

awk語法

- awk [options] 'scripts' var=value filename
 - -F 指定分隔符(可是字串或正規表達法), 預設是空白(space)
 - -f 從指令碼檔案(awk script)中讀取awk命令
 - -v var=value賦值變數, 將外部變數遞給awk

awk 算術運算子

awk '\$2 + \$3 >= 160 {print \$0}' filename

Operator	Meaning	Example
+	Add	x+y
-	Subtract	x-y
*	Multiply	x*y
/	Divide	x/y
%	Modulus	x%y
^	Exponential	x^y

```
(base) czw@czwangs-mbp 111_1_CS % cat people.txt
John 23 177
May 5 120
Tom 1 80
Jenny 15 150
David 27 185
Jennifer 18 158
Henry 38 186
Gary 41 172
Lulu 22 167
(base) czw@czwangs-mbp 111_1_CS %
```

```
base) czw@czwangs-mbp 111_1_CS % awk '$2 + $3 >= 160 {print $0}' people.txt
John 23 177
May 5 120
David 27 185
Jennifer 18 158
Henry 38 186
Gary 41 172
Lulu 22 167
base) czw@czwangs-mbp 111_1_CS %
```

11

awk 關係運算子

awk '{if (\$3 == 120) print \$0}' filename

Operator	Meaning	Example
<	Less than	x<y
<=	Less than or equal	x<=y
==	Equal to	x==y
!=	Not equal to	x!=y
>	Greater than	x>y
>=	Greater than or equal to	x>=y

#關係運算子

awk '{if (\$3 == 120) print \$0}' people.txt

awk '{if (\$3 != 120) print \$0}' people.txt

awk '{if (\$3 > 120) print \$0}' people.txt

```
(base) czw@czwangs-mbp 111_1_CS % awk '{if ($3==120) print $0}' people.txt
Lulu 22 167
(base) czw@czwangs-mbp 111_1_CS %
```

12

awk 邏輯運算子

awk '(\$2 > 6) && (\$3 >= 150) {print \$0}' filename

Operator	Meaning	Example
&&	Logical AND	x&&y
	Logical OR	x y

awk '(\$2 > 6) && (\$3 >= 150){print\$0}' people.txt

awk '(\$2 < 6) || (\$3 >= 150){print\$0}' people.txt

```
(base) czw@czwangs-mbp 111_1_CS % awk '($2 > 6) && ($3 >= 150) {print $0}' people.txt
John 23 177
May 5 120
David 27 185
Jennifer 18 158
Henry 38 186
Gary 41 172
Lulu 22 167
(base) czw@czwangs-mbp 111_1_CS %
```

```
(base) czw@czwangs-mbp 111_1_CS % awk '($2 < 6) || ($3 >= 150) {print $0}' people.txt
John 23 177
May 5 120
Tom 1 80
Jenny 15 150
David 27 185
Jennifer 18 158
Henry 38 186
Gary 41 172
Lulu 22 167
(base) czw@czwangs-mbp 111_1_CS %
```

13

awk 正則運算子

```
awk ' {if ($3 ~ /0/) print $0} ' filename
```

Operator	Meaning	Example
~	Matched by reg exp	x~/y/
!~	Not matched by req exp	x!~/y/

```
#正則運算子
awk '{if ($3 ~ /0/) print $0}' people.txt
```

```

root@kali:~/# curl -s http://192.168.1.100:8080/
{"id": 1, "name": "John", "age": 30, "gender": "Male", "email": "john.doe@example.com", "password": "123456"}

```

14

awk 賦値運算子

```
awk '{for (j=1; j <= NF; j++) { print $j } }' filename
```

Operator	Meaning
=	Assign result of right-hand-side expression to left-hand-side variable
++	Add 1 to variable
--	Subtract 1 from variable
+=	Assign result of addition
-=	Assign result of subtraction
*=	Assign result of multiplication
/=	Assign result of division
%=	Assign result of modulo
^=	Assign result of exponentiation

```
awk '{for (j=1; j<= NF; j++) { print sj }}' for.txt
```

```
(root) coolcourage~# ip 111.1.0.0 cat /etc/passwd  
world  
(root) coolcourage~# ip 111.1.0.0 awk '{for i=1; i<NF; i++} {print $i}' /etc/passwd  
world  
(root) coolcourage~# ip 111.1.0.0 N
```

15

sed文字分析工具

- 串流的編輯 (stream editor)
- 字串取代、錄製、刪除的功能
- 自動化的修改文字檔

Sed指令

```
sed [option] '[n1,n2] [command] / [pattern] / [replacement] / [flag]' file.txt
```

- ◆ **option**: 以 - 符號開頭的功能, 如 -n、-r、-i, 可省略
- ◆ **n1,n2**: 代表開始行數和結尾行數, 不輸入代表每行都會執行command
- ◆ **command**: 進行的動作, 如s, a, c, d, i
- ◆ **pattern**: 給command使用的參數
- ◆ **replacement**: 當使用s指令時會使用
- ◆ **flag**: 當使用s指令時會使用

Sed語法 – 常用選項

```
sed [-nefi] '[n1,n2] [command] / [pattern] / [replacement] / [flag]' file.txt
```

option:

- ◆ -n: 沉默模式, 只有經過sed處理的那行才會被印出
- ◆ -e: 直接在命令模式編輯, 預設
- ◆ -f: 直接將sed動作寫在一個檔案內, -f file會直接執行file裡面的動作
- ◆ -i: 修改檔案

grep, awk, sed比較

- **grep**: 文字搜尋工具
 - 可用正規表達法找出匹配的內容
- **awk**: 文字分析工具
 - 逐行的讀入, 可搭配正規表達法
 - 主要用在對文字和資料進行分析處理, 以空格為預設分隔符號
 - 也是程式語言
 - 用於處理有欄位規則的行內內容, 並支援格式化輸出
- **sed**: 是一種線上編輯器
 - 它一次處理一行內容, 可搭配正規表達法
 - 主要用來自動編輯一個或多個檔案, 簡化對檔案的反覆操作
 - 用於行間的內容操作, 如新增、刪除、修改、查詢、替換

Sed語法 – 常用指令

```
sed [-nefi] '[n1,n2] [command] / [pattern] / [replacement] / [flag]' file.txt
```

command:

- ◆ a: 新增, 在下一行插入字串, 未指定行數的話, 則是在每一行之後插入字串
- ◆ c: 取代, 取代指定行數的內容
- ◆ d: 刪除, 後面通常不接任何東西
- ◆ i: 插入, 在指定行數的前一行插入字串
- ◆ p: 印出, 只印出受影響的行, 常搭配-n使用
- ◆ s: 搜尋、取代, 最常用的指令, 可搭配正規表達式使用, 將搜尋的內容進行取代

Sed語法 – 常用旗幟

```
sed [-nefi] '[n1,n2] [command] / [pattern] / [replacement] / [flag]' file.txt
```

flag:

- ◆ g: 全部取代
- ◆ [0-9]: 每行第幾次出現
- ◆ I: 忽略大小寫
- ◆ w: 把符合的結果寫入檔案