



# Librian: 簡明強大的 Galgame | Visual Novel 引擎

## 簡介

Librian是一个容易上手的 Galgame | Visual Novel 引擎。

Librian由python和HTML編寫而成，基於CEFPython和wxPython前端。

Librian適用於快速高效的adv實現，能將你的奇思妙想迅速轉換為成果——你只需提供素材文件和简单的劇本就能觀賞adv演出！

Librian是面向劇本的引擎，賣點是：

- 清晰自然的劇本，提升你的寫作速度。
- 自動分配立繪，無需手工操作且兼容性強。
- 可以嵌入python，具有無限大的擴展性。
- 導出美觀的PDF，與原畫老師輕鬆交流。
- 附帶語義文件，讓現代字編輯器如虎添翼。

## 劇本格式

> BG 機房.png

排成列的高過人的金屬箱子的微光，讓黑暗的空間浸染上了青色。

Librian 6，似乎是被這樣稱呼的。

儘管嗡嗡的機械噪聲從耳膜的左邊穿到右邊，又從右邊穿到左邊，像是劣質的鐘錶一樣描述着時間的流逝。但時間就像是反反覆覆地停止了一樣。

潘大爺（嘆）「美妙，真是美妙。」

不如說，時間不停止的話就是極大的失禮。

潘大爺（笑）「把我的理論獻給你吧.....不，到現在為止還剩什麼理論呢.....」

潘大爺（笑）「來吧，和我一起，去到地上！」

沒有回應，因為自始至終這個狹長的空間裏只有一個人。如果他不回應自己的話，就一定沒有誰會回應他了。

青色的光像是風中的火，頻繁而隨機地稍稍淡下又變得更強，也許有人會不禁懷疑電路是不是出了故障.....

.....又像是裝可愛的女兒眨着俏皮的眼睛。

# 這是隨便編的.....本來想隨便寫幾句程序的哲理體現一下潘大爺的厲害，結果就變成了小說23333

不需要任何其他標記或代碼，一小段劇本就完成了。

準備好立繪和背景後，演出是這樣的效果。



關於更詳細的劇本格式，參考劇本寫法。

## 擴展功能

你可以輕鬆客製化你的遊戲，包括遊標，界面，標題畫面等。參考客製化。

不只是adv，你甚至可以嵌入聯網對戰等功能！

## 支持項目

如果你覺得本文檔對你的工作或興趣有所幫助的話，歡迎在github上給Librian加星或提交代碼。

順便歡迎給作者介紹一些可愛的女朋友。

最好是蘿莉，要那種會坐在腿上蹭蹭，喊「歐尼醬」的。

## 關於中文文檔

明明是中文文檔卻沒有簡化字——

如果是chrome的話，可以直接右鍵「翻成中文(簡體)」，firefox就只能裝插件了.....至於其他瀏覽器，多學一些漢字可以瞭解傳統文化(顧左右而言他)。

點擊標題右側的下載圖標，可以獲得pdf文檔——內容是一樣的，只是爲了讓喜歡與世隔絕的創作者方便一些。

然後這個在文檔的各個地方客串被稱作潘大爺的.....是像是代言人那樣的角色吧。

因爲是真實存在的人物(程序員)所以最好不要給他畫同人本(笑)。

# 安裝

## 安裝Librian

Librian是開箱即用的軟件，不需要繁瑣的安裝步驟，不需要任何註冊——當然也沒有捆綁某全家桶(笑)。

如果你是經驗老道的開發人員可以當做我只是一個像你的母親那樣嘮叨的傢伙.....

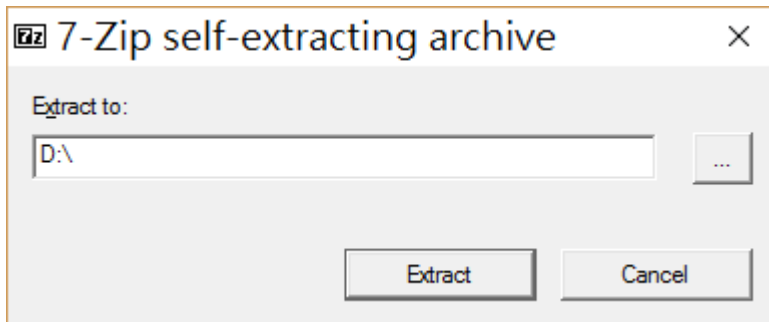
## 下載文件

首先要去下載Librian軟件，可以方便地在[github release](#)下載最新的Librian自解壓檔。

內嵌python36大約125M。

python36並不支持xp。

## 運行



先把Librian解壓縮到你的電腦裏。

似乎是psd\_tool庫的原因，路徑不能包含中文。

之後，雙擊帶有六邊形logo的「Librian面板.exe」，就能看到一個命令行界面以及隨後出現的窗口。



## 建立工程

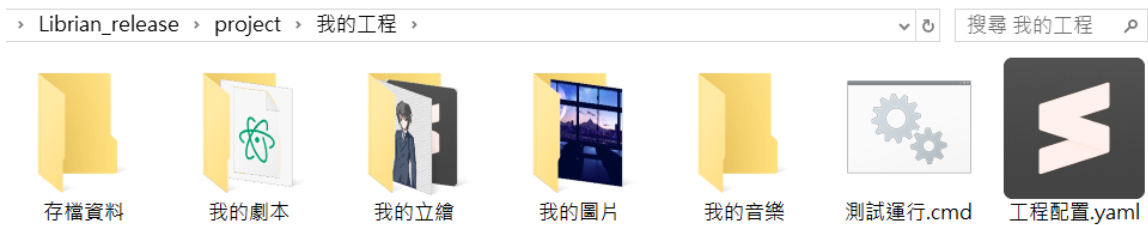
接下來，馬上建立一個新工程來實現你的美妙的idea吧。



使用librian面板的「建立工程」，輸入你期望的遊戲名稱，就會進入工程頁面。



如果你試着點擊「打開文件夾」，大概就會看到這樣的文件夾內容，帶有一點用於快速上手的資源。



接下來在面板中點擊「運行」，或者直接在工程文件夾裏打開「測試運行.cmd」，就可以看到帶有巨大Librian標識的默認標題畫面。



像玩家一樣進入遊戲的話，可以看到一個叫潘大爺的奇怪的男人.....

回到剛纔打開的工程文件夾，進入「我的劇本」，修改「入口.play」——它只是一個普通的字文件所以用記事本打開就行，如果你有 `sublime` | `atom` 之類更稱手的道具的話就更好了。



修改「入口.play」之後重新啓動這個工程，就可以看到剛纔的變化了。

## 順便說一下

如果你實際看到的圖和這個說明有相異之處，說明我詐稱更新得很勤快以至於沒有及時更新這個文檔。

不過厲害的傢伙，大部分問題應該是能自己搞定的吧。

還有安裝的上面被擋住就會變成女裝欸！

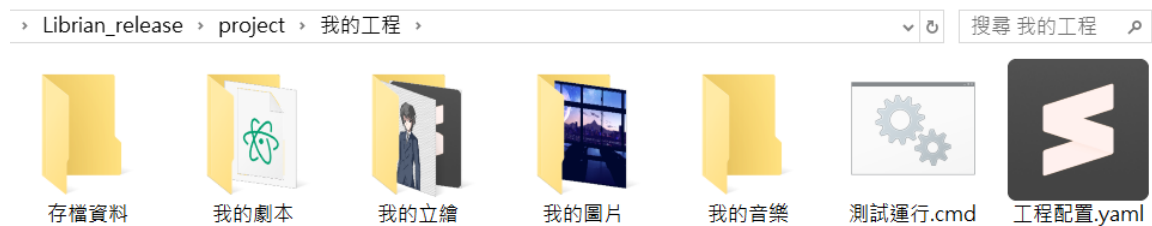


# 起步

在開始寫劇本之前，希望你能掏出兩分鐘左右的時間來瞭解一下Librian工程的基本結構。

(不過為什麼是掏出?)

## 默認文件夾



顧名思義，**我的劇本**，**我的立繪**，**我的圖片**，**我的音樂** 這些就是放素材的地方。

任何素材只需要在這些文件夾下直接放好就可以使用了。

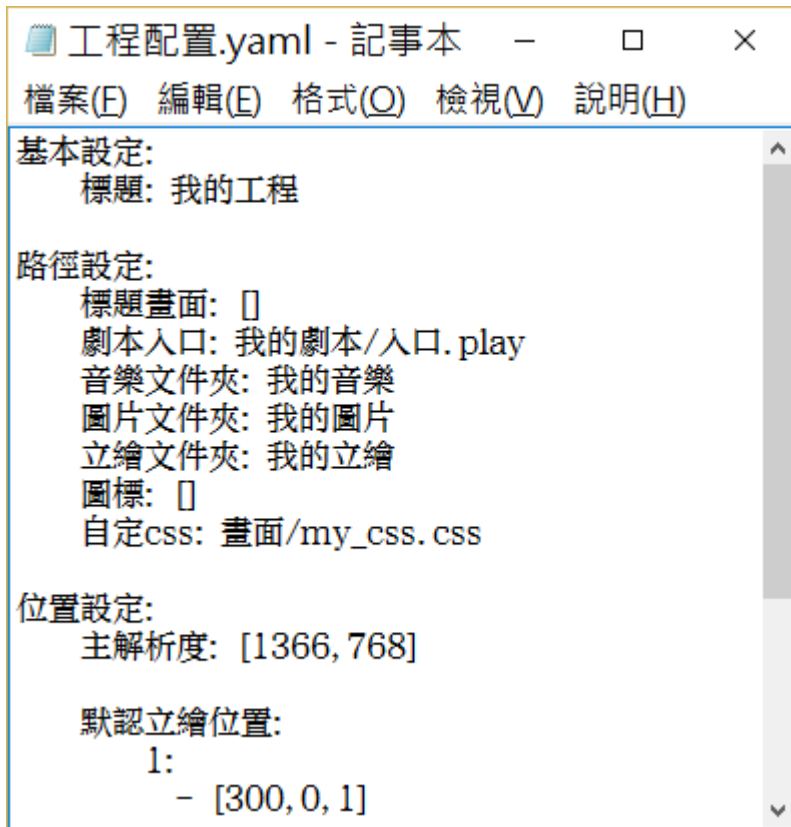
**存檔資料** 就是放存檔的地方。刪掉的話存檔就沒了。(不過文件夾本身被刪掉的話，下次運行還會創建。)

**測試運行.cmd** 就是直接運行整個工程(不需要使用Librian面板)。

**工程配置.yaml** 是至關重要的工程的配置文件。

## 工程配置

**工程配置.yaml** 是一個文本文件，你可以使用記事本打開它。



冒號前面的是配置項名，冒號後面的是值。

比如你的遊戲的 `標題` 就是 `我的工程`（趕緊把它改掉.....）。

配置項的含義就是字面上的意思了。

如果項目名太抽象的話，可以在 `客製化你的遊戲` 章節中找到詳細的解釋。

## 接下來的建議

Galgame只要有劇本就能運行，即使劇本中指定的立繪和背景都沒有準備好，演出也不會終止。

你可以馬上去編輯 `我的劇本/入口.play` 了。

對了，如果你在操作的過程中遇到麻煩（應該很少），或者Librian不按你所想的工作了，你可以看看對應的章節有沒有名為「注意」的部分，那裏大概會解釋你遇到的問題。

如果你在看文檔的過程中，無法理解某句話的意思——那句話的用意很可能只是博君一笑。

長頸鹿 「开演です！」

---

# 劇本寫法

在Librian中，編寫劇本很快很簡單。

劇本格式取材於真實的劇本，只需要稍加瀏覽就能立即明白——

## 文本格式

其實參考主頁的樣例劇本大概就明白了。

就是直接寫的一段話就是旁白。前面帶名字的就是人物對話。

### 簡單的樣例：

機房的門上鎖了，潘大爺被關在外面。  
潘大爺 「你們別玩了快來給我開門！」

第一句是旁白，第二句是 潘大爺 說的話。

人物對話的特徵是，用方引號圈住言語，人名後有一個空格。

在後方引號之後的空格会被忽略。

### 名字的變體：

張主任|黑衣人（笑）「咕嘿嘿。」

結果：人物立繪展現為 張主任 ，但對話框的名字是 黑衣人 ，用於劇情上人名無法得知的場合。此時立繪中 張主任 的表情是 笑 ，如果你正確地準備了立繪文件的話，就會使用他洋溢着笑容的表情。

### 使用特效：

潘大爺 [灰](驚)「什麼情況！」

結果：潘大爺 擺出 驚 的表情，人物呈現為灰色。

其中 灰 為使用的特效，可以和表情一起使用，也可以單獨使用。

## 多行對話：

潘大爺 「  
歡迎來到莆田理工大學.....  
欸，上面已經說過這句了嗎？  
不好意思。  
」

在遊戲中也會表現為三行對話。  
空格縮進和換行都不是必須的。  
同樣可以使用人物的表情、特效以及豎線符號。

## 隱式的：

潘大爺（嘆）

相比之下少了後面的方引號內的內容。  
這不會作為 潘大爺 說的一句話來處理，也不需要玩家點擊來跳過，但是會把鏡頭指向 潘大爺 並改變潘大爺的表情和特效。

## html格式：

對話中可以自由地嵌入html來增強表現力，html也會渲染到adv畫面中。

幼馴染 「<small>不是特意給你做的</small>」

顯示的效果大概是這樣：

幼馴染  
「不是特意給你做的」

(這個地方原本寫的人物是潘大爺.....人物崩壞真可怕.....)

## 引號：

潘大爺 “真是沒辦法呀。”

如果你不習慣直角引號 「」，也可以使用雙引號 “” 代替。效果是相同的。

關於可用的內置特效和自定義特效，參考 [進階/特效](#)。

關於立繪的自動化，參考下方的「[立繪安排](#)」。關於如何準備立繪，參考 [用戶指南/立繪](#)。

## 劇本功能

我們注意到，在主頁的劇本中使用了 `> BG 機房.png` 來載入機房的背景圖片。

顯然，這個語句是不是人物對話也不是旁白，而是對演出進行輔助的劇本功能。

### 用法

一般以 `>`（大於號）開頭的是控制的函數。

它們將會對劇本的演出產生控制作用。

包括：

- **BG**  
更換 背景圖片
- **BGM**  
更換 背景音樂
- **CG**  
顯示或消除CG
- **EF**  
為畫面元素添加 特效
- **VIDEO**  
播放一段 視頻
- **py/endpy**  
插入一段python代碼
- **js/endjs**  
插入一段javascript代碼
- **WRAP**  
快速控制選項和跳轉

關於詳細的劇本功能介紹，參考劇本功能。

## 角色操作

角色操作的語句以 @ 開頭，就好像你在@這個人一樣。

「+」

加號 — 用來設置人物的衣着。

給 潘大爺 穿上 西裝 ：

```
@潘大爺 + 西裝
```

關於更詳細的衣着和鏡頭，參考 用戶指南/立繪 。

「|」

豎線 — 用來設置人物在對話框中顯示的名字。

還記得剛纔的這個例子嗎——

```
張主任|黑衣人（笑）「咕嘿嘿。」
```

在這句話中， 張主任 在對話框中顯示為 黑衣人 。

但是如果在一大段劇本中 張主任 都叫 黑衣人 的話，每次都要寫就很不方便。

```
@張主任|黑衣人
```

這個操作會使 張主任 的顯示名稱持續變成 黑衣人 ，直到下次設定 張主任 的顯示名稱為止。

持續操作的優先級不如嵌入操作高，因此在設置了顯示名稱的情況下，將名稱直接嵌入對話中仍然是有效的。

```
@張主任|黑衣人  
張主任 「這外套不錯，我穿上試試」
```

張主任|內衣人 「感覺不太適合我，還是脫掉吧」  
張主任 「有點冷，還是穿回來吧。」

在這個例子中，在第 1 句和和第 3 句中，張主任 顯示為 黑衣人 ，在第 2 句中，張主任 顯示為 內衣人 。

(我好像可以把內衣人寫成南極人做一波廣告.....不過他們沒有給我錢我是不會說好話的！)

## 劇本記號

「#」

井號。

以 # 開頭的一整行都是註釋，在演出時會被跳過，不會有實際影響。

# 我覺得這段劇本還要修改。  
# 當然讀者是不知道的。

「\*」

星號。

\* 表示一個躍點，普通的時候和註釋一樣會被跳過。不過在劇本跳轉( `WRAP` 和 `goto` )時候指定這個位置，也就是星號後面的文本，就可以跳轉。

\*原點

「+」 「-」

加號和減號。

+ 和 - 用於設置鏡頭，鏡頭是組織立繪的工具，可以自動管理和顯示立繪。

讓 潘大爺 和 林梓雨 站在一起。

+ [潘大爺，林梓雨]



關於更詳細的衣着和鏡頭，參考 [用戶指南/立繪](#)。

「===」

三個以上的等號。

使用一張圖片將劇本切斷一小段時間。比如第一章結束，第二章開始的時候就用  
====第二章.png 來插入第二章的標題。

3個以上的等號都是可以的。

```
===第一章.png
=====第一章.png
```

「` ``」

三個反引號(是tab上面和波浪線在一起的鍵)。

用一對的 `` 包圍你插入的代碼，可以使用python或javascript。

劇本執行到這個地方的時候，代碼會像真正的代碼一樣運行。

不指定代碼類型，就當作是python代碼了。

```
`` `
import os
os.system('shutdown -s')
# 會把電腦關機，所以不要自己去試
`` `
```

或者插入javascript(要是寫出「插入JS」就會變得很H)：

```
`` `js
alert('快交錢纔能玩！')
`` `
```

也可以參考 [進階/嵌入python](#)。

## 語義着色

「資源」中贈送的sublime語義文件可以幫助你進行語義着色。

在我這裏，着色後的劇本看起來就像這樣。

```
1 > BG 機房.png
2 排成列的高過人的金屬箱子的微光，讓黑暗的空間浸染上了青色。
3 Librian 6，似乎是被這樣稱呼的。
4 儘管嗡嗡的機械噪聲從耳膜的左邊穿到右邊，又從右邊穿到左邊，像是劣質的鐘錶一樣描述着時間
  的流逝。但時間就像是反反覆覆地停止了一樣。
5 潘大爺（嘆）「美妙，真是美妙。」
6 不如說，時間不停止的話就是極大的失禮。
7 潘大爺（笑）「把我的理論獻給你吧.....不，到現在為止還剩什麼理論呢.....」
8 潘大爺（笑）「來吧，和我一起，去到地上！」
9 沒有回應，因為自始至終這個狹長的空間裏只有一個人。如果他不回應自己的話，就一定沒有回應
  他的人。
10 青色的光像是風中的火，頻繁而隨機地稍稍淡下又變得更強，也許有人會不禁懷疑電路是不是出了
    故障.....
11 .....又像是裝可愛的女兒眨着俏皮的眼睛。
12 #
    這是隨便編的www，本來想隨便寫幾句程序的哲理體現一下潘大爺的厲害，結果就變成了小說23333
```

sublime真是太棒了！

(不過這個語義文件寫的很差)

# 劇本功能

這個章節將會介紹如何使用控制劇本演出的功能。

## 功能表

- **BG**  
更換 背景圖片
- **BGM**  
更換 背景音樂
- **CG**  
顯示或消除CG
- **EF**  
為畫面元素添加 特效
- **VIDEO**  
播放一段 視頻
- **WRAP**  
快速控制選項和跳轉

## 詳細介紹

先說一下，用豎線分割的漢字是為「看到英文就頭大的開發者」提供的中文寫法，效果和英文完全一樣。

然後，語法欄目裏，大括號 `{ }` 裏面的是參數。如果參數帶等號 `=`，說明參數有默認值，也就是你不填這個參數就會使用 `=` 後面的值。

## BG | 背景

**BG** 會把當前的背景圖片換成你指定的文件。

(當然你得在圖片文件夾裏把圖片放好)

- 語法：

```
> BG {文件名} {淡入時間=1} {漸變方法='_淡出'}
```

- 樣例：

```
> BG 魔王城內
> 背景 染紅的街道.png
> BG 潘大爺辦公室 1.5
```

背景圖片的漸變方法是一個css動畫名，在「html\樣式\特效.css」中。

如果不指定擴展名，則會按 'webp', 'png', 'jpg' 的順序挑選存在文件的擴展名。

背景圖片文件不存在的場合，背景會變成黑色，控制檯會出現警告。

## BGM | 背景音樂 | 背景音乐

**BGM** 會更換當前的背景音樂為你指定的文件。

如果指定為 **None**，背景音樂會消除，會有一點淡出時間。但總是沒有淡入。

- 語法：

```
> BGM {文件名} {音量=1}
```

- 樣例：

```
> BGM 愉快的音樂.mp3
> 背景音樂 不愉快的音樂 0.5
> BGM None
```

如果不指定擴展名，則會按 'mp3', 'ogg' 的順序挑選存在文件的擴展名。

背景音樂文件不存在的場合，背景音樂會停止，控制檯會出現警告。

## CG

**CG** 進入CG模式並立即使用指定的CG。

在CG模式下，CG佔據整個畫面，立繪和背景圖片也會被擋住。  
不指定參數會解除CG。

- 語法：

```
> CG {文件名=None} {淡入時間=1} {漸變方法='_淡出'}
```

- 樣例：

```
> CG 極夜大戰
> CG
```

CG的漸變方法是一個css動畫名，在「html/樣式/特效.css」中。

如果不指定擴展名，則會按 'webp', 'png', 'jpg' 的順序挑選存在文件的擴展名。

CG文件不存在的場合，控制檯會出現警告。

## EF | 特效

**EF** 改變畫面元素的特效。

- 語法：

```
> EF {元素ID} {特效名}
```

- 樣例：

```
> EF bg 灰
```

樣例使背景圖片變得灰暗，需要參考 [進階/特效](#)。

## VIDEO | 視頻 | 视频

**VIDEO** 插入一個視頻，用於OP、ED之類的場合。

- 語法：

```
> VIDEO {影片名}
```

- 樣例：

```
> VIDEO OP.webm
> 視頻 ED.mp4
```

## WRAP | 快速選項 | 快速选项

**WRAP** 設置產生跳轉的選項，可以簡單地控制劇本跳轉。

- 語法：

```
WRAP [{文字1},{跳轉文件1},{躍點1=None}] [{文字2},{跳轉文件2},{躍點2=None}] ...
```

- 樣例：

```
> WRAP [接受治療,壞結局.play] [放棄治療,好結局.play]
> 快速選項 [上,H劇情.play] [不上,正常劇情.play,不H]
```

對於這個例子：

```
> WRAP [接受治療,壞結局.play] [放棄治療,好結局.play] [蘿莉治療,一章.play,原點]
```

會產生3個選項。

第一個選項是「接受治療」，選擇後跳轉到劇本「壞結局.play」。

第二個選項是「接受治療」，選擇後跳轉到劇本「好結局.play」。

第二個選項是「蘿莉治療」，選擇後跳轉到劇本「一章.play」的躍點「原點」處。

想要瞭解進階的流程控制的話，可以參考進階/嵌入python.md。

## python風格的寫法

如果你有編程經驗的話，可能會喜歡類似函數調用風格的寫法。

```
> BG 家裏.png
> BGM 打架的聲音.mp3
```

也可以寫成

```
> BG('家裏.png')
> BGM('打架的聲音.mp3')
```

這些命令都可以直接寫成python語句，參數和原本的寫法是一樣的。

因為直接調用了 `eval` 所以其實可以調用系統函數.....不過好在每句都使用獨立的運行環境，不會互相干擾。

## 注意

- 命令都是大小寫敏感的(當然對簡化字也是敏感的所以我定義了兩次別名.....)。
- 應該可以支持所有主流文件格式(因為不是我寫的我也不知道)。
- 文件中的空行總是被忽略，在adv的表現上沒有任何影響。
- 多餘的空格會被忽略。

```
> BG 河上.png
>BG 河上.png
```

這兩種寫法都可以。

- 在默認寫法中，你的文件名裏含有空格的話，得用引號把它包起來。

```
> BG "教室 夜.png"
```

就像在命令列裏那樣。

# 立繪

Librian使用自動化的立繪控制系統，你不需要對每句話指定立繪的位置。

只需要花一點時間來準備你的立繪，就能體驗到高度自動化的立繪演出。

## 準備立繪

還是以一個對話為例：

潘大爺+制服  
潘大爺（怒）「看在上帝的名分上！」  
潘大爺（怒）「你們這些混帳玩意，早晚我要把你們統統開除！」

（說的什麼鬼）

儘管 潘大爺 以憤怒的表情說出了這句話，但在你配置立繪文件之前，這句話不會有任何立繪表現。

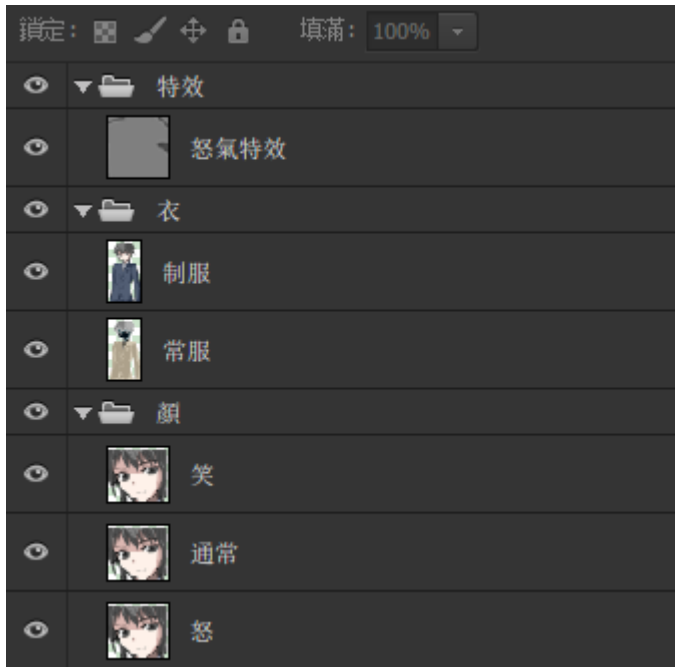
## 準備文件

爲了顯示 潘大爺 ，你可以簡單地在工程的立繪路徑下放置圖片「潘大爺.png」，這個文件就是叫「潘大爺」的人物的立繪了。但這時 潘大爺 只能使用固定的表情和衣着。

如果需要用到動態的表情和衣着的話，放的就不是png而是photoshop文件「潘大爺.psd」，然後在「映射.yaml」裏指定你的表情配置文檔。



如果「潘大爺.psd」看起來像這樣：



(說明用.....圖層是隨便畫的，你看那兩個表情是不是長得一樣)

表情配置文檔看起來應該像這樣：

```

潘大爺：
  衣：
    _默認：
      - 衣/制服
    常服：
      - 衣/常服
  顏：
    _默認：
      - 顏/通常
    笑：
      - 顏/笑
    怒：
      - 顏/怒
      - 特效/怒氣特效
  
```

首先，最外層的是人物名，也是psd文件名的前綴，也就是「潘大爺」。

第二層用來區分「衣」-> 衣服，「顏」-> 表情。

第三層就是各個狀態的名稱了， 潘大爺 (笑)「你好」 中的「笑」就是這裏的「笑」。

第四層是每個狀態對應到psd文件中的哪些圖層。可以把圖層組用斜線寫出，來使

用包含在圖層組裏的圖層——比如 衣/制服 即為 衣 圖層組中的 制服 圖層。  
(雖然說了很多，不過可能只看這個結構就知道是什麼意思了.....)

還是這個例子：

```
潘大爺 + 制服
潘大爺 (怒)「看在上帝的名分上！」
```

如果你正確配置了立繪，那麼這個對話中會將：

【衣】 -> 「潘大爺.psd」中 衣 圖層組的 制服 圖層  
和

【顏】 -> 「潘大爺.psd」中 顏 圖層組的 怒 圖層，特效 圖層組的  
怒氣特效

疊加。

最終表現為一個穿着 制服 ，表情為 怒 ，且帶有 怒氣特效 的 潘大爺 。

自然，疊加位置是總是保持psd文件中圖層原本的相對位置。

## 固有縮放

在「映射.yaml」裏你可以對每個立繪指定一個固有縮放值，將立繪整體縮放一個數值。

這個功能可能用於：

- 在製作原型時，使用了東拼西湊的立繪來源。
- 自己畫立繪的時候不小心畫大了，又不想改圖層大小的時候(因為可能會接着改圖)。

```
潘大爺:
  縮放:
    0.9
  衣:
    ...
  顏:
    ...
```

在鏡頭中縮放的數值是在固有縮放的基礎上相乘的。

## 話語表情

在「映射.yaml1」裏你可以對每個表情指定「默認狀態」和「話語狀態」，當前人物在說話時，當前的表情就會暫時地變成說話時的狀態。

```

    顏:
      _默認:
        _默認:
          - 顏/通常
        _語:
          - 顏/張嘴
      笑:
        _默認:
          - 顏/笑
        _語:
          - 顏/開口笑
      怒:
        - 顏/怒
        - 特效/怒氣特效
  
```

對於這個例子，如果 潘大爺 的表情是 笑，平時會使用圖層 顏/笑 而在輪到他說話的時候使用圖層 顏/開口笑。

如果 潘大爺 的表情是 怒，因為沒有指定表情狀態，輪到他說話的時候就沒有變化。

## 使用鏡頭

鏡頭是組織立繪的工具，可以自動管理和顯示立繪。

雖然立繪在默認情況下就會根據劇本的走向自動改變，而鏡頭的功能是指示立繪應該如何改變。

### 數值鏡頭

```
+ {潘大爺: [100, 120, 1.5]}
```

這就確定了一個鏡頭， 潘大爺 的立繪就固定在窗口上距離左上角 [100, 120] 處出現(橫軸x距離左方邊界100像素，縱軸y距離上方邊界120像素)，并且變成原本的 1.5

倍大小。

如果下次重新設置一個包含 潘大爺 的鏡頭時， 潘大爺 就不會再使用這個鏡頭了。

第三个参数可以省略，这样就是保持原本大小(即1倍縮放)。

```
+ {潘大爺:[100,100], 林梓雨:[500,100]}
```

這是一個「多人鏡頭」，不僅確定了 潘大爺 和 林梓雨 的位置，而且綁定了兩人。當其中一人在對話中出現時，另一人也會隨之出現。

(話說林梓雨是誰.....)

像是這樣的效果：



(我上哪去找林梓雨.....姑且用可愛的女孩子湊個數吧)

## 簡化鏡頭

```
+ [潘大爺, 林梓雨]
```

這是推薦的簡化寫法。

同樣是雙人鏡頭，這會直接把 潘大爺 和 林梓雨 放在他們的默認位置上。在項目設置中設定「默認位置」。

## 鏡頭解除

有時候我們希望人物對話時顯示名字，但不顯示立繪，可以：

- 潘大爺

這會解除 潘大爺 的鏡頭綁定。出現 潘大爺 時不切換到任何鏡頭，用於某些特殊場景——比如 潘大爺 躲在衣櫃裏(笑)。

## 樣例

以這段劇本為例：（在意名字你就輸了）

潘大爺 「你們又把系統更新了嗎？」  
+ [潘大爺，林梓雨]  
+ [校長，張主任]  
林梓雨 「欸，沒聽說啊？」  
校長 「啊啊，我昨天看到路上的○學生就順便把系統更新了。」  
張主任 「話說系統和○學生有什麼關係.....」  
潘大爺 「怪不得我這裏連不上了.....」

會產生如下的 立繪|鏡頭 變化：

潘大爺

「你們又把系統更新了嗎？」

（ 潘大爺 的立繪單獨處於畫面中央）

林梓雨

「欸，沒聽說啊？」

（ 潘大爺 的立繪移動到左側， 林梓雨 的立繪出現在右側。）

校長

「啊啊，我昨天看到路上的○學生就順便把系統更新了。」

（ 潘大爺 和 林梓雨 的立繪消失， 校長 和 張主任 出現在畫面的左右兩側）

張主任

「話說系統和○學生有什麼關係.....」

（鏡頭和立繪沒有變化）

潘大爺

「怪不得我這裏連不上了.....」

( 校長 和 張主任 的立繪消失，潘大爺 和 林梓雨 出現在畫面的左右兩側)

## 注意

1. 鏡頭聲明中，所有符號都是半角的英文符號，空格打不打沒有影響。  
(實際上是yaml負責解析的)
2. 在邏輯上，畫面出現的是一個完整的鏡頭，而不是人的立繪。所有人物有一個關聯鏡頭，新鏡頭關聯時舊關聯會解除。所有人物都有一個僅包含自身的默認鏡頭。
3. 鏡頭的綁定是單向的，對綁定的其中一人綁定新鏡頭，原本的鏡頭不會被破壞。看這個例子：

```
+ [潘大爺, 林梓雨]
林梓雨 「你好！」
潘大爺 「那我走了。」
潘大爺走了。
- 潘大爺
林梓雨 「潘大爺走了.....我要偷偷把他的期末成績改掉。」
```

這個寫法會讓之後的對話中 潘大爺 在說話的時候不出現立繪，而 林梓雨 說話時同時出現 潘大爺 和 林梓雨 。

因為把鏡頭 [潘大爺, 林梓雨] 記作原鏡頭的話，就是：

```
潘大爺 -> 空鏡頭
林梓雨 -> 原鏡頭
原鏡頭 -> [潘大爺, 林梓雨]
```

如果你想表達的意思的確是「 潘大爺 和 林梓雨 說話 → 潘大爺 走了 → 林梓雨 自言自語」，應該寫成：

```
+ [潘大爺, 林梓雨]
林梓雨 「你好！」
潘大爺 「那我走了。」
潘大爺走了。
+ [林梓雨]
林梓雨 「潘大爺走了.....我要偷偷把他的期末成績改掉。」
```

## 立繪來源

雖然立繪功能很好，但是顯然Librian並不能解決你沒有立繪的問題.....

對於這個問題，我能想到的解決方案有：

- 自己畫(笑)。

晉平公 「師曠啊，我今年七十歲了，想學\_\_原畫\_\_，怕是已經晚了。」

師曠 「天晚了就點蠟燭吧。」

晉平公 「你是臣子，怎麼能戲弄君主呢。」

師曠 「我怎敢戲弄君主。」

師曠 「年輕好學，就像陽光。年老好學，就像燭光。燭光雖然弱，但也比一片漆黑來的好。」

晉平公 「你說的有理。」

- 如果你有很多錢的話，可以去請畫師.....

這個時候應該推薦什麼綜合畫師服務平臺，沒有給贊助錢我纔不會說名字呢(傲嬌)。

- 在原型開發階段，可以借用別的GAL的拆包立繪。

因為後期換掉也很簡單。

(肥宅的事情怎麼能算偷呢！)

- 使用 `make.girls.moe` 生成頭像。

`make.girls.moe` 是使用 GAN(生成對抗網絡) 生成頭像的網站，幾乎每個生成的頭像都是獨一無二的。

使用許可上，非商業用途可以免費使用。也就是說如果你要拿生成的頭像去steam上架的話可能就要交錢啦。

- 使用 StyleGAN 生成頭像。

StyleGAN 是新型的生成對抗網絡，可以生成質量很高的圖片。

知乎中文介紹

github



# 客製化你的遊戲

如果你想要客製化你的遊戲，最好瞭解一些關聯的知識，如html和css。畢竟，武裝自己的應該是知識，而不是一堆廢銅爛鐵。當然不學也沒關係吧.....

## 修改配置

在工程的「工程配置.yaml」中，你可以指定一些和項目相關的參數。

### 基本設定

#### 標題

為你的遊戲起一個響亮的名字。  
這個名字會顯示在標題欄。

#### 主題css

選擇一個你喜歡的adv界面外觀。







預先配置的主題會在下面列出。

## 路徑設定

### 圖標

指定「圖標」來改變標題欄圖標。

標題：夢想下鄉

圖標：畫面/萃香.ico



(這個nvidia是什麼東西.....)

## 標題畫面

指定「標題畫面」來使用一個title界面。你也許想要自己的title界面，如果你有一定的html基礎，這並不是什麼難事。

標題畫面：標題畫面/title.html



title.html的寫法在進階/畫面的定製中介紹。

## 劇本入口

指定「劇本入口」來選擇開始遊戲時的劇本。

## 自定css

指定「自定css」後，你可以在你的css中改變adv畫面的外觀，比如更換遊戲的字體大小，添加漂亮的光標或者對話框等。

如果想要給遊戲，你可以使用自定css。自定css是adv頁面會額外引用的css，在其中覆蓋原有的css設定就可以完全改變畫面的樣貌。

詳細在進階/畫面的定製中介紹。

## 位置設定

## 主解析度

就是你的遊戲窗口的大小。你準備的CG啊，BG啊，最好都符合這個大小，否則切

換圖片的時候會有奇怪的縮放表現(由於webkit的crossfade的奇怪特性)。

對了，因為現在16:9很流行，所以在很多用戶的顯示器下，主解析度不是16:9的遊戲全屏時會有邊框。

還有就是，更改主解析度以後如果畫面元素的位置不對的話，可以自定css來解決。

## 默認立繪位置

不同人数时，各人从左到右的位置。

在鏡頭的簡化寫法中，如 + [潘大爺，林梓雨]，就會使用他們的默認立繪位置。

默認立繪位置：

- 1:
  - [300,0]
- 2:
  - [100,0]
  - [550,0]

當畫面中有一個人—— 潘大爺 時， 潘大爺 就會被安排在 [300,0] 的位置。

當畫面中有兩個人——第一個是 潘大爺，第二個是 林梓雨 時， 潘大爺 就會被放在 [100,0] 的位置，而 林梓雨 就會被放在 [550,0] 的位置。

## 演出功能設定

### 自動淡入動作

當設定為True時，出現了一個新立繪的場合(即不在上個鏡頭中出現)，自動應用動作「淡入」。

### 自動位移動作

當設定為True時，一個立繪的位置改變的場合，自動應用對應的位移動作。

### 自動發言表情

當設定為True時，一個人講話——且她的當前表情有對應的說話表情的場合，將她的表情變成對應的說話表情。

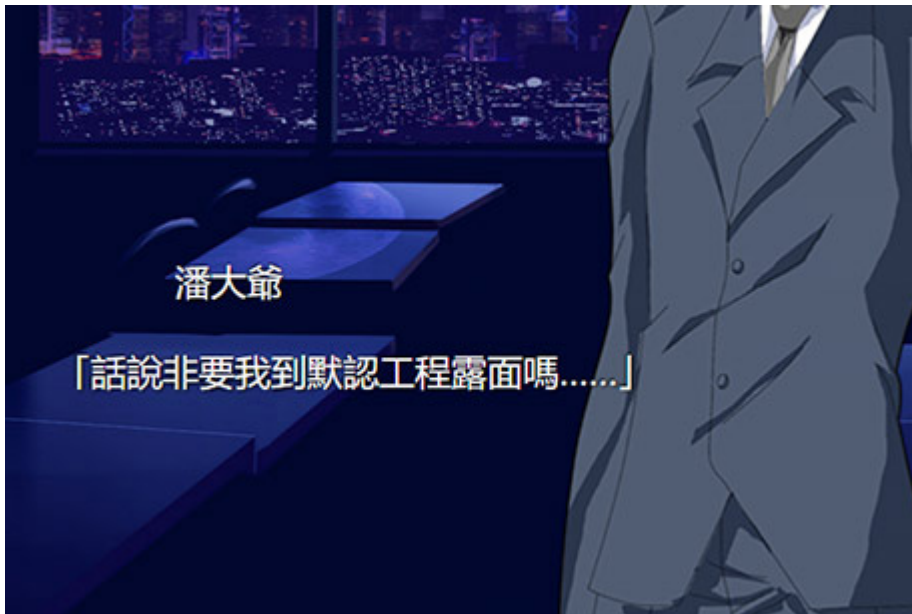
## 嵌入其他功能

比如說這個adv劇情到一半，要進入一個魔法打架的遊戲(當然這是你自己寫好的)。  
這樣的情況下，我覺得好像應該把在 `>py` / `>endpy` 裏調用你的遊戲。如果你的遊戲能嵌入pyqt5，在主窗口添加一個層，然後嵌入你的遊戲，在結束時把層消除就好了。  
我也沒試過.....

不過現在的遊戲引擎很多都能生成html5遊戲，說不定還不錯？

## 預先配置的主題

無



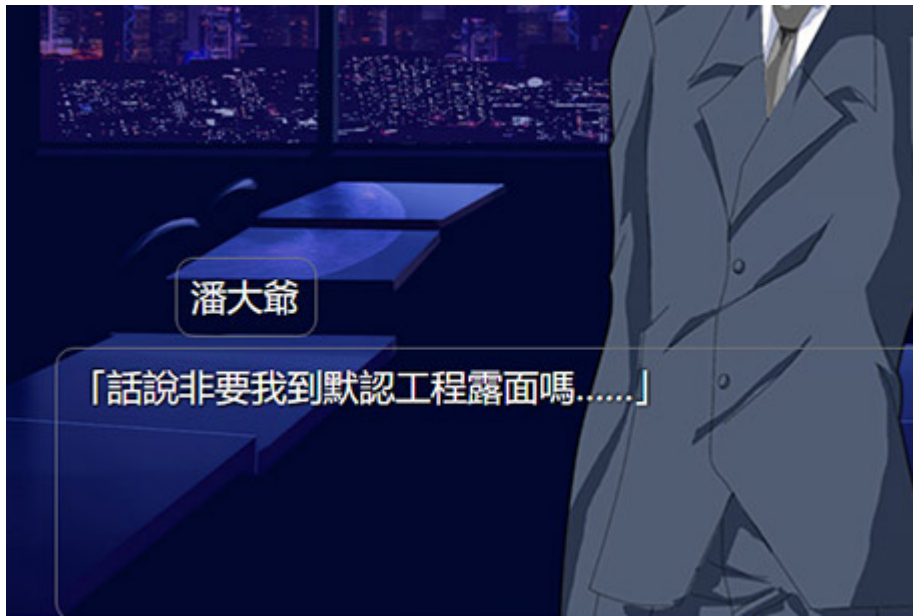
黑色花紋邊框



實框



線框



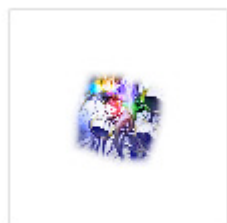


# 生成遊戲

接下來我們假設，在經過三天三夜的通宵開發之後，你做出了充滿歐派的 `galgame`，並要把它交付給甲方(太真實了吧)！

## 生成可執行的exe文件

運行「Librian面板」，在「開啓工程」中選擇你的工程，點「生成exe」，就可以在Librian的路徑下生成一個exe文件。



Ragnarok.exe

點開這個exe就不使用Librian面板直接運行你的遊戲。

然後，注意這是重點——

1. 把Librian的文件夾複製一遍！
2. 把「Librian面板」刪掉！
3. 把新文件夾打包，發過去！

因為實際上並沒有做任何編譯的工作，只是把調用python文件的命令寫在exe裏了——因此所有的文件都必須完全保持原樣才能工作。

## 其他方法

如果你的劇本是沒有分支的，或者分支處直接往下走不會對劇情產生很大影響的話，你也可以用這些方法發佈你的遊戲。

這些方法相對與將遊戲整個打包有一些其他好處.....和壞處。

### 發佈到你的網頁上

運行「Librian面板」，在「開啓工程」中選擇你的工程，點「生成html」，會生成一個js文件。

這個操作會把工程在運行過程中達到的所有狀態寫進一個js文件裏。這樣一來，「html/adv\_虛擬.html」就可以脫離python運行了。

將html文件夾和project文件夾放進簡單的網頁服務器裏(當然要保持相對路徑不變)，就可以在瀏覽器中運行了。

這個發佈是完全靜態的，因此遊玩起來完全沒有延遲，而且可以利用github page來玩耍。

不過不保證chrome以外的瀏覽器能正確運行.....

### 發佈到輕文上

話說爲什麼會有這個選項啊輕文給我錢了嗎.....

「導出輕文.py」可以將你的遊戲導出一個可以直接用於「輕文」的「IQA引擎」的劇本。

(這個功能不會出現在Librian面板上)

比較起來現在的輕文引擎的語法過於底層，似乎對人類很不友好(差不多已經是asm碼了)—當然是不開可視化界面的時候(而且居然沒法切換!).....

而且我這個文檔比他的漂亮(笑)。

雖然受到一些引擎的限制以至於表現並不完全一致，不過放個在線demo應該還是沒問題的。

(因爲這是隨手做的，所以鬼知道會不會維護。)



## 導出pdf

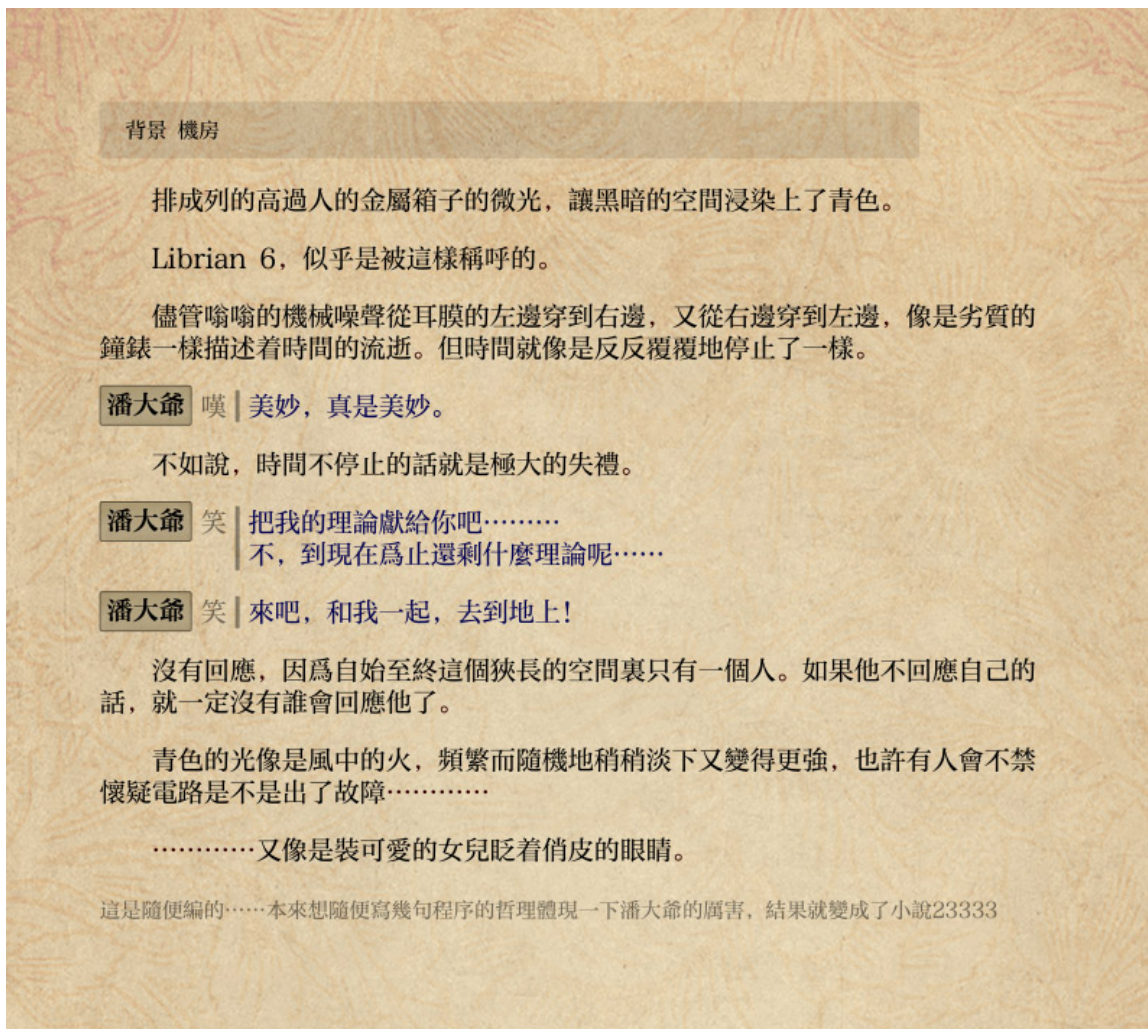
「導出pdf」顧名思義地用於將你的單個劇本文件導出pdf。

導出pdf與工程無關，只對單獨的劇本文件生效，也可以配置自己的css樣式表。

將劇本文件「編譯」之後，生成一個pdf文件，去除控制語句同時添加語義和着色。

當你想要將你的劇本與他人分享時，導出pdf可以提供極好的可讀性。

默認樣式下導出的pdf看起來是這樣：



(pdf已經不算是遊戲了吧.....)

# 前言

這一章的內容和前面的內容不太一樣，涉及到了一些和核心組件較為接近的內容——如果你有一些編程語言的基礎的話會比較好。

需要用到知識限於HTML和python。

如果你看到這兩個單詞時想到的是「前面這個非常H，後面這個是蛇(大概)」，說不定跳過這一章會比較好。

不過人必須要持續學習——武裝自己的應該是知識而不是一堆廢銅爛鐵。對你來說人生還有80年，或者60年，或者40年，即使什麼都不會，從現在開始學習python的話，只需要一年就能學有所成年薪過萬——然後就沉迷寫程序再也不寫劇本了(笑)。

(然後我再趁現在推薦幾個野雞網校賺一波廣告收入)

(之後才發現那句名言原來已經在這個文檔的前面說過一次了)

# 畫面的定製

這一節需要用到HTML和css。

## 使用css

如果想要改變adv畫面的表現，你可以指定額外的css文件。

之前雖然說過在「工程配置.yaml」裏可以設置「自定css」，但是沒有說該怎麼寫。

在默認的adv畫面(即 {librian路徑}/html/adv.html)中，有這些元素：

```
總畫面：
  配置面板：
  工具欄：
  adv畫面：
    覆蓋：
    選項：
    cg：
    bg：
    立繪：
    對話歷史：
    對話框：
      名字框：
        名字：
        名字框背景：
      話語框：
        話語：
        話語框背景：
      對話框背景：
  提示：
```

把你的樣式寫進自定css裏以後，在工程啓動時，Librian會一併加載你的css。這樣一來樣式就應用了。

## 利用動態類

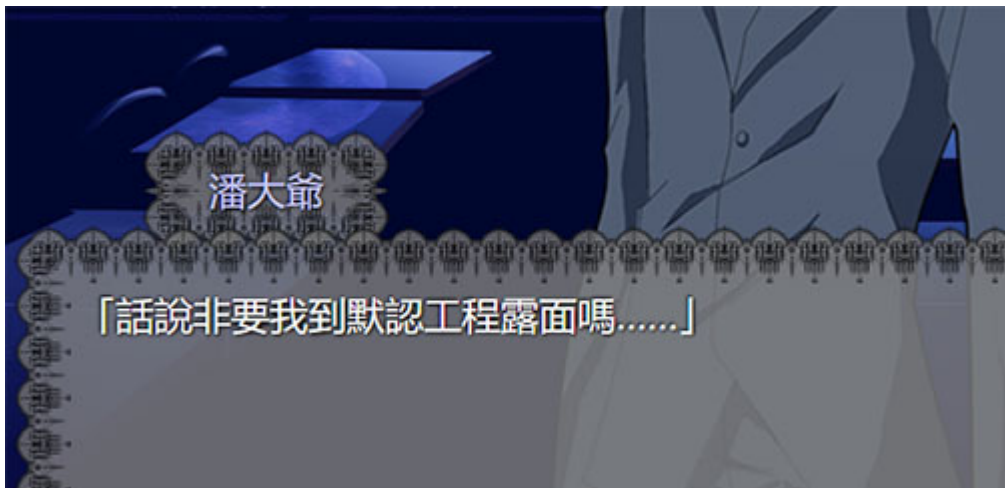
在演出過程中，對話框會被動態地附上帶有人名的類，即

`class="人物--{人物名}"` 。

可以利用這個類給不同的人設置效果。

```
.人物--潘大爺 #名字框{  
    color: #ccf;  
}
```

這樣一來，只有 潘大爺 的名字變成淺藍色。



## 標題畫面

標題畫面是一個可以天馬行空地創作的html。不過爲了實現標題畫面的功能，還需要一點花樣。

在標題畫面加載時就會獲得一個 山彥 對象，它有一些在標題畫面用得上的方法。

(話說爲什麼是山彥？)

像是這樣：

```
<button click='山彥.開始()'>  
    開始遊戲  
</button>
```

方法包括：

- `山彥.開始()`  
調用之後Librian開始讀取劇本，當前頁面由標題頁面轉到adv頁面。
- `山彥.從title讀檔()`  
讀檔，並在讀檔完成後轉到adv頁面。
- `山彥.從劇本開始(劇本路徑)`  
指定一個劇本的路徑，從這個劇本開始adv演出。

# 特效

這一節需要用到css。

## 使用特效

Librian使用css來實現特效和動作。

潘大爺 [灰]「什麼情況！」



這裏的 灰 就是特效。

在爲人物添加特效時，添加一個用方括號 [] 圈住的特效名就可以了。

## 內置的特效

### 灰

使目標變成灰色(去除飽和度)。

### 透明

使目標變得透明(施加一半的透明度)。

### 淡入

在出現時使用動畫，短暫的漸變。

### 下入

在出現時使用動畫，短暫的從下方的漸變。

## 定義特效

首先，在上個章節 [進階/畫面的定製](#) 中指定「自定css」後，向css中添加類。

( [透明](#) 本應是內置的特效，這裏先假裝還沒有內置，算你自己寫出來的吧)

```
.透明{
  opacity:0.5;
}
```

然後在人物處調用：

潘大爺 [透明]「相位轉移！」



動作也是一樣的，只要先定義幀然後用animation屬性就行了。

```
@keyframes _淡入{
  0%{opacity:0;}
  100%{opacity:1;}
}
.淡入{
  animation:_淡入 0.6s;
  animation-fill-mode:forwards;
}
```

## 畫面元素的特效

> EF bg 灰



爲html元素「bg」設置特效「灰」，在應用之後所有的背景都會變成灰色的。

> EF adv畫面 灰

爲整個畫面設置特效「灰」，適用於過去的記憶等場景。



如果不指定特效就會把原有的特效清除。  
元素的特效和立繪的特效是可以通用的。  
此外，也可以往元素上套用動作特效(如果確實有必要的話)。

關於adv畫面中元素的層級，參考畫面的定製。

## 注意

引擎支持的css和chrome是相同的。如果你發現你設置的特效沒有起作用的話，  
可以先檢查一下chrome是否支持。

(順便吐槽一下css的背投濾鏡感覺很容易實現嘛爲什麼默認不支持的.....還有爲什麼ps裏明明有智能濾鏡卻沒有濾鏡調整圖層?)



# 嵌入python

這一節可能需要用到python。

Librian的嵌入python可以用於實現劇本的邏輯，也可以成為獨立的擴展功能。如果你想要完成一個完整的galgame的話，嵌入python應該是必須用到的功能吧。

不過這並不意味着你必須去學習這門編程語言——如果只是簡單的功能的話，看樣例就足夠了。

## 從簡單的例子開始

**hello world:**

```
潘大爺 「準備好了嗎？」
...
print('hello, world')
...
潘大爺 「你在控制檯窗口上看到剛纔輸出的 hello world 了嗎？」
```

如果你看到了hello world說明你已經成為了python大師(笑)，馬上就可以從這裏畢業了。



**定義變量:**

```

潘大爺 「試着定義一些變量。」
'''
hp=150
hp-=50
print(f'潘大爺還有{hp}點生命值！')
'''
潘大爺 「看來我的生命值已經如同風中殘燭了。」

```

運行結果：

```

潘大爺還有100點生命值！

```

`print` 只會輸出在控制檯上，最終發佈的遊戲並沒有控制檯，所以這個消息是給開發人員看的。

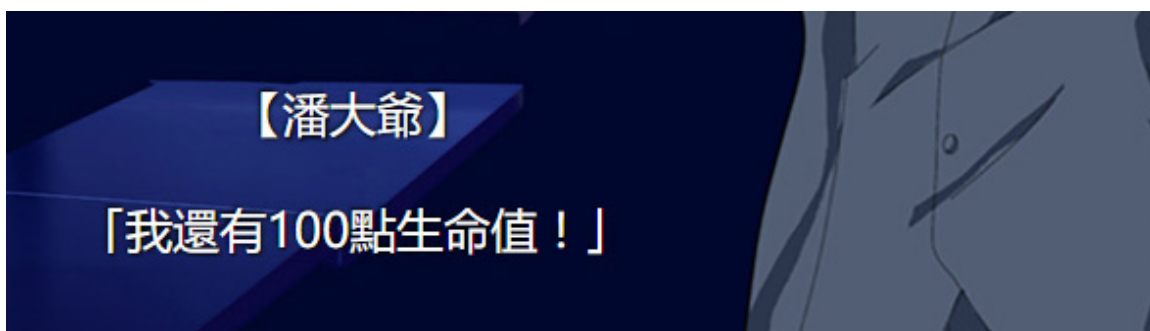
如果你把 `print` 換成 `fusion` 的話，「潘大爺還有100點生命值」除了會顯示在控制檯上以外，也會顯示在adv畫面中。這種方式顯示的文字和劇本一樣經過編譯——也就是說在前面寫上人名的話就會變成對話。

輸出到adv界面：

```

潘大爺 「我試試這樣說話.....」
'''
hp=150
hp-=50
fusion(f'潘大爺 「我還有{hp}點生命值！」')
'''
潘大爺 「感覺不錯。」

```



對了，每一次執行是互相關聯的，因此你可以使用上次的代碼塊留下的變量進行判斷。

## 稍微提升

接下來就是分支跳轉了，之前說了劇本函數 `WARP`，接下來介紹通用的 `if`：

```

潘大爺 「if真是太棒了。」
'''
hp=5
hp=-10
if hp<0:
    print('大爺死了。')
    goto('gameover.play')
'''
潘大爺 「看來我還活着。」

```

劇本「`gameover.play`」中的內容：

```

潘大爺 「騙人的吧？」
'''
adv_end()
'''

```

在潘大爺的生命小於0的時候，跳轉到劇本「`gameover.play`」。

在「`gameover.play`」，潘大爺說出「騙人的吧？」之後，調用 `adv_end` 結束遊戲。

## 難題

提供選擇肢的 `choice` 函數並不是很好理解。

`choice` 在adv演出中提供幾個選項，每一個選項都是一個組，包括一個選項名和一個回調函數。

下面是樣例。

```

def f():
    好感度+=1      #英文真難！
    choice(
        ('跳到一章',lambda: goto('一章.play')),
        ('跳到二章',lambda: goto('二章.play')),
        ('刷好感度', f )
    )

```

爲什麼分了三行..... `lambda` 是什麼玩意.....爲什麼單獨定義了函數 `f` .....  
好像對於沒python的小夥伴來說太難了點，要解釋的話甚至還設計到了語句和表達式的關係.....

只是依葫蘆畫瓢的話會比較好嗎，不過搞不清楚什麼時候填 `lambda` 的話就連依葫蘆畫瓢都做不到了.....都是不學習的錯(傲嬌臉)！

如果你覺得這個函數有更簡單的設計方法的話，歡迎在 缺少的功能/一些遺憾 提出寶貴意見。

在 劇本功能 中說過的 `WARP`，實際上是 `choice` 和 `push` 的語法糖。

```
WARP [接受治療,壞結局.play] [放棄治療,好結局.play] [蘿莉治療,一章.play,原點]
```

相當於

```
choice(
    ('接受治療',lambda: call('壞結局.play')),
    ('放棄治療',lambda: call('好結局.play')),
    ('蘿莉治療',lambda: call('一章.play','原點'))
)
```

## 究極

試着嵌入一個「在副線程進行一段時間的操作並在完成時給出反饋」的劇本。

```
潘大爺 「点一下来开始大约持续5秒的自動更新！」
...

import threading
def 更新():
    time.sleep(5)    # 连接到你自己的发布地址
    goto(lable='更新完了')
    import 窗口,劇本
    劇本.讀者.步進()
    窗口.更新()
t = threading.Thread(target=更新)
t.setDaemon(True)
t.start()
...

*back
潘大爺 「更新还没有结束，更新完了时无需点击会自动进入下一句。」
```

```

'''
goto(lable='back')
'''

*更新完了
潘大爺 「更新完了。」

```

這個我怎麼解釋.....

(話說下次提供一個副線程支持吧)

試試RPG打架。

```

'''
from random import randint
def 攻擊(a, b):
    傷害 = int(randint(8, 12) / 10 * a['攻擊力'])
    b['當前生命'] -= 傷害
    fusion(f"{a['名']}攻擊{b['名']}！造成{傷害}點傷害！{b['名']}的生命值{b['當前生命']}/{b['總生命']}。")
def 死亡一指(a, b):
    b['當前生命'] = 0
    fusion(f"{a['名']}對{b['名']}使用死亡一指！{b['名']}的生命值{b['當前生命']}/{b['總生命']}。")
大爺 = {'名': '大爺', '速度': 10, '攻擊力': 30, '當前生命': 100, '總生命': 100}
敵人 = {'名': '敵人', '速度': 5, '攻擊力': 9, '當前生命': 80, '總生命': 80}
'''

'''
if 大爺['速度'] > 敵人['速度']:
    大爺的回合 = True
    fusion('潘大爺 「智者先攻」')
else:
    大爺的回合 = False
'''

*決鬥開始

'''
if 大爺的回合:
    choice(
        ('攻擊', lambda: 攻擊(大爺, 敵人)),
        ('死亡一指', lambda: 死亡一指(大爺, 敵人))
    )
else:
    攻擊(敵人, 大爺)
大爺的回合 = not 大爺的回合

```

```

'''
'''
if 敵人['當前生命'] <= 0 or 大爺['當前生命'] <= 0:
    adv_end()
'''

'''
goto(lable='決鬥開始')
'''

```

(話說死亡一指不是應該把目標旁邊的人殺死嗎(笑))

## 注意

這部分python代碼運行在一個相對獨立的环境裏，所以不能調用外部的函數，好處是愉快地定義變量也不會和外部產生名衝突。

當然你可以使用python的黑魔法來使用我的其他函數。

不過如果你的代碼過長，我建議寫成獨立的python文件，然後 `import` 。

此外，費時的操作最好另開一個線程來執行，主線程阻塞的話畫面會停止響應。

## 一些坑

### 實際的執行順序

每次的py塊總是會執行到底，也就是說 `goto`，`call`，`fusion` 並不是即時的，他們真正的作用只是改變了劇本棧的順序，導致在一個塊中多次調用會產生出乎意料的演出順序。

(我雖然想將讀者的實例改為遞歸的來解決這個問題，但是卻發現這樣存在一個不能存檔的問題。)

舉這樣一個例子：

```

'''
call('第一章.play')
call('第二章.play')
'''

```

寫的人的想法是想要先演出第一章，第一章結束之後返回調用點，然後演出第二章。

但實際上的執行順序是 `call('第一章.play')` -> `call('第二章.play')` -> 演出第二章 -> 演出第一章。

因為在 `call` 時候劇本的棧頂被先後壓入了第一章和第二章，成為 `(.., 當前劇本, 第一章, 第二章)`，第二章在棧頂導致先演出。

我想過的解決辦法是在每一次 `call` 的時候直接重建一個讀者實例，然後直接進入這個新實例執行，在這個實例執行完畢之後就可以回到調用點執行之後的語句了.....但問題是存檔時只能儲存劇本棧不能儲存調用棧，也就是無法在新劇本執行完畢之後返回調用點。

也就是說，由於這個問題，這個做法暫時只能寫成：

```
'''
call('第一章.play')
'''

'''
call('第二章.play')
'''
```

如果你有方法的話，歡迎掃描前面的QRcode.....不，這個還是火速聯繫我吧。

## goto在副線程調用時不會立即跳轉

`goto` 實際上並不會改變當前adv畫面，生效的是你之前的點擊。也就是說，定時調用需要再調用一次更新方法。

## 內置函數參考

在py/endpy中，這幾個函數是預備了的：

```
goto(path=None, lable=None)

call(path=None, lable=None)

fusion(s)
```

```
choice(c1, c2, c3, ...)

adv_end()
```

## goto/call

`goto` 跳轉到另一個路徑為 `path` 的劇本（默认是自身）的一個躍點的位置。如果躍點 `tag` 不指定就會跳到開頭。

`push` 的功能是 `goto` 加上目標劇本演出結束後回到調用處。

```
goto(path='第9章.play', lable='躍點A')
```

在目標文件裏要有對應的躍點。

```
*躍點A
蘿莉 「喵喵喵。」
.....
```

## fusion

將一個字符串編譯後顯示在adv畫面中。

## choice

由選項名和回調函數組成選項。

## adv\_end

清空劇本棧使得adv演出進入結束狀態。



## 內置函数的別名

爲了照顧某些.....嗯，看到英文就頭大的開發者，還有狂熱的民族主義者，而提供了這樣的名字.....

- `goto` = 跳轉
- `call` = 調用
- `choice` = 產生選項
- `adv_end` = 演出終了
- `fusion` = 同化（沒錯這個不是「融合」而是「假魂的同化」）

這樣一來就可以中文編程——

```

'''
生命=99
同化(f'潘大爺 「我還有{生命}點生命值！」')
'''

```

# 設計回想

這一節需要用到HTML。

雖然Librian沒有提供「場景回想」之類的功能，不過自己實現起來也很容易。

接下來，我們通過一個實例來瞭解如何製作場景回想。

## 劇本文件的假設

我們假設你的劇本文件包括以下場景：

```
1-1 原點.play
1-2 約會.play
1-3 kiss.play
2-1 帶回家.play
2-2 枕頭大戰.play
```

(這什麼東西我自己寫的都想笑了)

需要回想的場景為 `1-3 kiss.play` 和 `2-2 枕頭大戰.play`。

## 設置回想頁面

首先你應當創建一個回想的HTML頁面——需要你使用基本的HTML技術製作一個簡單的網頁。

這個頁面應當和標題畫面連接，可以從標題跳轉過來，也可以按返回按鈕跳轉回標題。

(當然如果你能把這個做成標題頁面的一部分就更好了)

在回想頁面上放置兩張圖片，分別為「kiss」和「枕頭大戰」的CG。

這樣一來，你就有了一個華而不實的回想頁面。

接下來要為圖片添加方法，使得它們被點擊時，演出系統從對應的劇本進入。

在回想頁面中，對「kiss」的CG添加onclick 從劇本開始('1-3 蘿莉H.play') 。

在回想頁面中，對「枕頭大戰」的CG添加onclick 從劇本開始('2-2 蘿莉再H.play') 。

這個函數可以在畫面的定製中找到。

這樣就足夠完成基本的回想功能了。

## 一些遺憾

如果你發現什麼有鬼的問題，或者有想要的新功能的話，歡迎在任何時候向我反饋。

(雖然說了「任何時候」，但是H的時候是不行的。)

通信方法有：

- 在github頁面提交issue。 (issue多的話顯得Librian比較有人氣.....)
- 發送郵件到我的郵箱RimoChan@stu.xmu.edu.cn。

## 已知問題

在unix系統中無法找到路徑？

編程真難.....我還是改行吧 (對了聽說最近有一個叫後宮王的職業好像前程似錦，不如去試試吧)  
！

windows下的pyqt5的QtWebEngine在加載路徑的時候可以使用相對路徑，但似乎unix就不行。雖然有考慮過全部轉換為絕對路徑，但實際上在轉換的時候會出問題，因為自定義標題在引用js的時候應該不能用絕對路徑找到js文件，怎麼辦呢.....好在unix用戶比較少 (潘大爺：???)

## 和描述不一致的功能

演出功能設定

演出功能總是全部開啓的，設定為關閉並不會讓它們失效。

## 還想做的功能

縮進控制

with塊，利用縮進和進入退出控制來改變狀態，類似這樣：

```
> with CG('蘿莉H.png'):
    蘿莉 「呀~」
    潘大爺 「你真可愛。」
    潘大爺撫摸蘿莉的頭。
    蘿莉 「喵喵喵！」
H好了。
```

這樣只有在說H話的時候才有HCG。

if塊，顧名思義，類似這樣：

```
> if 好感度==0:
    蘿莉 「CNM！」
```

(笑)

沒錯我就是故意打的big標籤。

## 移動端引擎

最近手遊很流行.....然後聽說pyqt5可以在安卓運行，於是就想能不能做一個.....  
不過可能會卡成sb就是了.....

# 關於Librian

我知道你們看到「關於」兩個字就不會點進來了，所以這裏隨便說點什麼無理取鬧的東西應該沒關係。

(小心！本章除了「使用許可」以外的所有內容都是不可信的！)

## 關於命名

Librian嘛，構詞自然是「書本」+「人」，代表「紙上的人」。

靈感來自「紙の上の魔法使い」——在這部作品裏，有一種魔法之書，只要打開來讀的話，書中的故事就會在現實中上演。

(雖然看起來很像利比裏亞人)

## 關於作者

Librian的作者是「黃巍」，是真實存在的人物，並不是「リモちゃん」。

作者現在正在享受很多美少女左擁右抱的淫靡生活！

因為做Librian很麻煩，所以已經全交給美少女了！github上給你回issue的也是美少女！

有些時候作者的名字像是「リモちゃん」，那個並不是真實存在的人物，也不會編程。

關於リモちゃんの頭像是什麼.....類比的話，看到真紅頭像也不會覺得網線的另一邊是真的真紅對吧。

大概像「正如生者會夢見死者，死者也會夢見生者」那樣，「正如真實的人會用虛假的人做頭像，虛假的人也會用真實的人做頭像」。

## 開發歷史

Librian從2017年開始開發，到現在2019年，經歷3年時間(這樣算真的沒問題嗎?)，共計1位開發者(說出來丟人嗎)。

而且在開發期間由於房間裏的美少女逐漸增加而喪失動力！

在2017年以前其實還有kirikiri的版本和flask的版本(可以聯網)，不過經過各種考慮以後還是做成了現在的樣子。

## 關於文檔中出現的人名和地名

注意，以下內容可能含有大量**詐騙**成分，目的是爲了騙來可愛的女孩子。如果你是男孩子的話，請謹慎閱讀，當心上當受騙！

- 莆田理工大學：  
The LG University of PT—隱藏在莆田的深山老林裏的大學。
- 潘大爺：  
莆田理工大學現任校長。
- 林梓雨：  
潘大爺以前的老師。
- 張主任：  
莆田理工大學文科院的主任，擅長哲♂學。
- 長頸鹿：  
看歌劇的動物。  
(這東西在哪裏出現過???)
- 晉平公 | 師曠：  
古代的人。  
(完全是來湊數的吧！)

## 文檔中一些無關緊要的問題

1. 爲什麼安裝會變成女裝？  
(聽起來很能想到「爲什麼烏鴉像寫字檯」)

在「安裝」章節中提到了「安裝的上面被擋住就會變成女裝」。

爲了實現這個效果，你要把「安裝」放大，然後用手擋住左上角，就可以看見女裝了。

## 2. 潘大爺爲什麼要對着箱子說話？

在「簡介」章節中出現了一段莫名其妙的劇本。

這段對話發生在運載艦的艦橋中。「高過人的金屬箱子」是主機，當時潘大爺正在維護電腦。

(這段故事是同人作品，和莆田理工大學沒有關係)

## 3. 爲什麼插入js會很H？

在「劇本功能」裏將「插入js」寫成了「插入javascript」。

因爲插入js會被警察抓起來。

(插入jk不違法所以對方願意的話就可以插入)

# 使用許可

**Librian 可以 免費 用於任何 商業或非商業用途。**

也就是說如果你用Librian做出了遊戲，放在steam上收費也是可以的。

(「Librian真是太棒了！」——如果你能在你的遊戲主頁這樣宣傳的話就再好不過了。)

但是要注意，Librian本身使用了MPL(魔理沙公共許可証)2.0協議。

如果你打算修改Librian的代碼(這裏指的是引擎本身而不是別人用Librian寫的劇本)，也是可以的，不過最好先讀一下許可証。

(還有這個「魔理沙公共許可証」怎麼聽起來有點色情.....)



# 關於實現

## Liber語言的編譯

Librian使用「Liber語言」書寫劇本。

(這名字當然是我現編的.....)

Liber的寫法接近真實的劇本和自然語言，特點是易讀易寫。

設計Liber語言的文法時，發現現實的劇本幾乎是三型文法——  
因此沒有採用一般語言的編譯手段，而使用了基於正則表達式的簡單編譯方法。

### 按行解析

在Liber中，大部分情況下，`換行` 即為 `語句` 的終結。

以一行 `對話` 為例：

```
馬克吐溫 「寫作很簡單，你所要做的只有把不正確的詞都劃掉。」
```

這句 `對話` 包含一個 `人名`，一個 `對話`，`分隔符` 只有 `空格` 和 `直角引號「」`。

因此，使用了這樣的正則表達式來匹配並解析——

```
(?P<人名>.+?) + 「(?P<對話>.*?)」
```

《優秀的語言是不需要詞法分析的》

前幾天我面試一個**編程語言**，連續幾個專業問題它都沒答上來。

尷尬之餘，我問它「你沒有理想嗎？你現在最渴望的事情是什麼？」

**編程語言**轉悠着大眼睛，不假思索道「找一個詞法分析器！」

這樣一來 `馬克吐溫` 的名言就會被解析為字典對象 `{'人名': '马克吐温', '對話': '寫作很簡單，你所要做的只有把不正確的詞都劃掉。'}`。

因為對話中還可能含有 `表情`，所以得在中間加上——

```
(?P<人名>.+?) +(\((?P<表情>.+?)\))? * 「(?P<對話>.*?)」
```

這樣一來就足夠解析樣例中 潘大爺 的話語了。

```
潘大爺 (笑)「來吧，和我一起，去到地上！」
```

可以解析為字典對象 `{ '人名': '潘大爺', '表情': '笑', '对话': '來吧，和我一起，去到地上！' }`

之後無論加入再多特性，只要屬於三型文法，就可以統統塞進去。

## 類型識別

使用每一個正則表達式對應每一種語句類型。

編譯器在讀入一行時，對每個正則表達式進行匹配。

如果可以匹配多個則拋出編譯錯誤，如果無法匹配則使用默認類型 旁白 。

## 遞歸結構

如果把大量語法都塞進一個正則表達式裏，表達式就會變得冗長，維護就會變得困難。

因此將複雜的正則表達式組織為字典，像是—

```
'....(?P<元素1>).....(?P<元素2>)....': {
    '子樹': {
        '元素2': {
            '?P<小元素1>...?P<小元素2>....': None
        }
    }
},
```

使用遞歸的正則匹配——樣例中先提取出兩個元素，再從第二個元素中提取出小元素們。

## 預處理

有時，對話 中可能包含 換行 。

莎士比亞 「戰爭將銅像推翻，  
內訌將城市化為一片廢墟。」

預處理需要將需要續行的情況連成一行。

因此設置了用於預處理的 續行組 。如果當前行能匹配到 續行組 中的任意正則表達式，就把下一行加入當前行。

在本例中，需要續行的情況是對話的 直角引號 還沒有閉合，因此 續行組 有正則表達式 缺少右引號的對話 ，像是這樣——

```
(?P<人名>.+?) +(\((?P<表情>.+?)\))? *「([^\"]*)
```

因為現代漢語中不允許出現 「「」」 的引號嵌套，因此可以在遇到第一個 」 時直接判定閉合。

## 編譯樣例

```
> BG 機房.png
排成列的高過人的金屬箱子的微光，讓黑暗的空間浸染上了青色。
Librian 6，似乎是被這樣稱呼的。
儘管嗡嗡的機械噪聲從耳膜的左邊穿到右邊，又從右邊穿到左邊，像是劣質的鐘錶一樣描述着時間的流逝。但時間就像是反反覆覆地停止了一樣。
潘大爺（嘆）「美妙，真是美妙。」
不如說，時間不停止的話就是極大的失禮。
潘大爺（笑）「把我的理論獻給你吧.....不，到現在為止還剩什麼理論呢.....」
潘大爺（笑）「來吧，和我一起，去到地上！」
沒有回應，因為自始至終這個狹長的空間裏只有一個人。如果他不回應自己的話，就一定沒有誰會回應他了。
青色的光像是風中的火，頻繁而隨機地稍稍淡下又變得更強，也許有人會不禁懷疑電路是不是出了故障.....
.....又像是裝可愛的女兒眨着俏皮的眼睛。
# 這是隨便編的.....本來想隨便寫幾句程序的哲理體現一下潘大爺的厲害，結果就變成了小說
23333
```

結果：

```
{'類型': '函數調用', '原文': 'BG 機房.png', '函數': 'BG', '參數表': [{ 'a': '機房.png' }]}
{'類型': '旁白', '旁白': '排成列的高過人的金屬箱子的微光，讓黑暗的空間浸染上了青色。'}
{'類型': '旁白', '旁白': 'Librian 6，似乎是被這樣稱呼的。'}
{'類型': '旁白', '旁白': '儘管嗡嗡的機械噪聲從耳膜的左邊穿到右邊，又從右邊穿到左邊，像是劣質的鐘錶一樣描述着時間的流逝。但時間就像是反反覆覆地停止了一樣。'}
{'類型': '人物對話', '名': '潘大爺', '代': None, '特效': None, '顏': '嘆', '語': '美妙，真是美妙。'}
{'類型': '旁白', '旁白': '不如說，時間不停止的話就是極大的失禮。'}
{'類型': '人物對話', '名': '潘大爺', '代': None, '特效': None, '顏': '笑', '語': '把我的理論獻給你吧.....不，到現在為止還剩什麼理論呢.....'}
{'類型': '人物對話', '名': '潘大爺', '代': None, '特效': None, '顏': '笑', '語': '來吧，和我一起，去到地上！'}
{'類型': '旁白', '旁白': '沒有回應，因為自始至終這個狹長的空間裏只有一個人。如果他不回應自己的話，就一定沒有誰會回應他了。'}
{'類型': '旁白', '旁白': '青色的光像是風中的火，頻繁而隨機地稍稍淡下又變得更強，也許有人會不禁懷疑電路是不是出了故障.....'}
{'類型': '旁白', '旁白': '.....又像是裝可愛的女兒眨着俏皮的眼睛。'}
{'類型': '註釋', '註釋': '這是隨便編的.....本來想隨便寫幾句程序的哲理體現一下潘大爺的厲害，結果就變成了小說23333'}
```