**Mido Documentation**

**Release 1.2.9**

1. **安裝套件**

安裝：pip3 install mido

更新：pip3 --upgrade mido

1. **分析MIDI檔案**

**CODE**

**from mido import MidiFile**

**mid = MidiFile('twinkle\_twinkle.mid')**

**for i, track in enumerate(mid.tracks):**

**print('Track {}: {}'.format(i, track.name))**

**for msg in track:**

**print(msg)**

<https://mido.readthedocs.io/en/latest/midi_files.html>

**RUN**

Track 0: Twinke Twinkle Little Star (檔案名稱)

**time\_signature：拍號**

<meta message time\_signature numerator=4 denominator=4 clocks\_per\_click=24 notated\_32nd\_notes\_per\_beat=8 time=0>

* 屬性定義與預設值：

|  |  |  |  |
| --- | --- | --- | --- |
| 屬性 | 定義 | 範圍值 | 預設值 |
| numerator | 每小節有幾拍(分子) | 0-255 | 4 |
| denominator | 幾分音符為一拍(分母) | 1 | 4 |
| clocks\_per\_click |  | 0-255 | 24 |
| notated\_32nd\_notes\_per\_beat |  | 0-255 | 8 |

**key\_signature：調號**

<meta message key\_signature key='C' time=0>

* Valid values:

A A#m Ab Abm Am B Bb Bbm Bm C# C#m Cb Cm D D#m Db Dm

E Eb Ebm Em F# F#m Fm G G#m Gb Gm

**set\_tempo：節拍**

<meta message set\_tempo tempo=600000 time=0>

* Tempo BPM 轉換

Tempo : 每一拍多少微秒(μs)。

BPM : 每分鐘多少拍。

1 μs = 10-6 s

* 1. BPM轉Tempo

**>>> mido.bpm2tempo(bpm)**

Returns microseconds per beat 回傳每一拍多少微秒

* 1. Tempo轉BPM

**>>> mido.tempo2bpm(tempo)**

Returns BPM (beats per minute) 回傳每一分鐘有多少拍

執行CODE結果：



tempo = 600000

∴ 600000 μs / beat = 0.6 s / beat

∴ tempo = 600000表示每一拍是600000微秒(60萬)，相當於0.6秒。

BPM=120，每分鐘演奏120個四分音符，每個四分音符的長度等於1分鐘除以120等分＝0.5秒，1小節就是0.5秒乘4拍＝2秒長。

<meta message track\_name name='Twinke Twinkle Little Star' time=0>

<meta message text text='trad.' time=0>

<meta message end\_of\_track time=1>

**Bytes表示**

定義：所有屬性預設均為0，除了velocity，預設為64 (middle velocity)。

優點：Easy to view便於查看、Easy to analyze便於分析、Vectorize the MIDI input輸入向量化。

* 把回傳的屬性用Bytes表示，方法如下：

**>>> msg**

**<message note\_on channel=0 note=0 velocity=64 time=0>**

**>>> msg.bytes()**

**[144, 0, 64]**

* 回傳的bytes添加其他資訊，方法如下：

return [ note\_on/off , note , velocity , time ]

**if i == 1:**

**for msg in track2:**

**if msg.type == 'note\_on' or msg.type == 'note\_off':**

**print(msg)**

**time = msg.time**

**bytes\_list = msg.bytes()**

**bytes\_list.append(time)**

**print(bytes\_list)**

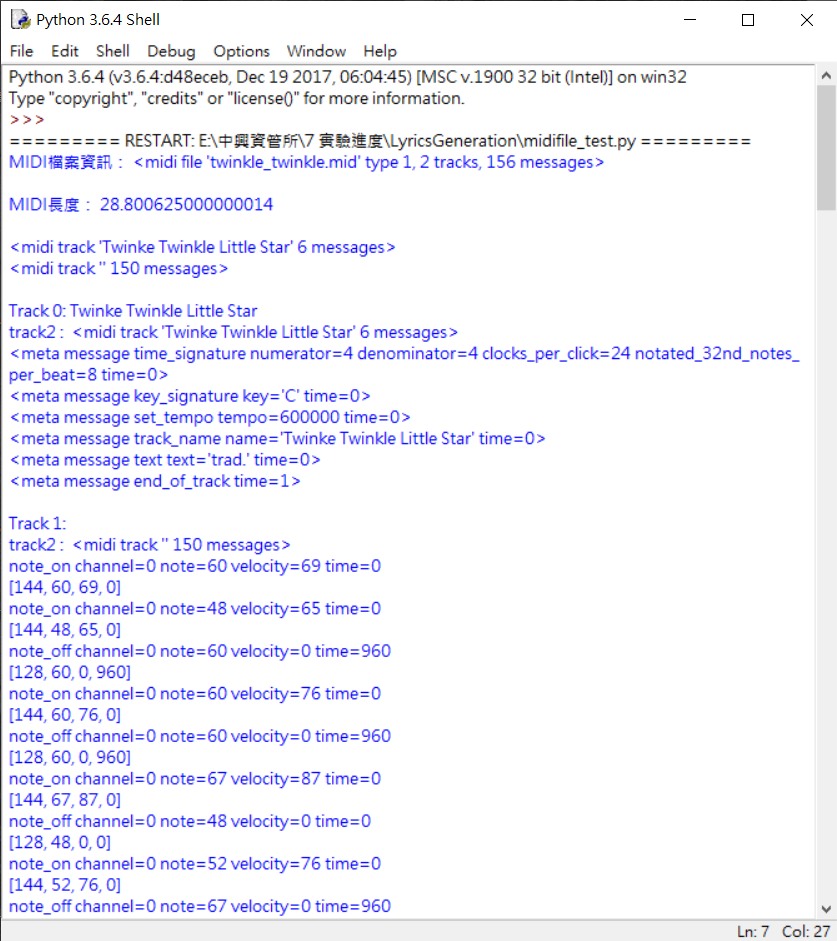
**Playback Length MIDI長度**

目的：以秒為單位取得總播放時間

**>>> mid.length**

備註：

1. 只支援type1和type2的MIDI檔案
2. type2的MIDI檔案會產生ValueError

****

**資料型態Message Types**

參考：<https://mido.readthedocs.io/en/latest/message_types.html>

**Supported Messages**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Definition** | **Keyword Arguments / Attributes** | **Value** |
| note\_off | 按下 | channel note velocity | 144 |
| note\_on | 放開 | channel note velocity | 128 |
| polytouch |  | channel note value |  |
| control\_change |  | channel control value |  |
| program\_change |  | channel program |  |
| aftertouch |  | channel value |  |
| pitchwheel |  | channel pitch |  |
| sysex |  | data |  |
| quarter\_frame |  | frame\_type frame\_value |  |
| songpos |  | pos |  |
| song\_select |  | song |  |
| tune\_request |  |  |  |
| clock |  |  |  |
| start |  |  |  |
| continue |  |  |  |
| stop |  |  |  |
| active\_sensing |  |  |  |
| reset |  |  |  |

**Parameter Types**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Definition** | **Valid Range** | **Default Value** |
| channel |  | 0..15 | 0 |
| frame\_type |  | 0..7 | 0 |
| frame\_value |  | 0..15 | 0 |
| control |  | 0..127 | 0 |
| note | 音符值 | 0..127 | 0 |
| program |  | 0..127 | 0 |
| song |  | 0..127 | 0 |
| value |  | 0..127 | 0 |
| velocity | 速度 | 0..127 | 64 |
| data |  | (0..127, 0..127, …) | () (empty tuple) |
| pitch |  | -8192..8191 | 0 |
| pos |  | 0..16383 | 0 |
| time | 時間 | any integer or float | 0 |

※ 備註

velocity：音符被敲擊或放開的速度，默認為64。

time：參數不包含在編碼資訊中，並且Mido目前不使用。

實際MIDI檔案回傳的資訊如下：(以Twinke Twinkle Little Star歌曲為例)

note\_on：按下

note\_off：放開

以此首為例：

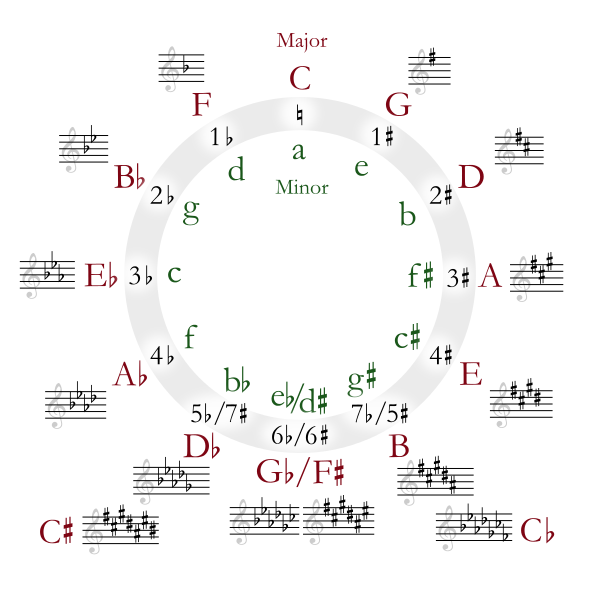


**MIDI分析結果：**



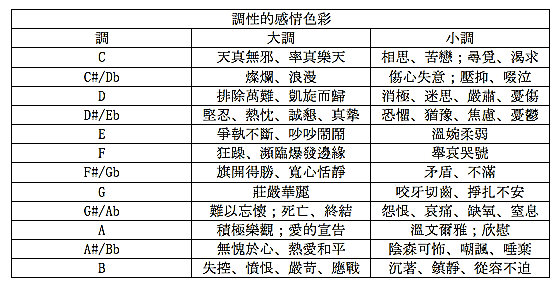
* **Key Signature 調號**

1. 調號分析圖：



1. 調性和情緒之間關係：

生成歌詞的時候，可以對應旋律是什麼調性，然後就會生成相對應情感的歌詞。



1. 參考：

<https://blog.playzmith.com/2017/02/09/63-7-elements-of-musical-composition-key/>

<https://kknews.cc/zh-tw/news/b22ge4o.html>

<https://iceya.pixnet.net/blog/post/24535240>

**參考資料**

* 這個人工智能模型能夠由歌詞來生成樂曲

<http://shuzix.com/15935.html>

* Working with MIDI data in Python using Mido

<https://www.twilio.com/blog/working-with-midi-data-in-python-using-mido>

* python處理midi文件

<http://bloodwolf.github.io/2015/08/19/python%E5%A4%84%E7%90%86midi%E6%96%87%E4%BB%B6/>

* 使用機器學習創建生成音樂的模型

<http://www.atyun.com/27792.html>

* 簡單音樂MIDI下載

<http://yuan.yocjh.kh.edu.tw/midi>[/](http://yuan.yocjh.kh.edu.tw/midi/)

<https://www.8notes.com/>

* 流行音樂MIDI下載

<https://www.midishow.com/zh-tw>