Facebook API
and
Regular Expression

- Maggie Chang
- 2016.06.18

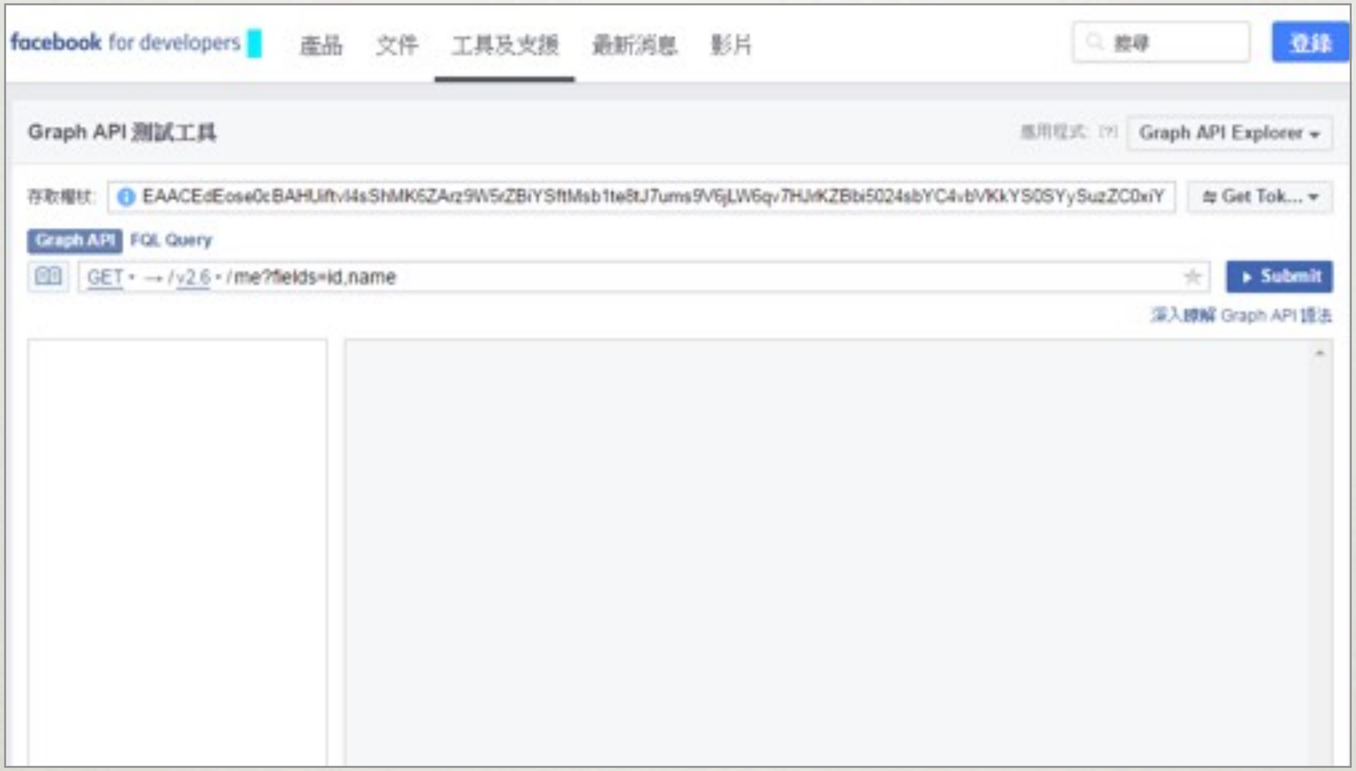
Agenda

- Facebook Graph API Introduction
- Facebook Graph API Crawling
- Regular Expression

Agenda

- Facebook Graph API Introduction (30 mins)
    - UI 介面
    - Syntax
    - Components
    - Metadata=1
    - Modifier

Facebook Graph API Explorer



Graph API 測試工具

應用程式: [?] Graph API Explorer

存取權杖: EAACEdEose0cBAHUiftvl4sShMK6ZArz9W5rZBiYSftMsb1te8tJ7ums9V6jLW6qv7HJrKZBbi5024sbYC4vbVKkYS0SYySuzZC0xiY

GET → /v2.6 /me?fields=id,name

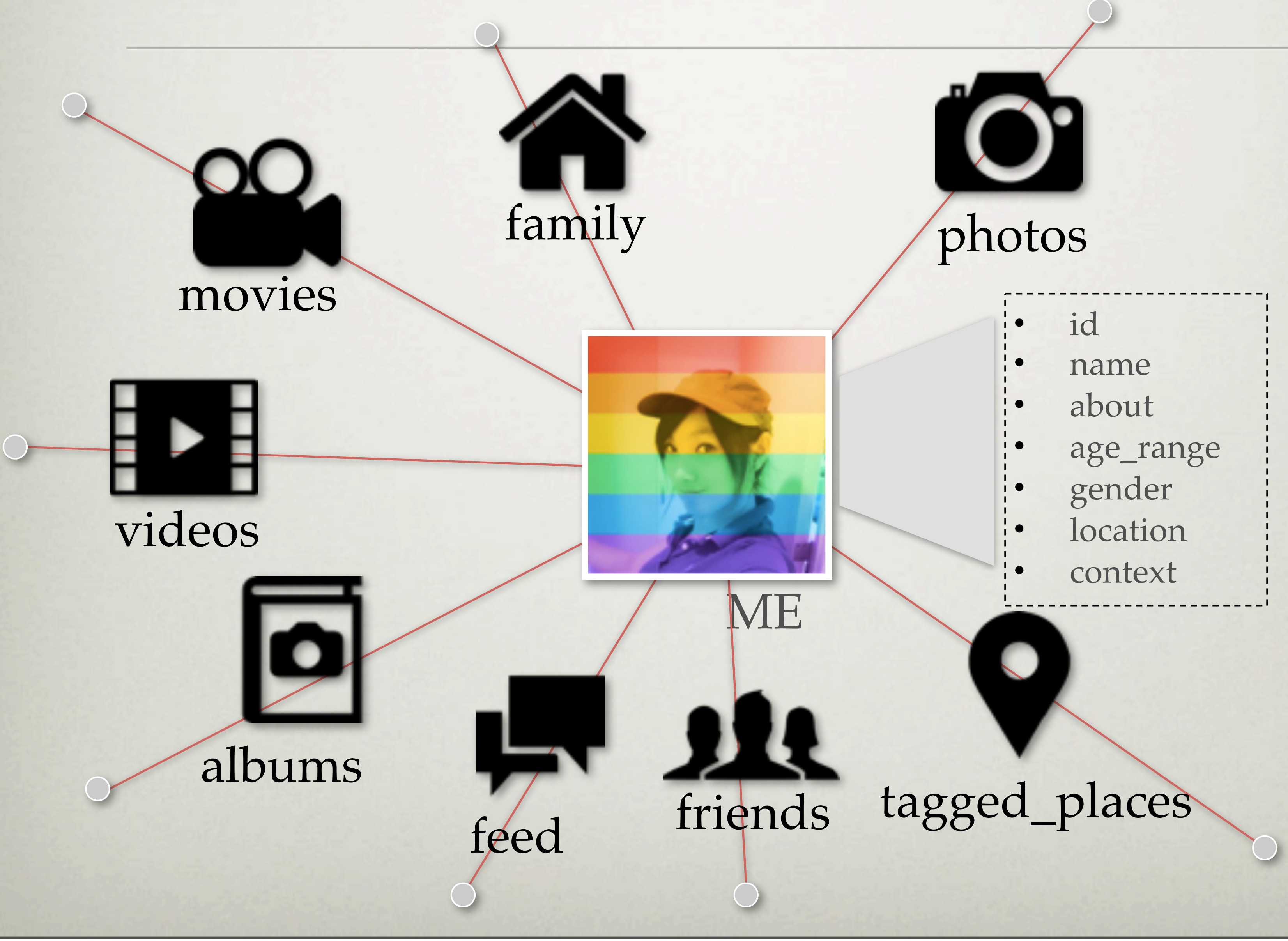
- https://developers.facebook.com/tools/explorer
- need a valid *access token*
- the token will expire in hour (and can be refreshed)

Baisc Syntax

- /<node_id>/<edge_name>?fields=f1,f2,...
- 釋例：me/photos?fields=from

- nodes
  - **objects**: a User, a Photo, a Page, a Comment, …
- edegs
  - **connections** between objects
- fields
  - **information** about objects

Components of the API



movies

family

photos

videos

albums

feed

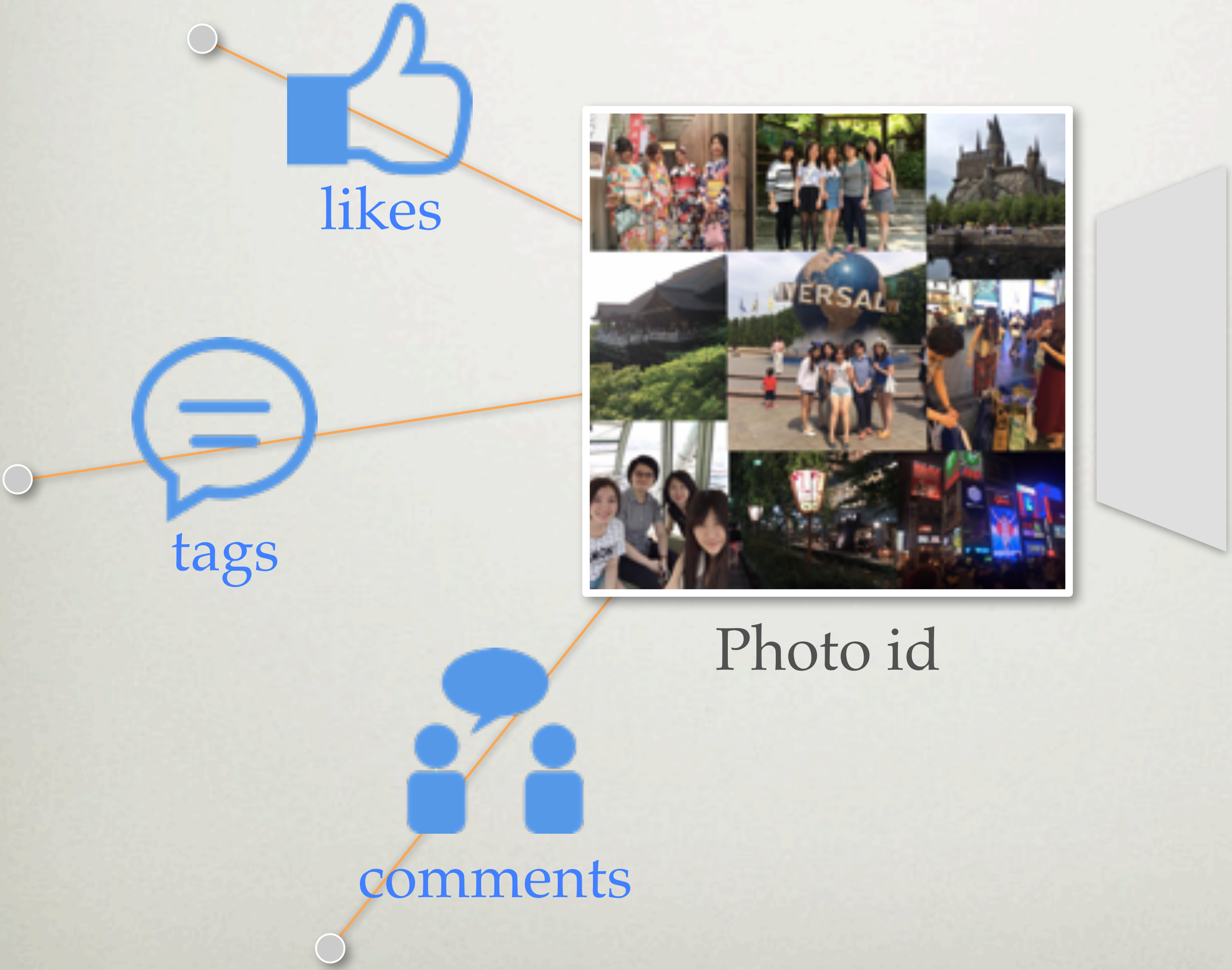
friends

tagged_places

ME

- id
- name
- about
- age_range
- gender
- location
- context

Node? Edge?



- id
- album
- Created_time
- from
- name
- link

Node? Edge?

- edge is the *population,* node is the *individual*
  - “all of my photos” => edge (no id)
  - “this photo of mine” => node (with id)

- difference in syntax (and returned json):
  - **field querying:** me?fields=photos.fields(from)
  - **edge querying:** me/photos?fields=from

Nested Query

允許巢狀結構

- <node_id>?fields=f1.fields(subf1,subf2,...),f2,...
- 釋例： me?fields=photos.fields(from,id)

Find all Fields/Edges

- <node_id>?metadata=1
- 釋例： me?metadata=1

- 常用：

photos, posts, feed, likes, reactions, tagged_place, comments

Limit Modifiers

limit: 限制單頁顯示資料個數

- on field:
  - me?fields=photos.limit(10)
- on edge:
  - me/photos?limit=10

Summary Modifiers

summary: 總計

- on field:
  - <photo_id>?
    fields=likes.summary(true)
- on edge:
  - <photo_id>/likes?summary=true

Since Modifiers

since: 設定起始日期，需使用unix時間格式

- on field:
  - won’t work
- on edge:
  - me/posts?since=1451648891

- what is unix time? (unix time converter)
  - total seconds elapsed since 1970-01-01 UTC time
  - 1451648891 is 2016-01-01 UTC

Query Modifiers

允許複數Modifiers

- on field:
  - <photo_id>?files=likes.limit(1).summary(true)
- on edge:
  - <photo_id>/likes?limit=1&summary=true

Try it :

找出是誰在按我的文章讚

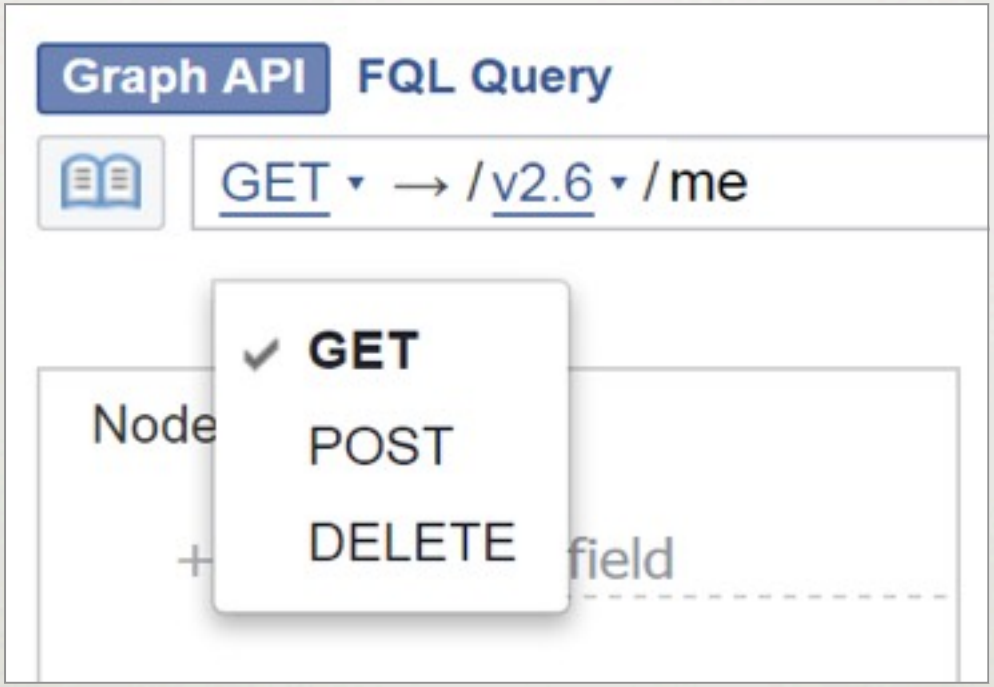
思考步驟

- start from *me*
- is posts an edge or a field of me?
    - either of the following query works:
    - me/posts
    - me?fields=posts
- pickup an arbitrary post
    - how many fields does it have?
    - use metadata=1 query string to investigate!
- is likes an edge or a field of posts?
- use the since modifier to specify a starting time
- use the summary modifier to get a total counts

參考解答

- 參考解答：
  me/posts?fields=likes.summary(true)&since=1451648891&limit=100

GET, POST, and DELETE



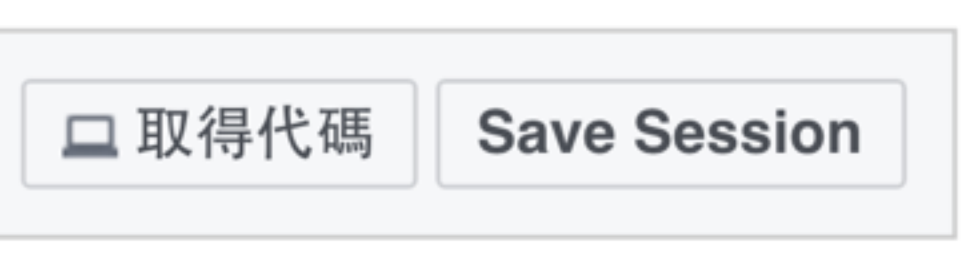
- GET: 查詢
- POST: OP文, 按讚
- DELETE: 刪文, 取消讚

Agenda

- Facebook Graph API Crawling (90 mins)
  - HTTP request
  - Exercise1: 抓取metadata fields
  - Exercise2: 抓取所有分頁結果
  - Exercise3: 我的按讚趨勢
  - Exercise4: Facebook機器人
  - Exercise5: 幫朋友按讚，成為狂讚士
  - Exercise6: 抓取所以文章comments紀錄

HTTP REQUEST

- Get method
- Facebook Graph API 右下角：取得代碼 (Get Code)



```
curl -i -X GET \
 "https://graph.facebook.com/v2.6/me?fields=id
%2Cname&access_token=EAACEdEose0cBAIVrZBQPdVwfFBoV1n1wHiJYhVS
5MJCOUsASmO2iRtZCJSdP8xU79JokvsBVOMWKpxbedrfVR5ybEpRl5EZB3uh
EgnVOXXPT8KgXk2gKpkZBVWe4ZAr1AJ32GajuFbyZAmJWBkHkkdNmQRg
MXn01CnrJoCpH5kTQZDZD"
```

Exercise1:
抓取”me”所有metadata fields

Exercise1: 思考步驟

- start from *me*
  - *me?metadata=1*
- 觀察http request
- Get method
- Data parser

Exercise1: 參考解答

- R code參考解答

Exercise2:
抓取所有分頁結果

Exercise2: 思考步驟

- 以一篇文章按讚者清單為例
  - *me/feed* （get post id）
  - *<post_id>/reactions* or *<post_id>?fields=likes*
- 觀察 資料位置 與 URL
  - 如果觀察不到分頁情況，可使用limit()設定
  - *<post_id>/reactions?limit=*2 or *<post_id>?fields=reactions.limit(*2*)*
- Get method
- Data parser

Exercise2: 參考解答

- R code參考解答

Exercise3:

我的按讚趨勢

Exercise3: 思考步驟

- 特定時間內，所有文章按讚趨勢圖
  - 需要設定時間、抓取所有文章 ID、文章按讚者資料
  - *me/feed?fields=created_time,reactions&since=1420070400*
- 觀察！！！
- Get method
- Data parser

Exercise3: 參考解答

- R code參考解答

Exercise4:
Facebook機器人

Exercise4: 思考步驟

- 自動PO文、按讚、刪文
  - me
- 觀察！！！
- POST method

Exercise4: 參考解答

- R code參考解答

Exercise5:

成為狂讚士

Exercise5: 思考步驟

- 挑選一位朋友，對其所有文章按讚
- 觀察！！！
- POST method

Exercise5: 參考解答

- R code參考解答

Exercise6:
抓取文章Comments紀錄

Exercise6: 思考步驟

- 挑選單篇文章，抓取所有Comments紀錄
- 觀察！！！
  - 注意：comments中的reply
- GET method
- Data parser

Exercise6: 參考解答

- R code參考解答

Agenda

- Regular Expression (60 mins)
  - What is Regex?
  - Why we need it?
  - Replace/ Filter功能
  - Basic Component
  - Exercise: 抓取日期資料格式
  - Exercise: 抓取PTT版特殊pattern

What is Regex?

…a sequence of characters that define a search pattern, mainly for use in pattern matching with strings, or string matching, i.e. "find and replace"-like operations.

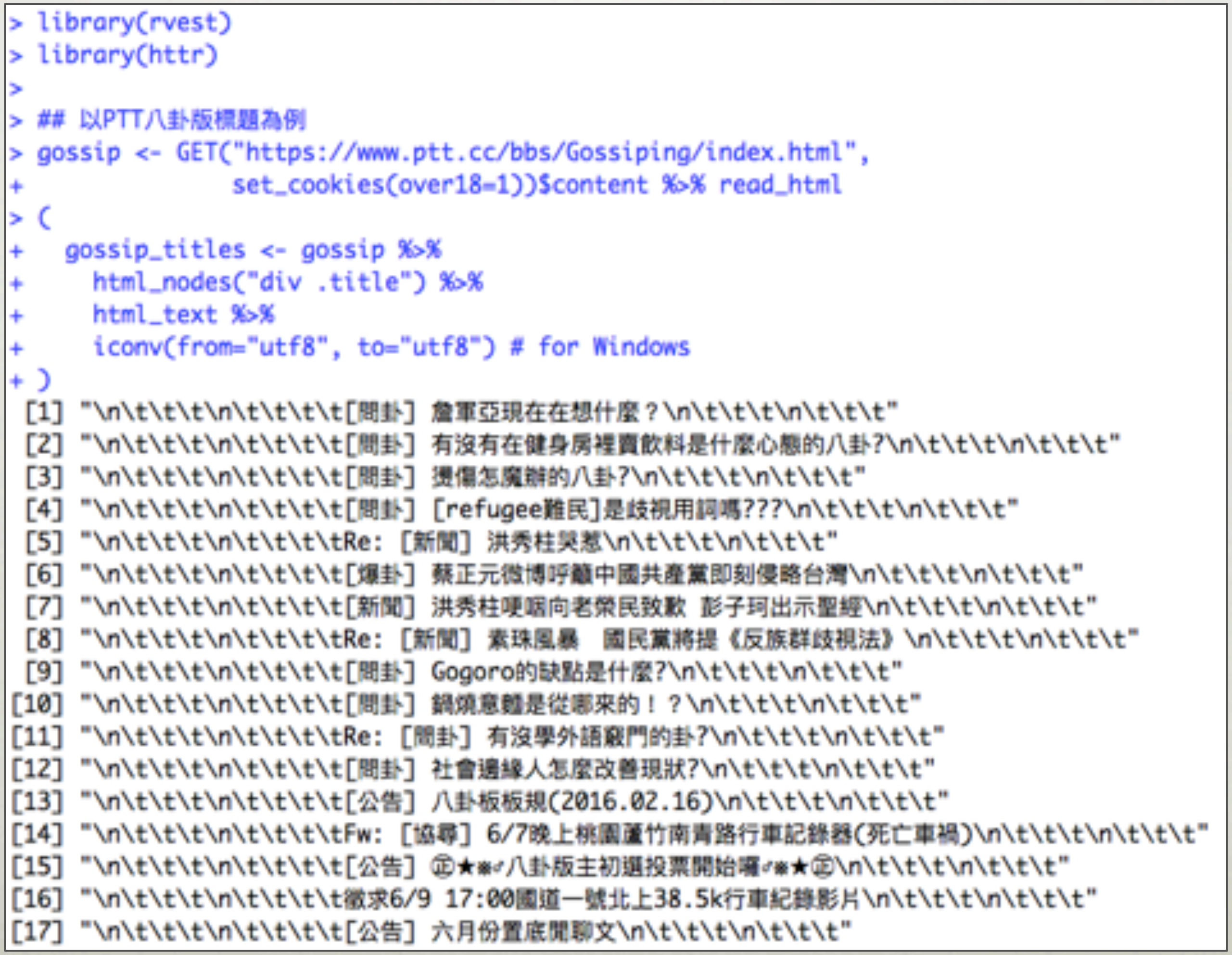
-wikipedia

Why we need Regex?

- 當資料結構混亂不一致，只用CSS selector/ Xpath selector無法有效取得資料。
- 處理大量字串時使用，例如log資料
- 需要抓取大量特定pattern時，例如日期、電話號碼、email、IP等

以PTT八卦版標題為例

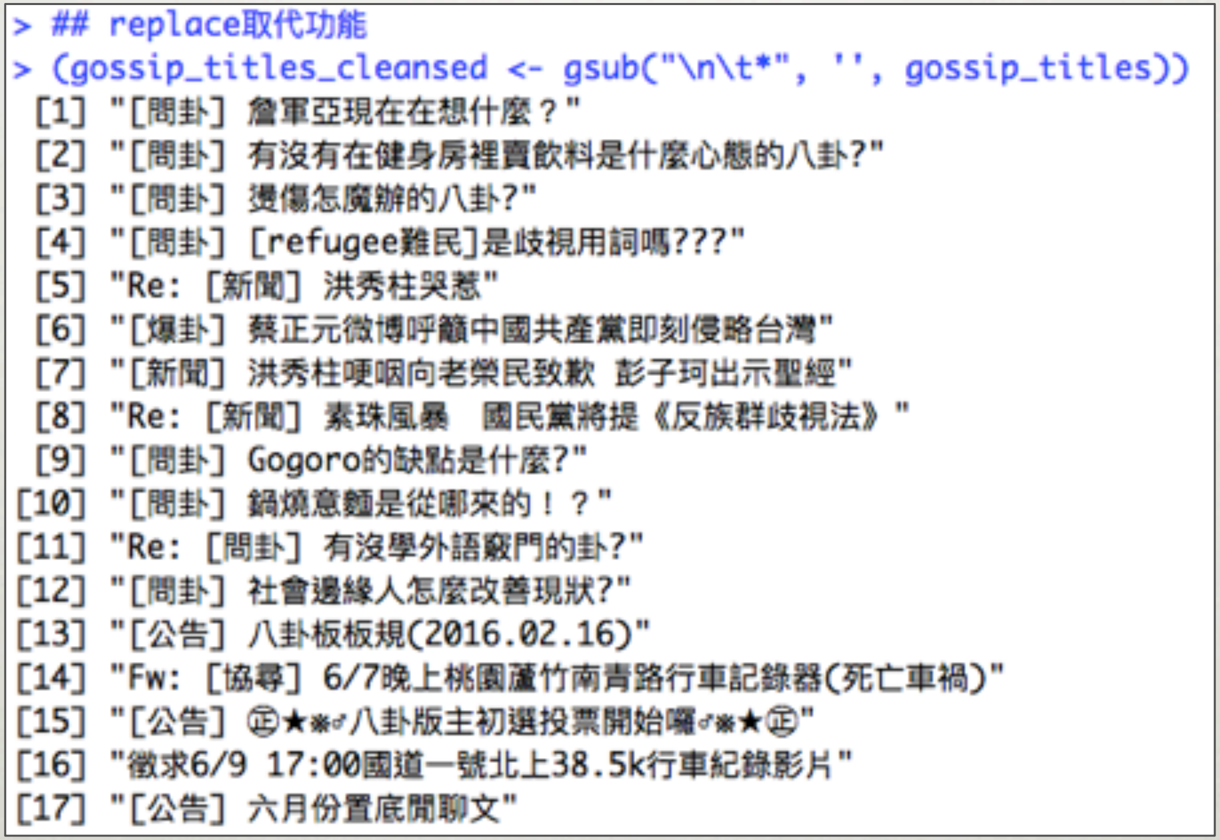
- R Code連結



```
> library(rvest)
> library(httr)
>
> ## 以PTT八卦版標題為例
> gossip <- GET("https://www.ptt.cc/bbs/Gossiping/index.html",
+               set_cookies(over18=1))$content %>% read_html
> (
+   gossip_titles <- gossip %>%
+     html_nodes("div .title") %>%
+     html_text %>%
+     iconv(from="utf8", to="utf8") # for Windows
+ )
 [1] "\n\t\t\t\n\t\t\t\t[問卦] 詹軍亞現在在想什麼？\n\t\t\t\n\t\t\t"
 [2] "\n\t\t\t\n\t\t\t\t[問卦] 有沒有在健身房裡賣飲料是什麼心態的八卦?\n\t\t\t\n\t\t\t"
 [3] "\n\t\t\t\n\t\t\t\t[問卦] 燙傷怎麼辦的八卦?\n\t\t\t\n\t\t\t"
 [4] "\n\t\t\t\n\t\t\t\t[問卦] [refugee難民]是歧視用詞嗎???\n\t\t\t\n\t\t\t"
 [5] "\n\t\t\t\n\t\t\t\tRe: [新聞] 洪秀柱哭惹\n\t\t\t\n\t\t\t"
 [6] "\n\t\t\t\n\t\t\t\t[爆卦] 蔡正元微博呼籲中國共產黨即刻侵略台灣\n\t\t\t\n\t\t\t"
 [7] "\n\t\t\t\n\t\t\t\t[新聞] 洪秀柱哽咽向老榮民致歉 彭子珂出示聖經\n\t\t\t\n\t\t\t"
 [8] "\n\t\t\t\n\t\t\t\tRe: [新聞] 素珠風暴  國民黨將提《反族群歧視法》\n\t\t\t\n\t\t\t"
 [9] "\n\t\t\t\n\t\t\t\t[問卦] Gogoro的缺點是什麼?\n\t\t\t\n\t\t\t"
[10] "\n\t\t\t\n\t\t\t\t[問卦] 鍋燒意麵是從哪來的！？\n\t\t\t\n\t\t\t"
[11] "\n\t\t\t\n\t\t\t\tRe: [問卦] 有沒學外語竅門的卦?\n\t\t\t\n\t\t\t"
[12] "\n\t\t\t\n\t\t\t\t[問卦] 社會邊緣人怎麼改善現狀?\n\t\t\t\n\t\t\t"
[13] "\n\t\t\t\n\t\t\t\t[公告] 八卦板板規(2016.02.16)\n\t\t\t\n\t\t\t"
[14] "\n\t\t\t\n\t\t\t\tFw: [協尋] 6/7晚上桃園蘆竹南青路行車記錄器(死亡車禍)\n\t\t\t\n\t\t\t"
[15] "\n\t\t\t\n\t\t\t\t[公告] ㊣★※♂八卦版主初選投票開始囉♂※★㊣\n\t\t\t\n\t\t\t"
[16] "\n\t\t\t\n\t\t\t\t徵求6/9 17:00國道一號北上38.5k行車紀錄影片\n\t\t\t\n\t\t\t"
[17] "\n\t\t\t\n\t\t\t\t[公告] 六月份置底閒聊文\n\t\t\t\n\t\t\t"
```

Replace 取代功能

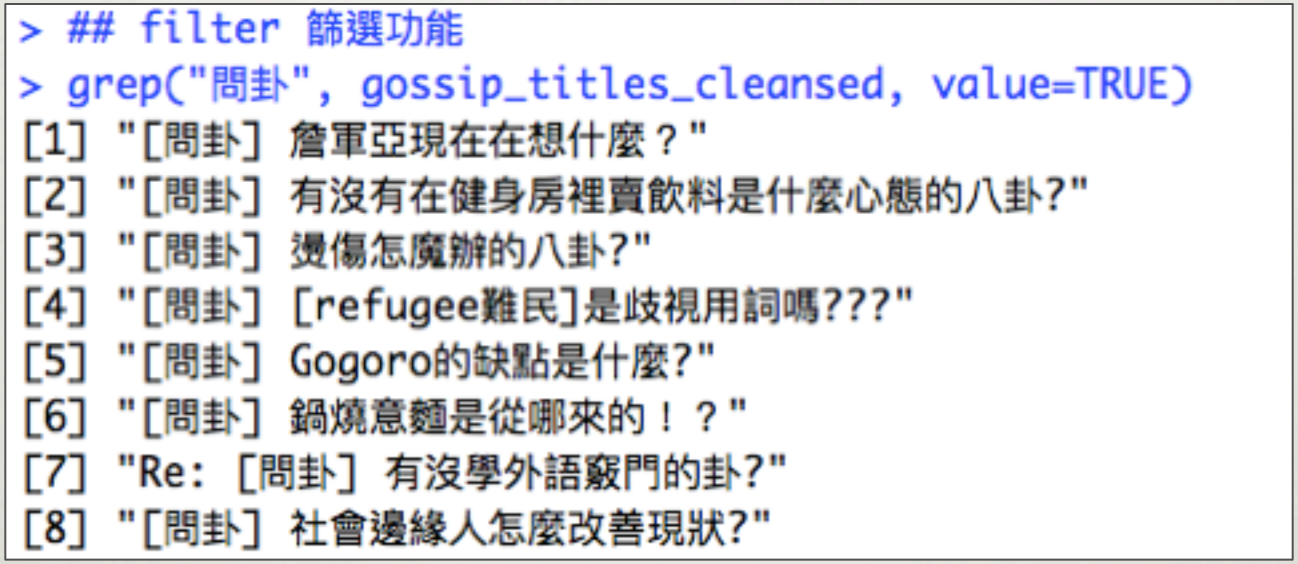
- 取代將標題中的”\n與\t”



```
> ## replace取代功能
> (gossip_titles_cleansed <- gsub("\n\t*", '', gossip_titles))
 [1] "[問卦] 詹軍亞現在在想什麼？"
 [2] "[問卦] 有沒有在健身房裡賣飲料是什麼心態的八卦?"
 [3] "[問卦] 燙傷怎魔辦的八卦?"
 [4] "[問卦] [refugee難民]是歧視用詞嗎???"
 [5] "Re: [新聞] 洪秀柱哭惹"
 [6] "[爆卦] 蔡正元微博呼籲中國共產黨即刻侵略台灣"
 [7] "[新聞] 洪秀柱哽咽向老榮民致歉 彭子珂出示聖經"
 [8] "Re: [新聞] 素珠風暴  國民黨將提《反族群歧視法》"
 [9] "[問卦] Gogoro的缺點是什麼?"
[10] "[問卦] 鍋燒意麵是從哪來的！？"
[11] "Re: [問卦] 有沒學外語竅門的卦?"
[12] "[問卦] 社會邊緣人怎麼改善現狀?"
[13] "[公告] 八卦板板規(2016.02.16)"
[14] "Fw: [協尋] 6/7晚上桃園蘆竹南青路行車記錄器(死亡車禍)"
[15] "[公告] ㊣★※♂八卦版主初選投票開始囉♂※★㊣"
[16] "徵求6/9 17:00國道一號北上38.5k行車紀錄影片"
[17] "[公告] 六月份置底閒聊文"
```

FILTER 篩選功能

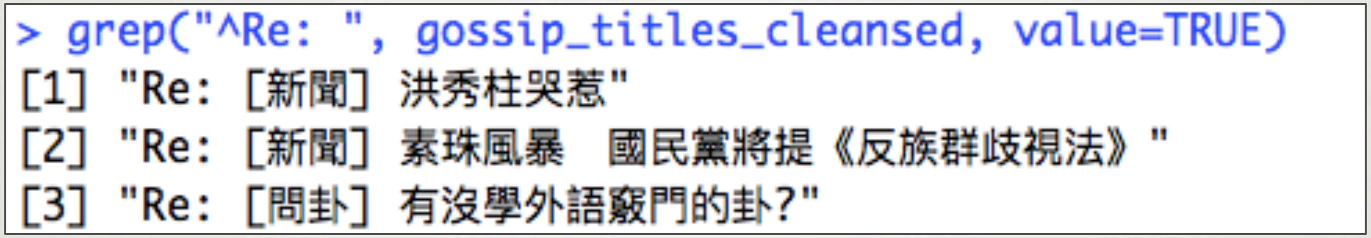
- 抓取含"問卦"的標題



```
> ## filter 篩選功能
> grep("問卦", gossip_titles_cleansed, value=TRUE)
[1] "[問卦] 詹軍亞現在在想什麼？"
[2] "[問卦] 有沒有在健身房裡賣飲料是什麼心態的八卦?"
[3] "[問卦] 燙傷怎魔辦的八卦?"
[4] "[問卦] [refugee難民]是歧視用詞嗎???"
[5] "[問卦] Gogoro的缺點是什麼?"
[6] "[問卦] 鍋燒意麵是從哪來的！？"
[7] "Re: [問卦] 有沒學外語竅門的卦?"
[8] "[問卦] 社會邊緣人怎麼改善現狀?"
```

Filter 篩選功能

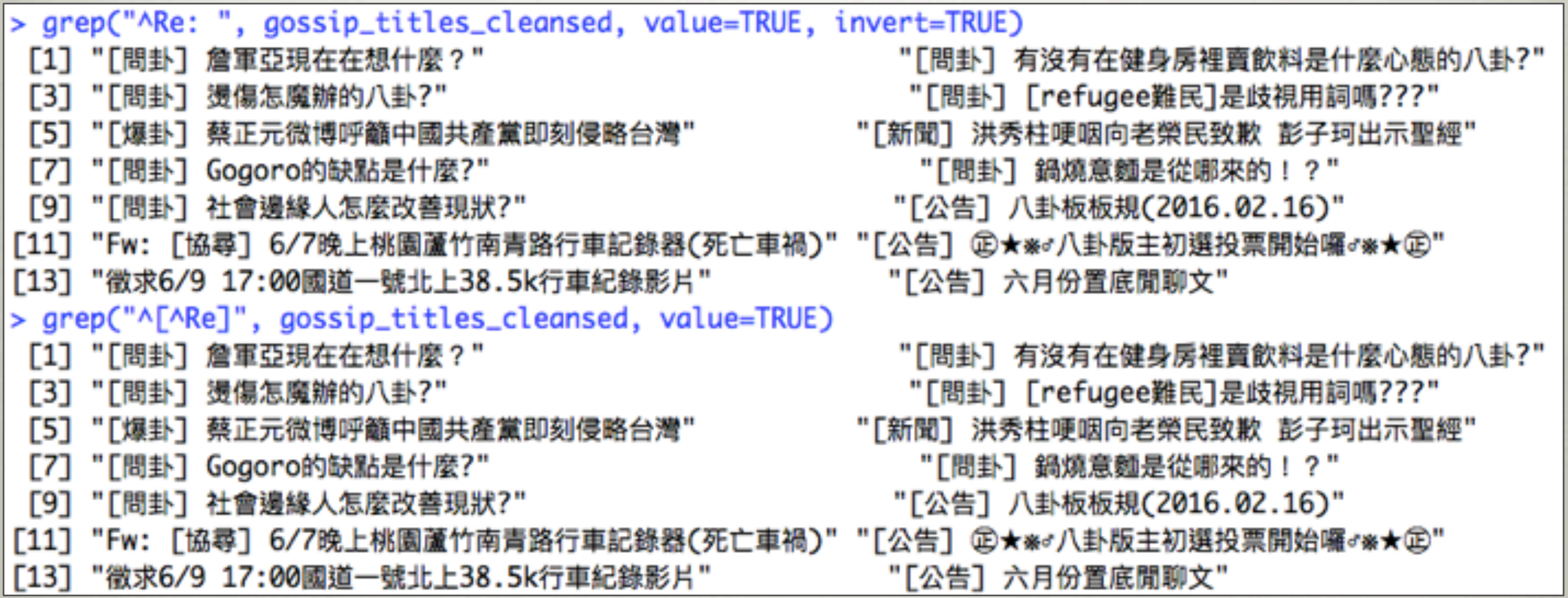
- 抓取”Re開頭”的標題



```
> grep("^Re: ", gossip_titles_cleansed, value=TRUE)
[1] "Re: [新聞] 洪秀柱哭惹"
[2] "Re: [新聞] 素珠風暴　國民黨將提《反族群歧視法》 "
[3] "Re: [問卦] 有沒學外語竅門的卦?"
```

Filter 篩選功能

- 抓取”非Re開頭”的標題



```
> grep("^Re: ", gossip_titles_cleansed, value=TRUE, invert=TRUE)
 [1] "[問卦] 詹軍亞現在在想什麼？"                            "[問卦] 有沒有在健身房裡賣飲料是什麼心態的八卦?"
 [3] "[問卦] 燙傷怎麼辦的八卦?"                                "[問卦] [refugee難民]是歧視用詞嗎???"
 [5] "[爆卦] 蔡正元微博呼籲中國共產黨即刻侵略台灣"           "[新聞] 洪秀柱哽咽向老榮民致歉 彭子珂出示聖經"
 [7] "[問卦] Gogoro的缺點是什麼?"                              "[問卦] 鍋燒意麵是從哪來的！？"
 [9] "[問卦] 社會邊緣人怎麼改善現狀?"                         "[公告] 八卦板板規(2016.02.16)"
[11] "Fw: [協尋] 6/7晚上桃園蘆竹南青路行車記錄器(死亡車禍)" "[公告] ㊣★※♂八卦版主初選投票開始囉♂※★㊣"
[13] "徵求6/9 17:00國道一號北上38.5k行車紀錄影片"             "[公告] 六月份置底閒聊文"
> grep("^[^Re]", gossip_titles_cleansed, value=TRUE)
 [1] "[問卦] 詹軍亞現在在想什麼？"                            "[問卦] 有沒有在健身房裡賣飲料是什麼心態的八卦?"
 [3] "[問卦] 燙傷怎麼辦的八卦?"                                "[問卦] [refugee難民]是歧視用詞嗎???"
 [5] "[爆卦] 蔡正元微博呼籲中國共產黨即刻侵略台灣"           "[新聞] 洪秀柱哽咽向老榮民致歉 彭子珂出示聖經"
 [7] "[問卦] Gogoro的缺點是什麼?"                              "[問卦] 鍋燒意麵是從哪來的！？"
 [9] "[問卦] 社會邊緣人怎麼改善現狀?"                         "[公告] 八卦板板規(2016.02.16)"
[11] "Fw: [協尋] 6/7晚上桃園蘆竹南青路行車記錄器(死亡車禍)" "[公告] ㊣★※♂八卦版主初選投票開始囉♂※★㊣"
[13] "徵求6/9 17:00國道一號北上38.5k行車紀錄影片"             "[公告] 六月份置底閒聊文"
```

Basic components

- 字面字元 (string-literals)

例如：ABCabc123

- 詮釋字元 (meta-characters)

例如：. ^ $ * + ? { } [ ] \ | ( )

- 特殊字元 (short-cuts)

例如：\d \D \s \S \w \W \b \B

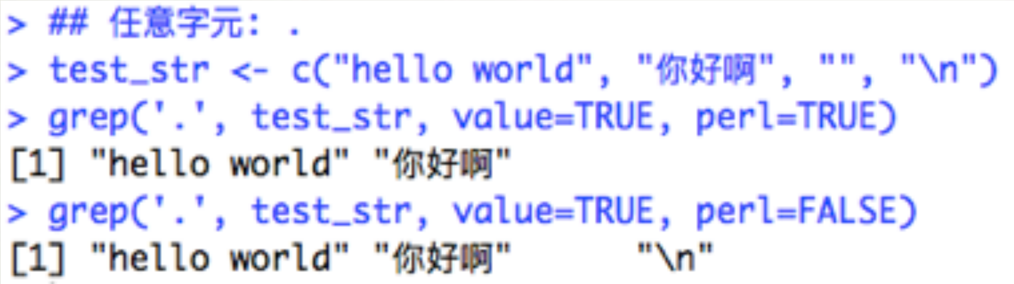
Basic components

- R Code連結

Meta-characters .

- 任意字元：符合任意字元，不包含空字元

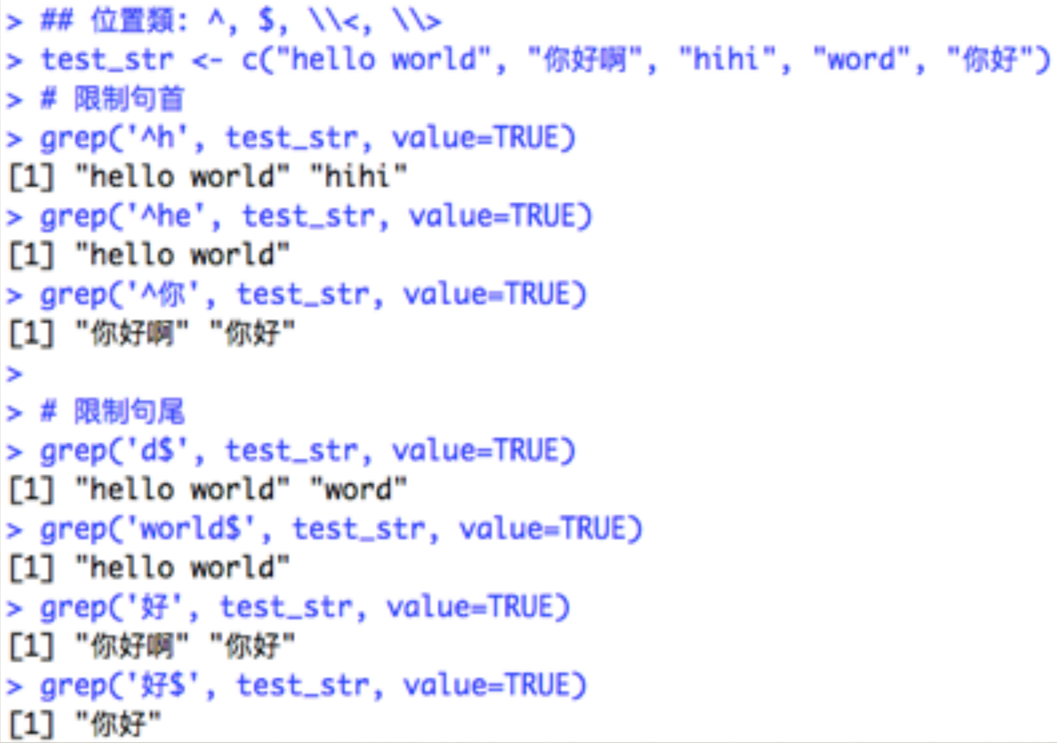
範例



```
> ## 任意字元: .
> test_str <- c("hello world", "你好啊", "", "\n")
> grep('.', test_str, value=TRUE, perl=TRUE)
[1] "hello world" "你好啊"
> grep('.', test_str, value=TRUE, perl=FALSE)
[1] "hello world" "你好啊"      "\n"
```

Meta-characters ^,$,\\<,\\>

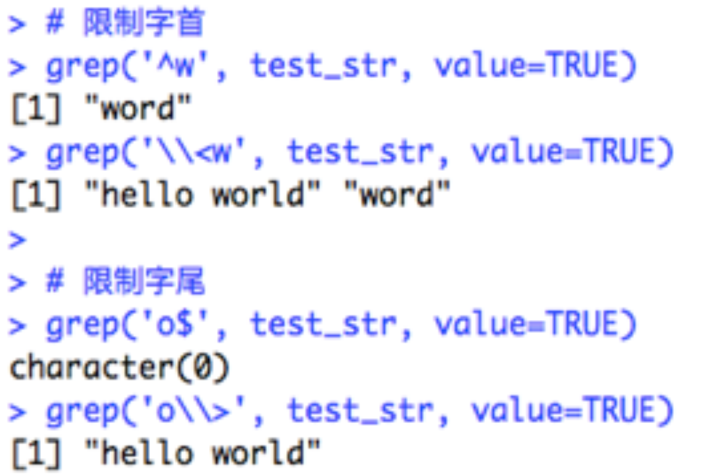
- 位置類：出現為行首行尾、字首字尾



```
> ## 位置類: ^, $, \\<, \\>
> test_str <- c("hello world", "你好啊", "hihi", "word", "你好")
> # 限制句首
> grep('^h', test_str, value=TRUE)
[1] "hello world" "hihi"
> grep('^he', test_str, value=TRUE)
[1] "hello world"
> grep('^你', test_str, value=TRUE)
[1] "你好啊" "你好"
>
> # 限制句尾
> grep('d$', test_str, value=TRUE)
[1] "hello world" "word"
> grep('world$', test_str, value=TRUE)
[1] "hello world"
> grep('好', test_str, value=TRUE)
[1] "你好啊" "你好"
> grep('好$', test_str, value=TRUE)
[1] "你好"
```

Meta-characters ^,$,\\<,\\>

- 位置類：出現為行首行尾、字首字尾

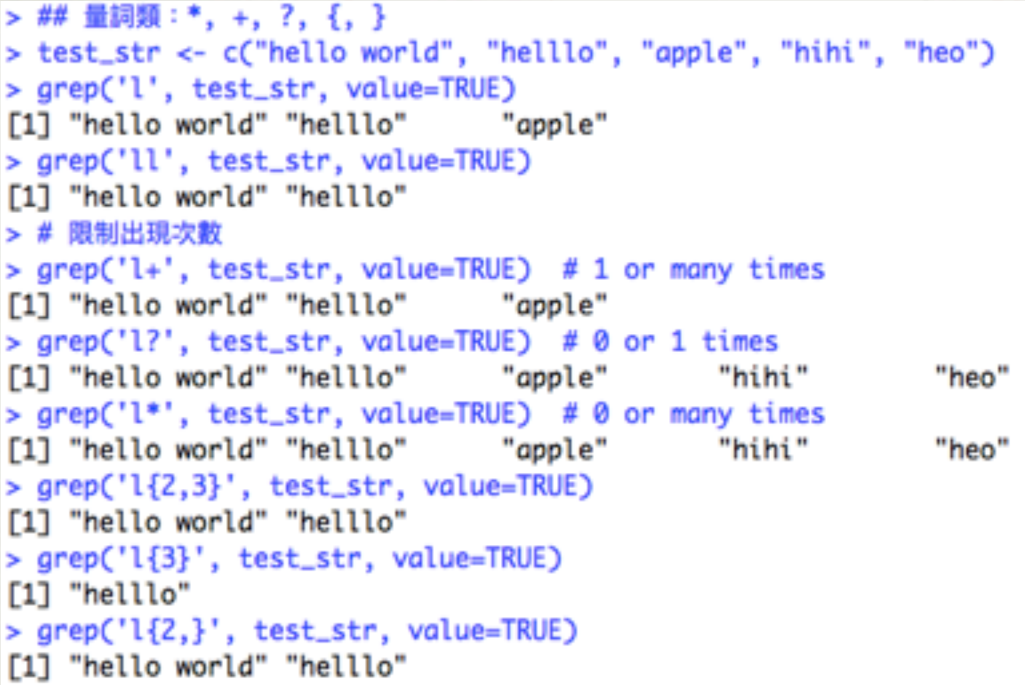


```
> # 限制字首
> grep('^w', test_str, value=TRUE)
[1] "word"
> grep('\\<w', test_str, value=TRUE)
[1] "hello world" "word"
>
> # 限制字尾
> grep('o$', test_str, value=TRUE)
character(0)
> grep('o\\>', test_str, value=TRUE)
[1] "hello world"
```

Meta-characters * + ? { }

- 量詞類：滿足出現特定次數

範例



```
> ## 量詞類：*, +, ?, {, }
> test_str <- c("hello world", "helllo", "apple", "hihi", "heo")
> grep('l', test_str, value=TRUE)
[1] "hello world" "helllo"      "apple"
> grep('ll', test_str, value=TRUE)
[1] "hello world" "helllo"
> # 限制出現次數
> grep('l+', test_str, value=TRUE)  # 1 or many times
[1] "hello world" "helllo"      "apple"
> grep('l?', test_str, value=TRUE)  # 0 or 1 times
[1] "hello world" "helllo"      "apple"       "hihi"        "heo"
> grep('l*', test_str, value=TRUE)  # 0 or many times
[1] "hello world" "helllo"      "apple"       "hihi"        "heo"
> grep('l{2,3}', test_str, value=TRUE)
[1] "hello world" "helllo"
> grep('l{3}', test_str, value=TRUE)
[1] "helllo"
> grep('l{2,}', test_str, value=TRUE)
[1] "hello world" "helllo"
```

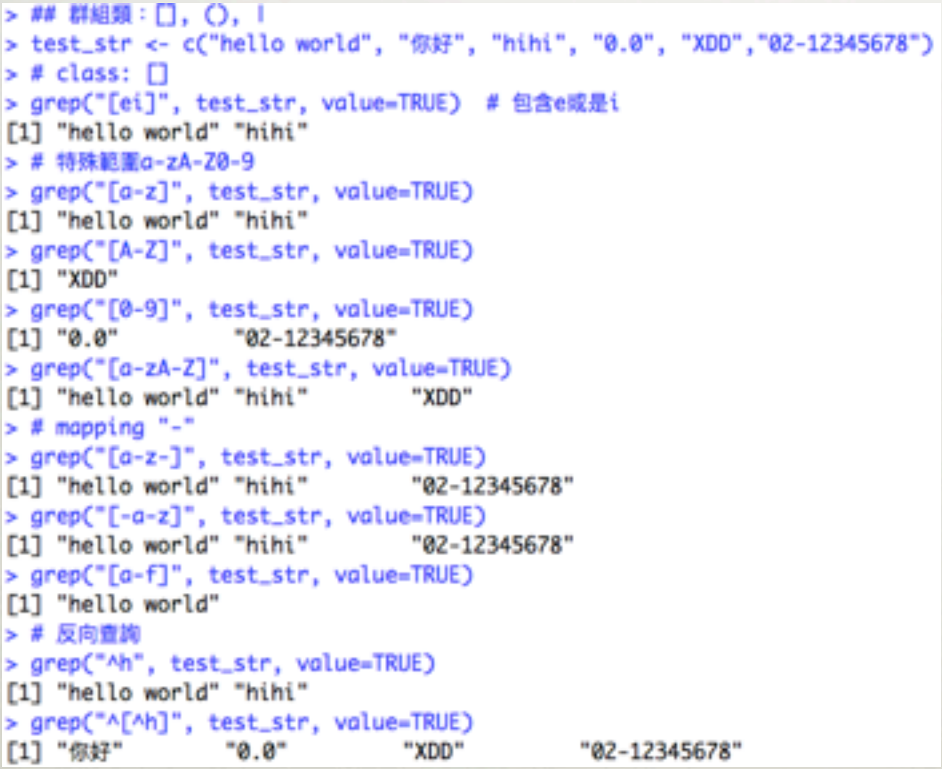
Meta-characters * + ? { }

- 量詞類：滿足出現特定次數

量詞		
?	問號(question)	允許0個或1個
*	星號(star)	允許0個或多個
+	加號(plus)	允許1個或多個
{下限, 上限}	指定範圍(specified range)	至少要下限，最多允許上限

Meta-characters [], (), |

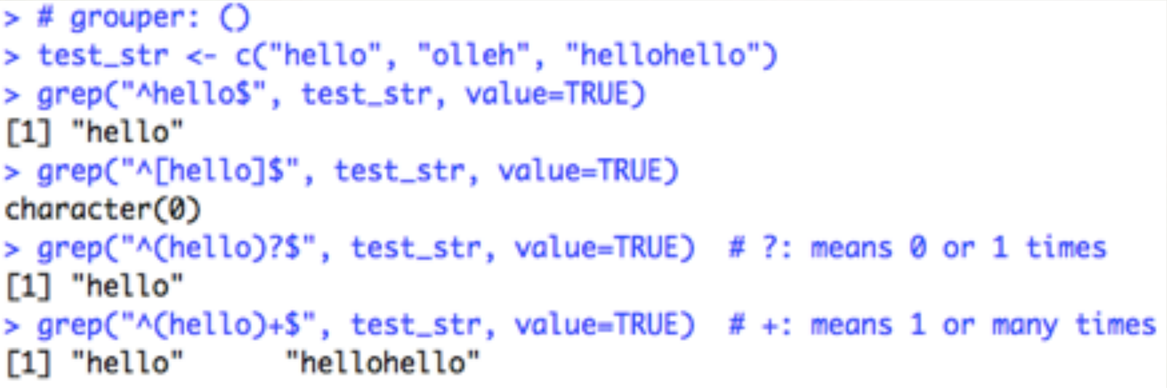
- 群組類：允許特定特定範圍、特定字母



```
> ## 群組類：[], (), |
> test_str <- c("hello world", "你好", "hihi", "0.0", "XDD","02-12345678")
> # class: []
> grep("[ei]", test_str, value=TRUE)  # 包含e或是i
[1] "hello world" "hihi"
> # 特殊範圍a-zA-Z0-9
> grep("[a-z]", test_str, value=TRUE)
[1] "hello world" "hihi"
> grep("[A-Z]", test_str, value=TRUE)
[1] "XDD"
> grep("[0-9]", test_str, value=TRUE)
[1] "0.0"         "02-12345678"
> grep("[a-zA-Z]", test_str, value=TRUE)
[1] "hello world" "hihi"        "XDD"
> # mapping "-"
> grep("[a-z-]", test_str, value=TRUE)
[1] "hello world" "hihi"        "02-12345678"
> grep("[-a-z]", test_str, value=TRUE)
[1] "hello world" "hihi"        "02-12345678"
> grep("[a-f]", test_str, value=TRUE)
[1] "hello world"
> # 反向查詢
> grep("^h", test_str, value=TRUE)
[1] "hello world" "hihi"
> grep("^[^h]", test_str, value=TRUE)
[1] "你好"        "0.0"         "XDD"         "02-12345678"
```

Meta-characters [], (), |

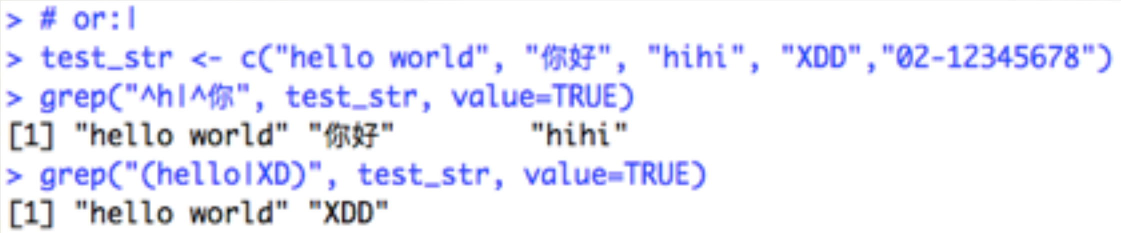
- 群組類：允許特定特定範圍、特定字母



```
> # grouper: ()
> test_str <- c("hello", "olleh", "hellohello")
> grep("^hello$", test_str, value=TRUE)
[1] "hello"
> grep("^[hello]$", test_str, value=TRUE)
character(0)
> grep("^(hello)?$", test_str, value=TRUE)  # ?: means 0 or 1 times
[1] "hello"
> grep("^(hello)+$", test_str, value=TRUE)  # +: means 1 or many times
[1] "hello"      "hellohello"
```

Meta-characters [], (), |

- 群組類：允許特定特定範圍、特定字母

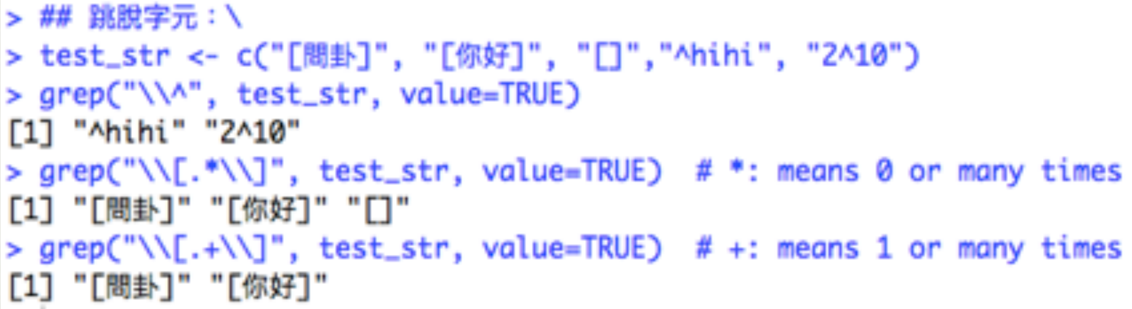


```
> # or:|
> test_str <- c("hello world", "你好", "hihi", "XDD","02-12345678")
> grep("^h|^你", test_str, value=TRUE)
[1] "hello world" "你好"        "hihi"
> grep("(hello|XD)", test_str, value=TRUE)
[1] "hello world" "XDD"
```

Meta-characters \\

- 跳脫字元：跳脫預設功能

範例



```
> ## 跳脫字元：\
> test_str <- c("[閒卦]", "[你好]", "[]","^hihi", "2^10")
> grep("\\^", test_str, value=TRUE)
[1] "^hihi" "2^10"
> grep("\\[.*\\]", test_str, value=TRUE)  # *: means 0 or many times
[1] "[閒卦]" "[你好]" "[]"
> grep("\\[.+\\]", test_str, value=TRUE)  # +: means 1 or many times
[1] "[閒卦]" "[你好]"
```

特殊字元

- \\d: any decimal digit => [0-9]
- \\D: any non-digit => [^0-9]
- \\s: any white space => [\t\n\r\f\v]
- \\S: any non-white space => [^ \t\n\r\f\v]
- \\w: any alphanumeric => [a-zA-Z0-9_]
- \\W: any non-alphanumeric => [^a-zA-Z0-9_]
- \\b: matches the empty string, but only at the beginning or end of a word (\\w)

RegEx in R

- base::grep and base::grepl
  基本篩選功能
- base::gsub
  基本取代功能
- stringr package
  進階篩選功能
  (str_extract , str_extract_all ,str_match and str_match_all)

- R code連結

Exercise:
抓取日期資料格式

Exercise: 參考解答

- Data:

02/22/16

02-22-16

02.22.16

- R Code連結

Exercise:
抓取PTT版特殊pattern

Exercise: 參考解答

R Code連結