分詞 斷詞

In [8]:

```
import urllib.request
import jieba
import matplotlib.pyplot as plt
from wordcloud import WordCloud
import numpy as np
import pandas as pd
```

TF-IDF

In [9]:

```
import jieba.analyse
url = "https://raw.githubusercontent.com/cjwu/cjwu.github.io/master/courses/n
text = urllib.request.urlopen(url).read().decode("utf-8")
result = jieba.analyse.extract_tags(text, topK=100, withWeight=True)

for i in result:
    print('word:', i[0], 'TF-IDF:', i[1])
```

```
Building prefix dict from the default dictionary ...
Dumping model to file cache /var/folders/ry/phxc250s3lx_m2h646xkjb1h0000gn/T/j
ieba.cache
Loading model cost 0.682 seconds.
Prefix dict has been built successfully.
word: 什麼 TF-IDF: 0.19464509600157406
word: 八卦 TF-IDF: 0.1942050114055646
word: 台灣 TF-IDF: 0.12081408131496592
word: 怎麼 TF-IDF: 0.11186701189176337
word: 肥宅 TF-IDF: 0.07336596927026089
word: 現在 TF-IDF: 0.03800903805058438
word: 不會 TF-IDF: 0.036229733848907944
word: 還是 TF-IDF: 0.03568381096884812
word: 是不是 TF-IDF: 0.0355764318510935
word: 一個 TF-IDF: 0.03550183667549485
word: 中國 TF-IDF: 0.034474018907481
word: 這樣 TF-IDF: 0.0325902479818425
word: 怎樣 TF-IDF: 0.029860633581543415
word: 時候 TF-IDF: 0.02967528939386879
word: 一樣 TF-IDF: 0.028691280252032575
word: 真的 TF-IDF: 0.026822655692091746
word: 沒有 TF-IDF: 0.026099831518909124
word: ... TF-IDF: 0.02543259244328046
word: 應該 TF-IDF: 0.02359936993246231
word: 可以 TF-IDF: 0.022958278161416924
word: 喜歡 TF-IDF: 0.02285462328744244
word: 因為 TF-IDF: 0.022416537025666042
word: 一堆 TF-IDF: 0.021680457381807062
word: 問題 TF-IDF: 0.020815837223021518
word: 感覺 TF-IDF: 0.020714740393380813
word: 哪個 TF-IDF: 0.020643972612632316
word: 女生 TF-IDF: 0.0205553136507414 5
word: 這麼 TF-IDF: 0.019963253959718225
word: 覺得 TF-IDF: 0.01862540591413954
word: 這種 TF-IDF: 0.01824460785582621
word: 美國 TF-IDF: 0.018055893773830226
word: 正妹 TF-IDF: 0.016286699255117856
word: 知道 TF-IDF: 0.015811205555870015
word: 其實 TF-IDF: 0.01557902144763291
word: 為何 TF-IDF: 0.015043208250537165
```

```
word: 還有 TF-IDF: 0.014736547867293687
word: 東西 TF-IDF: 0.014719698395686903
word: 比較 TF-IDF: 0.01451750473640549
word: 那麼 TF-IDF: 0.014369229386265786
word: 到底 TF-IDF: 0.013855929767928526
word: 有人 TF-IDF: 0.013674560834915146
word: 自己 TF-IDF: 0.013466423468451874
word: QQ TF-IDF: 0.013445878342213998
word: 時間 TF-IDF: 0.013361630984180075
word: 開始 TF-IDF: 0.013206615845397658
word: 這個 TF-IDF: 0.012974093137224033
word: 哪裡 TF-IDF: 0.01282581778708433
word: 不是 TF-IDF: 0.012758688475732879
word: 沒人 TF-IDF: 0.012552856347054421
word: 甚麼 TF-IDF: 0.012468608989020498
word: 出來 TF-IDF: 0.012320333638880795
word: 日本 TF-IDF: 0.012127572493474573
word: 那個 TF-IDF: 0.012054111987493602
word: 發現 TF-IDF: 0.011993453889709177
word: 中國人 TF-IDF: 0.011976604418102393
word: 國家 TF-IDF: 0.011976604418102393
word: 如果 TF-IDF: 0.01195142064483163
word: 不要 TF-IDF: 0.011371977189677437
word: 就是 TF-IDF: 0.011178381502120506
word: 他們 TF-IDF: 0.010851059714769191
word: 大家 TF-IDF: 0.010829155333451544
word: 朋友 TF-IDF: 0.010388757432612226
word: 很多 TF-IDF: 0.010254718969681581
word: 台北 TF-IDF: 0.010090151731776114
word: 已經 TF-IDF: 0.009971517296895042
word: 老師 TF-IDF: 0.009917598987753332
word: 大學 TF-IDF: 0.00989737962182519
word: 變成 TF-IDF: 0.009695185962543778
word: 邊緣 TF-IDF: 0.009459293360048795
word: 我們 TF-IDF: 0.0093885255793003
word: 結果 TF-IDF: 0.0093885255793003
word: 遊戲 TF-IDF: 0.009065015724450039
word: 不用 TF-IDF: 0.00885884470039901
word: 手機 TF-IDF: 0.008744875763921134
word: 一點 TF-IDF: 0.008717916609350278
word: 看到 TF-IDF: 0.008670900198239207
word: 多少 TF-IDF: 0.008670105043160773
word: 男生 TF-IDF: 0.00855216230787021
word: 別人 TF-IDF: 0.008535942315997006
word: 當然 TF-IDF: 0.008482024006855295
word: 10 TF-IDF: 0.008434845486356299
word: 女友 TF-IDF: 0.008252324115040545
word: 如何 TF-IDF: 0.008088109356327164
word: 還好 TF-IDF: 0.008067527005328399
word: 電影 TF-IDF: 0.008013608696186688
word: 新聞 TF-IDF: 0.008013608696186688
word: 還要 TF-IDF: 0.008003499013222618
word: 韓國 TF-IDF: 0.007986649541615832
word: 鄉民 TF-IDF: 0.007952950598402264
word: 起來 TF-IDF: 0.007770976305048992
word: 根本 TF-IDF: 0.007719534763027918
word: XD TF-IDF: 0.007700208524300497
word: 好吃 TF-IDF: 0.007579158623013259
word: 妹妹 TF-IDF: 0.007550327706300244
word: 的掛 TF-IDF: 0.0075384535968753665
word: 不過 TF-IDF: 0.0073497395148793805
word: 一直 TF-IDF: 0.007303212591178013
word: .. TF-IDF: 0.0072486426852386735
```

```
word: ptt TF-IDF: 0.007110477018063041
word: 最強 TF-IDF: 0.006979051139530123
```

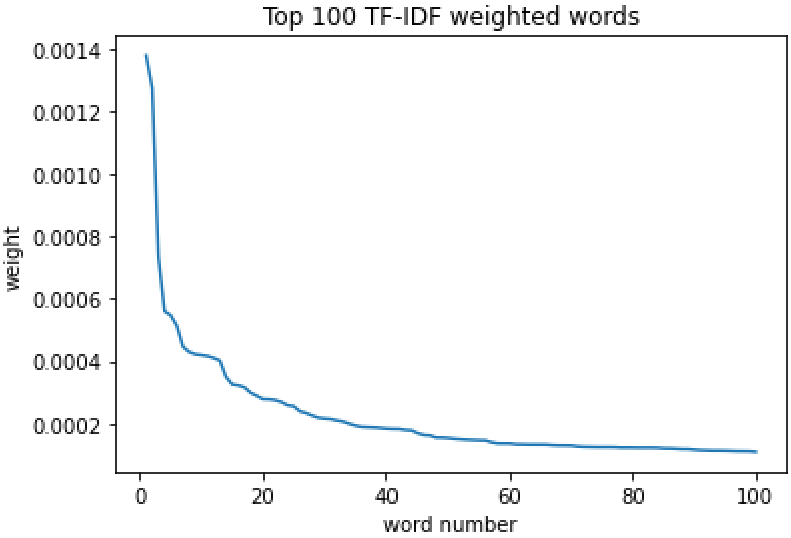
fig#1 x軸: 字詞編號, y軸: 權重

In [ ]:

```
import nltk
nltk.download('stopwords')
nltk.download('punkt')
```

In [5]:

```
import urllib.request
import numpy as np
import pandas as pd
from sklearn.feature_extraction.text import TfidfVectorizer
import matplotlib.pyplot as plt

url = "https://raw.githubusercontent.com/cjwu/cjwu.github.io/master/courses/n
response = urllib.request.urlopen(url)
text = response.read().decode()
docs = text.split('\n') # 一行算一個文章

vectorizer = TfidfVectorizer() # 算TF-IDF權重
X = vectorizer.fit_transform(docs)

weights = np.asarray(X.mean(axis=0)).ravel().tolist() # 所有字詞的TF-IDF權重平均
df = pd.DataFrame({'term': vectorizer.get_feature_names(), 'weight': weights}
df = df.sort_values(by='weight', ascending=False)


top100 = df[:100] # 取前100個

plt.plot(range(1, 101), top100['weight'], marker='', linestyle='-')
plt.title('Top 100 TF-IDF weighted words')
plt.xlabel('word number')
plt.ylabel('weight')
plt.show()
```

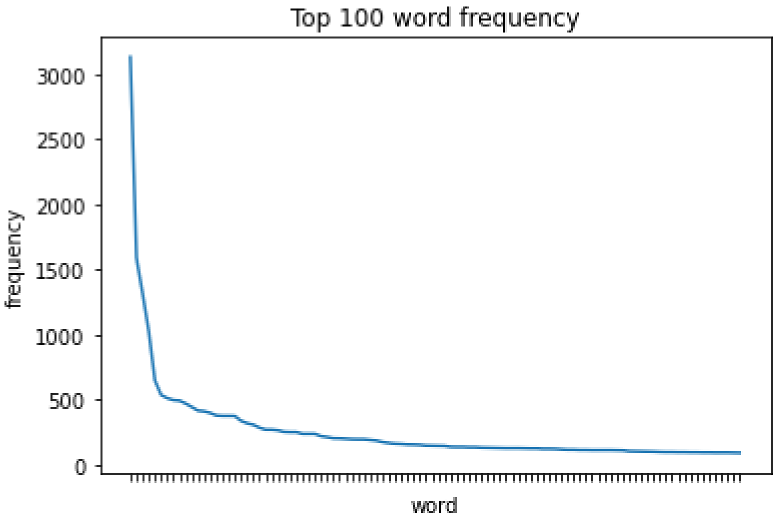


fig#2 x軸: 字詞編號, y軸: 出現頻率

In [7]:

```
import urllib.request
import numpy as np
import matplotlib.pyplot as plt
```

```
url = 'https://raw.githubusercontent.com/cjwu/cjwu.github.io/master/courses/n
response = urllib.request.urlopen(url)
text = response.read().decode()

# 將語料轉換成一個詞彙列表
words = text.split()

# 使用 numpy 轉換成一維陣列，計算出現頻率
unique, counts = np.unique(words, return_counts=True)
freqs = np.asarray((unique, counts)).T
freqs = freqs[np.argsort(-freqs[:, 1].astype(int))]  # 將出現頻率轉換為整數

# 取出前 100 項並畫出統計圖形
top_freqs = freqs[:100]
x = range(1, len(top_freqs)+1)
y = top_freqs[:, 1].astype(int)  # 只取詞頻
labels = top_freqs[:, 0]  # 保留詞彙，用作 x 軸標籤

plt.plot(x, y, linestyle='-')
plt.xticks(x, [])
plt.title('Top 100 word frequency')
plt.xlabel('word')
plt.ylabel('frequency')
plt.show()
```

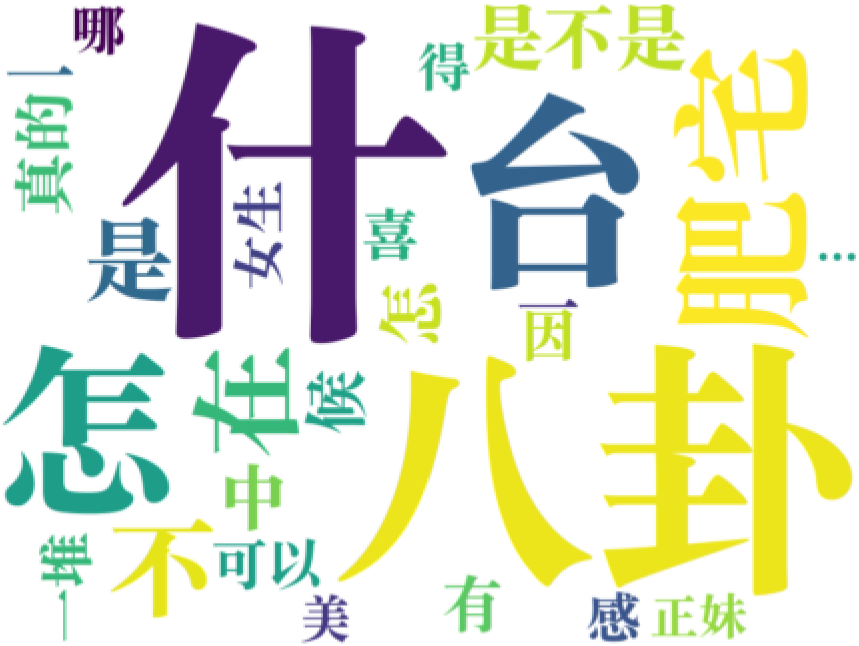


fig#3 文字雲

In [6]:

```
import jieba.analyse
from wordcloud import WordCloud
import matplotlib.pyplot as plt

url = "https://raw.githubusercontent.com/cjwu/cjwu.github.io/master/courses/n
text = urllib.request.urlopen(url).read().decode("utf-8")
result = jieba.analyse.extract_tags(text, topK=100, withWeight=True)

# 取前 32 個詞彙
result = result[:32]

# 將詞彙及權重轉成 dictionary
words_dict = dict(result)

# 產生 WordCloud 物件
wc = WordCloud(font_path='/System/Library/Fonts/Supplemental/Songti.ttc', wid

# 將 dictionary 中的詞彙及權重傳給 WordCloud 物件
wc.generate_from_frequencies(words_dict)
```

```
# 繪製文字雲
plt.figure(figsize=(10, 6))
plt.imshow(wc, interpolation='bilinear')
plt.axis('off')
plt.show()
```



```
In [ ]:
```