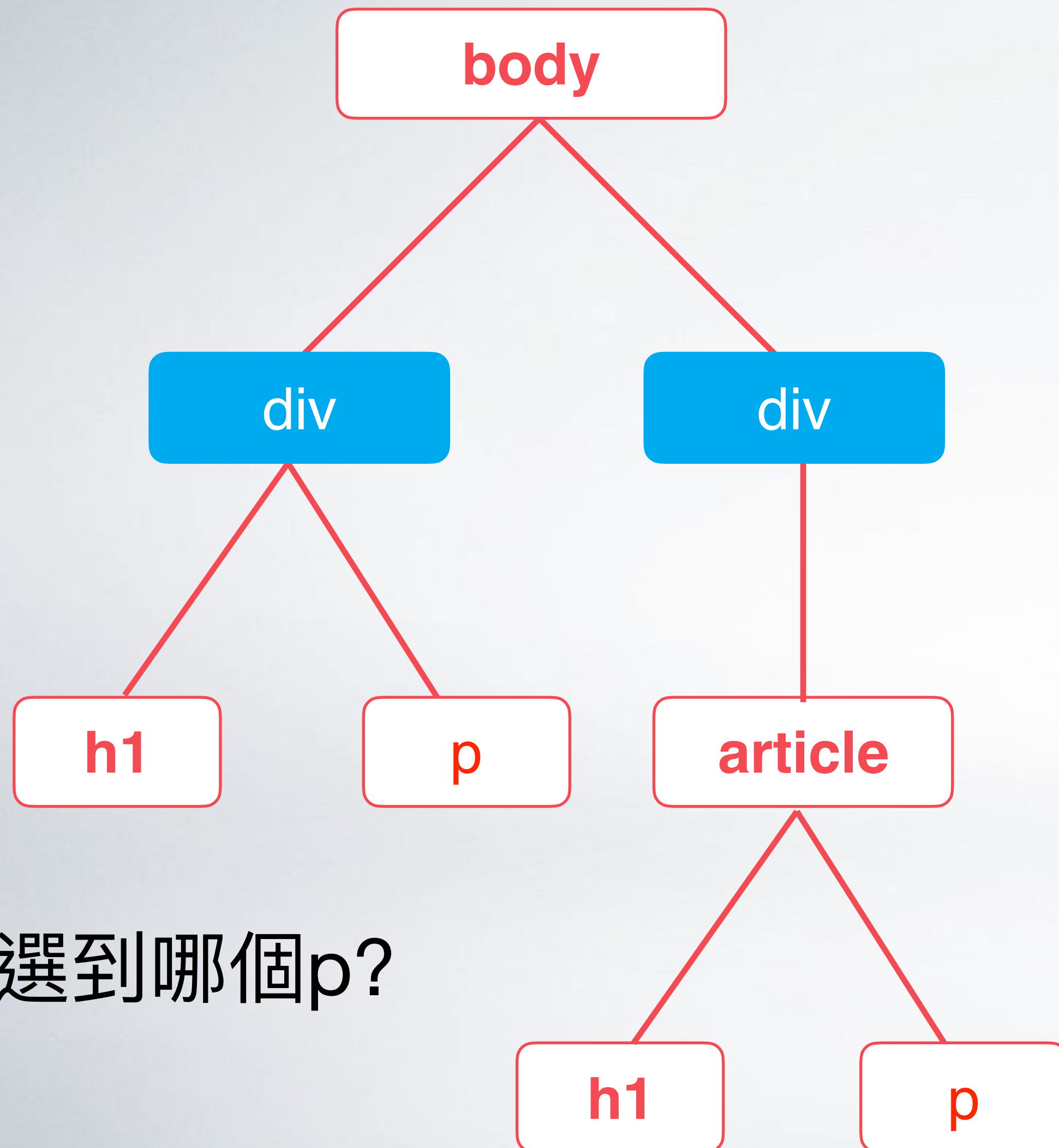


# 使用進階選擇器

用法：PyQuery物件(選擇器)

選擇器	用法舉例	選取元素舉例
孩子選擇器	<code>doc("#side-menu&gt;li")</code>	<code>&lt;li class="active"&gt;...&lt;/li&gt;</code> <code>&lt;li&gt;...&lt;/li&gt; &lt;li&gt;...&lt;/li&gt;</code>
子孫選擇器	<code>doc("#side-menu a")</code>	<code>&lt;a href="..."&gt;Home&lt;/a&gt;</code> <code>&lt;a href="..." class="category-link "&gt;&lt;/a&gt;</code> <code>&lt;a href="..." class="category-link "&gt;&lt;/a&gt;</code>
同層鄰接選擇器	<code>doc('img+div')</code>	幾個元素???
同層全體選擇器	<code>doc('img~div')</code>	幾個元素???
全選擇器	<code>doc('body&gt;*')</code>	幾個元素???

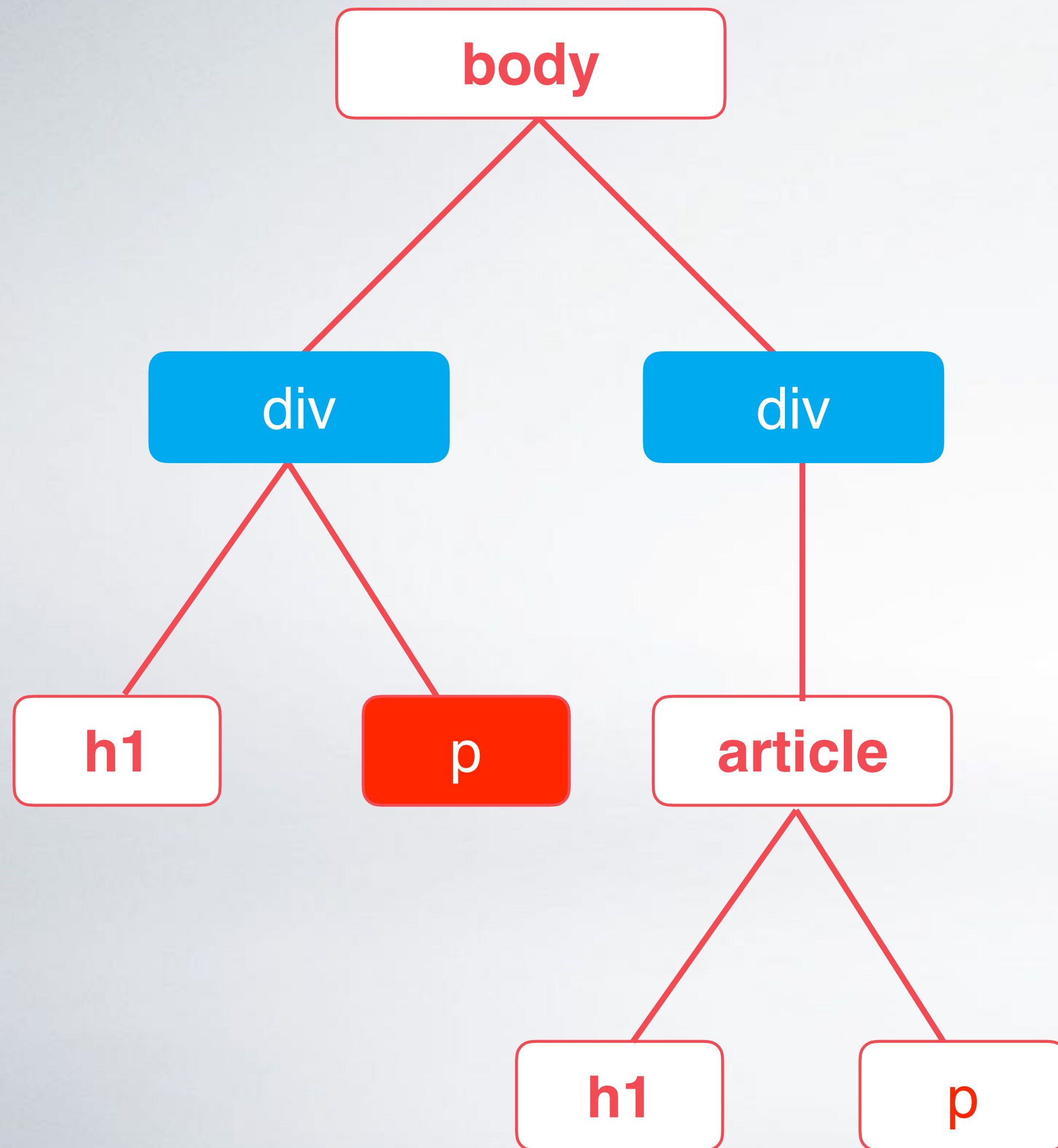
# 孩子選擇器 Child Selector



**div>p** = div的孩子中為p者  
= 選擇所有p，他在div下一層



# 孩子選擇器 Child Selector



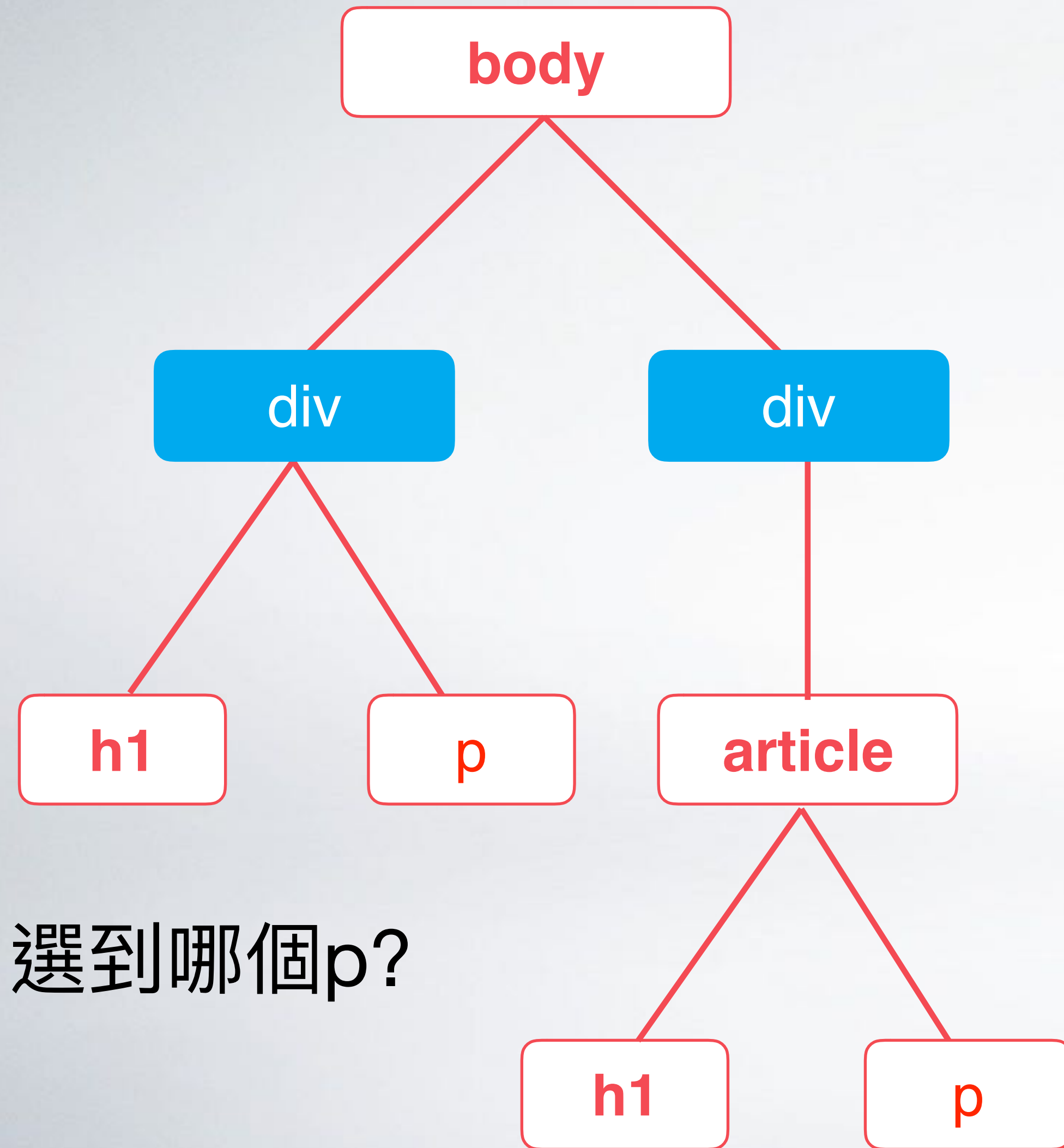
**div>p** = div的孩子中為p者  
= 選擇所有p，他在div下一層

# 子孫選擇器

## Descendant Selector

CSS樣式規則：

`div p` = `div`的子孫中為`p`者  
= 選擇所有`p`，他在`div`底下



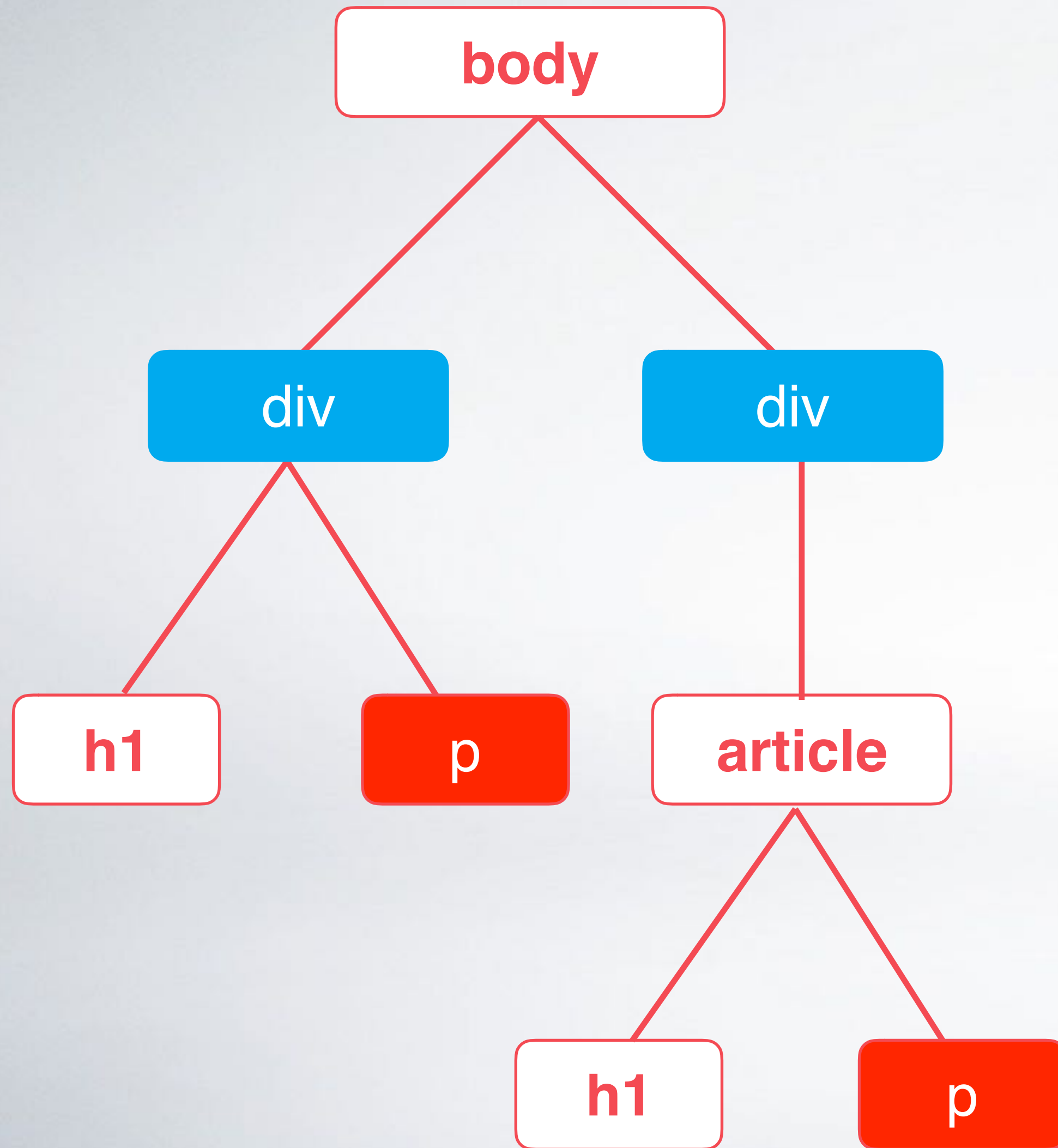


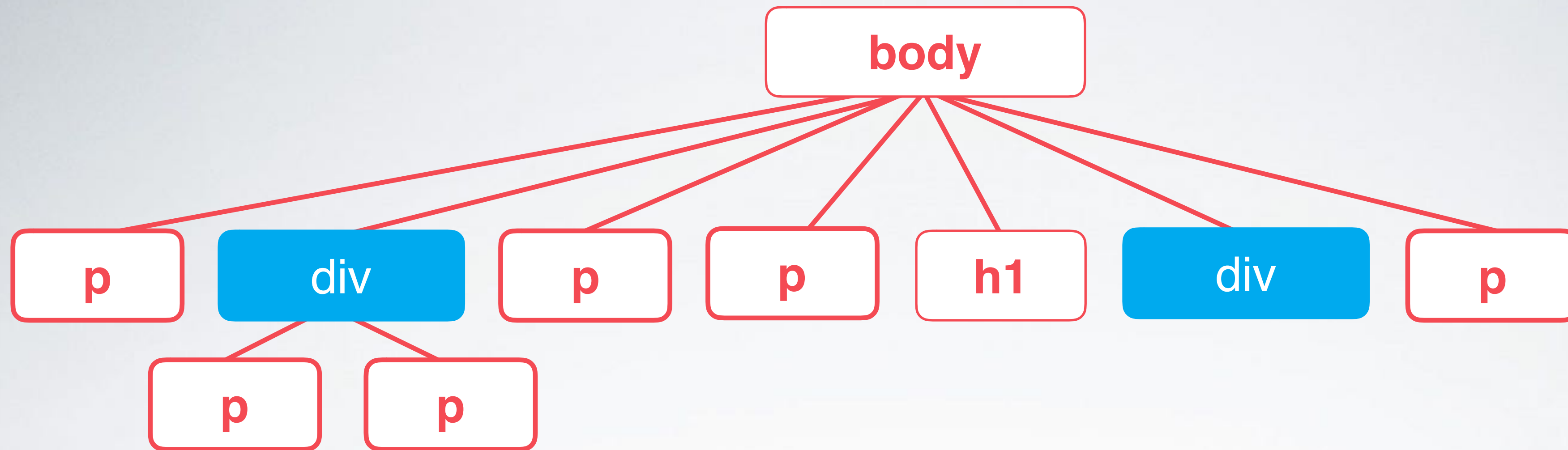
# 子孫選擇器

## Descendant Selector

CSS樣式規則：

**div p** = div的子孫中為p者  
= 選擇所有p，他在div底下

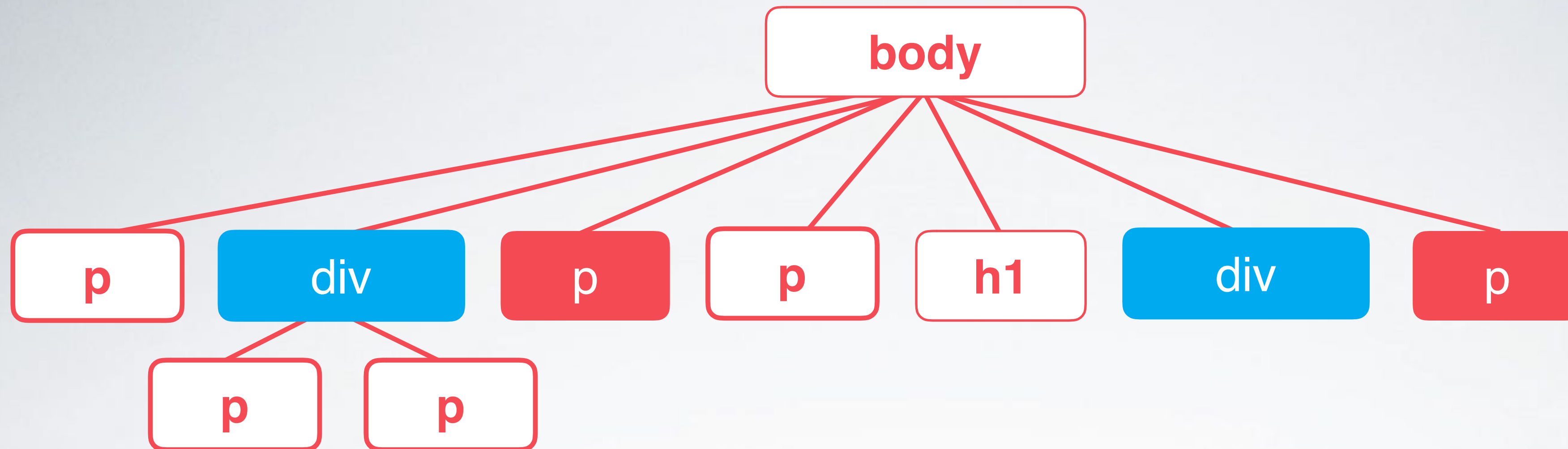




# 同層鄰接選擇器

## Adjacent Silbing Selector

**div+p** = 緊接在div後為p者  
= 選擇所有p，他前一個是div

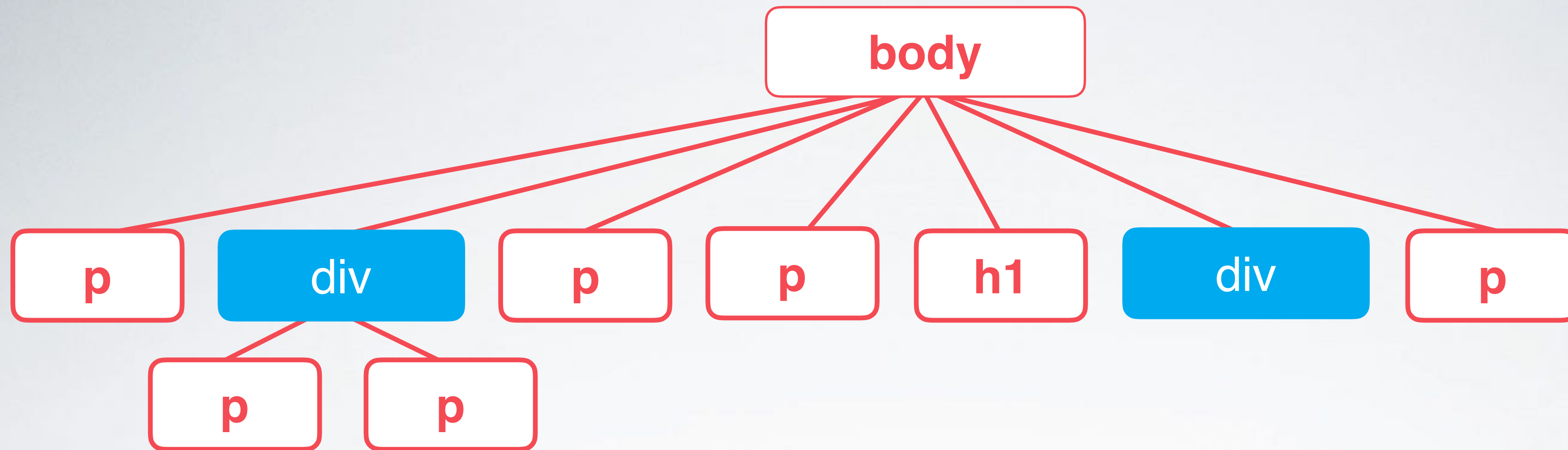


# 同層鄰接選擇器

## Adjacent Silbing Selector

**div+p** = 緊接在div後為p者  
= 選擇所有p，他前一個是div

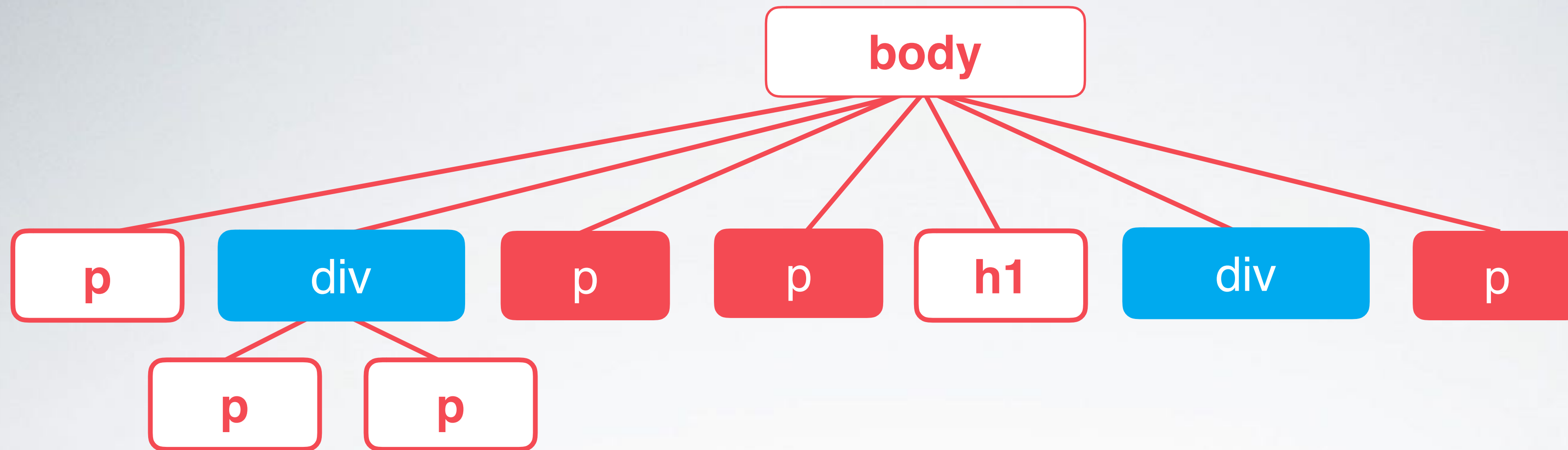




## 同層全體選擇器

### General Sibling Selector

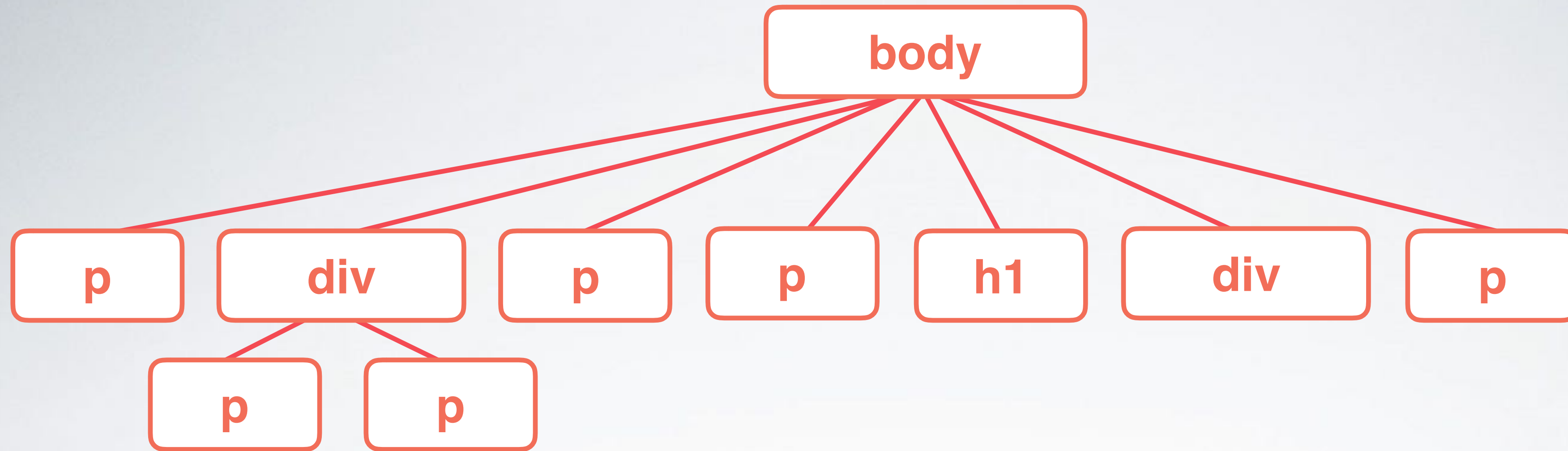
**div~p** = 在div後為p者  
= 選擇所有p，他前面有div



## 同層全體選擇器

### General Sibling Selector

**div~p** = 在div後為p者  
= 選擇所有p，他前面有div

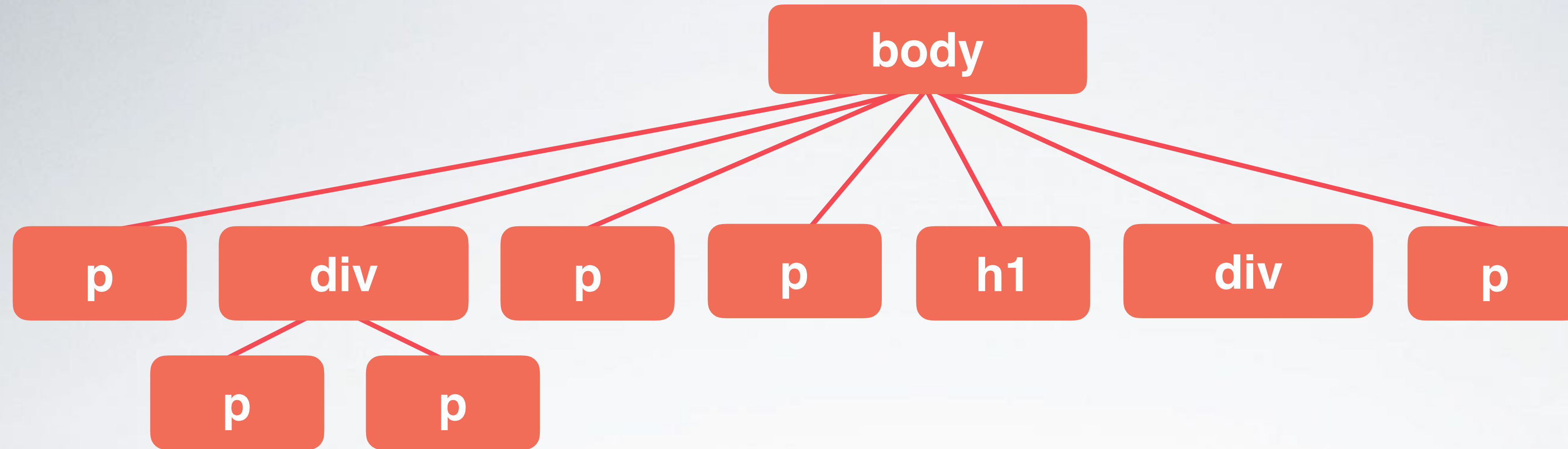


# 全選擇器

## Universal Selector

\* 代表任意元素  
選擇 任意元素

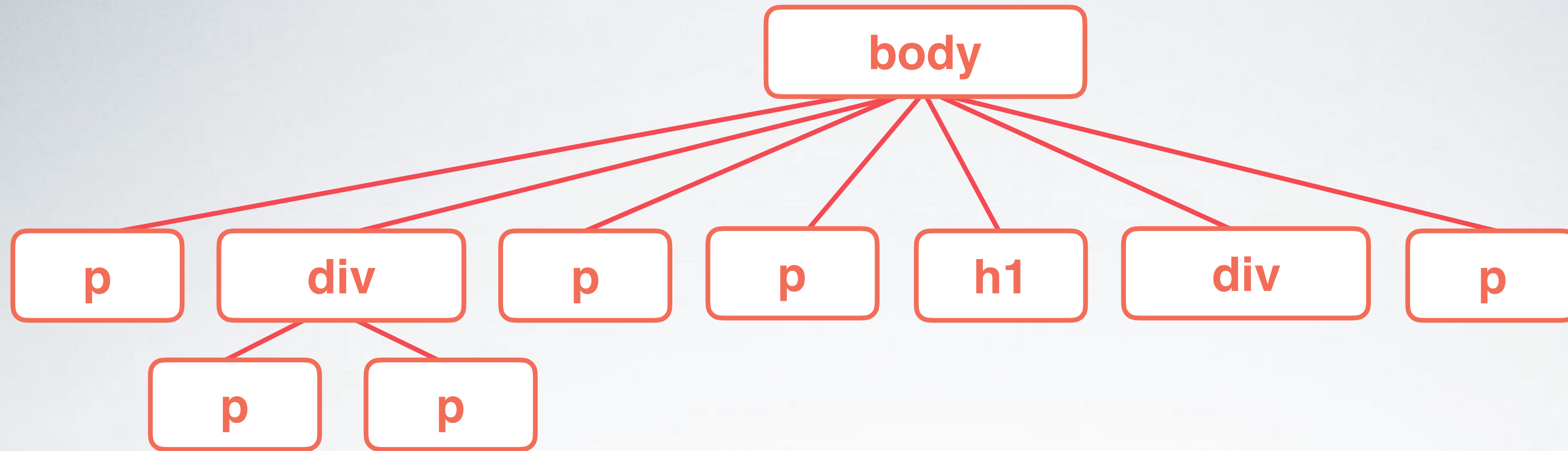




# 全選擇器

## Universal Selector

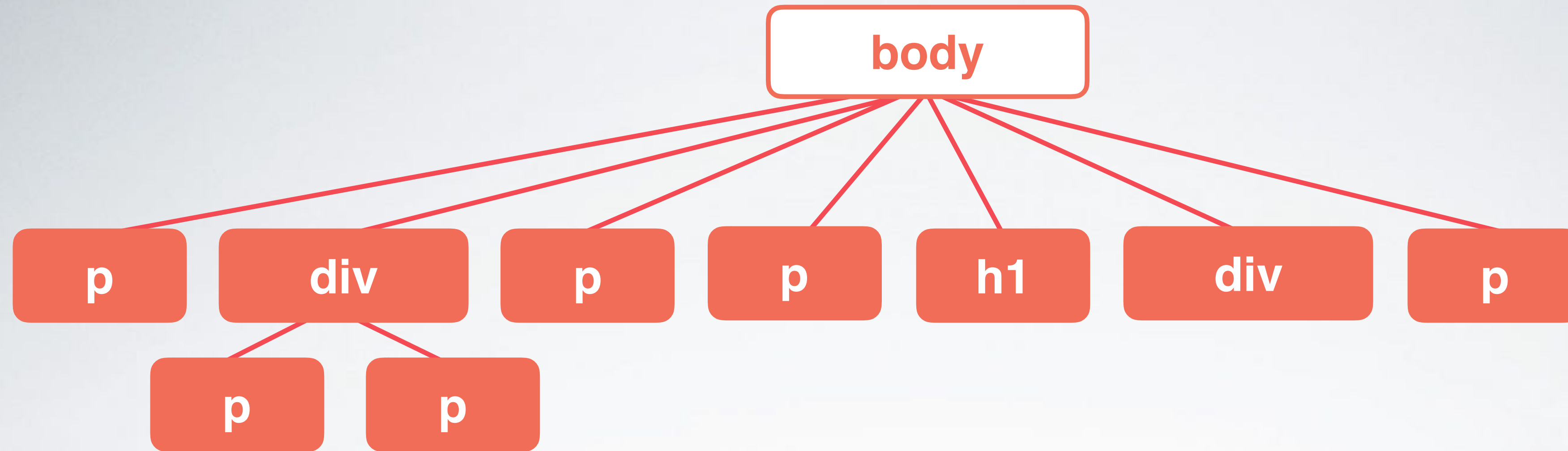
\* 代表任意元素  
選擇 任意元素



全選擇器

Universal Selector

**body \***

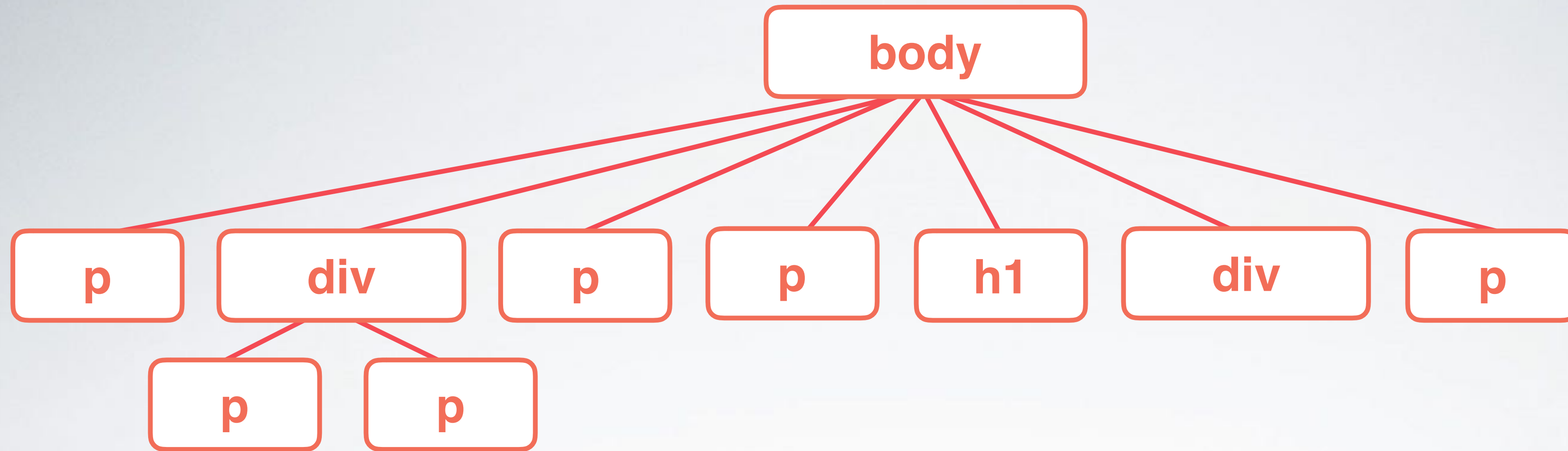


# 全選擇器

## Universal Selector

`body *` = 選擇 `body` 的子孫們

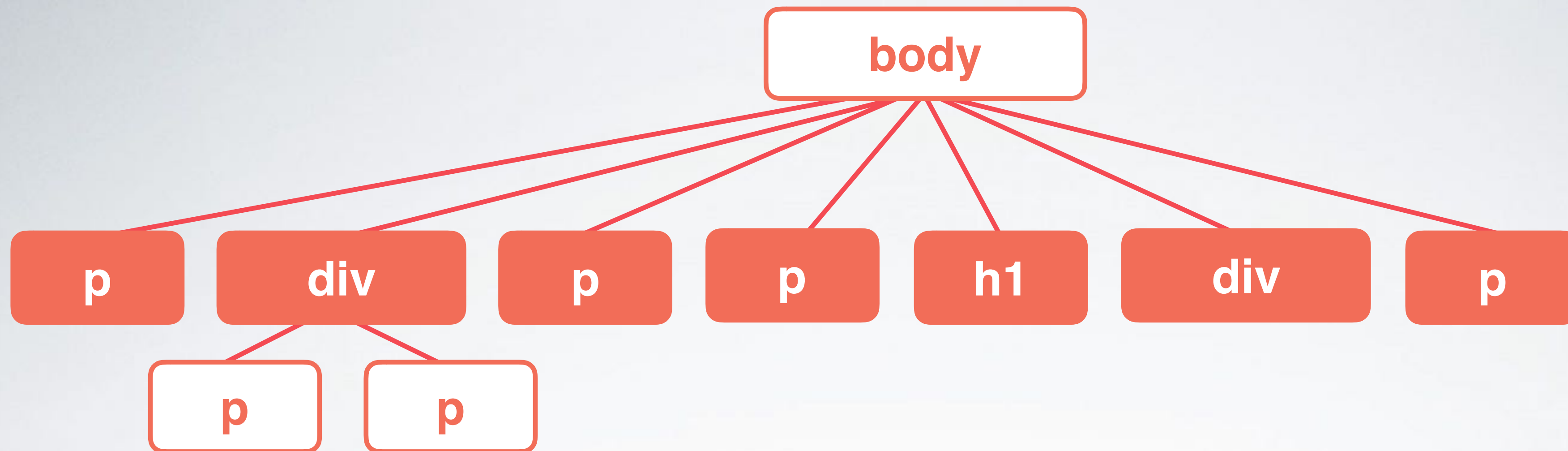




全選擇器

Universal Selector

`body>*`



# 全選擇器

## Universal Selector

`body>*` = 選擇 body 的孩子們

# N孩後，選

老大 :`first-child`

老么 :`last-child`

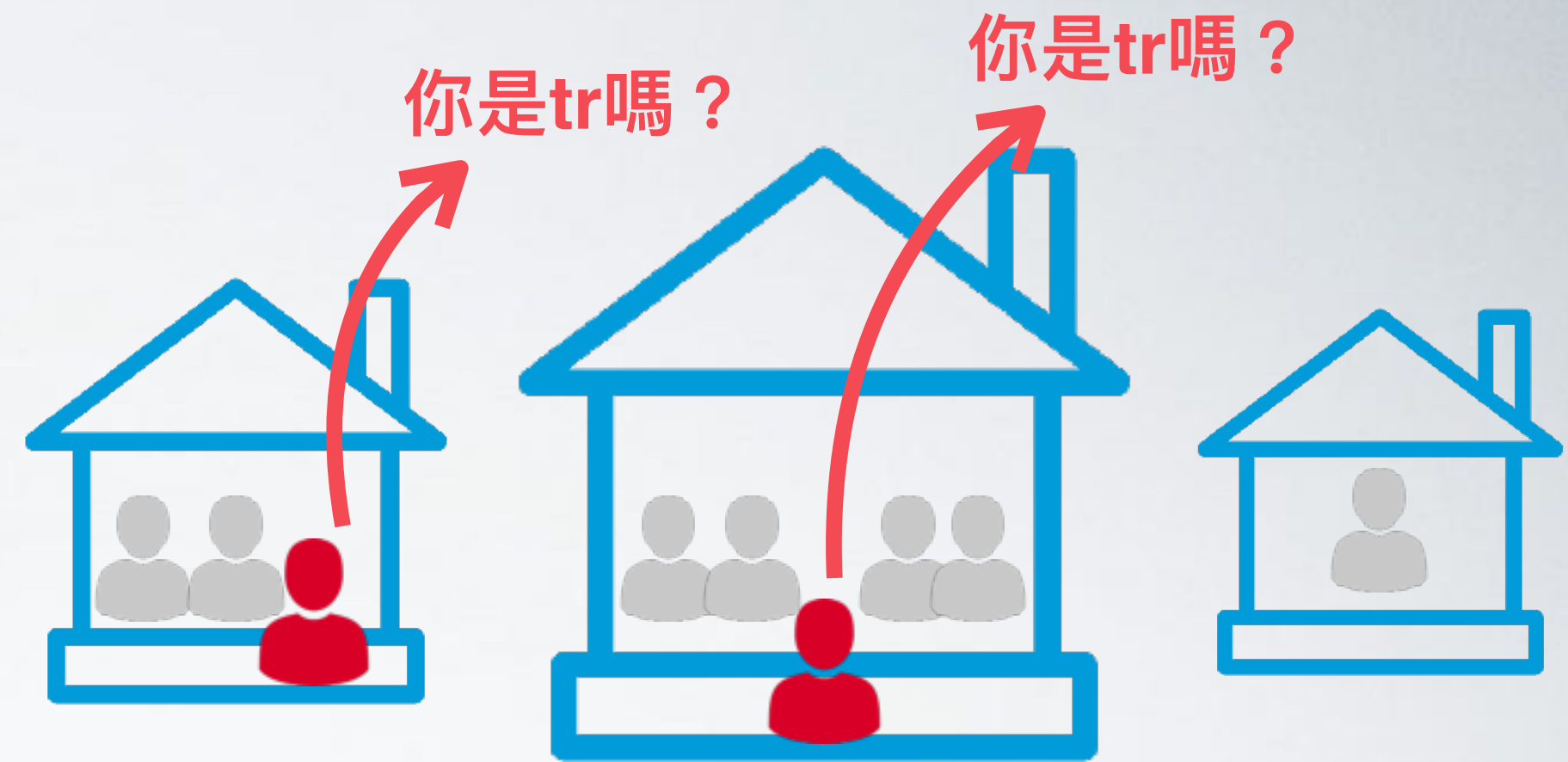
獨子 :`only-child`

老N :`nth-child(數字)`

倒著數老N :`nth-last-child(數字)`



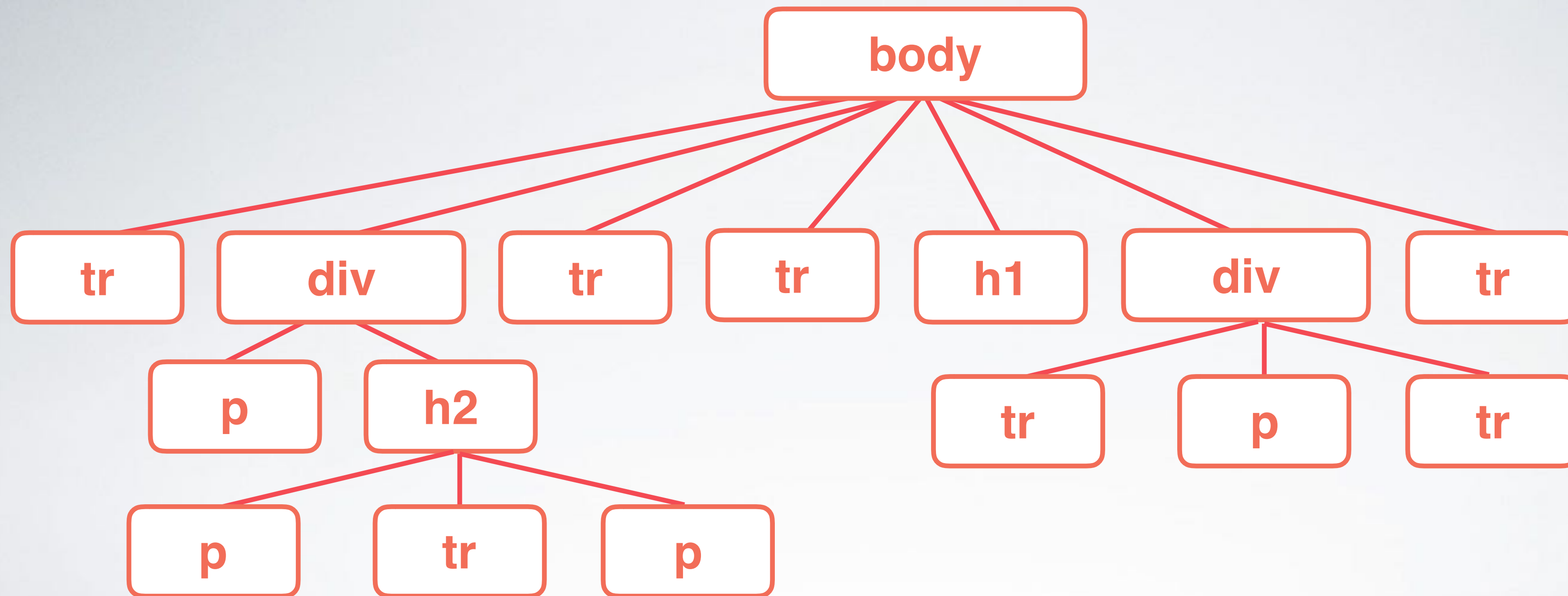
第3個出列!!



**tr: `nth-child(3)`**

選擇每個標籤裡的第3個孩子，而且是tr





## N孩後，選 —

老大 :`first-child`

老么 :`last-child`

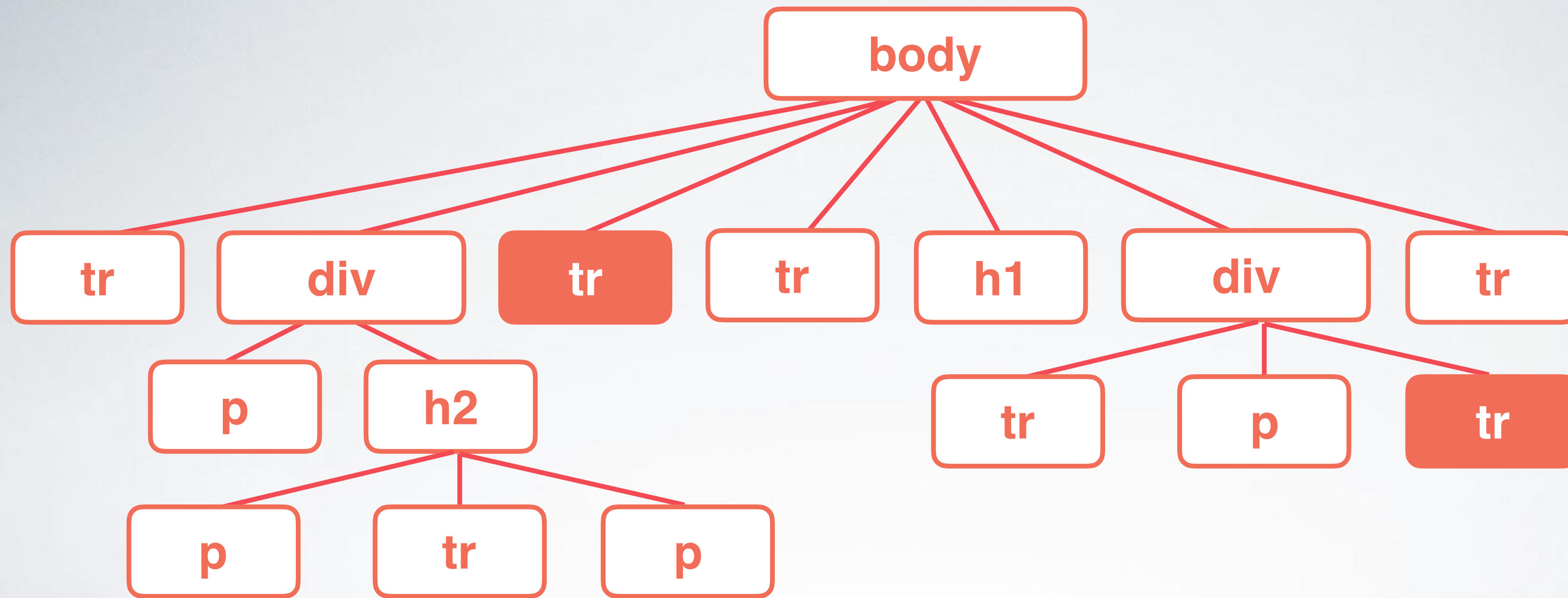
獨子 :`only-child`

老N :`nth-child(數字)`

倒著數老N :`nth-last-child(數字)`

**`tr: nth-child(3)`**

選擇每個標籤裡的第3個孩子，而且是tr



## N孩後，選 —

老大 :`first-child`

老么 :`last-child`

獨子 :`only-child`

老N :`nth-child(數字)`

倒著數老N :`nth-last-child(數字)`

**tr: `nth-child(3)`**

**選擇每個標籤裡的第3個孩子，而且是tr**

# 進階用法

## `:nth-child(an+b)`

**`tr: nth-child(an+b)`**  
選擇第 $an+b$ 個tr的所有子元素

CSS ▼

```
tr{  
  background: #ccc;  
}  
tr:nth-child(2n+1) {  
  background: lightgreen;  
}
```

Output

食物名稱	熱量(kcal)
雞腿飯	700
炒麵	400
炒米粉	400
牛肉麵	470
起司三明治	200

**Q `:nth-last-child(2n+1)`**



# 歸類後，選N \_\_\_\_

老大 :first-of-type

老么 :last-of-type

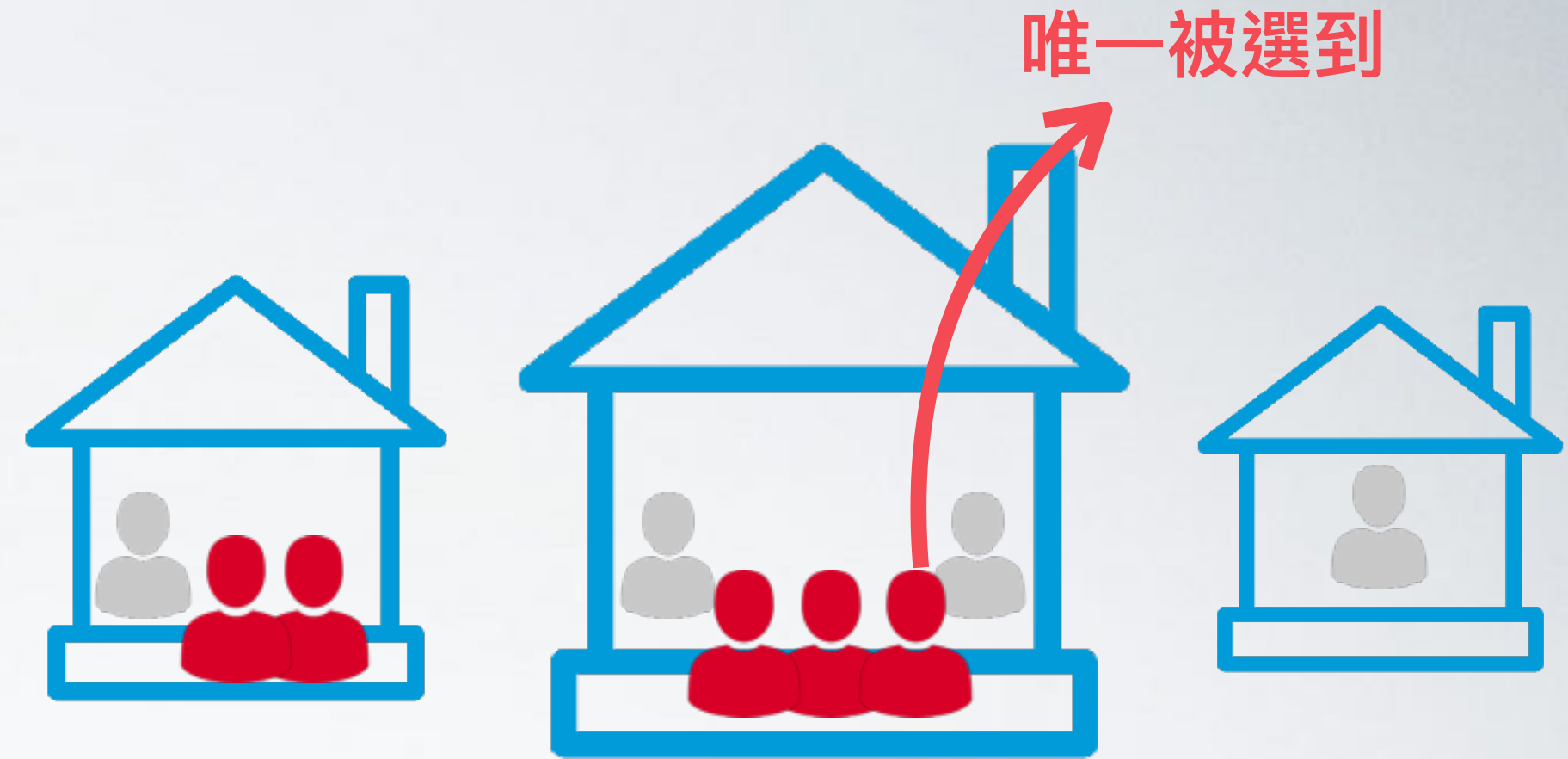
獨子 :only-of-type

老N :nth-of-type (數字)

倒著數老N :nth-last-of-type(數字)

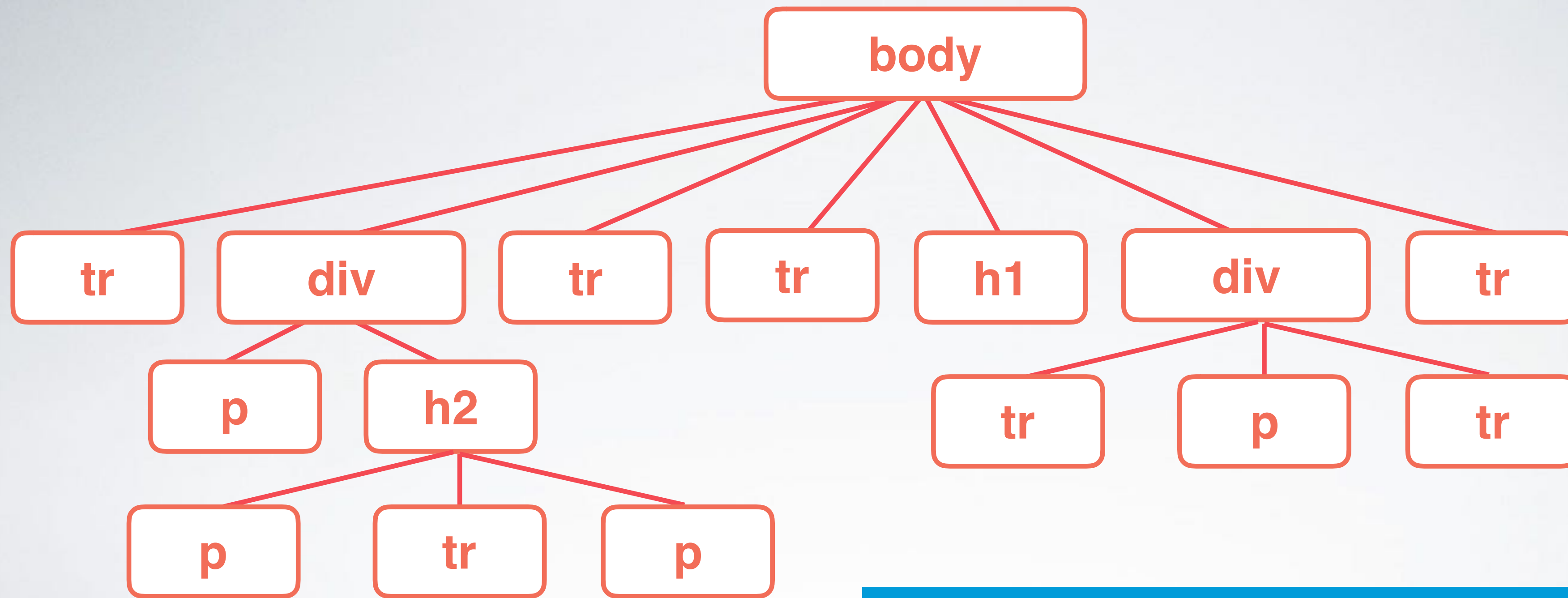


tr出列！！



**tr: nth-of-type(3)**

選擇每個標籤裡的tr，第3個



## 歸類後，選N \_\_

老大 :`first-of-type`

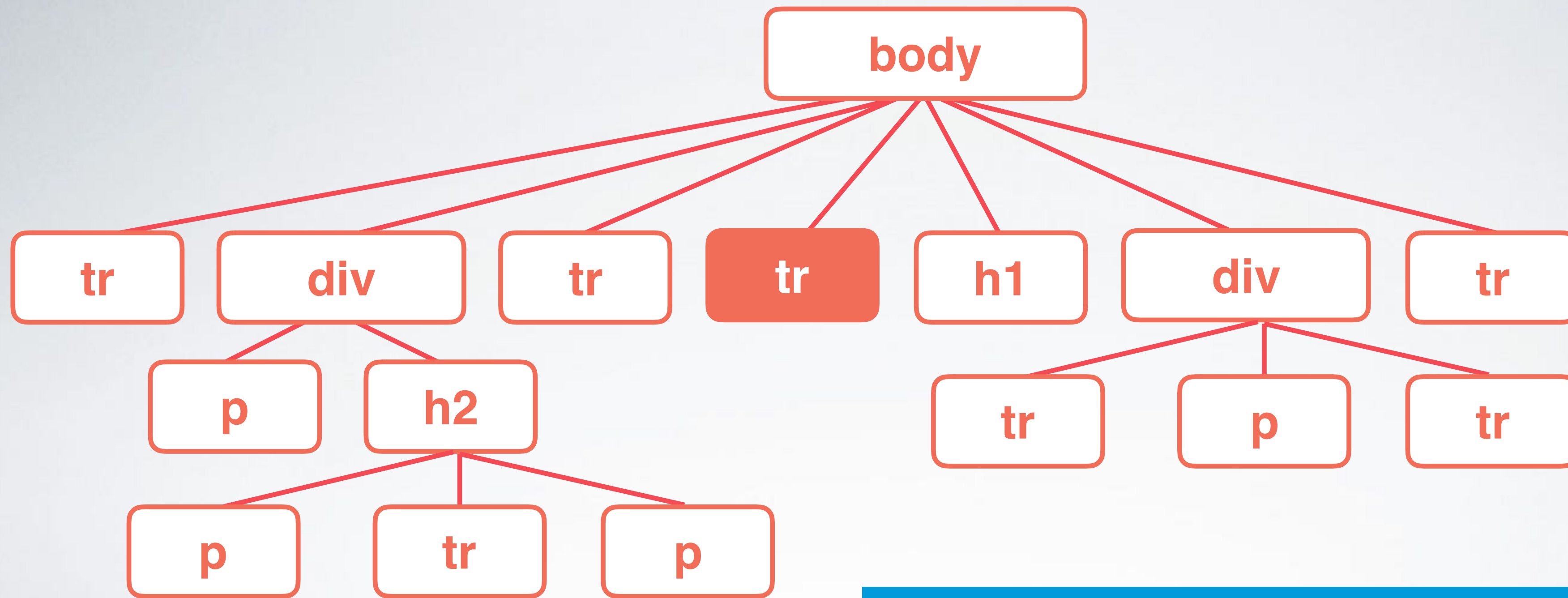
老么 :`last-of-type`

獨子 :`only-of-type`

老N :`nth-of-type` (數字)

倒著數老N :`nth-last-of-type`(數字)

**`tr: nth-of-type(3)`**  
選擇每個標籤裡的tr，第3個



## 歸類後，選N \_\_

老大 :`first-of-type`

老么 :`last-of-type`

獨子 :`only-of-type`

老N :`nth-of-type` (數字)

倒著數老N :`nth-last-of-type`(數字)

**`tr: nth-of-type(3)`**  
選擇每個標籤裡的tr，第3個



# 哪裡不一樣：nth-child V.S. nth-of-type

N孩後，選 \_\_：

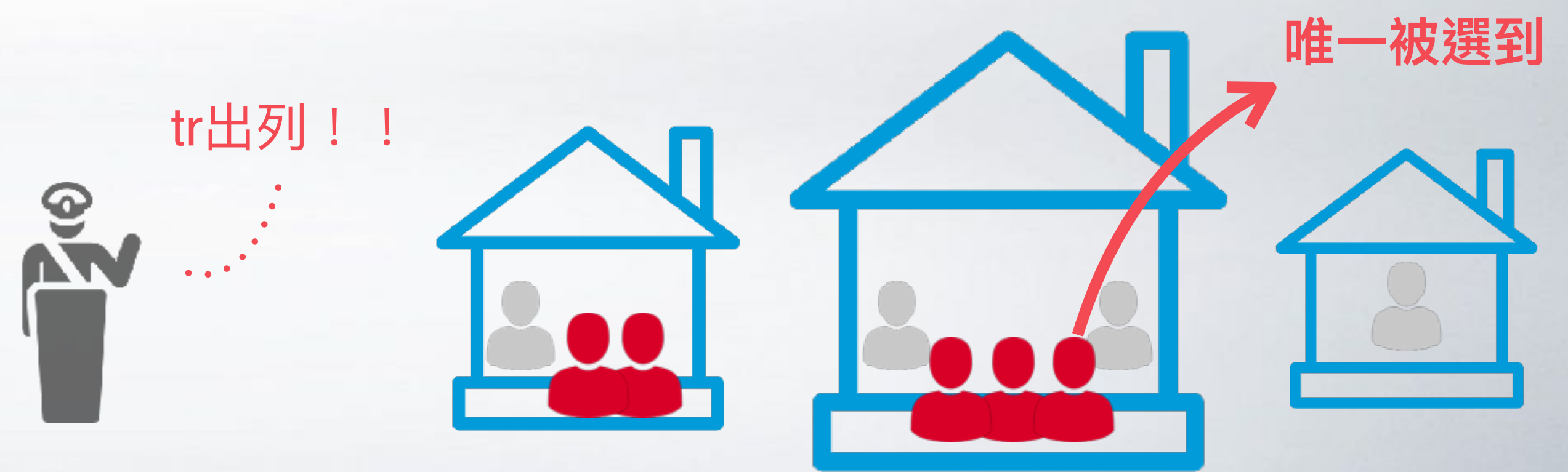
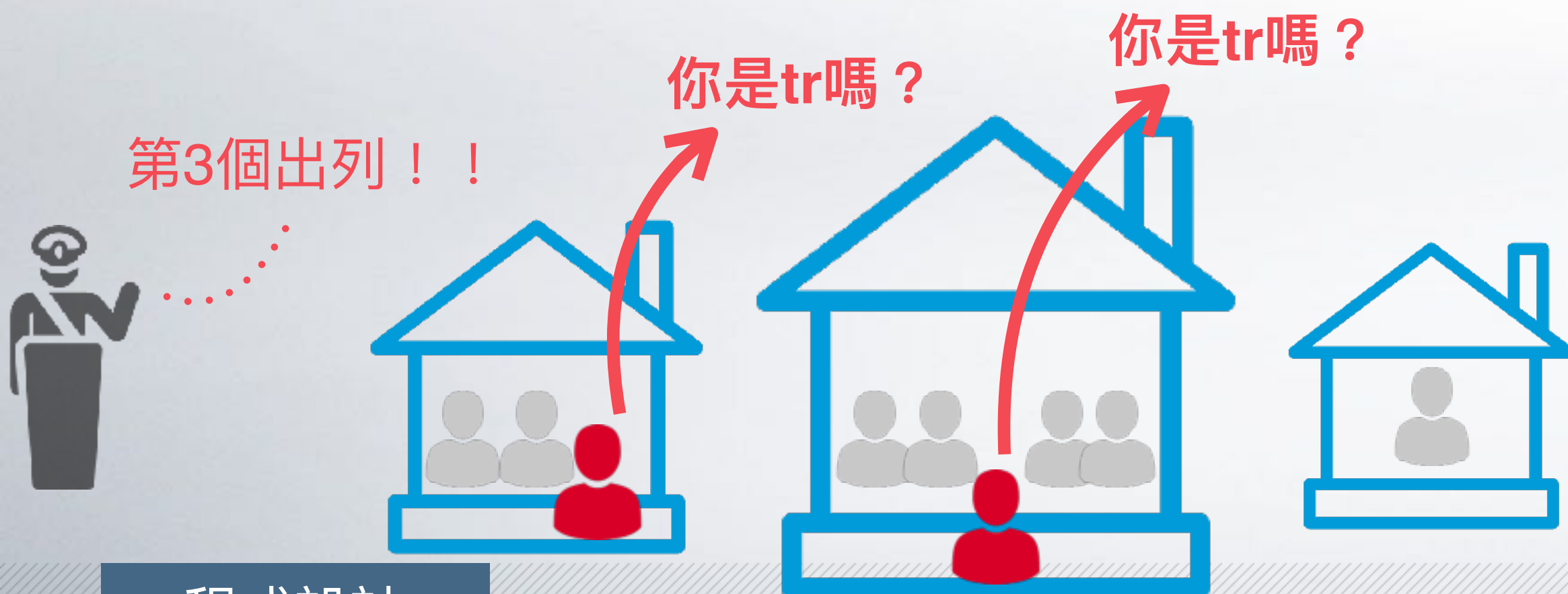
**tr: nth-child(3)**

選擇每個標籤裡的第3個孩子，而且是tr

歸類後，選 N\_\_：

**tr: nth-of-type(3)**

選擇每個標籤裡的tr，第3個



# 練功坊#1-遍歷

爬取 Computers與Phones類別內所有分頁(不包含各大類別3個預覽商品)的商品資訊含以下欄位：類別、產品名稱、描述、價格、星等、評論數

```
productList
[{'Category': 'Computers / Laptops',
  'Description': '15.6", Pentium N3520 2.16GHz, 4GB, 500GB, Linux',
  'Name': 'Aspire E1-510',
  'Price': '$306.99',
  'Reviews': 9,
  'Stars': 4},
 {'Category': 'Computers / Laptops',
  'Description': '15.6", AMD E2-3800 1.3GHz, 4GB, 500GB, Windows 8.1',
  'Name': 'Packard 255 G2',
  'Price': '$416.99',
  'Reviews': 6,
  'Stars': 2},
 {'Category': 'Computers / Laptops',
  'Description': '15.6", Core i5-4210U, 4GB, 500GB, Windows 8.1',
  'Name': 'HP 250 G3',
  'Price': '$520.99',
  'Reviews': 11,
  'Stars': 4},
 {'Category': 'Computers / Laptops',
  'Description': '15.6", Core i5-4200U, 4GB, 750GB, Windows 8.1',
  'Name': 'HP 350 G1',
  'Price': '$577.99',
  'Reviews': 11,
  'Stars': 4},
 {'Category': 'Computers / Laptops',
  'Description': '15.6", Core i5-4200U, 8GB, 1TB, Radeon HD 8670M 2GB, Windows 8.1',
  'Name': 'Aspire E1-572G',
  'Price': '$581.99',
  'Reviews': 8,
  'Stars': 4},
 {'Category': 'Computers / Laptops',
  'Description': '15.6", Core i5-4200U, 8GB, 750GB, Windows 8.1',
  'Name': 'Pavilion',
  'Price': '$609.99',
  'Reviews': 3,
  'Stars': 2},
 {'Category': 'Computers / Laptops',
  'Description': '14", Core i5 2.6GHz, 4GB, 500GB, Win7 Pro 64bit',
  'Name': 'ProBook',
  'Price': '$739.99',
  'Reviews': 0,
  'Stars': 0},
 {'Category': 'Computers / Laptops',
  'Description': 'Moon Silver, 15.6", Core i7-4510U, 8GB, 1TB, Radeon HD R7 M265 2GB',
  'Name': 'Inspiron 15',
  'Price': '$745.99',
  'Reviews': 0,
  'Stars': 0},
 {'Category': 'Computers / Laptops',
  'Description': '12.5" Touch, Core i3-4010U, 4GB, 500GB + 16GB SSD Cache',
  'Name': 'ThinkPad Yoga',
  'Price': '$1033.99',
  'Reviews': 0,
  'Stars': 0}]
```

類別

爬取目標

[按我連結](#)



# 細說PyQuery

## 基本函式

函式	說明
<code>.eq(索引值)</code>	根據索引號(從0)指定pyquery物件中某一元素
<code>.children([選擇器])</code>	獲取pyquery物件之所有(或符合條件的)子元素們
<code>.parent()</code>	獲取pyquery物件之父元素
<code>.siblings([選擇器])</code>	獲取pyquery物件之相鄰元素(或符合條件的相鄰元素)
<code>.next()</code>	獲取pyquery物件之下一個元素
<code>.nextAll()</code>	獲取pyquery物件之後面全部元素



# 細說PyQuery

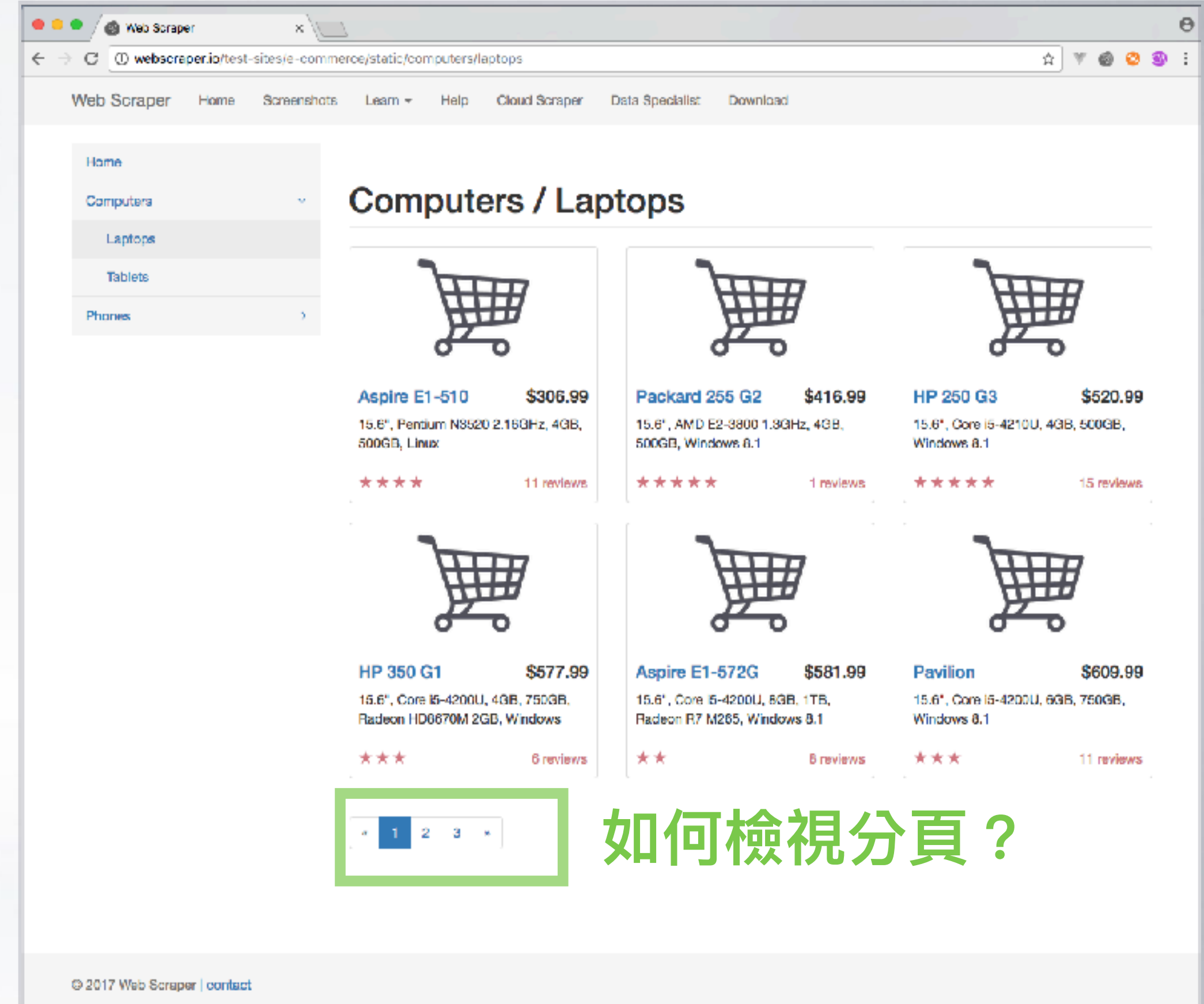
## 基本函式

函式	說明
<code>.find(選擇器)</code>	尋找pyquery物件裡所有指定的元素
<code>.filter(選擇器)</code>	只顯示符合條件的pyquery物件
<code>.attr(屬性, [新值])</code>	獲取、修改元素的屬性與值
<code>.hasClass(名稱)</code>	判斷是否包含指定的class，返回True/False
<code>.not_(選擇器)</code>	回傳不符合選擇器條件之元素
<code>for i in pyquery物件.items([選擇器]):</code>	遍歷pyquery物件中的元素(或指定的子元素)
<code>pyquery物</code>	針對<a>將其路徑顯示改為絕對路徑

# 練功坊#2-分頁

繼續沿用剛剛程式，現在修改一下初始網站，網站裡有一部分商品被藏在分頁中

```
productList
[{'Category': 'Computers / Laptops',
  'Description': '15.6", Pentium N3520 2.16GHz, 4GB, 500GB, Linux',
  'Name': 'Aspire E1-510',
  'Price': '$306.99',
  'Reviews': 9,
  'Stars': 4},
 {'Category': 'Computers / Laptops',
  'Description': '15.6", AMD E2-3800 1.3GHz, 4GB, 500GB, Windows 8.1',
  'Name': 'Packard 255 G2',
  'Price': '$416.99',
  'Reviews': 6,
  'Stars': 2},
 {'Category': 'Computers / Laptops',
  'Description': '15.6", Core i5-4210U, 4GB, 500GB, Windows 8.1',
  'Name': 'HP 250 G3',
  'Price': '$520.99',
  'Reviews': 11,
  'Stars': 4},
 {'Category': 'Computers / Laptops',
  'Description': '15.6", Core i5-4200U, 4GB, 750GB, Windows 8.1',
  'Name': 'HP 350 G1',
  'Price': '$577.99',
  'Reviews': 6,
  'Stars': 2},
 {'Category': 'Computers / Laptops',
  'Description': '15.6", Core i5-4200U, 8GB, 1TB, Radeon R7 M265, Windows 8.1',
  'Name': 'Aspire E1-572G',
  'Price': '$581.99',
  'Reviews': 8,
  'Stars': 2},
 {'Category': 'Computers / Laptops',
  'Description': '15.6", Core i5-4200U, 8GB, 750GB, Windows 8.1',
  'Name': 'Pavilion',
  'Price': '$609.99',
  'Reviews': 11,
  'Stars': 3}]
```



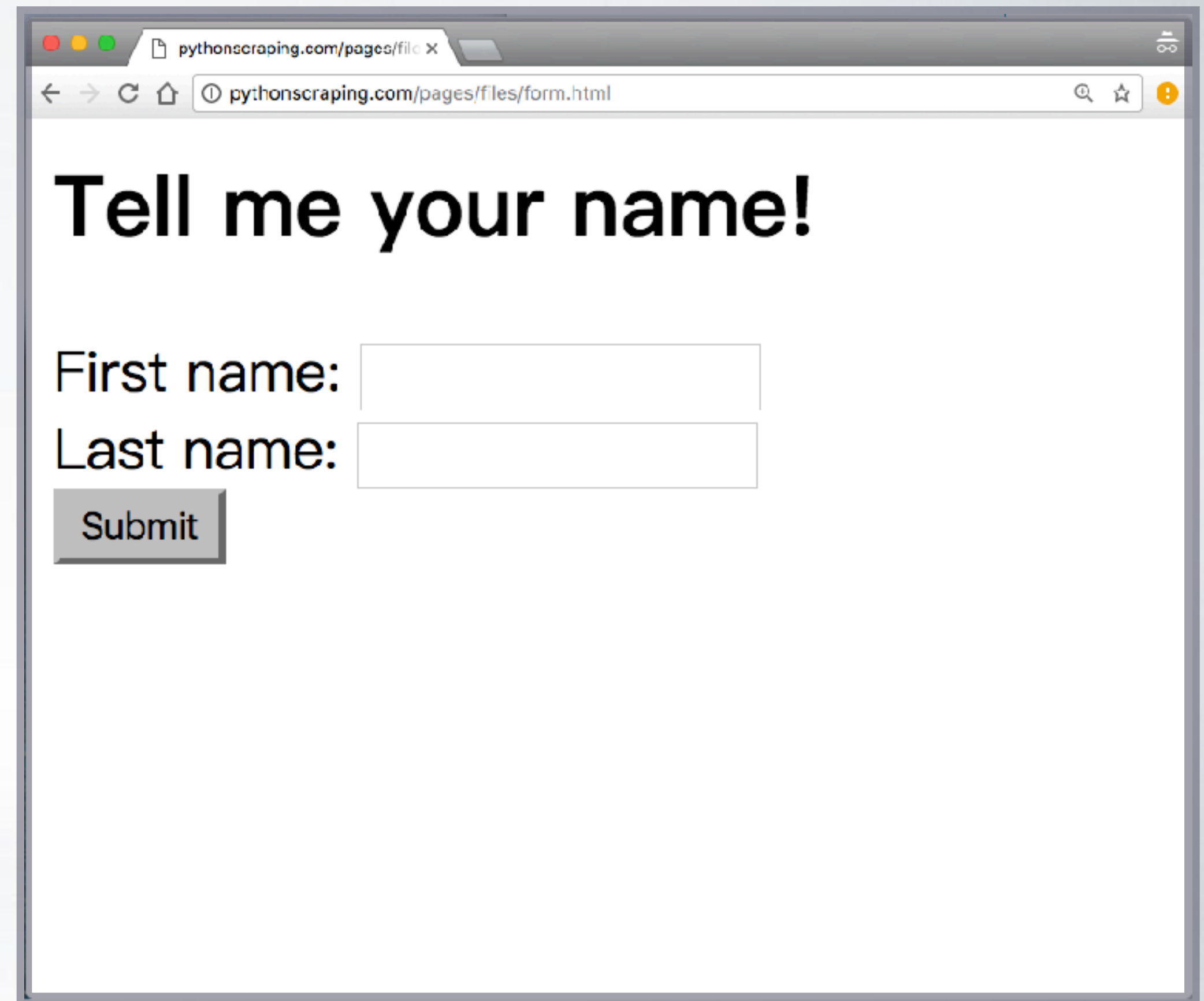
如何檢視分頁？

按我連結



# 練功坊#3-提交表單

- 表單<form>基本上就是讓使用者傳送POST請求，來將填寫資料傳送至伺服器中，並得到回應結果。
- 我們也可以用requests模組中的post()來模擬使用者填寫表單並送出的動作
- 試著取得填寫表單後的網頁內容



The screenshot shows a web browser window with the address bar displaying 'pythonscraping.com/pages/files/form.html'. The page content includes the heading 'Tell me your name!' followed by two text input fields labeled 'First name:' and 'Last name:'. Below these fields is a 'Submit' button.

[按我連結](#)



如何使用

## Requests中的post函式

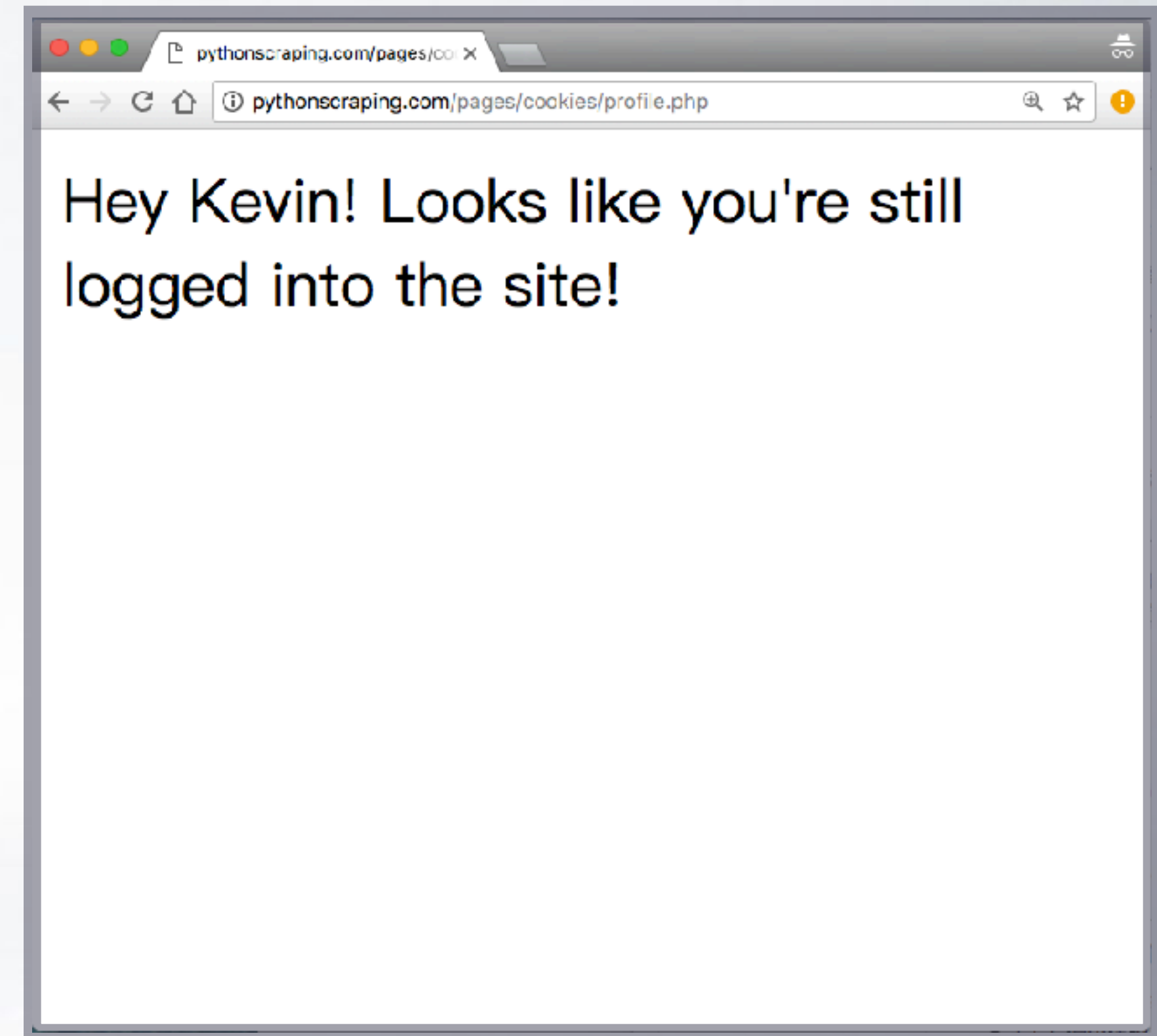
- 在使用者和伺服器之間進行請求時，兩種最常被用到的方法是：GET 和 POST。
  - GET：從指定的資源請求資料
  - POST：向指定的資源送出要被處理的資料

使用 **requests.post** 送出表單

```
import requests
params = {"firstname": "Kevin",
          "lastname": "Lin"}
res= requests.post(url=???,
data=params)
res.text
```

# 練功坊#4- 處理登入與Cookies

- **Cookie:** 是由你拜訪的網站儲存在你電腦裡的資料，裡面記載著您開啟的頁面記錄或是登入資訊
- 試著取得登入後如右的網頁內容
- 步驟：
  - 1.找尋登入頁面
  - 2.填寫並送出表單
  - 3.保存Cookies供後續瀏覽使用



[按我連結](#)

# 用Requests處理登入與Cookies

- Cookie對網頁開發者來說是很棒的工具，但對爬蟲來說就比較棘手
- 透過requests，你可以將爬蟲在某網站上的操作而產生的Cookies保留下來，以供後續瀏覽同個網站持續使用

## 使用 `.cookies` 取得Cookies內容

```
import requests
from pyquery import PyQuery as pq

.....

params = {"username": "Kevin", "password":
"password"}
res = requests.post("http://pythonscrapping.com/
pages/cookies/welcome.php", data=params)
res.cookies.get_dict()

profileRes = requests.get("http://
pythonscrapping.com/pages/cookies/profile.php",
cookies=res.cookies)
profileDoc = pq(profileRes.text)
profileDoc.html()
```



# 實戰演練#1

## 來爬PTT八卦版

- 如何解決「詢問是否18歲按鈕」問題？-> 用Cookies
- 爬取末三頁的所有標題與作者



<https://www.ptt.cc/bbs/Gossiping/index.html>

# 實戰演練#2

## 來爬Yahoo購物中心

- 爬取此站所有類別中的所有商品資訊，包含：商品名稱、價格



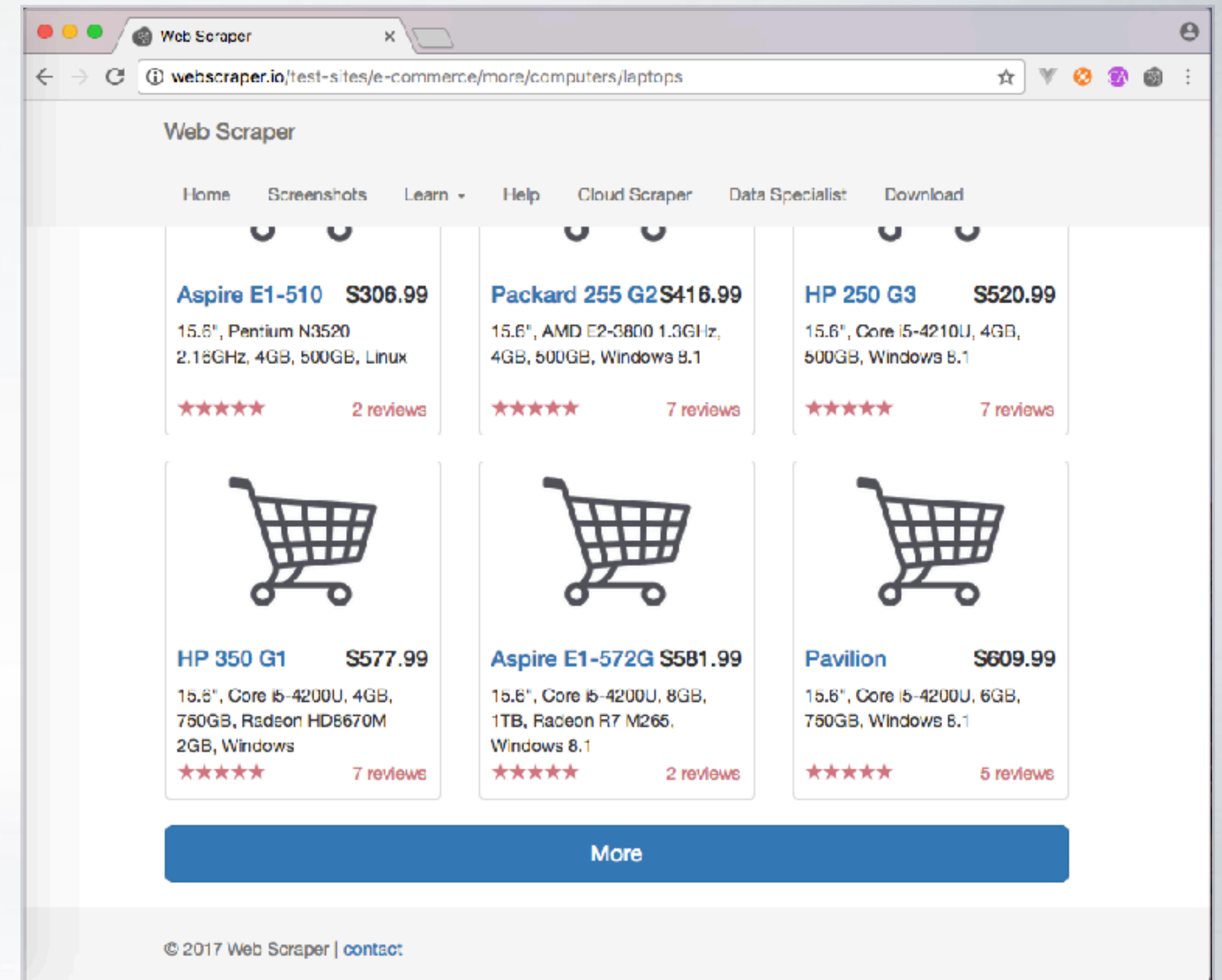
<https://tw.buy.yahoo.com/>



# 動態HTML

## dynamic HTML (DHTML)

HTML與CSS網頁內容隨著用戶端  
使用及操作而產生變化



[查看範例](#)



# 使用Selenium

## 執行真實的瀏覽器

- 一套網站測試工具，為瀏覽器自動化需求所設計，可以直接驅動瀏覽器進行各種網站操作
- 它能夠直接獲取即時的内容，讓程式可以直接與網頁元素即時互動並執行 JavaScript 程式
- Selenium沒有自己的瀏覽器，需要借助Firefox, Chrome 等瀏覽器

### 前置作業

```
pip3 install selenium
```

Chrome Driver 檔案下載連結

```
from selenium import webdriver
```

```
driver = webdriver.Chrome(檔案路徑)  
driver.get( URL )
```

```
...
```

```
driver.current_url  
driver.quit()
```

# Selenium 瀏覽器常用控制函式

函式	說明
<code>driver.get(URL)</code>	將URL填入瀏覽器，發出頁面請求
<code>driver.forward()</code>	在瀏覽器中，將網頁移至上一頁
<code>driver.back()</code>	在瀏覽器中，將網頁移至下一頁
<code>driver.quit()</code>	關閉目前driver中的瀏覽器
<code>driver.refresh()</code>	將driver中的瀏覽器重新整理
<code>driver.current_url</code>	獲取目前瀏覽器所在的URL
<code>driver.title</code>	獲取目前瀏覽器中網頁的標題
<code>driver.execute_script(JavaScript程式碼)</code>	在瀏覽器中執行引數中的JS程式碼
<code>driver.save_screenshot('/Screenshots/foo.png')</code>	在瀏覽器中截圖，並儲存到/Screenshots/foo.png

# Selenium 元素選取與操作函式

函式	說明
<code>driver.find_element_by_css_selector(選擇器)</code>	使用CSS選擇器，選取瀏覽器中單一元素
<code>driver.find_elements_by_css_selector(選擇器)</code>	使用CSS選擇器，選取瀏覽器中所有元素
<code>driver.選擇元素.get_attribute(屬性)</code>	得到元素的 attribute/property 值
<code>driver.選擇元素.text</code>	得到元素之內容
<code>driver.選擇元素.size</code>	得到元素之大小(寬高值)
<code>driver.選擇元素.clear()</code>	清除內容
<code>driver.選擇元素.click()</code>	模擬滑鼠左鍵，點擊元素一次



# Selenium + PyQuery

```
from selenium import webdriver  
from pyquery import PyQuery as pq
```

```
driver = webdriver.Chrome()  
driver.get(URL)
```

```
html = driver.find_element_by_css_selector("*").get_attribute("outerHTML")  
doc = pq(html)
```

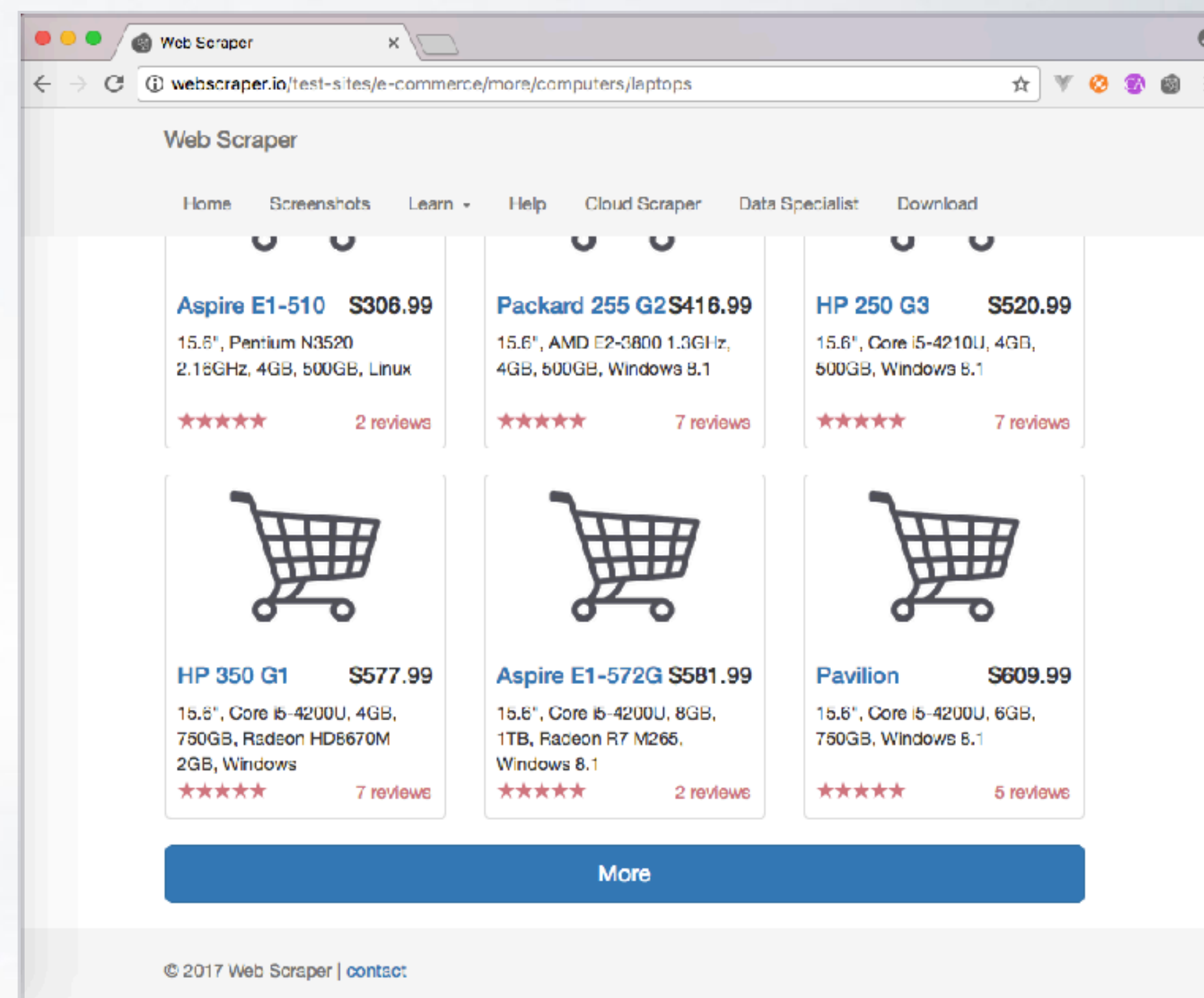
```
...
```

```
driver.quit()
```

# 動手練習

使用Selenium+PyQuery突破範例  
網站內More按鈕限制，把所有商  
品名稱都扒下來

[提示] `driver.選擇元素.click()`



[查看範例](#)

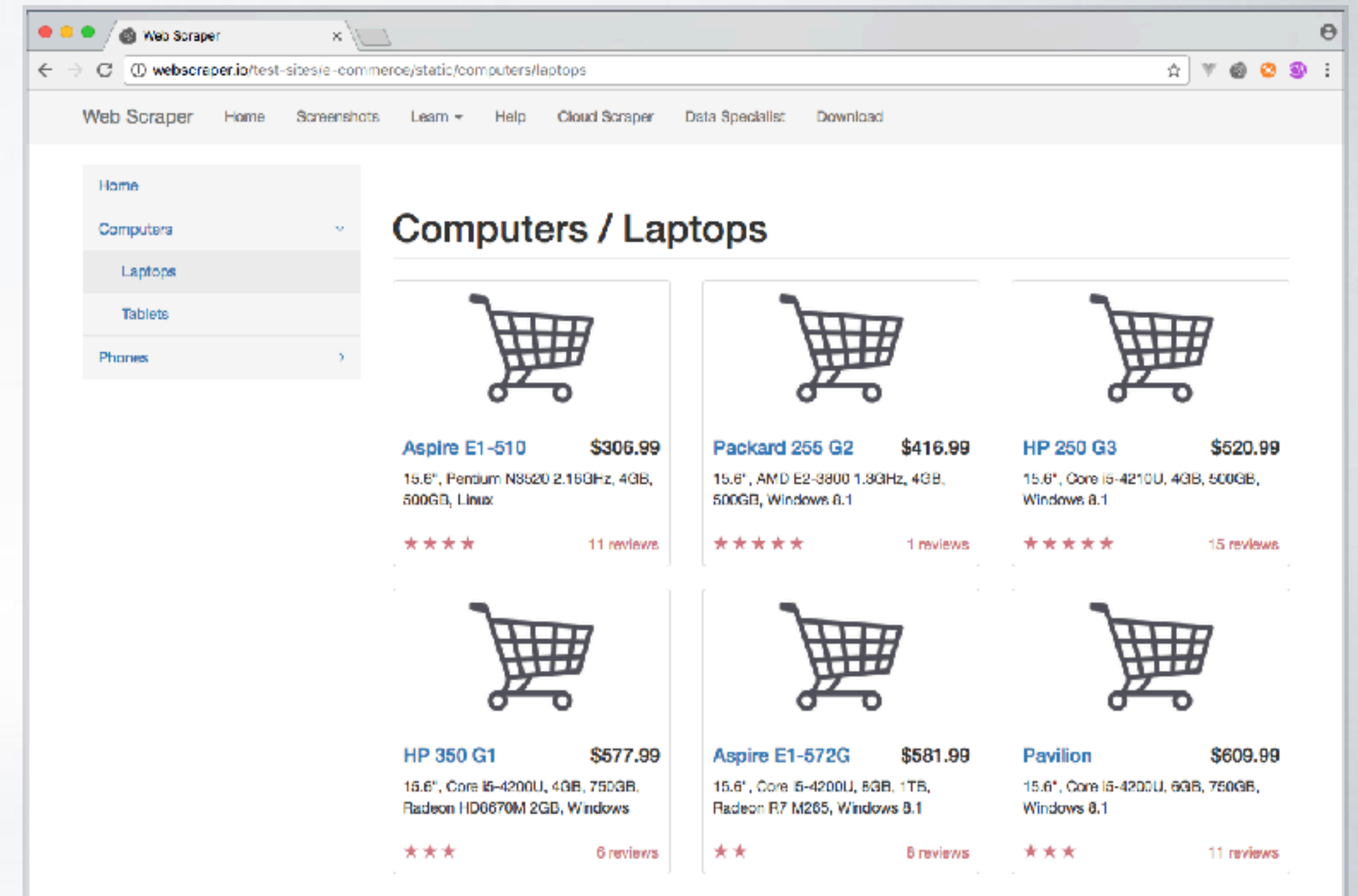


# 動手練習

使用Selenium+PyQuery突破範例  
網站內頁面捲動限制，把所有商品  
名稱都扒下來

[提示1] `driver.execute_script(JS程式碼)`

[提示2] 捲動指令: `window.scrollTo(X方向,  
Y方向)`



目標網站



# Selenium 元素滑鼠操作函式- 點擊與拖曳

函式	說明
<code>driver.選擇元素.click()</code>	模擬滑鼠左鍵，點擊元素一次
<code>driver.選擇元素.click_and_hold()</code>	模擬滑鼠左鍵，點擊元素並按住
<code>driver.選擇元素.release()</code>	模擬滑鼠左鍵，放開左鍵
<code>driver.選擇元素.double_click()</code>	模擬滑鼠左鍵，雙擊元素
<code>driver.選擇元素.context_click()</code>	模擬滑鼠右鍵，點擊元素一次
<code>driver.drag_and_drop(source, target)</code>	模擬滑鼠拖曳source元素並在target元素上放開
<code>driver.drag_and_drop_by_offset(source, xoffset, yoffset)</code>	模擬滑鼠拖曳source元素並在網頁中xoffset, yoffset位置上放開

# Selenium 元素滑鼠操作函式- 移動

函式	說明
<code>driver.選擇元素.move_by_offset(xoffset, yoffset)</code>	模擬滑鼠移動(但不可見)，移至xoffset, yoffset
<code>driver.選擇元素.move_to_element(某元素)</code>	模擬滑鼠移動(但不可見)，移至某元素上
<code>move_to_element_with_offset(某元素, xoffset, yoffset)</code>	模擬滑鼠移動(但不可見)，移至某元素上，並且相對平移xoffset, yoffset

# Selenium 元素鍵盤操作函式

函式	說明
driver.選擇元素.key_down(“d”)	模擬鍵盤按鍵，點擊”d”鍵
driver.選擇元素.key_up(“d”)	模擬鍵盤按鍵，放開”d”鍵
driver.選擇元素.send_keys(字串)	模擬鍵盤按鍵，一次送出多個鍵



# Selenium 的Cookies操作

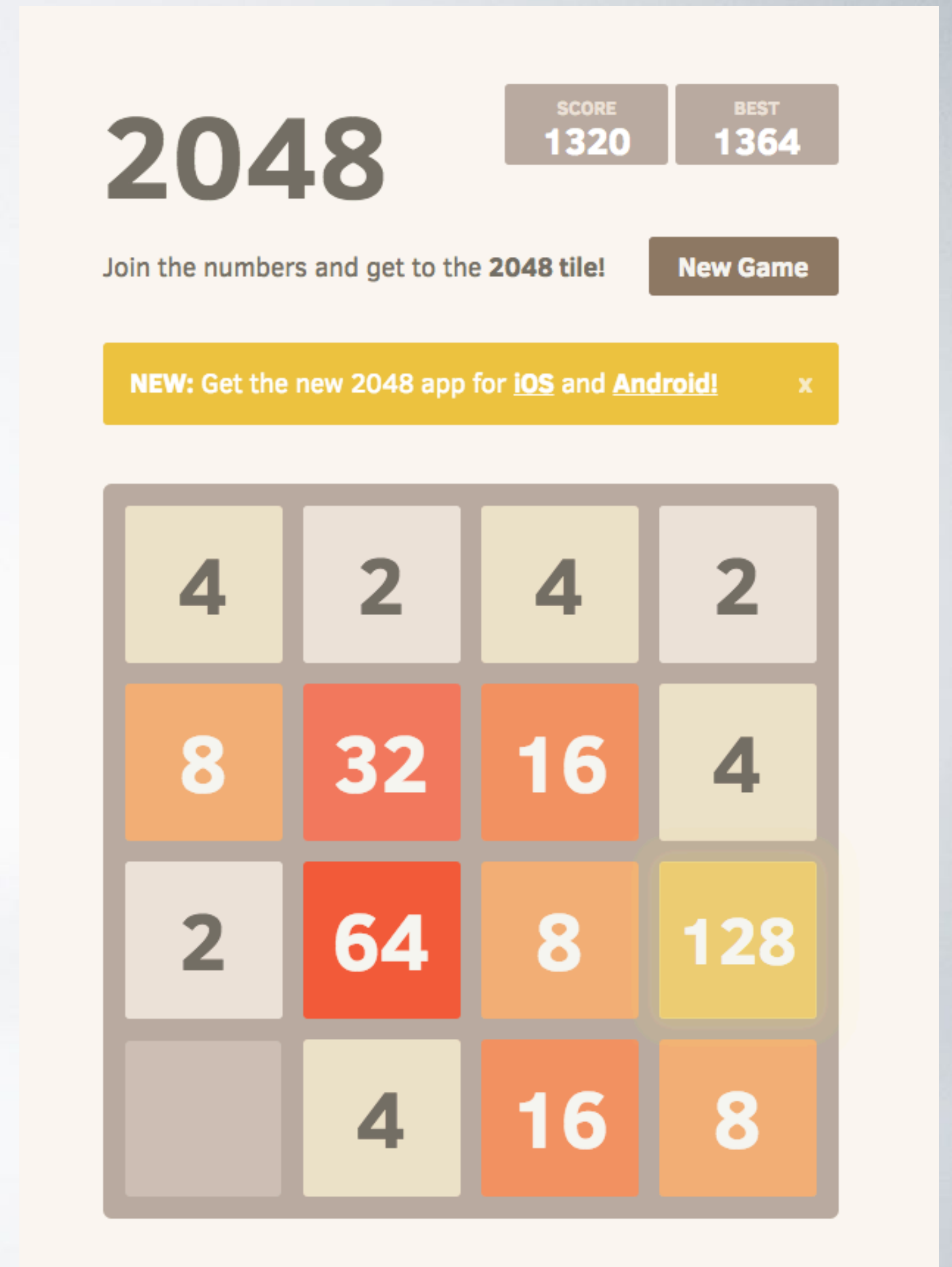
函式	說明
<code>driver.add_cookie({'name':'over18', 'value':'1', 'path': '/'})</code>	加入Cookies至driver瀏覽器中
<code>driver.get_cookies()</code>	獲得目前瀏覽器中的所有Cookies
<code>driver.delete_cookie("over18")</code>	刪除目前瀏覽器中name為'over18'的Cookie
<code>driver.delete_all_cookies()</code>	刪除目前瀏覽器中所有的Cookies

# 爬蟲也能當遊戲快手

透過Selenium中的鍵盤控制  
.send\_keys()來完成2048遊戲

`from selenium.webdriver.common.keys import Keys`

[按我連結](#)





# CAPTCHA - 驗證碼

## 政府專家示警：驗證碼已被破解 金融業要當心



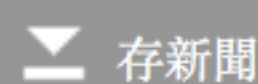
分享



留言



列印



存新聞

A-

A+

2017-08-17 20:54 經濟日報 記者陳怡慈／即時報導



讚 927



分享



傳送

法務部調查局電腦偵辦科科长周台維17日在一場研討會上提醒，國內金融業者要小心，不少銀行、證券公司視為保護機制的圖形或文字驗證碼，已經被破解了，「仍然用這機制在作驗證的機構，希望盡快換掉」。

他並表示，可以改用照片辨識，這是新的方法。例如，在一張九宮格的照片上，詢問客戶，照片裡的汽車是在哪一格，然後去點它，這方式目前還沒有被破解。

[報導連結](#)



# 驗證碼機制種類繁多



名字	<input type="text" value="请输入用户名"/>
	<input type="text" value="66+9=?"/>
验证码计算结果	<input type="text" value="请输入验证码计算结果"/>
	<input type="button" value="登录"/>

Qualifying question

Just to prove you are a human, please answer the following math challenge.

Q: Calculate:

$$\frac{\partial}{\partial x} \left[ 4 \cdot \sin \left( 7 \cdot x - \frac{\pi}{2} \right) \right] \Big|_{x=0}$$

A:

mandatory

Note: if you do not know the answer to this question, reload the page and you'll get another question.

☐ 我不是機器人

  
reCAPTCHA  
隱私權 - 條款

請選取圖片中含有 透抽 的圖片



# 驗證碼練習

經濟部國際貿易局 -  
廠商基本資料查詢系統為例

```
params = {'queryType':'C',  
          'basic_select':'2',  
          'chinese_name':'食品',  
          'ccc_select':'1',  
          'ccc_num':'8',  
          'pname_select':'1',  
          'txtCheckCode':'???'}
```

廠商基本資料查詢 操作手冊

統一編號：

廠商中文名稱： (請至少輸入二碼中文名稱)

廠商英文名稱：

廠商代表人姓名： (請至少鍵入二碼中文姓名)

出進口貨品： 進口 8碼 (請鍵入CCC Code 前4碼或前8碼)

出進口貨品名稱： 進口

圖片驗證碼： 9k5gxx 重查圖片驗證碼

為預防不當駭客以程式攻擊本系統或蓄意進行大量資料之查詢，並保障您的權益，請依圖示內容輸入驗證碼。

查詢 重新輸入

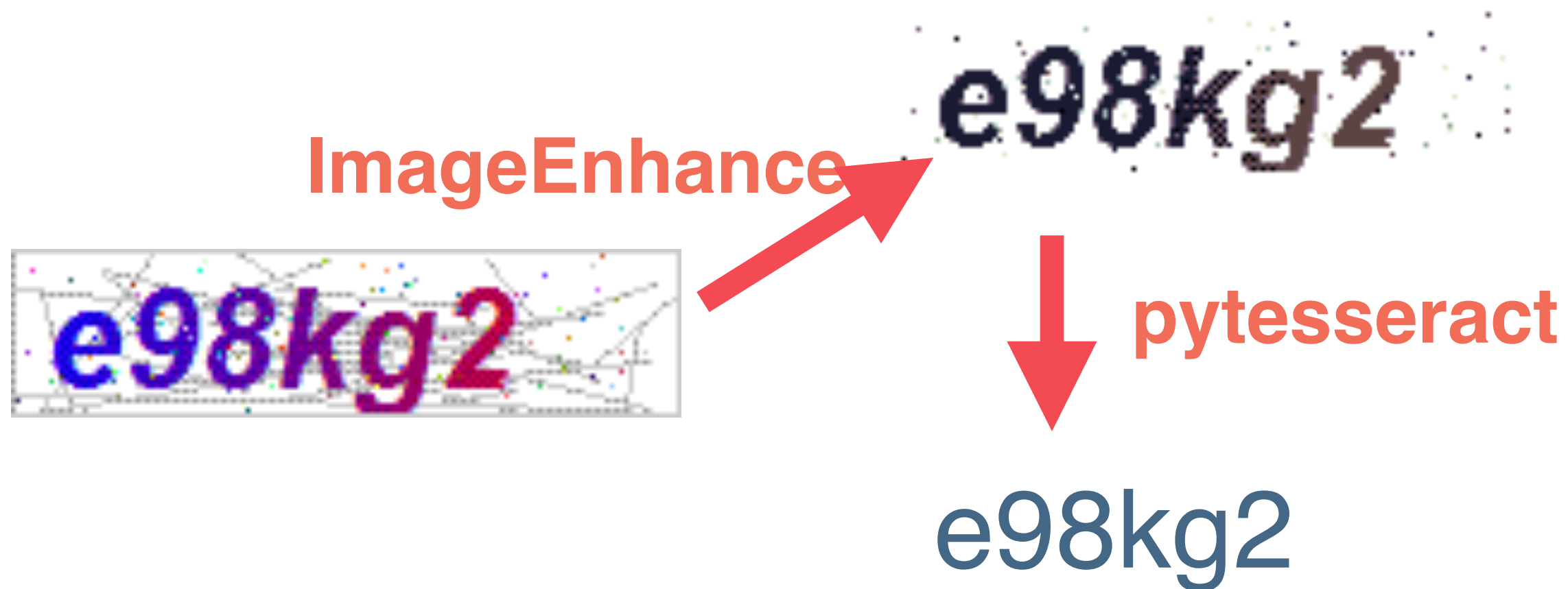
備註：  
1.網站公開之依據：出進口廠商登記辦法第7條之1。  
2.廠商可向經濟部國際貿易局申請不公開以出進口貨品CCC Code前8碼查詢出進口廠商登記基本資料，以維護廠商權益。

ARPHIC 注 OFF ?

<https://fbfh.trade.gov.tw/rich/text/fbj/asp/fbje140Q.asp>

# 如何用最簡單且自動的方式 辨識網站中的驗證碼

- PIL(Python Imaging Library)影像資料處理套件，具備數十種圖檔格式的讀寫能力、基本的影像與色彩處理、濾鏡效果
- Tesseract 是一款被廣泛使用的開源 OCR 工具，辨識影像中文字



[點我前往Tesseract-OCR 安裝連結](#)

```
pip3 install pytesseract  
pip3 install Pillow
```

```
import pytesseract  
from PIL import Image, ImageEnhance  
# import cv2
```

```
im = Image.open("captcha.png")  
im = im.convert("RGBA")  
im = ImageEnhance.Contrast(im).enhance(3.0)  
im = ImageEnhance.Brightness(im).enhance(35.0)  
im = ImageEnhance.Color(im).enhance(0.1)  
im.show()  
Capt = pytesseract.image_to_string(im).lower()
```