



Python 物件導向簡介



http, _ = start_http_server(8000) 這是什麼用法，我對python的物件導向不熟，你能教我嗎，或是你能給更簡單的範例



這段程式碼：

```
http, _ = start_http_server(8000)
```

是 Python 的 **tuple unpacking**（元組解包）用法，`start_http_server(8000)` 回傳一個 tuple（例如 `(http_server_instance, None)`），然後 `http` 會接收第一個值，而 `_` 則是用來忽略第二個值（常見的 Python 慣例）。

但你不熟 Python 物件導向，所以我們先從 **基本的 Python 物件導向概念** 開始，然後再來解析 `start_http_server(8000)` 的用法。

1. Python 物件導向的基本概念

在 Python 中，**類別（Class）** 是用來建立物件的模板，而 **物件（Object）** 是類別的實例（Instance）。每個類別都可以有 **屬性（Attributes）** 和 **方法（Methods）**。

簡單範例：定義一個 `Server` 類別

```
class Server:
    def __init__(self, port):
        self.port = port
```



```
def start(self):
    print(f"Server started on port {self.port}")

# 建立 Server 物件
server = Server(8000)
server.start() # 輸出: Server started on port 8000
```

- `__init__` 是建構函式 (Constructor)，用來初始化物件的屬性。
- `self` 代表該物件本身，用來存取類別內的變數與方法。
- `server = Server(8000)` 創建一個 `Server` 物件，並將 `port` 設定為 8000。
- `server.start()` 會執行 `start` 方法，輸出 `Server started on port 8000`。

2. `start_http_server(8000)` 可能的用法

在 Prometheus client 庫裡，`start_http_server(port)` 是用來啟動一個 HTTP 伺服器，讓 Prometheus 可以來收集指標 (Metrics)。但 `start_http_server(8000)` 可能回傳的是一個 HTTP 伺服器物件 (類似 `Server`)。

假設 `start_http_server` 定義如下：

```
def start_http_server(port):
    server = Server(port) # 假設 Server 是一個類別
    server.start()
    return server, None # 回傳一個 tuple
```

當你執行：

```
http, _ = start_http_server(8000)
```

- `http` 會取得 `server` 這個物件。
- `_` 會忽略 `None` (這是 Python 常見用來表示不重要的值的方式)。



這裡 `http` 就像是一個伺服器物件，假設你想要存取它的 `port`，你可以：

```
print(http.port) # 8000
```

3. 簡化版 `start_http_server` 實作

如果你希望完全理解這個概念，這裡是一個 **超簡單版** 的 `start_http_server`，可以自己測試：

```
class SimpleServer:
    def __init__(self, port):
        self.port = port

    def start(self):
        print(f"SimpleServer started on port {self.port}")

def start_http_server(port):
    server = SimpleServer(port)
    server.start()
    return server, None # 回傳一個 tuple

# 解包回傳值
http, _ = start_http_server(8000)

# 存取 http 物件的屬性
print(http.port) # 8000
```

這段程式碼模擬 `start_http_server(8000)`：

1. `SimpleServer` 類別代表一個簡單的伺服器，並有 `start()` 方法。
2. `start_http_server(port)` 建立 `SimpleServer` 物件並啟動它，然後回傳 `(server, None)`。



3. `http, _ = start_http_server(8000)` 透過 **tuple unpacking** 取得 `server` 物件，並忽略 `None`。
-

這樣你對 **Python 物件導向**、**tuple unpacking** 以及 `start_http_server` 的回傳值 有更清楚的理解了嗎？如果有不清楚的地方，我可以再提供其他範例來幫助你學習！ 🚀