

## Python 面試題目

### 1. 請舉例三個程式開發上使程式碼好維護的原則

默想：我現在這樣寫，不久後我或別人看得懂嗎？

- 1.) 加上註解，提醒自己或告知其他使用者這段程式碼的用意，註解最好能簡明扼要。
- 2.) 命名語意要能讓人快速理解，可參考蛇行命名法與駝峰式命名法。
- 3.) 製作說明文件，最好能圖文並茂（附上表格為佳），並說明引用的套件有哪些、及操作流程。
- 4.) 寫法盡量統一，雖然有多種解法但盡量只使用一種。比方說讀 Excel 檔案，可以使用 `xlrd` 或 `pandas`，但盡量擇一使用；或者 `import datetime` / `from datetime import datetime` 擇一使用。
- 5.) 不要把程式碼寫死，且預測可能發生的變化。縱使現在程式可以執行，但若資料型態不同，是否會有 Bug。

### 2. 請舉例並說明一個使用過的裝飾器其功能為何

Flask 中為路由 (Route)，說明如下。

- a. 透過函式的裝飾器設定路由
- b. 決定網址路徑和處理函式的對應關係，前端輸入不同路徑時，後端程式要決定對應的處理函式。
- c. 範例如下：（完整程式碼請參考 GitHub：[https://github.com/AcerPing/Ace\\_Pycharm\\_Project\\_2020-08-04Renewed/blob/master/Flask\\_%20MessageBoard/index.py](https://github.com/AcerPing/Ace_Pycharm_Project_2020-08-04Renewed/blob/master/Flask_%20MessageBoard/index.py)）

```
@app.route('/')
def hello():
    page = request.args.get('page', default=1, type=int)
    print("請求方法", request.method)
    print("頁數:", page)
    list = load_data_page(page)
    return render_template("index.html", list=list)
```

### 3. 請定義一個 function 名為 `append_to_list`，功能說明如下：

傳入參數為 `n`, `list`，當 `list` 沒提供的時候自動用空陣列作為 default

此 function 功能為 將 `n` 附加到 `list` 最後傳出

```
append_to_list(n=1)
>>> [1]
append_to_list(n=1, lst=[1,2])
>>> [1,2,1]
```

```
def append_to_list(n, lst=[]):
    lst.append(n)
    return lst

lst = append_to_list(n=1)
print(lst)

lst = append_to_list(n=1, lst=[1,2])
print(lst)
```

### 4. 已知網頁 API 回應內容如下

```
{
  "total": 5,
  "data": [
    {"id": "001", "name": "productA", "price": 25, "category": "AAA", "stock": 7},
    {"id": "002", "name": "productB", "price": 30, "category": "AAA", "stock": 17},
    {"id": "003", "name": "productC", "price": 60, "category": "BBB", "stock": 2},
    {"id": "004", "name": "productD", "price": 10, "category": "CCC", "stock": 26},
    {"id": "005", "name": "productE", "price": 15, "category": "BBB", "stock": 16},
  ]
}

# 請按照 category 欄位進行整理，算出每個 category 的 stock 加總：
{
  "AAA": {"products": ["ProductA", "ProductB"], "stock": 24},
  "BBB": {"products": ["ProductC", "ProductE"], "stock": 18},
  "CCC": {"products": ["ProductD"], "stock": 26}
}
```

```
# 題目
API_Response = {"total": 5,
                "data": [
                    {"id": "001", "name": "productA", "price": 25, "category": "AAA", "stock": 7},
                    {"id": "002", "name": "productB", "price": 30, "category": "AAA", "stock": 17},
                    {"id": "003", "name": "productC", "price": 60, "category": "BBB", "stock": 2},
                    {"id": "004", "name": "productD", "price": 10, "category": "CCC", "stock": 26},
                    {"id": "005", "name": "productE", "price": 15, "category": "BBB", "stock": 16}],
                ]}

# Solution
Result = {}
for each_id in API_Response["data"]:
    # print(each_id["category"])
    if each_id["category"] not in Result:
        Sub_Result = {}
        Sub_Result.setdefault("products", [each_id["name"]])
        Sub_Result.setdefault("stock", each_id["stock"])
        Result[each_id["category"]] = Sub_Result
    else:
        Result[each_id["category"]]["products"].append(each_id["name"])
        Result[each_id["category"]]["stock"] += each_id["stock"]
print(Result)
```

5. 請定義一個 function，爬取以下網站：<https://tw.stock.yahoo.com/rank/volume?exchange=TAI> 並根據名次欄位設計成 dict 格式回傳

```
In [59]: import requests_html
         session = requests_html.HTMLSession()
```

```
In [88]: def get_rank_stock():
```

```
    # 爬取 Yahoo! 台股成交量排名
    url = "https://tw.stock.yahoo.com/rank/volume?exchange=TAI"
    r = session.get(url)
    # 解析 HTML
    data = r.json()
    return data
```

```
In [89]: get_rank_stock()
```

```
{
  'rank': 1,
  'stock': '212.00',
  'price': '187.00',
  'volume': '25.00',
  'value': '420,313'},
  '2': {'stock': '大成鋼',
        'price': '2027.TW',
        'volume': '58.40',
        'value': '1.50',
        'high': '2.64%',
        'low': '61.60',
        'price': '57.40',
        'volume': '4.20',
        'value': '130,636'},
  '3': {'stock': '聯電',
        'price': '2303.TW',
        'volume': '53.40',
        'value': '1.20',
        'high': '2.30%',
        'low': '1.10'}}
```

# 原答案方法

```
def get_rank_stock():
    url = 'https://tw.stock.yahoo.com/rank/volume?exchange=TAI'
    response = session.get(url)
    title_line = response.html.xpath("//div[(contains(@class, 'table-header-wrapper'))]") # 標題列
    title_line_list = title_line[0].text.split('\n') # 標題列 -> list 型式
    rows = response.html.xpath("//li[@class='List(n)']") # 每一筆數據資料
    data = {}
    for each_row in rows:
        Sub_Result = {}
        each_row_list = each_row.text.split('\n') # 每一筆數據資料 -> list 型式
        for i in range(len(title_line_list)):
            Sub_Result.setdefault(title_line_list[i], each_row_list[i])
        del Sub_Result['名次'] # 名次要當 dict.key
        data.setdefault(each_row_list[0], Sub_Result)
    return data
```

# 但原答案欄位抓取錯誤，改進後如下

```
def get_rank_stock():
    url = 'https://tw.stock.yahoo.com/rank/volume?exchange=TAI'
    response = session.get(url)
    title_line = response.html.xpath("//div[(contains(@class, 'table-header-wrapper'))]") # 標題列
    title_line_list = title_line[0].text.split('\n') # 標題列 -> list 型式
    rows = response.html.xpath("//li[@class='List(n)']") # 每一筆數據資料
    data = {}
    for each_row in rows:
        Sub_Result = {}
        each_row_list = each_row.text.split('\n') # 每一筆數據資料 -> list 型式
        each_row_list[1] = each_row_list[1] + ' / ' + each_row_list[2] # 修改數據
        del each_row_list[2]
        for i in range(len(title_line_list)):
            Sub_Result.setdefault(title_line_list[i], each_row_list[i])
        del Sub_Result['名次'] # 名次要當 dict.key
        data.setdefault(each_row_list[0], Sub_Result)
    return data
```