

```
import pandas
```

```
//導入 Pandas 函式庫
```

```
from bs4 import BeautifulSoup
```

```
//導入 BeautifulSoup 函式庫
```

```
import requests
```

```
//導入 requests 函式庫
```

```
url = 'http://npb.jp/scores/2017/1104/h-db-06/playbyplay.html'
```

```
//將要爬蟲之網址存入 url 這個變數
```

```
res = requests.get(url)
```

```
//將網址利用 requests.get 這個 API 讀取，並將傳回值放入 res 這個  
變數
```

```
soup = BeautifulSoup(res.text)
```

```
//將 res 用 BeautifulSoup 來解析網頁原始碼並將傳回值放入 soup 這  
個變數
```

```
test1 = pandas.read_html(url)[3]
```

```
//將網址用 pandas.read_html 來讀取，回傳值為 list，PlayByPlay 的資  
料是從 list 中 index = 3 開始
```

```
for i in range(4,len(pandas.read_html(url)),1):
    test2 = pandas.read_html(url)[i]
    test1 = pandas.concat([test1,test2],axis=0,ignore_index=True)

//用 for 迴圈，將整個網頁，從底層開始的 Table 一系列組合起來

放入 test1 這個變數
```

```
baseball = pandas.DataFrame(test1)

//test1 是一個 list，須將它轉成 DataFrame 才能順利存檔

baseball.columns =
[soup.select('.w1')[0].text,soup.select('.w1')[1].text,soup.select('.w1')[2].t
ext,soup.select('.w1')[3].text,soup.select('.w2')[0].text]

//因為將網頁組合起來時，最上面的 Title 會讀到空白，造成 Title 位
移，因此我們必須利用 BeautifulSoup 另外組合，Title 在網頁原始碼
Tag 是.w1 及.w2，針對這兩的標籤，我們可以單獨取出 Title 來設定
baseball 這個 DataFrame 的 column 名字
```

```
baseball.to_excel('20171104.xlsx')

//最後依照場次年月日做為檔名，及完成該場次的爬蟲
```