玩聲音，數位音訊的無限可能

MIDI訊號處理

本課程四週教材安排

- 本課程將兩種音樂數位訊號（音波與MIDI)，以兩週為一主題段落指導學生。第一週用基礎程式語言和套件讀取基本數位訊號的格式和訊息，第二週以高階的圖像化音樂編輯程式指導如何編排和應用這些數位訊號。
- 第一週：音樂資料類型概述與基本MIDI 訊號讀取分析
- 第二週：用圖像化音樂編輯程式進階編輯 MIDI 訊號
- 第三週：數位音訊原理 （基本音波撰寫）與頻譜檢視
- 第四週：用圖像化音樂編輯程式進階編輯聲音訊號

簡介 MIDI 與 MIDI 訊號讀取分析

- 古代創作音樂的方式
  - 作曲家創作（可經過出版商出版）
  - 演奏家練習
  - 演奏家演出

第一週 音樂資料類型概述與基本MIDI 訊號讀取分析

- 1. Digital Audio Workstation 簡介
- 2. 簡介 MIDI 與 MIDI 訊號讀取分析

古代創作音樂的方式

- 作曲家創作（可經過出版商出版）
- 演奏家練習
- 演奏家演出

現代數位音樂創作音樂製作方式與流程

- 作曲（主旋律）
- 編曲（配器）、伴奏等 loop
- 錄棚（主唱、真實樂器演奏）
- 混音
- 過帶（輸出）
- 出版發行

- 上述流程除了錄棚外，其餘可由 Digital Audio Workstation (DAW) 完成

聲音編輯的兩種基本方式：MIDI 與音訊

MIDI 是什麼？

- 音樂數位介面（Musical Instrument Digital Interface，簡稱MIDI）是一個工業標準的電子通訊協定。
- 並未帶有任何音訊訊息，僅有音符開始、結束之時間點、力度、音色、拍號等資料。

簡介 MIDI 與 MIDI 訊號讀取分析

- 上網搜尋任一MIDI to Text online converter 試試看：http://flashmusicgames.com/midi/mid2txt.php
- 上網搜尋任一 MIDI sequencer ：https://onlinesequencer.net/import
- Play with Chrome Music Lab: https://musiclab.chromeexperiments.com/Song-Maker/

Music21 介紹

- Music21 為一款由美國麻省理工學院製作之處理 MIDI 訊號之套件，此外其有完整功能可協助分析或編輯音樂 MIDI 資料。

Music21 實作方式

- 1. 打開 Google Colab (須註冊 Google 帳號）：https://colab.research.google.com/notebooks/welcome.ipynb
- 2. 在 Google Colab 內新增一個 Python3 之記事本

Music21 實作方式

將以下程式碼複製貼上，然後按左側三角執行：

```
!pip3 install music21
from music21 import *
environment.set('autoDownload','allow' )
s = converter.parseURL('http://yuan.yocjh.kh.edu.tw/midi/td.mid')
s.show('text')
```

程式碼解說：

- `!pip3 install music21` 安裝 Music 21 之環境
- `from music21 import *` 把Music21 載入使用
- `environment.set('autoDownload','allow' )`
  Music21 自動下載 URL 資料設定
- `s = converter.parseURL`
  ('http://yuan.yocjh.kh.edu.tw/midi/td.mid')
  自動下載 URL 的 MIDI 檔案並轉成 Music21 之格式
- `s.show('text')` 以文字顯現 MIDI 檔案之內容

Digital Audio Workstation

- 課後可自行嘗試：
- 下載任一免費或試用版本之 DAW: Garage Band, Pro Tools, Reaper, Cubase, …etc.
- 在其官網或 youtube 觀看教學影片
- 自行嘗試使用 piano roll 作點旋律

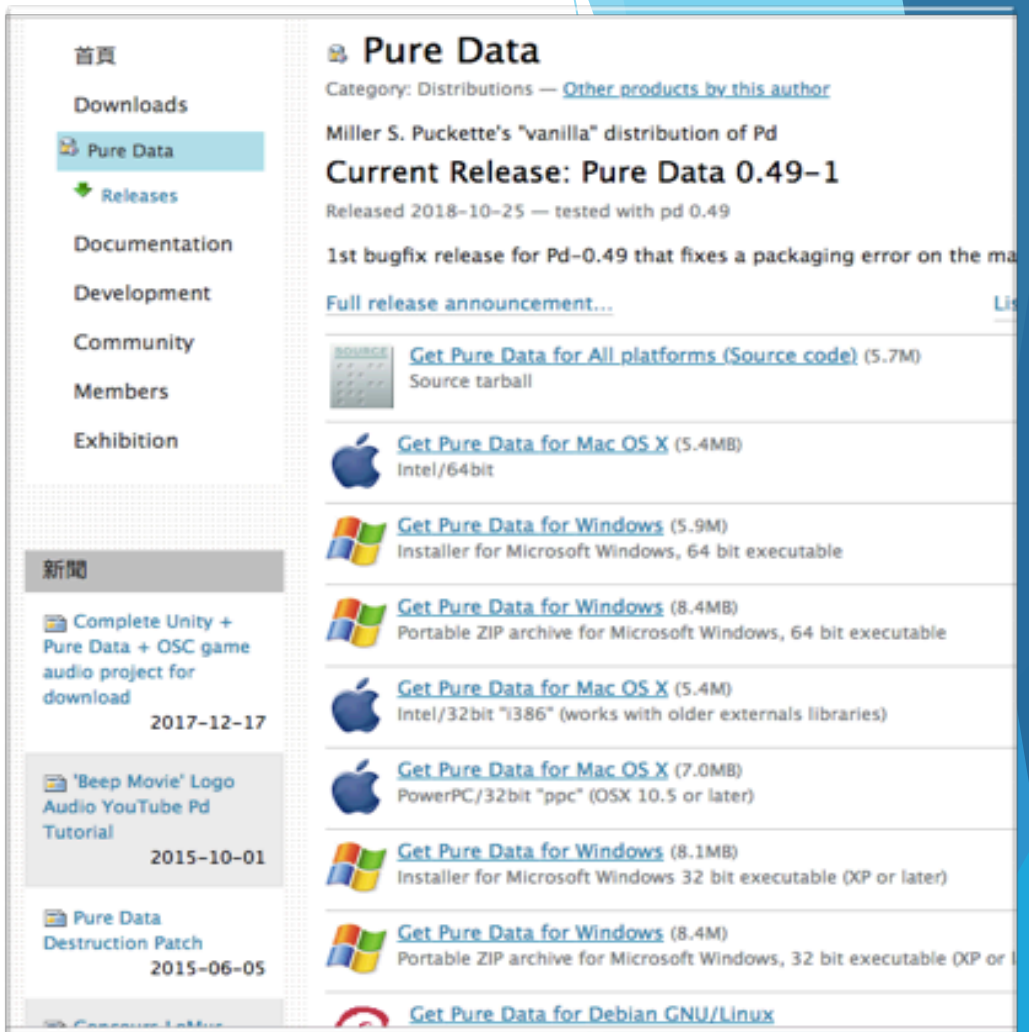
Pure Data（免費版）或 Max（試用版）圖像化音樂編輯程式使用: MIDI 訊號編輯與創作

Pure Data 歷史、下載

- Pure Data（或稱作PD）是米勒．帕克特在90年代為創造交互的計算機音樂和多媒體作品而開發的視覺化程式設計語言。雖然帕克特是Pd的主要作者，但是它是一個多數開發者為起開發新擴展的開放原始碼項目。它以一個類似於 BSD許可證 類似許可證下發行，可運行在GNU/Linux、Mac OS X、iOS、Android和Windows。
- Max/MSP 屬於 Pure Data 商業化的版本，同學可自行斟酌是否採用。

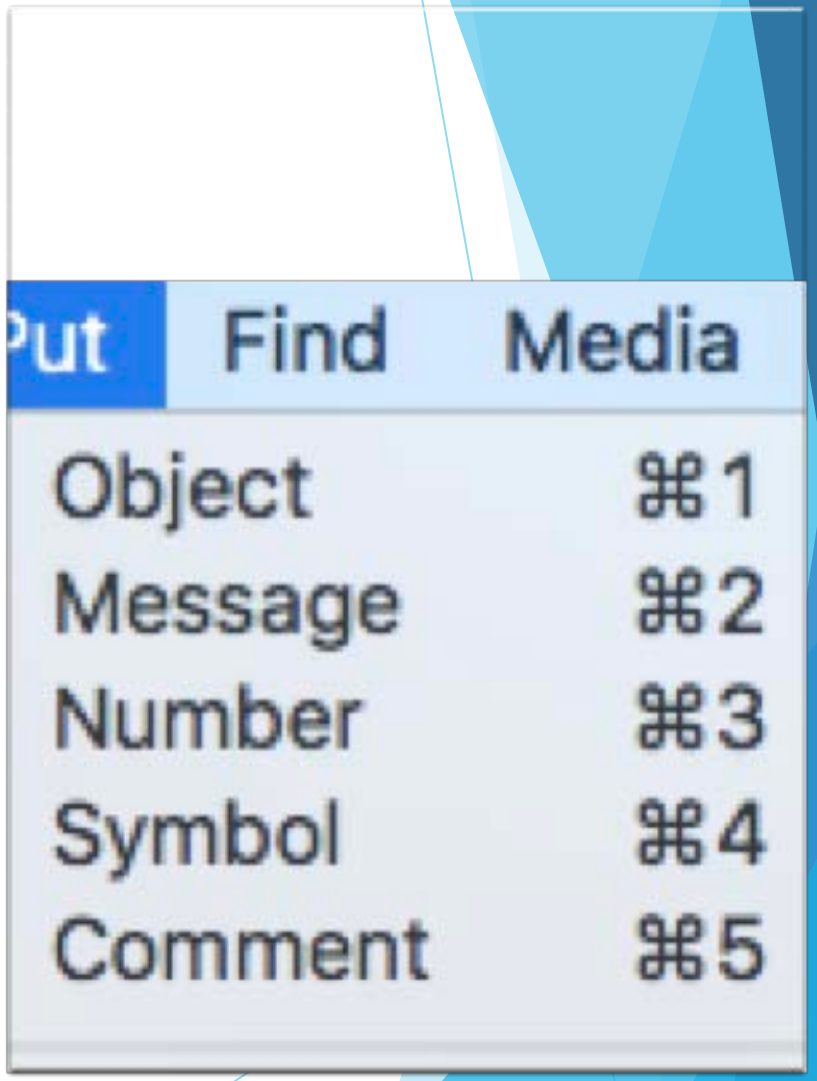
Pure Data 下載

- 請至 https://puredata.info/ 下載對應 OS 之版本
- 下載後解壓縮
- 打開解壓縮後的檔案



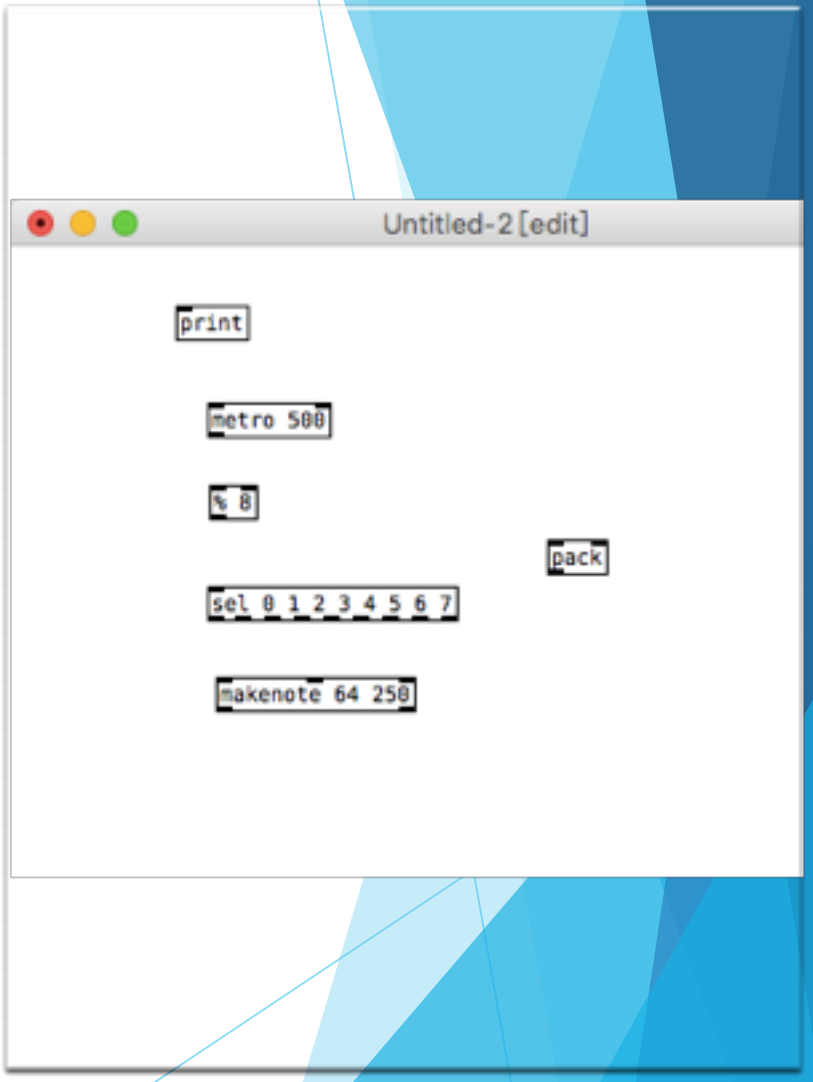
PD 資料型態介紹

- PD 有自己獨特的資料型態，包括：Object, Message, Number, Symbol, 和 Comment
- 可用右邊的快捷鍵叫各種資料型態之物件
- 將這些物件連接起來，行程 Patch，便可執行許多數學運算、MIDI、甚至音訊等運算。



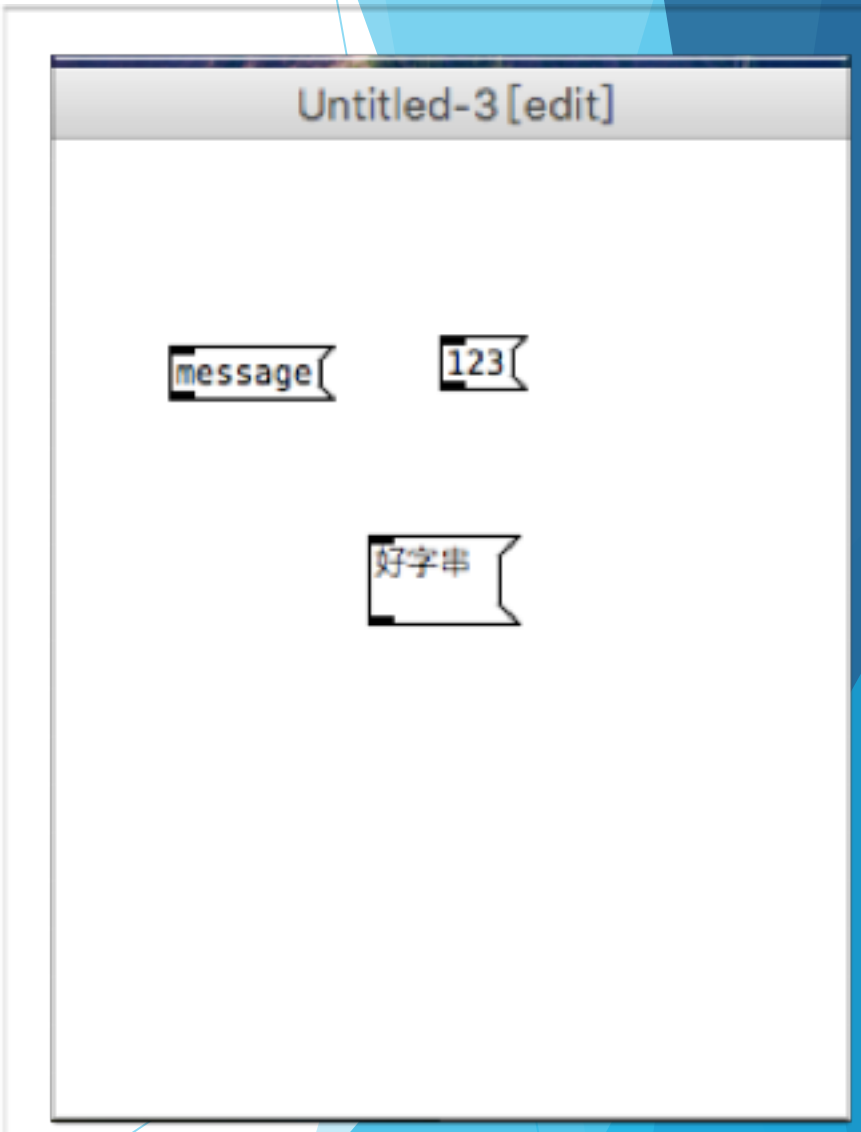
Object

- 最主要資料型態，屬於傳統程式語言之 function 類別，本身具有邏輯運算的功能
- 當輸入正確的 API (與參數）時，會出現帶有粗黑短線條之黑色框
- 粗黑短線條為訊號輸入和輸出之接口
- 上端接口為輸入端、下為輸出端
- 所有 Object List 可查詢：http://blazicek.net/list_of_pure_data_objects.html



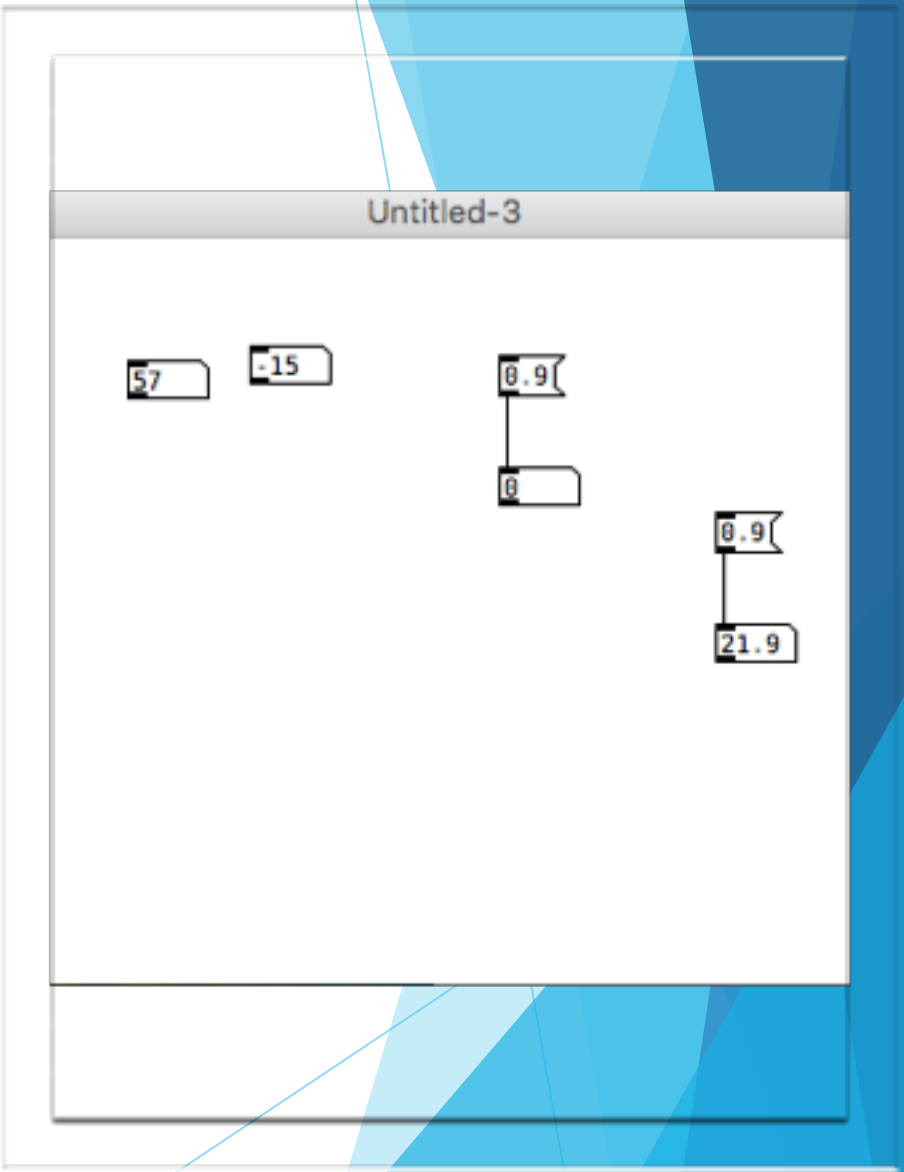
Message

- 屬靜態字串類別
- 旗標狀
- 左側上下各有粗線接口，上為輸入端、下為輸出端



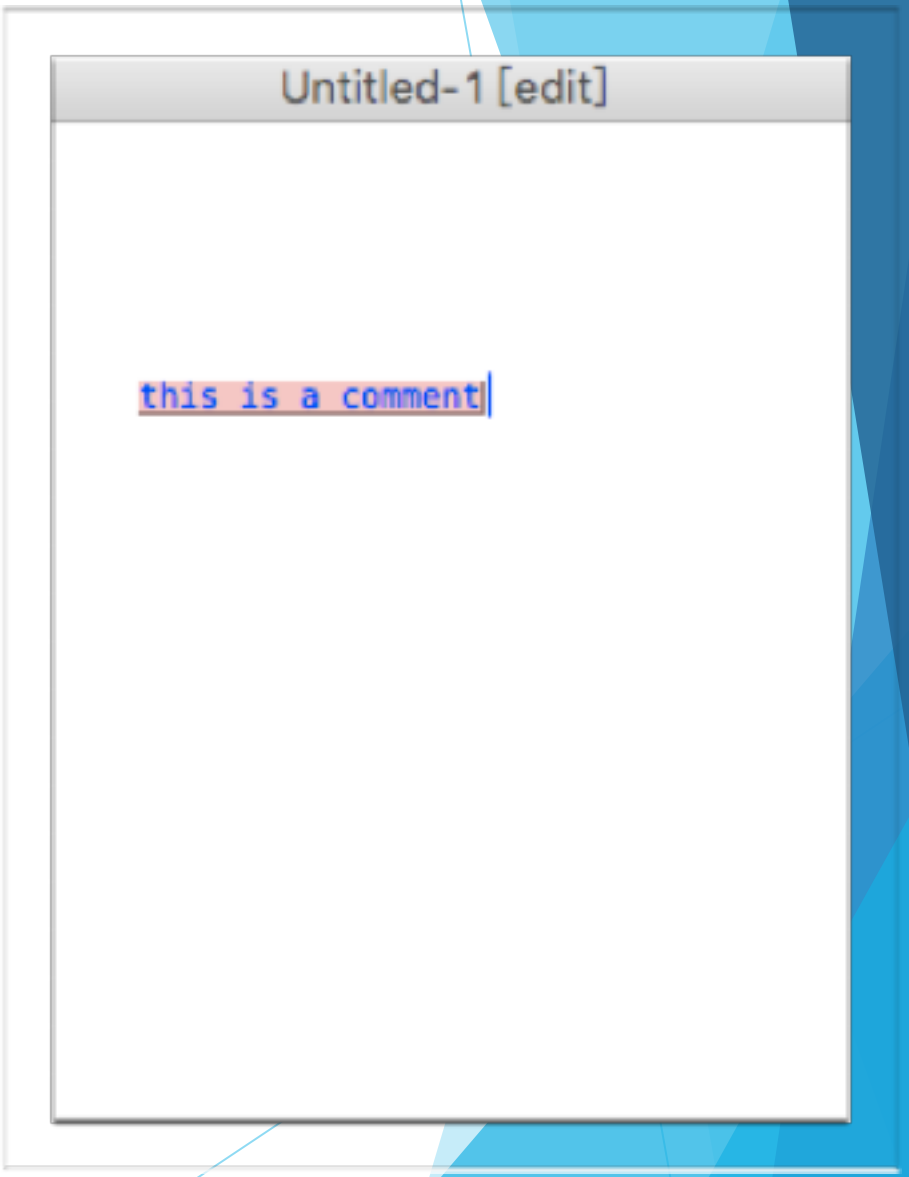
Number

- 屬數字類別
- 可有負數、浮點值
- 右上方有缺角之框格狀
- 左側上下各有粗線接口，上為輸入端、下為輸出端
- 在 edit mode 中只能呈現預設值 0
- 切換到 play mode 時（crl+e or cmd+e），以滑鼠左鍵點按於該數字框格上下拖曳後，可更改數字
- 以message 輸入連接浮點數字後，在 edit mode 點按message 框格後，number 即變成為可拖曳更改之浮點值（圖中最右）



Comment

- 註解

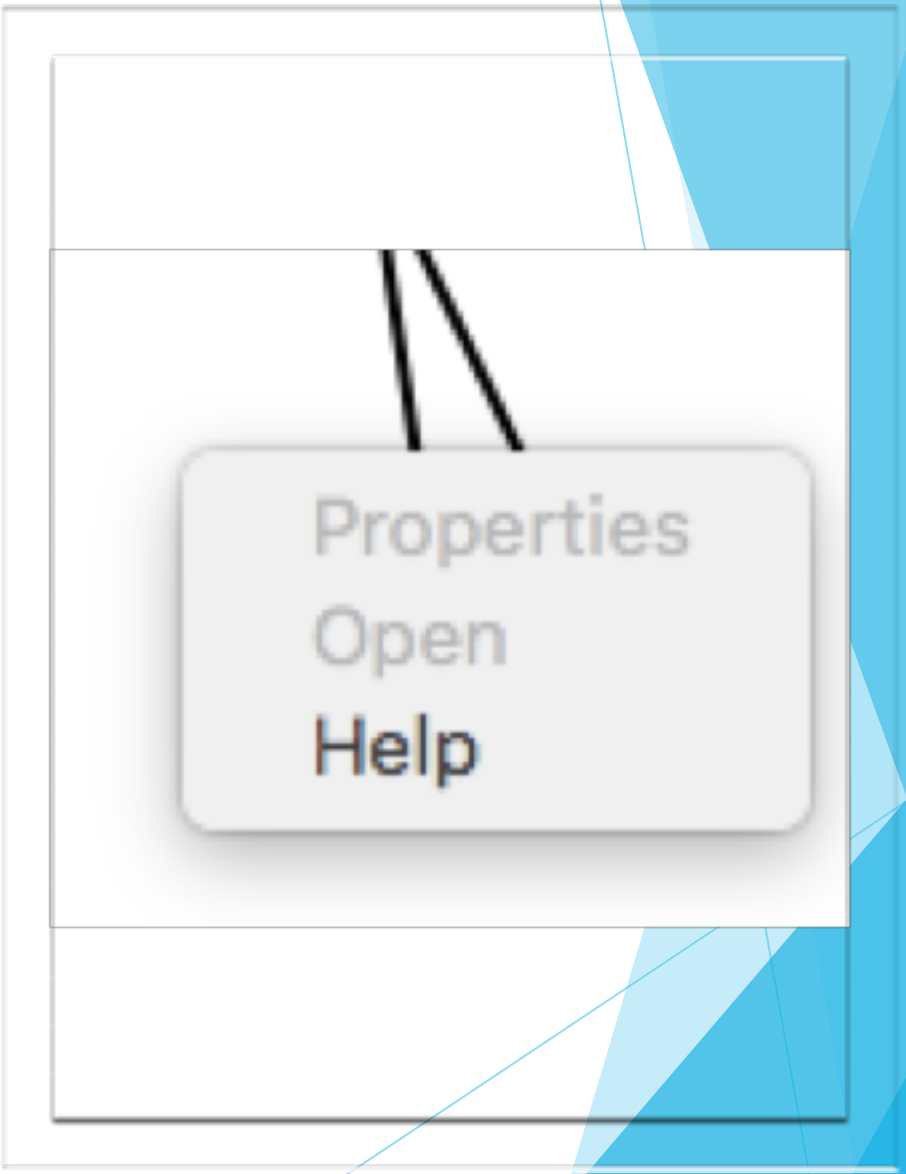


Symbol

- A symbol is a string that is cached (or in fact, permanently stored in a hash table) inside PureData. It can also have some data attached to it. Symbols are a very important part of PureData since a lot of information (e.g. type info) is identified by using them.
- The fact that a symbol is permanently resident has the following consequences: * In a PureData external symbol strings can be compared by comparing their pointer values, since there's only one copy in the system. This makes comparison much faster than comparing string characters. Data attached to a symbol won't get lost, since once a created, a symbol stays in the system. If many symbols are created (e.g. automatically in a patch), symbol lookup in the hash table will get slower - this can slow down the whole system as long as PureData is running.
- 資料引述自：https://puredata.info/community/pdwiki/symbol

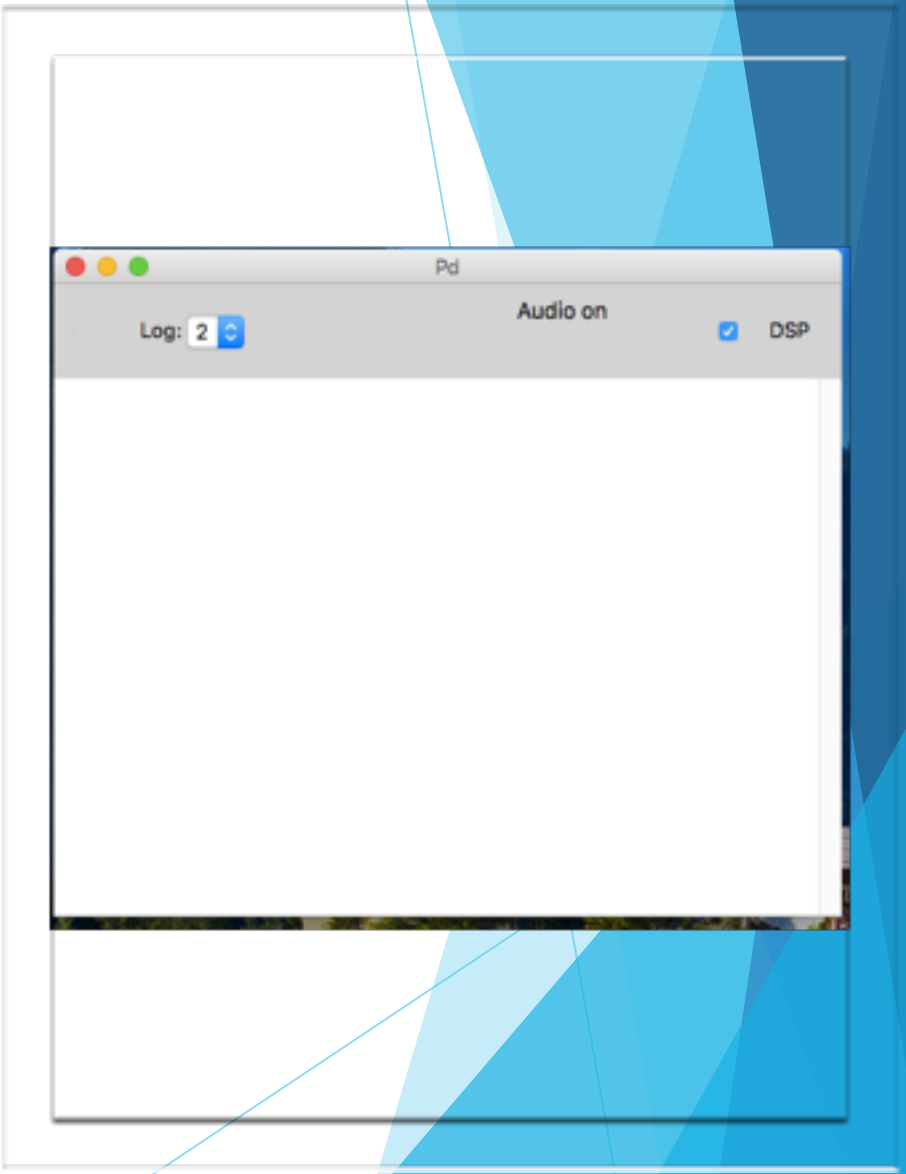
Help 功能

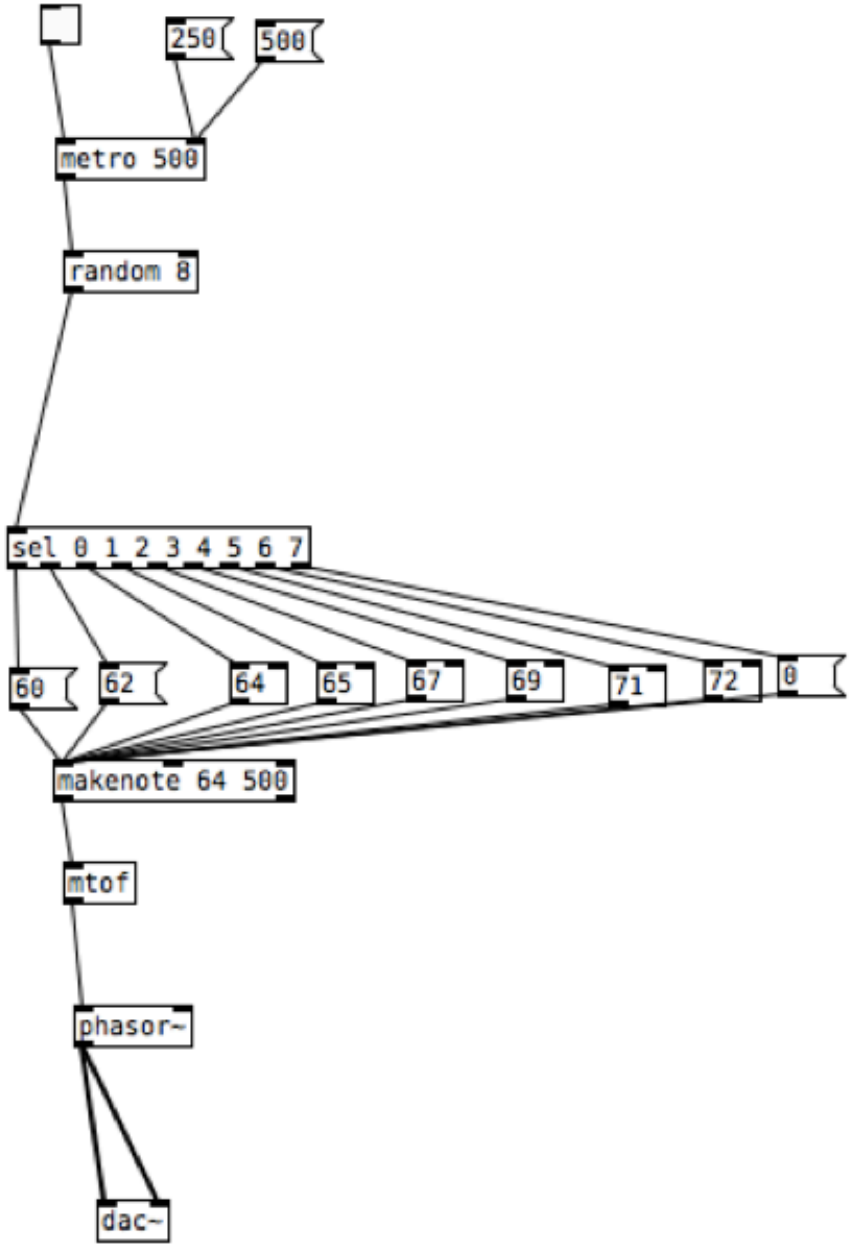
- 僅須將滑鼠以右鍵點按上述物件便能看到關於該物件的 *Help* 解釋



PD 實作（旋律產生器）

- 在 Log 視窗勾選右上方的 DSP 來開啟音訊
- 依照下方圖示產生各種物件，並以滑鼠左鍵點按個物件的輸出或輸入端，以產生線條連結個物件。




250
500
metro 500
random 8
sel 0 1 2 3 4 5 6 7
60
62
64
65
67
69
71
72
0
makenote 64 500
mtof
phasor~
dac~

玩聲音，數位音訊的無限可能

數位音訊原理（基本音波撰寫）與頻譜檢視

第三週 *數位音訊原理（基本音波撰寫）與頻譜檢視*

- 1. 數位聲音訊號原理
- 2. 頻譜檢視與分析
- 3. 基本 sine wave 數位訊號程式撰寫

數位聲音訊號原理

- 聲音是一種波動，當演奏樂器、拍打一扇門或者敲擊桌面時，聲音的振動會引起介質——空氣分子有節奏的振動，使周圍的空氣產生疏密變化，形成疏密相間的縱波，這就產生了聲波，這種現象會一直延續到振動消失為止。
- 一般的聲音總是包含一定的頻率範圍。人耳可以聽到的聲音的頻率範圍在20到2萬赫茲（Hz）之間（每秒震動循環次數）。高於這個範圍的波動稱為超音波，而低於這一範圍的稱為次聲波。
- 資料引用：https://zh.wikipedia.org/wiki/%E5%A3%B0%E9%9F%B3

類比音訊與數位音訊

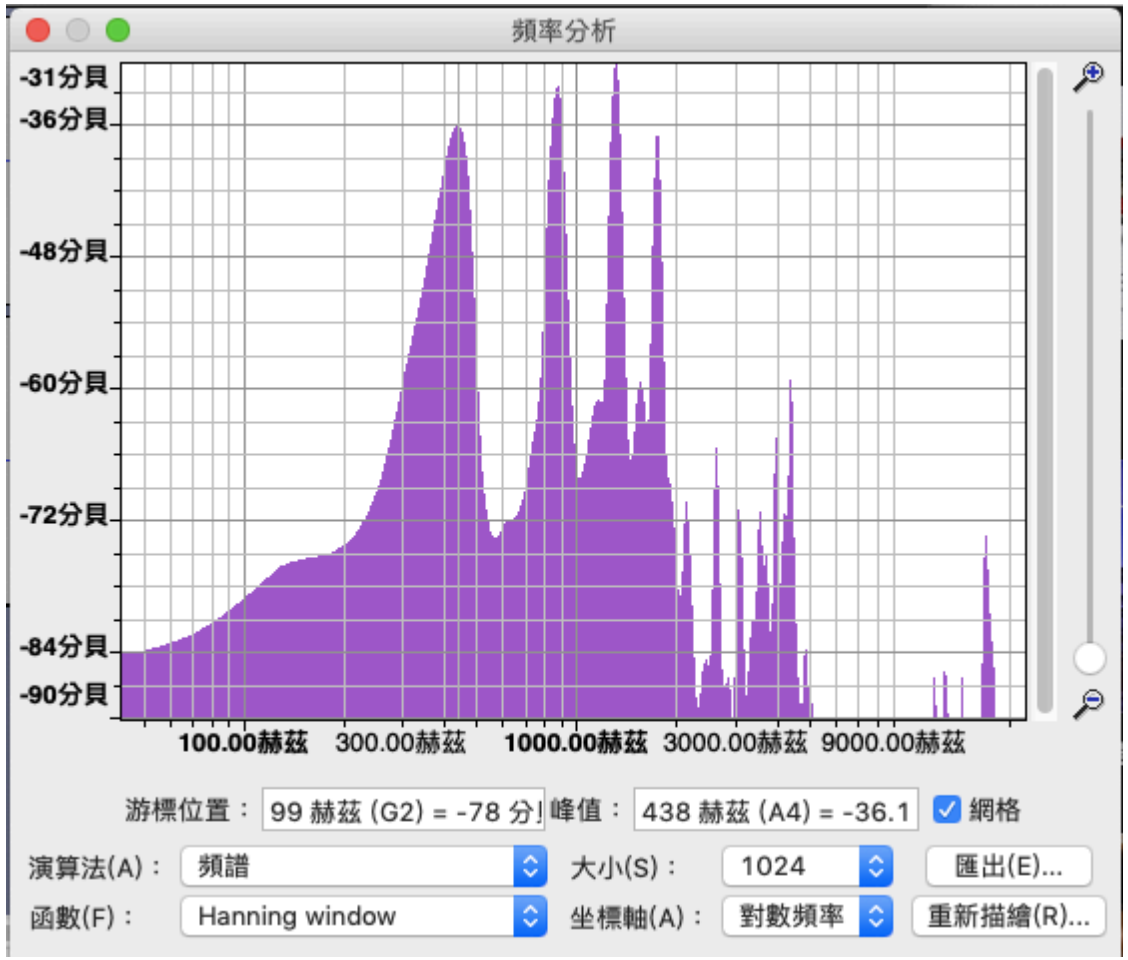
- 如同圖像一樣，有類比圖像（例如傳統底片式相片）與數位圖像的分別，類比圖像所有的線條為連續性，當一條斜線無限放大，還是斜線，但數位則是不連續性，當一條斜線被放大解讀，則會看到不連續之階梯狀線條。
- 類比音訊就比如日常生活的聲音，所有音波是連續的，但數位音訊則是指採取某些不連續的點以儲存入電腦中。

數位音訊採樣頻率

- 要將音波資料儲存入電腦等數位裝置，需要將一段音波的細分成許多點，將個點的聲壓數值依序儲存起來。細分成多少點，稱之為取樣頻率，有鑒於人類可聽頻率最高為兩萬赫茲，一個波循環，至少需有兩個取樣點，因此能聽頻率最高所需的取樣為 2萬 乘以 2(取樣點) = 4 萬取樣點，依電腦的位元數，則為建議為每秒至少需取 44100 個取樣點以上，方能將人類能聽頻率全數收錄。
- 現 CD 採樣頻率為 44100 Hz, 其他如搭配影像或高音值聲音作品可將採樣頻率提高到 48000 ~ 192000 之間。採樣頻率越高，所需儲存空間則越多。

頻譜檢視與分析

- 用不同樂器演奏同樣的音高，聲音頻率雖然相同，但由於波形不同，形成不同的音色。
- 各式各樣的波形，只要是循環波，便可以被分解為不同頻率不同強度正弦波的疊加。這種變換（或分解）的過程，稱為傅立葉變換。
- 透過傅立葉轉換成頻譜，我們可看到一個循環音波被分解成由低至高頻的多個正弦波之分布情形，以理解音色的特性。例如：聲音渾厚者的歌聲與聲音纖細者的歌聲，雖唱同樣音高，頻譜分析下，聲音渾厚者之低頻分佈會較多，聲音纖細者低頻分佈會較少。
- 資料引用：https://zh.wikipedia.org/wiki/%E5%A3%B0%E9%9F%B3

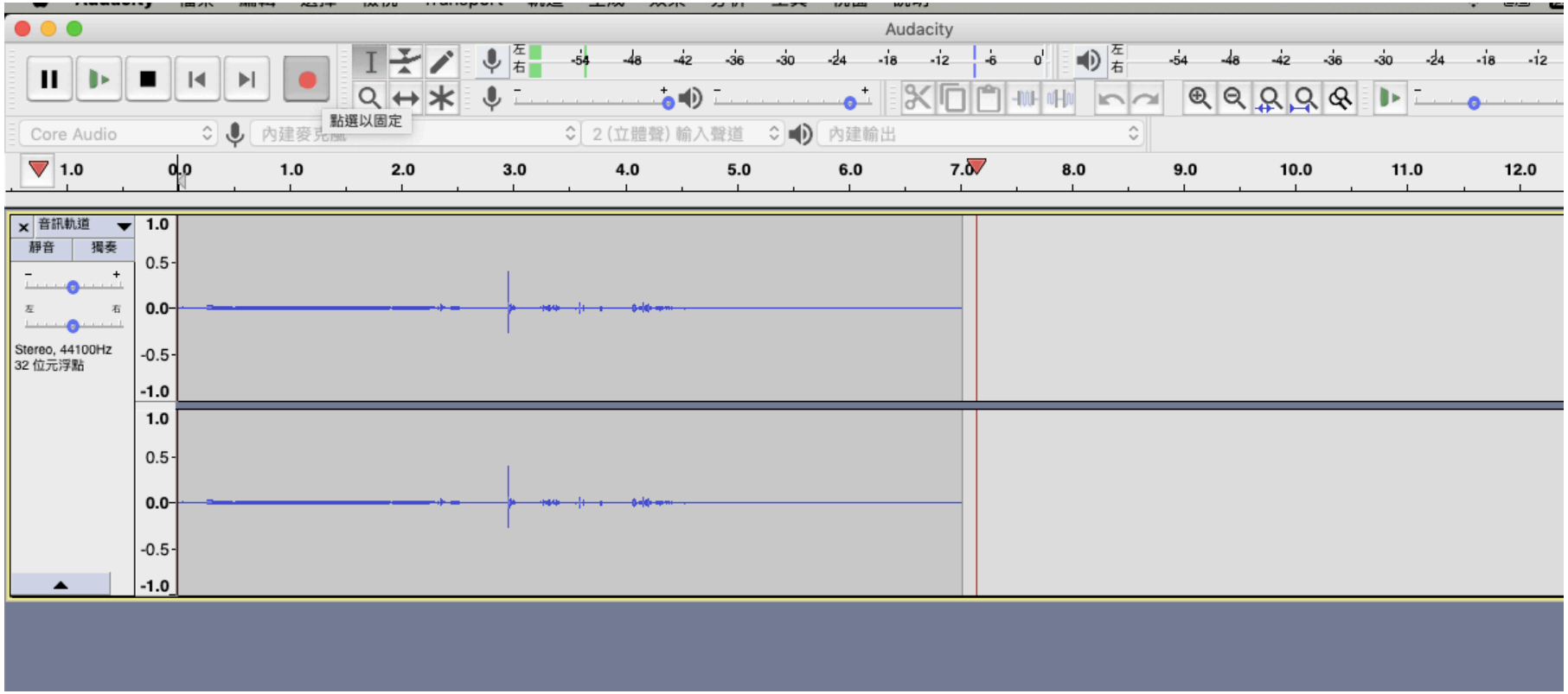


頻譜範例

此為筆者人聲哼唱 標準音 A 音（440 Hz) 的頻譜分析，可見頻譜分佈並非僅於 440 赫茲，還有440 的泛音與其他非泛音之雜訊頻率（沙啞聲）。越清亮渾厚的好歌喉產生的聲音，頻譜分佈會分佈在泛音列之上，非泛音列之雜訊頻率會較少。

下載 Audacity 與分析頻譜實作

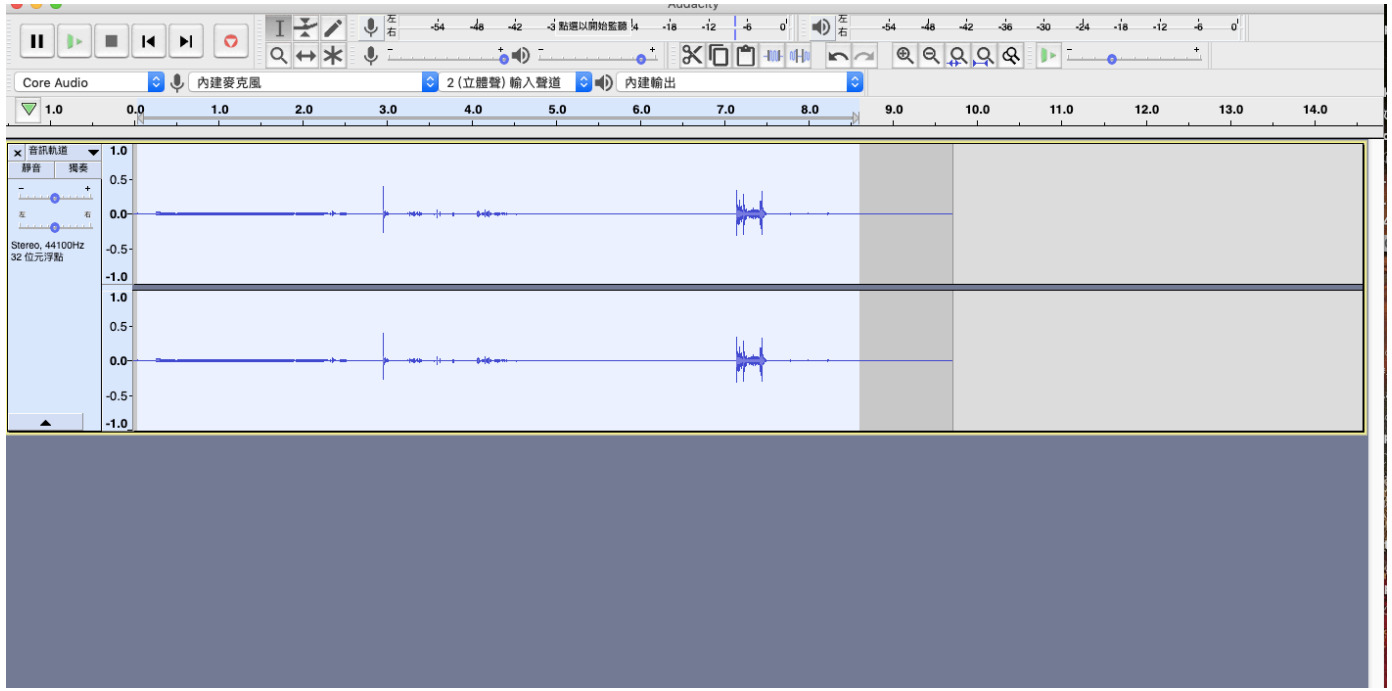
- Audacity 是一套免費的音頻分析與編輯軟體，可致其官方網站下載：https://www.audacityteam.org/download/
- 下載安裝並執行此軟體後，點按錄音鍵，便可用電腦的麥克風開始錄音，請哼唱一段小曲。



點按上方紅色錄音鈕開始錄音，結束錄音請按停止鈕

下載 Audacity 與分析頻譜實作

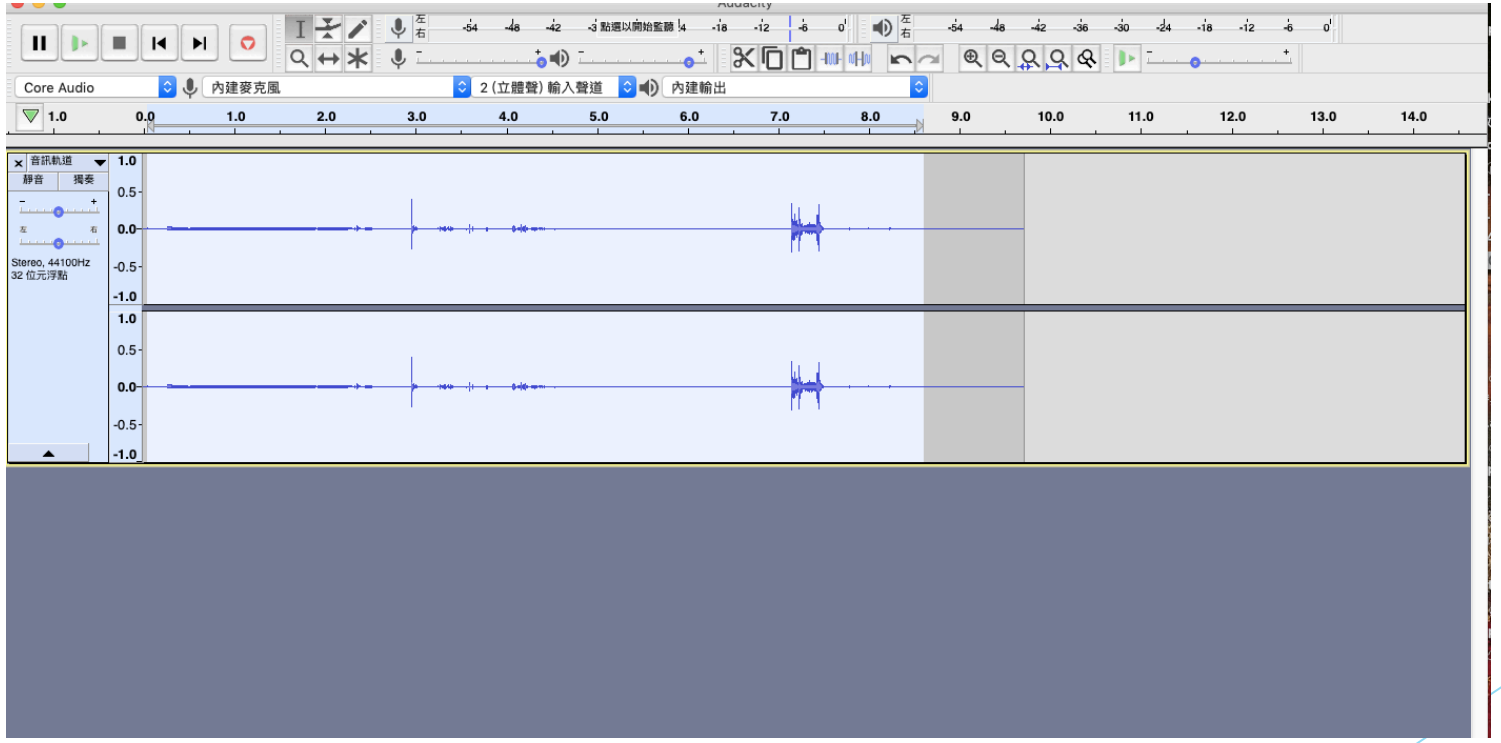
- 點按停止鈕後，以滑鼠長按左鍵一邊選取想要分析的段落，被選取之段落會反白，如下圖：



打字來輸入說明。

下載 Audacity 與分析頻譜實作

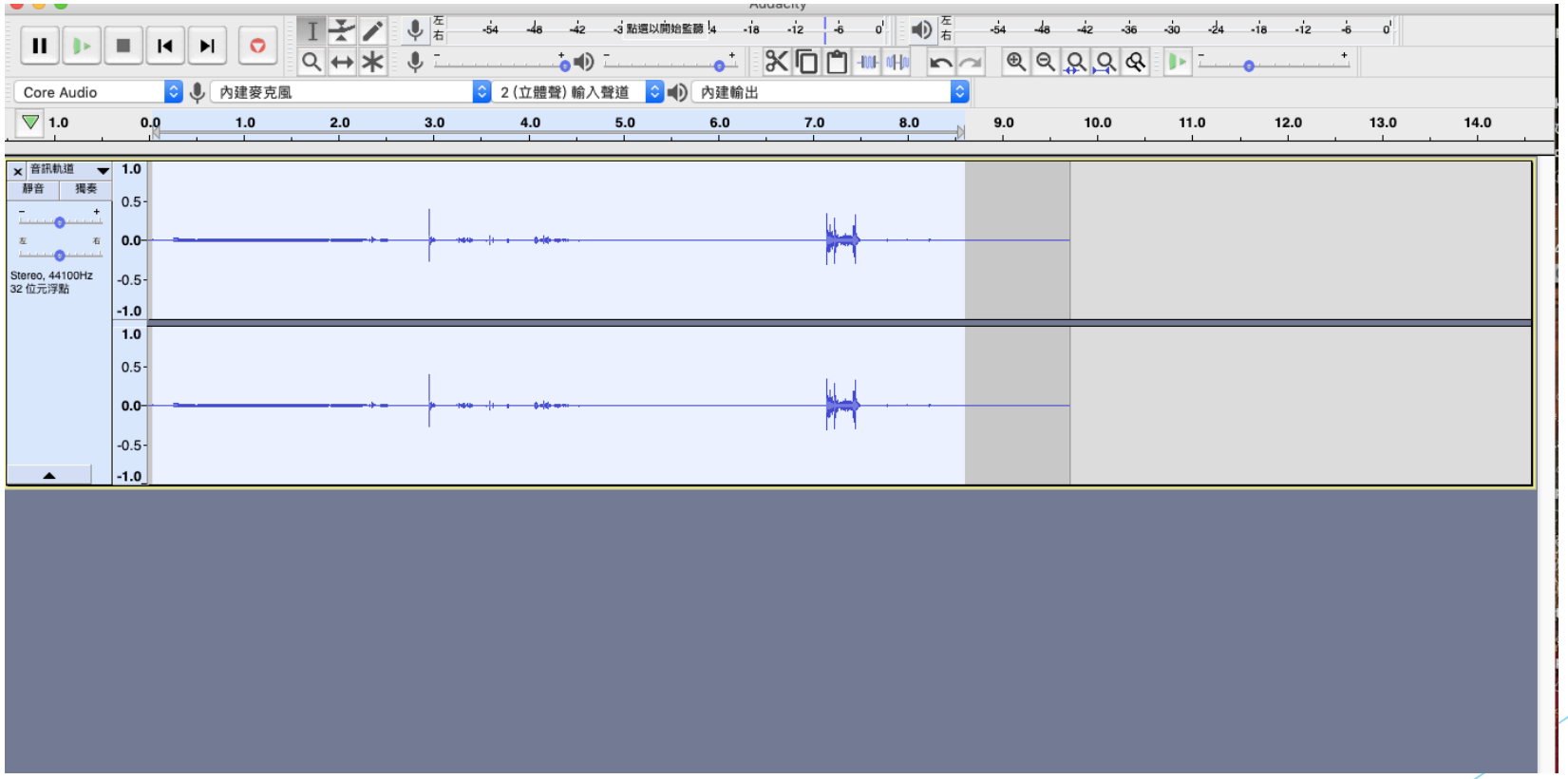
- 選取後，於上方工具列中點按『分析—>描繪頻譜』，如下圖：



打字來輸入說明。

下載 Audacity 與分析頻譜實作

- 完成之頻譜分析，如下圖



打字來輸入說明。

基本 sine wave 音訊程式撰寫

- 了解音訊原理後，接下來實際用 python 來寫一個 sine wave 音頻訊號
- *1. 打開 Google Colab (須註冊 Google 帳號）：https://colab.research.google.com/notebooks/welcome.ipynb*
- *2. 在 Google Colab 內新增一個 Python3 之記事本*

基本 sine wave 音訊程式撰寫

- 將以下程式碼複製貼上

```
import numpy as np
from scipy.io.wavfile import write
from scipy.io.wavfile import read
from google.colab import files
```

基本 sine wave 音訊程式撰寫

- 將以下程式碼複製貼上

```
#sample rate per seconds
sps = 44100

#frequency
freq_hz = 440.0

# duration in seconds
duration_s = 4.0

each_sample_number = np.arange(duration_s * sps)
waveform = np.sin(2*np.pi * each_sample_number * freq_hz / sps)
amplitude = 0.3
waveform_volume = amplitude * waveform
waveform_intergers = np.int16(waveform_volume * 32767) #32767 為振幅定值
write("sinewave.wav", sps, waveform_intergers)
```

基本 sine wave 音訊程式撰寫

- 將以下程式碼複製貼上

```
read('sinewave.wav')
files.download('sinewave.wav')
```

- 最後執行所有程式碼，便會自動下載一個sinewave.wav的檔案

程式碼解說：

以下為環境設定：

```
import numpy as np

#scipy 套件內有一個 wavfile 可執行 wav 檔案的讀寫
from scipy.io.wavfile import write

from scipy.io.wavfile import read
from google.colab import files
```

程式碼解說：

1. Sine wave 的數學式為：y(t) = A * sin(2*pi*f*t + phi)

2. 先設定基本參數，sps （sample rate per seconds） = 44100; 欲產生的 sine wave 頻率 freq_hz = 440.0，以及時長 duration_s = 4.0 （四秒），振幅值(A) 為 0-1，在此為 0.3，amplitude = 0.3

3. 由於在數位處理時，必須給電腦每一個取樣的時間值 (t)， 因此對於每一次波的循環 t = 1/sps

4. 為時 4 秒的 sine wave 將會有 4 * sos 的取樣點 (each_sample_number)，此外之後方便運算，將這些取樣點以 np.arange 方式計算，each_sample_number = np.arange (duration_s * sps)

5. 套上公式：
   waveform = np.sin(2*np.pi * each_sample_number * freq_hz /sps)

程式碼解說：

6. 乘上力度（振幅）waveform_volume = amplitude * waveform

7. 轉為電腦 16 位元的資料： waveform_intergers = np.int16(waveform_volume * 32767) #32767 為振幅定值
8. 寫入檔案 write(“sinewave.wav", sps, waveform_intergers)

9. 讀取檔案 read(‘sinewave.wav')

10. 下載檔案 files.download(‘sinewave.wav')

玩聲音，數位音訊的無限可能

Max/MSP 或 Pure Data 合成器製作

第四週 *Max/MSP 或 Pure Data 合成器製作*

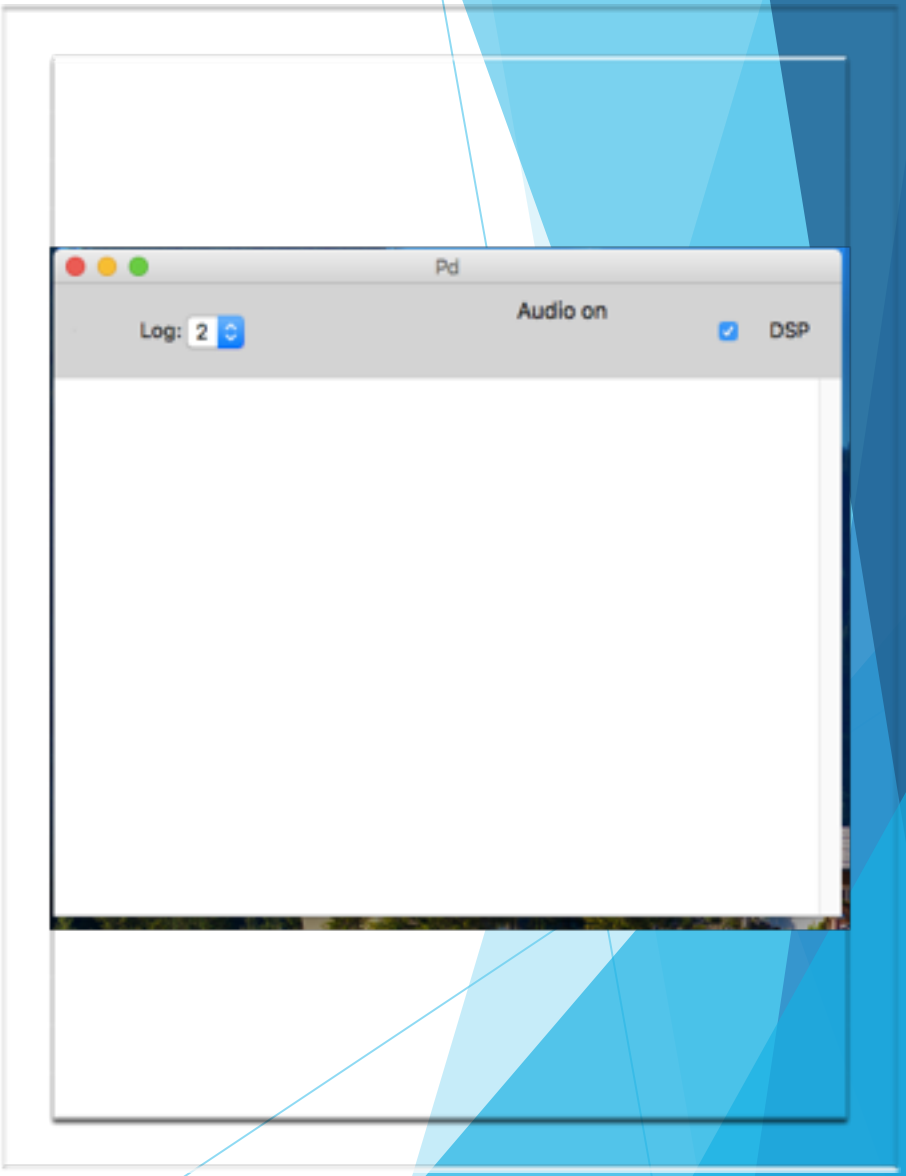
- 1. 運用圖像式高階語言進階編輯音訊
- 2. 製作音訊疊加式合成器 (additive synthesis)

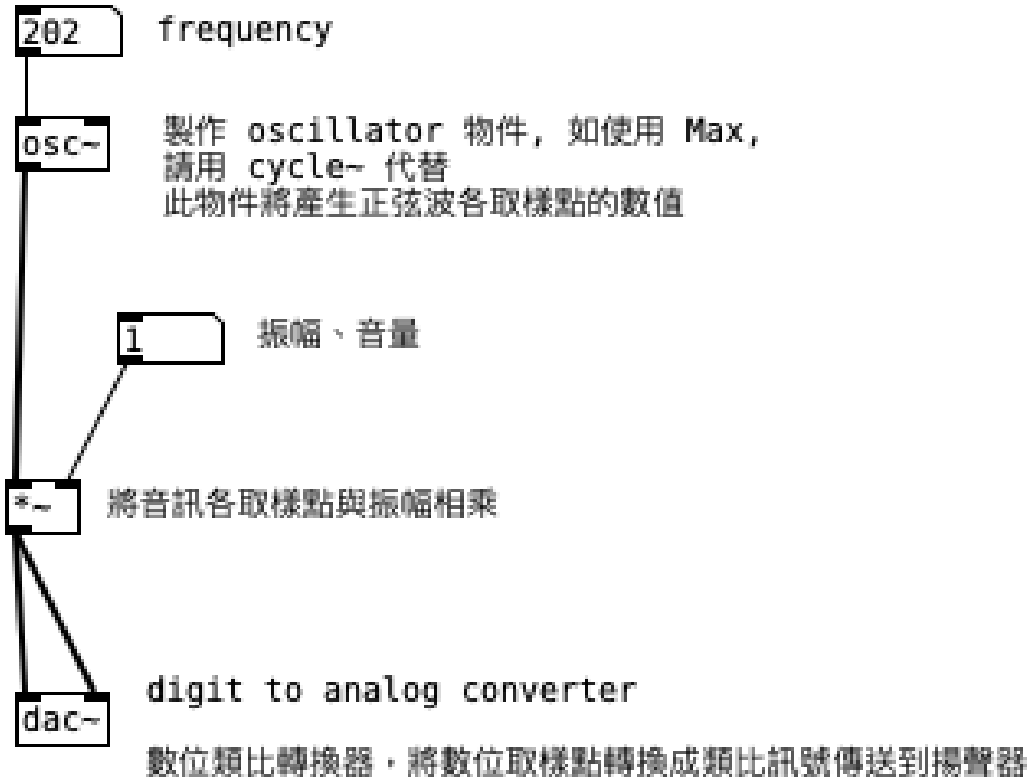
音訊疊加式合成器

- 一般日常聽到的器樂演奏之聲音，含有具循環性樂音頻率和非循環性的噪音頻率（例如擦弦、氣聲、或敲擊物件的噪音）所組成。
- 可由正弦波的組合來模擬循環性樂音頻率的部分。
- 循環性的樂音通常為某音的基音頻率以及該基因的倍數頻率（又稱泛音）所疊加合成
- 音訊上僅要將各取樣點的數值相加，便可製作。

PD 實作（音訊疊加式合成器）

- （呈第二週的 PD 學習知識）在 *Log* 視窗勾選右上方的 *DSP* 來開啟音訊
- 依照下方圖示產生各種物件，並以滑鼠左鍵點按個物件的輸出或輸入端，以產生線條連結個物件。



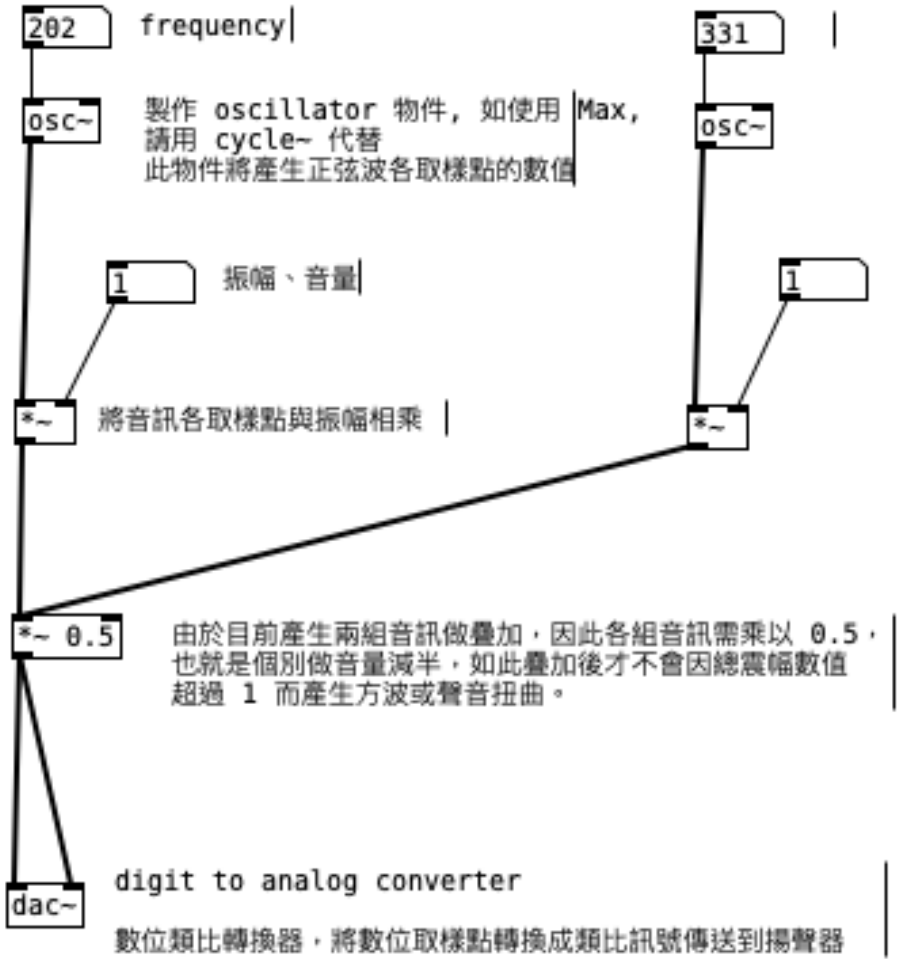


單一正弦波

1. 首先依照圖示製作一個單一的正弦波

2. 運用 crtl + e 或是 comment + e 切換編輯/執行模式

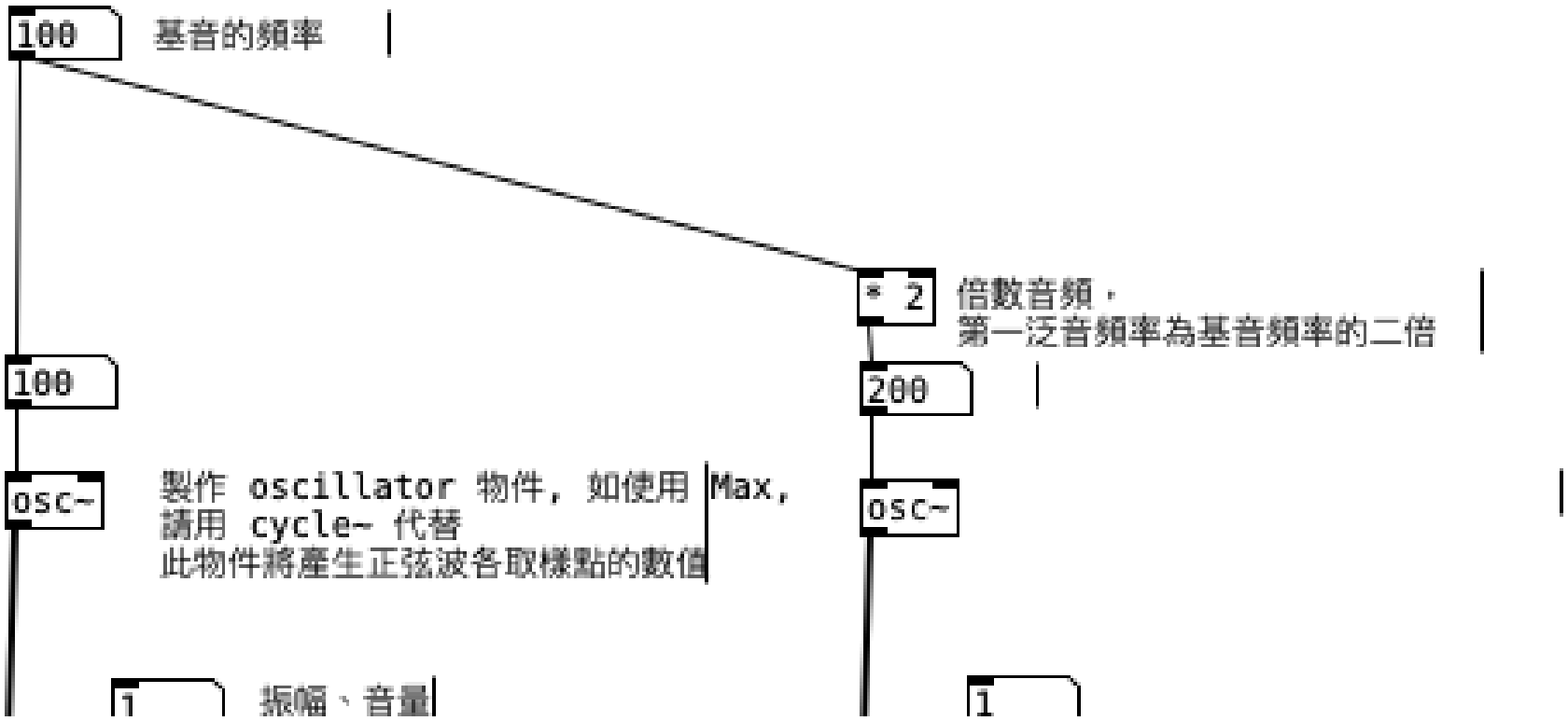
3. 在執行模式中，可用滑鼠左鍵拖曳調整 frequency 和振幅的數值，frequency 須在人耳可聽範圍方能被聽見，震幅則須介於 0~1 之間的浮點數



第二組正弦波

1. 可用滑鼠圈選欲重製的物件，按 crtl + c 以及 crtl + v 複製貼上，並拖曳到要放置的位置，以產生另一組正弦波。

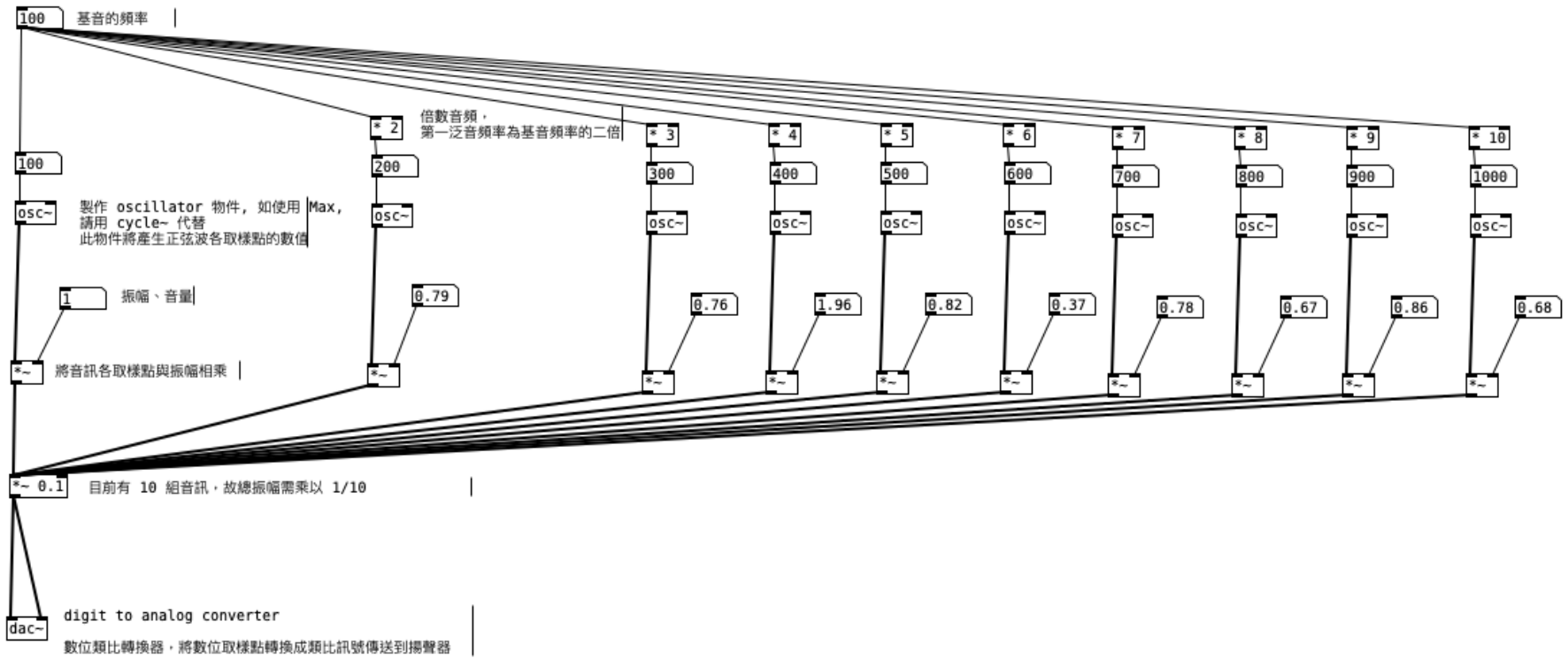
2. 由於兩組正弦波之振幅最高數值震幅為 1 的情況下，疊加後振幅為 2 ，因此需再將疊加後的振幅減半（乘以 0.5）使最高振福部超過 1。



使第二個正弦波與第一個正弦波為倍數關係

1. 在上方加入一個數字物件作為基音的頻率，並連至第一個正弦坡的頻率數字物件上（則第一個正弦波之頻率數字物件僅會重現基音頻率的數值）

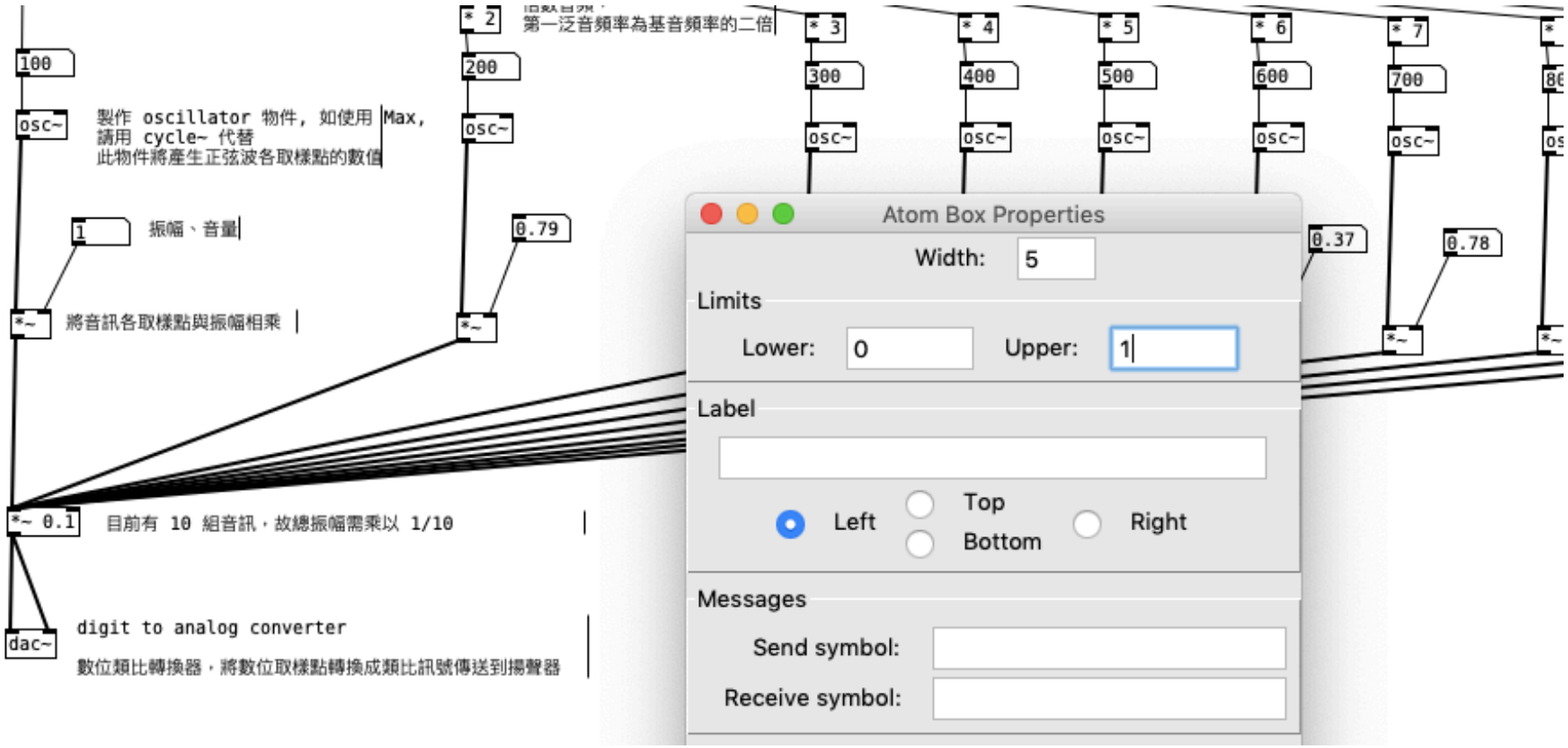
2. 第二個正弦波上方加入相乘物件，在此因為要做第一泛音，其頻率為基音之兩倍，故乘以二，並將基音頻率物件連接到此相乘物件。



製作其他九組泛音正弦波

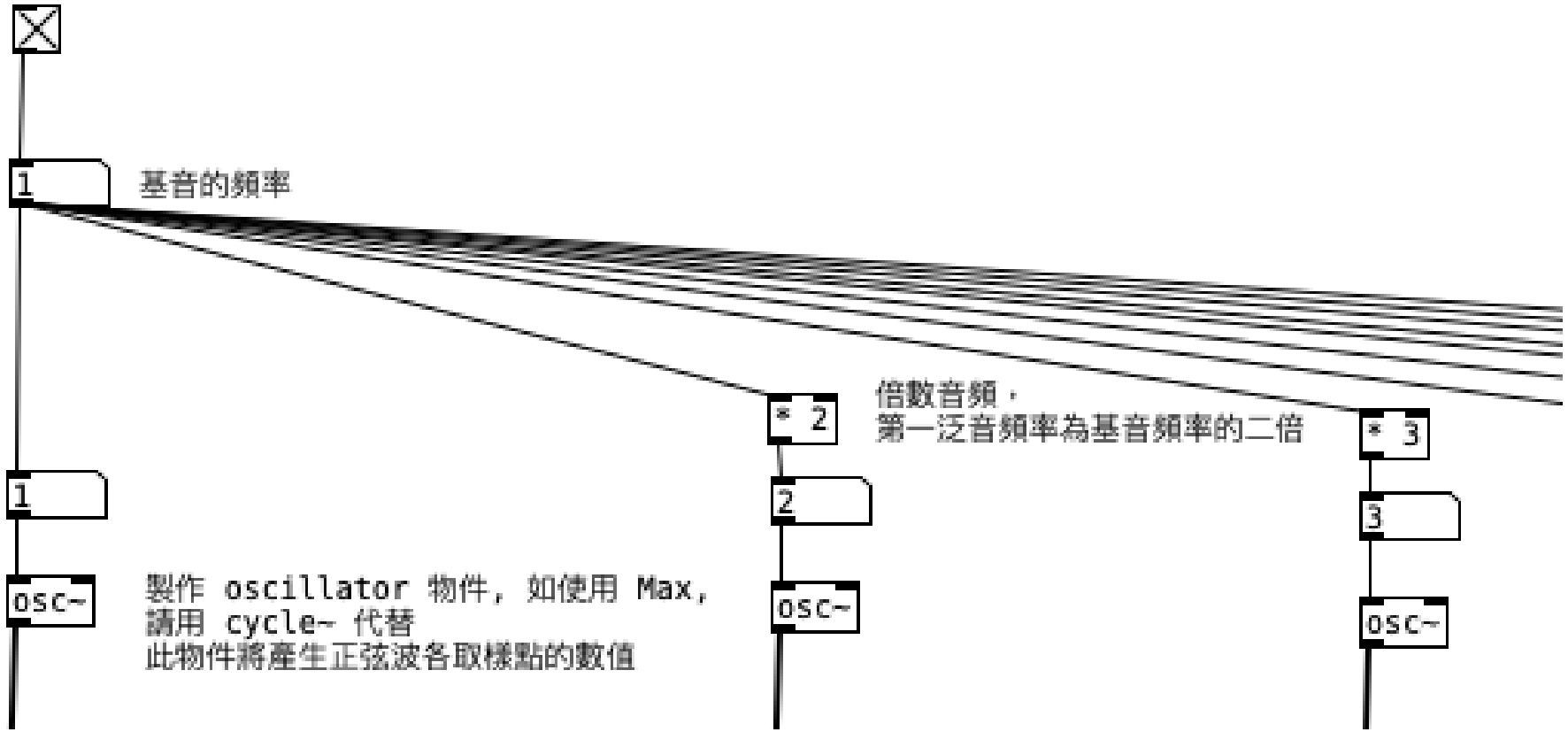
1. 繼續製作其他九組泛音正弦波，並調整倍數值。
2. 由於目前有 10 組正弦波，故將總振幅數值改乘以 0.1 (1/10)
3. 在執行模式下，對各組振幅數值之物件，按住 shift 鍵，

並以滑鼠左鍵點按上下拖曳，便能產生浮點值（小數點），滑鼠的位置偏右一點，可調整小數點的位數。



設定數字物件的範圍

以滑鼠右鍵點按振幅數字物件，便會出現對話方塊，在方塊中設定數字物件的值範圍，在此設定為 1，如此在執行模式下，更易操縱。



總開關設置

1. 在基音頻率上方新增一物件，並打上 toggle，此時便會自動產生方塊狀的開關物件
2. 將開關物件連接到基音頻率
3. 在執行模式下，可開啟開關（空格狀）或關閉開關（有 X 字狀）
4. 調整基音頻率，此時所有正弦波之數值將被連動，產生樂音
5. 透過調整各正弦波之振幅值，以改變音色，基音的振幅需高過其他泛音之振幅，如此，此總音波感知頻率才會維持在基音上

其他參考資料

- 更多 pure data 的操作方式，可參考以下連結：https://tuftsdev.github.io/MusicAppsOnTheIpad/readings/reading1.pdf