

## 手機應用程式排行分析

### 一、輸出應用程式下載量的排名

· 程式碼：

```
def print_app_rankings(app_rankings):  
    """  
    輸出應用程式的下載排名和分數排名  
    """  
    sorted_by_downloads = sorted(app_rankings['apps'], key=lambda x: x['downloads'], reverse=True)  
    print("下載排名:")  
    for i, app in enumerate(sorted_by_downloads, start=1):  
        print(f"{i}. {app['name']} - {app['downloads']} downloads")
```

· 輸出結果：

下載排名：

```
1. WhatsApp - 6000000000 downloads  
2. YouTube - 5000000000 downloads  
3. Facebook - 3000000000 downloads  
4. Instagram - 1000000000 downloads  
5. Snapchat - 1000000000 downloads  
6. TikTok - 500000000 downloads
```

->由此可知WhatsApp的下載量最多，是最普遍被使用的應用程式。

### 二、輸出用戶給予應用程式的評分排名

· 程式碼：

```
sorted_by_rating = sorted(app_rankings['apps'], key=lambda x: x['rating'], reverse=True)  
print("\n分數排名:")  
for i, app in enumerate(sorted_by_rating, start=1):  
    print(f"{i}. {app['name']} - Rating: {app['rating']}")
```

· 輸出結果：

分數排名：

```
1. YouTube - Rating: 4.8  
2. WhatsApp - Rating: 4.7  
3. Instagram - Rating: 4.6  
4. TikTok - Rating: 4.5  
5. Snapchat - Rating: 4.3  
6. Facebook - Rating: 4.0
```

->可以得知YouTube是最讓使用者滿意的應用程式

### 三、綜合下載量和排名找到整體排名最高的應用程式

程式碼：

```
def print_top_app(app_rankings):  
    """  
    輸出綜合下載次數與分數排名相對最高的應用程式  
    """  
    sorted_by_combined_rank = sorted(app_rankings['apps'], key=lambda x: (x['downloads'] + x['rating']), reverse=True)  
    top_app = sorted_by_combined_rank[0]  
    print(f"\n整體排名最高的應用程式: {top_app['name']}")
```

· 輸出結果：

**整體排名最高的應用程式：WhatsApp**

->雖然WhatsApp的使用者評分只排在第二，但兩項數據的綜合表現是由WhatsApp佔居第一，由此可知WhatsApp對於使用者而言是被頻繁使用的應用程式，但可能使用上有一些小問題導致評分較YouTube低。