

# 資料欄位命名方式

具有一致性的資料欄位命名很重要。若無命名原則，我們將耗費大量的時間在重新命名、處理命名衝突以及衝突所造成的bug。

由於程式會依賴正則表達式運行，以下原則規範也是在確保我們可以順利地運用正則表達式對相關資訊進行處理。

原則的制定根據：

1. 清晰與易讀性。
2. 是否可以順利地以正則表達式(regular expression)從字串中提取想要的資訊。

## 原則

### 一般

1. 物理量最放前面、其他附屬資訊放後面
2. 除了縮寫外，不使用不必要的大寫
3. 避免籠統 (例如 `air_temperature` 比 `temperature` 好)
4. 不同類型的資訊以底線(`_`)分隔。
5. 附屬資訊以底線分隔。同一類型的附屬資訊須有一致性的字首(前綴)。
  - 例如有一個新的氣溫溫度計，可以命名為 `air_temperature_CWB`。若又再新增一個溫度計，可以命名為 `air_temperature_CWB1` (同一類別的測站) 或 `air_temperature_NCU` (不同類別的測站)。
  - 規則須滿足[原則-字母與數字](#)。
6. 純數字不要成為單獨的一區(e.g. `_1_`)；這會造成難以透過正則表達式提取該項資訊。
7. 是單字的話不要拼錯。

### 字母與數字

在大多數時候，數字 `[0-9]` 應在字母 `[A-Za-z]` 之後；在少數特定情況，數字也可在字母之前或交錯出現。

數字在字母之前的場合：

- 值+單位。例如：`..._10cm_...`

字母在數字之前的場合：

- 代號=字首+版本號。例如：`..._CWB1_...`，`..._CWBv1_...`

- 純版本號。例如 `..._v1_...`。

數字與字母交錯出現的場合：

- 中英混和的代號。例如台中農試所的溼度：`humidity_G2F820`
- 這是一個特例情況。在這種場合，新資訊必須再以底線分隔。例如 `humidity_G2F820_v1`。

## 附屬資訊順序

- 附屬資訊的順序基本上可由重要性排列，但不嚴格要求。
- 同一物理量的附屬資訊順序需始終一致。

## 範例

### 原則-一般

- `CWB_Humidity` ❌ (物理量應放最前面、附註放後面；不必要的大寫)。
- `CWB_humidity` ❌ (物理量應放最前面、附註放後面)
- `Humidity_CWB` ❌ (不必要的大寫)
- `humidity_CWB` ○
- `humidity` ○
- `Soil_water_content_blabla` ❌ (不必要的大寫)
- `soil_water_content_blabla` ○
- `air_temperature` ○
- `soil_water_content_2_10cm` ❌ (數字不能成為單獨的一區)
- `Temperature` ❌ (不必要的大寫；溫度有土水氣溫，`temperature`過於籠統)
- `precipitation` ❌ (拼錯單字)

### 原則-字母與數字

- `humidity_CWB1`、`humidity_CWB2` ○。
- `humidity_CWBv1`、`humidity_CWBv2` ○。
- `humidity_G2F820_v1`、`humidity_G2F820_v2` ○。
- `humidity_G2F820v1` ❌：(若為字母與數字交錯的代號(e.g., `G2F820`)，則該代號不得與其他資訊(例如代表版本的資訊 `v1`)未經底線分隔合併)

### 原則-附屬資訊順序

- `soil_water_content_v2_10cm` ○ (附屬資訊順序不拘，但同一物理量需始終保持一致性)。
- `soil_water_level_10cm_v2` ○ (附屬資訊順序不拘，但同一物理量需始終保持一致性)
- `soil_water_content_v1_10cm` 且 `soil_water_content_20cm_v1` ❌：(同一物理量的附屬資訊順序並沒有保持一致性)

# 日誌

- 雨量的部分，`Rain01` 改成 `precipitation_01mm` ；原本精度 0.5mm的 `precipitation` 改成 `precipitation_05mm`
- 氣溫一律使用 `air_temperature` (在2019、2020的資料為 `Temperature` )
- `Soil_water_content` 改成 `soil_water_content` (其他井水位、土溫也比照辦理取消不必要的大寫)