



Python 與 LINE Bot 機器人 全面實戰 特訓班 Flask 最強應用

碁峯資訊

【投影片使用規範與聲明】本投影片僅供非營利教學用途，教師得搭配用書授課講解使用，可於用書期間內將投影片置放於學校內部網站，但需有帳號權限機制，且僅供修課學生瀏覽使用。敬請老師善盡著作權保護之責，請勿將投影片任意散布與販售，亦不得以任何形式或方法轉載內容使用。

LINE Bot 進階互動功能

6.1 回應樣板訊息

6.2 圖片地圖及日期時間

LINE Bot 設定

LINE Business ID

使用LINE帳號登入

或

使用商用帳號登入

建立帳號

使用LINE商用ID須遵從服務條款等規定，登入後即視為您同意相關條款內容。

🔗 關於LINE商用ID

繁體中文 ▾

[Help](#) [服務條款](#) © LINE Corporation

<https://developers.line.biz/zh-hant/>

LINE

透過行動條碼登入



🔄 更新行動條碼

請開啟Android / iOS版本的LINE行動條碼掃描器，並掃描此行動條碼。

透過電子郵件帳號登入

[掃描行動條碼的方法請見此處](#)

<https://developers.line.biz/zh-hant/>

Connect with CLOVA

Console

Documentation

About LINE Developers site

Products



LINE Login

讓 LINE 用戶輕鬆登入並連接開發者的服務。



Messaging API

開啟與 LINE 用戶的雙向溝通。



LINE MINI App

Connect your service and users with LINE MINI App.



LINE Things

透過 LINE 連接 Bluetooth® 低功耗裝置。



LINE CLOVA

A problem solver and user experience optimizer,



CLOVA Extensions Kit

透過 CLOVA 擴充套件，整合

https://developers.line.biz/console/provider/1657406077

LINE Developers

[News](#)[Products](#)[Documentation](#)[FAQ](#)[Glossary](#)[Community](#)[Blog](#)[Console home](#)[Providers](#)[Admin](#)[NKUST_YiWen](#)[Tools](#)[TOP](#) > [NKUST_YiWen](#)

NKUST_YiWen

[Channels](#)[Roles](#)[Settings](#)

Create a new channel

[Admin](#)

NKUST_YiWen

Messaging API

Console home

Providers

Search...

Admin

NKUST_YiWen

Tools

Support

TOP

Providers (1)

Create

Search...



Provider

✓ NKUST_YiWen

LINEBot教室



Messaging API

Messaging API LIFF Security Statistics Roles

ings

ation

You can change your app name and icon in [LINE Official Account Manager](#).

Channel ID

1657395872

Channel icon



主頁 分析 聊天 基本檔案 LINE VOOM 擴充功能 口袋商店

進階影片訊息

多頁訊息

推廣相關

優惠券

集點卡

問卷調查

聊天室相關

加入好友的歡迎訊息

圖文選單

資料管理

受眾

追蹤 (LINE Tag)

增加好友人數

加入好友指南

加好友廣告

圖文選單

您可建立具有視覺效果的互動型選單，於聊天室中提供優惠券、網址連結及相關行銷資訊以進行宣傳。
本頁面僅可顯示透過本頁面所建立的圖文選單。

目前顯示的選單

此為顯示於聊天室供用戶點選的圖文選單。

※若透過Messaging API設定圖文選單，用戶實際看到的選單可能與此有所差異



傳送聲音



傳送影片

標題 LINEBot測試2

✓ 使用期間 2022/08/24 16:50 - 2022/08/30 23:59 ✗

動作

- 文字 - @傳送聲音
- 文字 - @傳送影片

預約 / 公開

待設定



YYYY/MM/DD

- + 訊息項目
 - 圖文訊息
 - 進階影片訊息
 - 多頁訊息
- 推廣相關
 - 優惠券
 - 集點卡
 - 問卷調查
- 聊天室相關
 - 加入好友的歡迎訊息
 - 圖文選單
- 資料管理
 - 受眾
 - 追蹤 (LINE Tag)
- 增加好友人數
 - 加入好友指南
 - 加好友廣告

圖文選單

您可建立具有視覺效果的互動型選單，於聊天室中提供優惠券、網址連結及相關行銷資訊以進行宣傳。

刪除 儲存草稿

基本設定

標題

LINEBot測試2

標題為方便後台管理用，不會向用戶顯示。

使用期間

2022/08/24

16:50

~

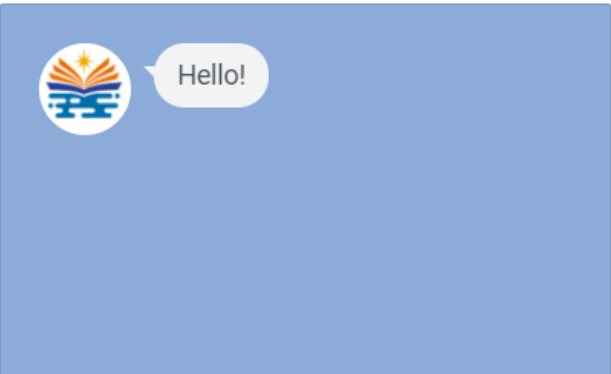
2022/08/30

23:59

清除

內容設定

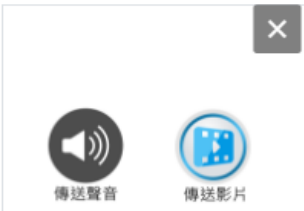
預覽 ?



版型

變更

圖片



建立新選單

LINE Official Account Manager



KNUST_YiWen @861tvqqn

輕用量

1

回應模式：聊天機器人



Sophie (廖奕雯)

[主頁](#) [分析](#) [聊天](#) [基本檔案](#) [LINE VOOM](#) [擴充功能](#) [口袋商店](#)

設

+ 訊息項目

圖文訊息

進階影片訊息

多頁訊息

🗑 推廣相關

優惠券

集點卡

問卷調查

🔗 聊天室相關

加入好友的歡迎訊息

圖文選單

📁 資料管理

受眾

追蹤 (LINE Tag)

圖文選單

您可建立具有視覺效果的互動型選單，於聊天室中提供優惠券、網址連結及相關行銷資訊以進行宣傳。
本頁面僅可顯示透過本頁面所建立的圖文選單。

目前顯示的選單

此為顯示於聊天室供用
※若透過Messaging AI

圖文選單

您可建立具有視覺效果的互動型選單，於聊天室中提供優惠券、網址連結及相關行銷資訊以進行宣傳。



傳送聲音

儲存草稿

預約 / 公開

待審

基本設定

標題

LINEBot測試三

標題為方便後台管理用，不會向用戶顯示。

使用期間

2022/08/25

00:00

~

2022/08/26

23:59

清除

選擇版型

大型

適合顯示版面較大或項目較多的圖文選單。



顯示版型框線 ☒

小型

適合顯示版面較小或項目較少的圖文選單。



動作

▼ A

類型

文字

可設定為「關鍵字回應」

@按鈕樣板

▼ B

類型

文字

可設定為「關鍵字回應」

@確認樣板

▼ C

類型

文字

可設定為「關鍵字回應」

@轉盤樣板

▼ D

類型

文字

可設定為「關鍵字回應」

@圖片轉盤

取消

套用

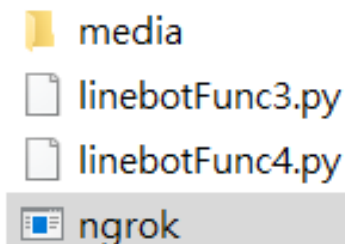
執行 ngrok.exe

ngrok http 5000

<https://934b-140-133-74-88.ngrok.io>

<https://934b-140-133-74-88.ngrok.io/callback>

<https://934b-140-133-74-88.ngrok.io>




D:\高科大_1111課程\111-1_程式設計(三)\Python與LINE Bot機器人全面實戰特訓班--Flask最強應用學習資源\本書範例\ch06\ngrok.exe - ngrok

ngrok by @inconshreveable

```
Session Status      online
Session Expires    1 hour, 59 minutes
Update              update available (version 2.3.40, Ctrl-U to update)
Version             2.3.18
Region              United States (us)
Web Interface        http://127.0.0.1:4040
Forwarding           http://934b-140-133-74-88.ngrok.io -> http://localhost:80
Forwarding           https://934b-140-133-74-88.ngrok.io -> http://localhost:80

Connections
  ttl    opn    rtl    rt5    p50    p90
    0      0    0.00   0.00   0.00   0.00
```

Console home

Providers 

Search...

Admin

NKUST_YiWen

TOP > NKUST_YiWen > NKUST_YiWen > Messaging API




NKUST_YiWen

Admin

Messaging API

<https://934b-140-133-74-88.ngrok.io/callback>

Basic settings

 Messaging API

LIFF


Security

Statistics

Roles


TOP > NKUST_YiWen > NKUST_YiWen > Messaging API


Webhook settings

Webhook URL  <https://e82c-140-133-74-88.ngrok.io/callback>

Verify


Edit

Use webhook 

Webhook redeliv 

Error

The webhook returned an HTTP status code other than 200.(404 Not Found)

Confirm that your bot server returns status code 200 in response to the HTTP POST request sent from the LINE Platform. For more information, see [Response](#)  in the Messaging API Reference.

OK

Webhook settings

Webhook URL ?

https://934b-140-133-74-88.ngrok.io/callback

- ✓ Don't leave this empty
- ✓ Enter a valid HTTPS URL
- ✓ Enter no more than 500 characters

Update

Cancel

Webhook settings

Webhook URL ?

https://934b-140-133-74-88.ngrok.io/callback

Verify

Edit

Use webhook ?



Webhook redelivery ?



Success

OK

樣板回應訊息 TemplateSendMessage()

按鈕樣板

Button Template

確認樣板

Confirm Template

轉盤樣板

Carousel Template

圖片轉盤樣板

ImageCarousel Template

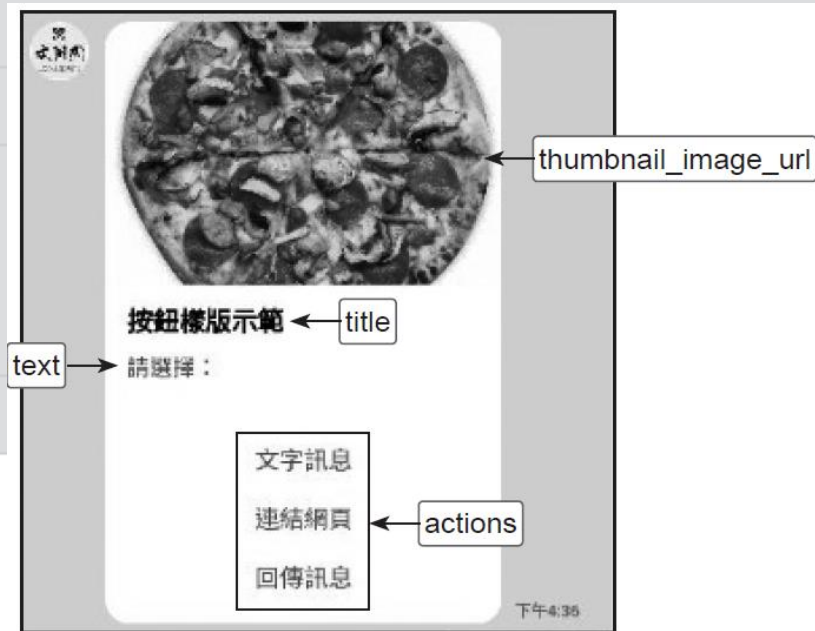
6.1 回應樣板訊息

6.1.1 按鈕樣板 (Button Template)

本節範例：在LINE 開發者頁面建立 ehappyFunc3 LINE Bot，加入圖文選單：版型使用「大型」的第二個版型(四個項目)，圖形上傳本章範例 <media/func3.png>，四個項目的類型皆選擇「文字」，傳送的文字分別設定為@按鈕樣板、@ 確認樣板、@ 轉盤樣板及@ 圖片轉盤。接著建立Flask 程式 <linebotFunc3.py>，讓LINE Bot回應使用者點選圖文選單的各項功能。

按鈕樣板的基本語法為：

```
訊息變數 = TemplateSendMessage(  
    alt_text= 取代文字 ,  
    template=ButtonsTemplate(  
        thumbnail_image_url= 圖片網址 ,  
        title= 主標題 ( 字型較大 ) ,  
        text= 說明文字 ( 字型較小 ) ,  
        actions=[  
            按鈕元件一 ,  
            按鈕元件二 ,  
            .....  
        ]  
    )  
)
```



6.1.1 按鈕樣板 (Button Template)

定義訊息變數

```
message = TemplateSendMessage(  
    alt_text = 取代文字,  
    template = ButtonsTemplate(  
        thumbnail_image_url = 圖片網址,  
        title = 主標題(字型較大),  
        text = 說明文字(字型較小),  
        actions = [  
            按鈕元件1,  
            按鈕元件2,  
            .....  
        ]  
    )  
)
```

按鈕樣板

Button Template

- 文字按鈕(MessageTemplateAction)
- 連結按鈕(URITemplateAction)
- 回傳按鈕(PostbackTemplateAction)
- 日期時間按鈕(DatetimePickerTemplateAction)

按鈕樣板

文字按鈕

MessageTemplateAction

連結按鈕

URITemplateAction

回傳按鈕

PostbackTemplateAction

按鈕樣板

Button Template

```
MessageTemplateAction(  
    label = 按鈕文字,  
    text = 回傳文字  
)
```

```
URITemplateAction(  
    label = 按鈕文字,  
    uri = 網址  
)
```

```
PostbackTemplateAction(  
    label = 按鈕文字,  
    text = 回傳文字,  
    data = Postback資料  
)
```

文字按鈕(MessageTemplateAction)

文字按鈕會回傳文字訊息，語法為：

```
MessageTemplateAction(  
    label= 按鈕文字 ,  
    text= 回傳文字  
)
```

執行結果為：

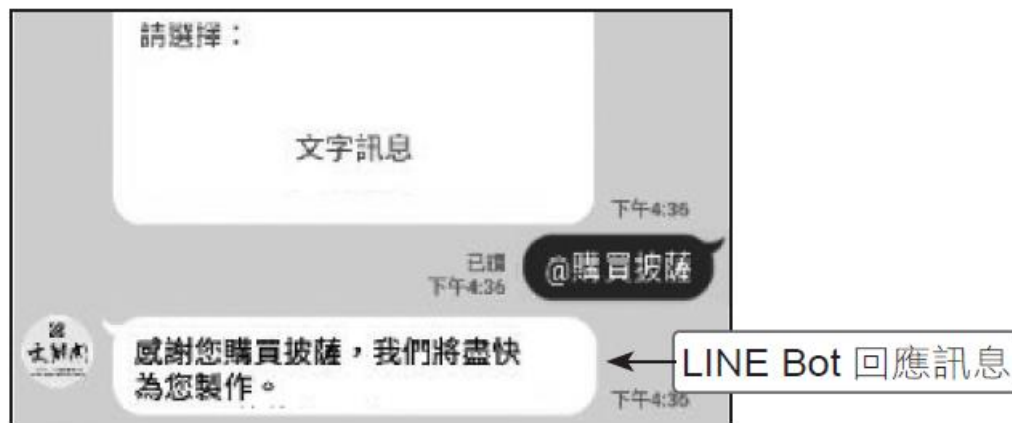


- 文字按鈕(MessageTemplateAction)
- 連結按鈕(URITemplateAction)
- 回傳按鈕(PostbackTemplateAction)
- 日期時間按鈕
(DatetimePickerTemplateAction)

設計者可以根據回傳值「@ 購買披薩」做後續處理，例如回應使用者已收到訊息，會盡快製作披薩：

```
if mtext == '@ 購買披薩':  
    message = TextSendMessage(  
        text = '感謝您購買披薩，我們將盡快為您製作。'  
    )  
    line_bot_api.reply_message(event.reply_token, message)
```

執行結果為：



連結按鈕(URITemplateAction)

連結按鈕會開啟指定的網頁，語法為：

```
URITemplateAction(  
    label= 按鈕文字 ,  
    uri= 網址  
)
```

回傳按鈕 (PostbackTemplateAction)

回傳按鈕除了可以回傳文字訊息外，還可以回傳一個Postback 資料，語法為：

```
PostbackTemplateAction(  
    label= 按鈕文字 ,  
    text= 回傳文字 ,  
    data=Postback 資料  
)
```

label 及text 與文字按鈕相同，「Postback 資料」的格式與網址列附加的參數雷同，語法為：

```
名稱一 = 值一 & 名稱二 = 值二 &……
```

Postback 資料不會在LINE 中顯示。

回傳按鈕的缺點是其處理程序較為複雜：使用點選回傳按鈕會觸發Postback 事件，後續動作必須在Postback 事件中處理。

```
1 @handler.add(PostbackEvent)
2 def handle_postback(event):
3     backdata = dict(parse_qsl(event.postback.data))
4     if backdata.get('action') == 'buy':
5         message = TextSendMessage(
6             text = '感謝您購買披薩，我們將盡快為您製作。'
7         )
8         line_bot_api.reply_message(event.reply_token, message)
```

使用者按回傳按鈕的執行結果為：



```
1 from flask import Flask
2 app = Flask(__name__)
3
4 from flask import request, abort
5 from linebot import LineBotApi, WebhookHandler
6 from linebot.exceptions import InvalidSignatureError
7 from linebot.models import MessageEvent, TextMessage, \
8     PostbackEvent, TextSendMessage, TemplateSendMessage, \
9     ConfirmTemplate, MessageTemplateAction, ButtonsTemplate, \
10     PostbackTemplateAction, URITemplateAction, \
11     CarouselTemplate, CarouselColumn, \
12     ImageCarouselTemplate, ImageCarouselColumn
13 from urllib.parse import parse_qs
14
15 #你的 CHANNEL_ACCESS_TOKEN
16 line_bot_api = LineBotApi('2MEVypC1TIUqmn1f18rrG89F6TBONrqkYF')
17 #你的 CHANNEL_SECRET
18 handler = WebhookHandler('737b5bf1884ff412f43fcc6ceade40d6')
19
20 @app.route("/callback", methods=['POST'])
21 def callback():
22     signature = request.headers['X-Line-Signature']
23     body = request.get_data(as_text=True)
24     try:
25         handler.handle(body, signature)
26     except InvalidSignatureError:
27         abort(400)
28     return 'OK'
```



```
29
30 @handler.add(MessageEvent, message=TextMessage)
31 def handle_message(event):
32     mtext = event.message.text
33     if mtext == '@按鈕樣板':
34         sendButton(event)
35
36     elif mtext == '@確認樣板':
37         sendConfirm(event)
38
39     elif mtext == '@轉盤樣板':
40         sendCarousel(event)
41
42     elif mtext == '@圖片轉盤':
43         sendImgCarousel(event)
44
45     elif mtext == '@購買披薩':
46         sendPizza(event)
47
48     elif mtext == '@yes':
49         sendYes(event)
50
51 @handler.add(PostbackEvent) #PostbackTemplateAction觸發此事件
52 def handle_postback(event):
53     backdata = dict(parse_qsl(event.postback.data)) #取得Postback資料
54     if backdata.get('action') == 'buy':
55         sendBack_buy(event, backdata)
56     elif backdata.get('action') == 'sell':
57         sendBack_sell(event, backdata)
```



```
59 def sendButton(event): #按鈕樣版
60     try:
61         message = TemplateSendMessage(
62             alt_text='按鈕樣板',
63             template=ButtonsTemplate(
64                 thumbnail_image_url='https://i.imgur.com/4QfKuz1.png', #顯示的圖片
65                 title='按鈕樣版示範', #主標題
66                 text='請選擇:', #副標題
67                 actions=[
68                     MessageTemplateAction( #顯示文字訊息
69                         label='文字訊息',
70                         text='@購買披薩'
71                     ),
72                     URITemplateAction( #開啟網頁
73                         label='連結網頁',
74                         uri='http://www.e-happy.com.tw'
75                     ),
76                     PostbackTemplateAction( #執行Postback功能,觸發Postback事件
77                         label='回傳訊息', #按鈕文字
78                         #text='@購買披薩', #顯示文字訊息
79                         data='action=buy' #Postback資料
80                     ),
81                 ]
82             )
83         )
84         line_bot_api.reply_message(event.reply_token, message)
85     except:
86         line_bot_api.reply_message(event.reply_token, TextSendMessage(text='發生錯誤!'))
```

```
88 def sendConfirm(event): #確認樣板
89     try:
90         message = TemplateSendMessage(
91             alt_text='確認樣板',
92             template=ConfirmTemplate(
93                 text='你確定要購買這項商品嗎?',
94                 actions=[
95                     MessageTemplateAction( #按鈕選項
96                         label='是',
97                         text='@yes'
98                     ),
99                     MessageTemplateAction(
100                         label='否',
101                         text='@no'
102                     )
103                 ]
104             )
105         )
106         line_bot_api.reply_message(event.reply_token, message)
107     except:
108         line_bot_api.reply_message(event.reply_token, TextSendMessage(text='發生錯誤!'))
```

```
110 def sendCarousel(event): #轉盤樣板
111     try:
112         message = TemplateSendMessage(
113             alt_text='轉盤樣板',
114             template=CarouselTemplate(
115                 columns=[
116                     CarouselColumn(
117                         thumbnail_image_url='https://i.imgur.com/4QfKuz1.png',
118                         title='這是樣板一',
119                         text='第一個轉盤樣板',
120                         actions=[
121                             MessageTemplateAction(
122                                 label='文字訊息一',
123                                 text='賣披薩'
124                             ),
125                             URITemplateAction(
126                                 label='連結文淵閣網頁',
127                                 uri='http://www.e-happy.com.tw'
128                             ),
129                             PostbackTemplateAction(
130                                 label='回傳訊息一',
131                                 data='action=sell&item=披薩'
132                             ),
133                         ]
134                     ),
```

```
135     CarouselColumn(  
136         thumbnail_image_url='https://i.imgur.com/qaAdBkR.png',  
137         title='這是樣板二',  
138         text='第二個轉盤樣板',  
139         actions=[  
140             MessageTemplateAction(  
141                 label='文字訊息二',  
142                 text='賣飲料'  
143             ),  
144             URITemplateAction(  
145                 label='連結台大網頁',  
146                 uri='http://www.ntu.edu.tw'  
147             ),  
148             PostbackTemplateAction(  
149                 label='回傳訊息二',  
150                 data='action=sell&item=飲料'  
151             ),  
152         ]  
153     )  
154 ]  
155 )  
156 )  
157 line_bot_api.reply_message(event.reply_token,message)  
158 except:  
159     line_bot_api.reply_message(event.reply_token,TextSendMessage(text='發生錯誤!'))
```

7

```
161 def sendImgCarousel(event): #圖片轉盤
162     try:
163         message = TemplateSendMessage(
164             alt_text='圖片轉盤樣板',
165             template=ImageCarouselTemplate(
166                 columns=[
167                     ImageCarouselColumn(
168                         image_url='https://i.imgur.com/4QfKuz1.png',
169                         action=MessageTemplateAction(
170                             label='文字訊息',
171                             text='賣披薩'
172                         )
173                     ),
174                     ImageCarouselColumn(
175                         image_url='https://i.imgur.com/qaAdBkR.png',
176                         action=PostbackTemplateAction(
177                             label='回傳訊息',
178                             data='action=sell&item=飲料'
179                         )
180                     )
181                 ]
182             )
183         )
184         line_bot_api.reply_message(event.reply_token,message)
185     except:
186         line_bot_api.reply_message(event.reply_token,TextSendMessage(text='發生錯誤!'))
```

8

```
188 def sendPizza(event):
189     try:
190         message = TextSendMessage(
191             text = '感謝您購買披薩，我們將盡快為您製作。'
192         )
193         line_bot_api.reply_message(event.reply_token, message)
194     except:
195         line_bot_api.reply_message(event.reply_token, TextSendMessage(text='發生錯誤！'))
196
197 def sendYes(event):
198     try:
199         message = TextSendMessage(
200             text='感謝您的購買，\n我們將盡快寄出商品。',
201         )
202         line_bot_api.reply_message(event.reply_token, message)
203     except:
204         line_bot_api.reply_message(event.reply_token, TextSendMessage(text='發生錯誤！'))
```

9

```
205
206 def sendBack_buy(event, backdata): #處理Postback
207     try:
208         text1 = '感謝您購買披薩，我們將盡快為您製作。\\n(action 的值為 ' + backdata.get('action') + ')'
209         text1 += '\\n(可將處理程式寫在此處。)'
210         message = TextSendMessage( #傳送文字
211             text = text1
212         )
213         line_bot_api.reply_message(event.reply_token, message)
214     except:
215         line_bot_api.reply_message(event.reply_token, TextSendMessage(text='發生錯誤!'))
216
217 def sendBack_sell(event, backdata): #處理Postback
218     try:
219         message = TextSendMessage( #傳送文字
220             text = '點選的是賣 ' + backdata.get('item')
221         )
222         line_bot_api.reply_message(event.reply_token, message)
223     except:
224         line_bot_api.reply_message(event.reply_token, TextSendMessage(text='發生錯誤!'))
225
226 if __name__ == '__main__':
227     app.run()
```

■ D:\高科大_1111課程\111-1_程式設計(三)\Python與LINE Bot機器人全面實戰特訓班--Flask最強應用學習資源\本書範例\ch06\ngrok

ngrok by @inconshreveable

```
Session Status      online
Session Expires    1 hour, 44 minutes
Update              update available (version 2.3.40, Ctrl-U to update)
Version             2.3.18
Region              United States (us)
Web Interface        http://127.0.0.1:4040
Forwarding           http://a105-140-133-74-88.ngrok.io -> http://localhost:80
Forwarding           https://a105-140-133-74-88.ngrok.io -> http://localhost:80
```

Connections	ttl	opn	rt1	rt5	p50	p90
	5	0	0.00	0.00	130.06	180.26

HTTP Requests

```
-----
POST /callback      404 Not Found
POST /               405 Method Not Allowed
POST /               405 Method Not Allowed
POST /               405 Method Not Allowed
POST /               405 Method Not Allowed
POST /callback      404 Not Found
POST /callback      404 Not Found
POST /callback      404 Not Found
POST /callback      404 Not Found
POST /callback      404 Not Found
```


<linebotFunc3.py>

檢查圖文選單傳送
文字的程式碼為：

程式碼：linebotFunc3.py

```
.....
8 from urllib.parse import parse_qs
.....
23 @handler.add(MessageEvent, message=TextMessage)
24 def handle_message(event):
25     mtext = event.message.text
26     if mtext == '@ 按鈕樣板 ':
27         sendButton(event)
28
29     elif mtext == '@ 確認樣板 ':
30         sendConfirm(event)
31
32     elif mtext == '@ 轉盤樣板 ':
33         sendCarousel(event)
34
35     elif mtext == '@ 圖片轉盤 ':
36         sendImgCarousel(event)
37
38     elif mtext == '@ 購買披薩 ':
39         sendPizza(event)
40
41     elif mtext == '@yes':
42         sendYes(event)
43
44 @handler.add(PostbackEvent) #PostbackTemplateAction 觸發此事件
45 def handle_postback(event):
46     backdata = dict(parse_qs(event.postback.data)) #取得 Postback 資料
47     if backdata.get('action') == 'buy':
48         sendBack_buy(event, backdata)
49     elif backdata.get('action') == 'sell':
50         sendBack_sell(event, backdata)
.....
```

關於按鈕樣板的程式：

程式碼：linebotFunc3.py (續)

```
.....
52 def sendButton(event): # 按鈕樣版
53     try:
54         message = TemplateSendMessage(
55             alt_text=' 按鈕樣板 ',
56             template=ButtonsTemplate(
57                 thumbnail_image_url='https://i.imgur.com/
58                     4QfKuz1.png', # 顯示的圖片
59                 title=' 按鈕樣版示範 ', # 主標題
60                 text=' 請選擇： ', # 副標題
61                 actions=[
62                     MessageTemplateAction( # 顯示文字訊息
63                         label=' 文字訊息 ',
64                         text='@ 購買披薩 '
65                     ),
66                     URITemplateAction( # 開啟網頁
67                         label=' 連結網頁 ',
68                         uri='http://www.e-happy.com.tw'
69                     ),
70                     PostbackTemplateAction( # 執行 Postback 功能 ,
71                         觸發 Postback 事件
72                         label=' 回傳訊息 ', # 按鈕文字
73                         #text='@ 購買披薩 ', # 顯示文字訊息
74                         data='action=buy' #Postback 資料
75                     ),
76                 ]
77             )
78         line_bot_api.reply_message(event.reply_token, message)
```

```
78     except:
79         line_bot_api.reply_message(event.reply_token,
            TextSendMessage(text=' 發生錯誤！ '))
.....
181 def sendPizza(event):
182     try:
183         message = TextSendMessage(
184             text = ' 感謝您購買披薩，我們將盡快為您製作。'
185         )
186         line_bot_api.reply_message(event.reply_token, message)
187     except:
188         line_bot_api.reply_message(event.reply_token,
            TextSendMessage(text=' 發生錯誤！ '))
.....
199 def sendBack_buy(event, backdata):  # 處理 Postback
200     try:
201         text1 = ' 感謝您購買披薩，我們將盡快為您製作。\\n
            (action 的值為 ' + backdata.get('action') + ' )'
202         text1 += '\\n( 可將處理程式寫在此處。 )'
203         message = TextSendMessage(  # 傳送文字
204             text = text1
205         )
206         line_bot_api.reply_message(event.reply_token, message)
207     except:
208         line_bot_api.reply_message(event.reply_token,
            TextSendMessage(text=' 發生錯誤！ '))
.....
```

使用者按圖文選單中按鈕樣板的執行結果：



確認樣板的基本語法為：

▲ 點選按鈕樣板

▲ 點選連結網頁按鈕

```
訊息變數 = TemplateSendMessage(  
    alt_text= 取代文字 ,  
    template=ConfirmTemplate(  
        text= 提示文字  
        actions=[  
            按鈕元件一 ,  
            按鈕元件二 ,  
            .....  
        ]  
    )  
)
```

關於確認樣板的 程式碼為：

程式碼：linebotFunc3.py (續)

```
.....
29     elif mtext == '@ 確認樣板 ':
30         sendConfirm(event)
31     .....
41     elif mtext == '@yes':
42         sendYes(event)
43     .....
81 def sendConfirm(event): # 確認樣板
82     try:
83         message = TemplateSendMessage(
84             alt_text=' 確認樣板 ',
85             template=ConfirmTemplate(
86                 text=' 你確定要購買這項商品嗎？ ',
87                 actions=[
88                     MessageTemplateAction( # 按鈕選項
89                         label=' 是 ',
90                         text='@yes'
91                     ),
92                     MessageTemplateAction(
93                         label=' 否 ',
94                         text='@no'
95                     )
96                 ]
97             )
98         )
99         line_bot_api.reply_message(event.reply_token, message)
```

```
100     except:
101         line_bot_api.reply_message(event.reply_token,
                                     TextSendMessage(text=' 發生錯誤！ '))
.....
190 def sendYes(event):
191     try:
192         message = TextSendMessage(
193             text=' 感謝您的購買，\n我們將盡快寄出商品。',
194         )
195         line_bot_api.reply_message(event.reply_token, message)
196     except:
197         line_bot_api.reply_message(event.reply_token,
                                     TextSendMessage(text=' 發生錯誤！ '))
.....
```

使用者按圖文選單中確認樣板的執行結果：



6.1.3 轉盤樣板 (Carousel Template)

轉盤樣板的基本語法為：

```
訊息變數 = TemplateSendMessage(  
    alt_text= 取代文字 ,  
    template=CarouselTemplate(  
        columns=[  
            CarouselColumn(  
                按鈕樣板一  
            ),  
            CarouselColumn(  
                按鈕樣板二  
            ),  
            .....  
        ]  
    )  
)
```

關於轉盤樣板的程式碼為：

程式碼：linebotFunc3.py (續)

```
.....  
32     elif mtext == '@ 轉盤樣板':  
33         sendCarousel(event)  
.....  
44 @handler.add(PostbackEvent) #PostbackTemplateAction 觸發此事件  
45 def handle_postback(event):  
46     backdata = dict(parse_qsl(event.postback.data)) #取得 Postback 資料  
.....  
49     elif backdata.get('action') == 'sell':  
50         sendBack_sell(event, backdata)  
.....
```

```
103 def sendCarousel(event): # 轉盤樣板
104     try:
105         message = TemplateSendMessage(
106             alt_text=' 轉盤樣板 ',
107             template=CarouselTemplate(
108                 columns=[
109                     CarouselColumn(
110                         thumbnail_image_url='https://
111                             i.imgur.com/4QfKuz1.png',
112                         title=' 這是樣板一 ',
113                         text=' 第一個轉盤樣板 ',
114                         actions=[
115                             MessageTemplateAction(
116                                 label=' 文字訊息一 ',
117                                 text=' 賣披薩 '
118                             ),
119                             URITemplateAction(
120                                 label=' 連結文淵閣網頁 ',
121                                 uri='http://www.e-happy.com.tw'
122                             ),
123                             PostbackTemplateAction(
124                                 label=' 回傳訊息一 ',
125                                 data='action=sell&item= 披薩 '
126                             )
127                         ]
128                     )
129                 ]
130             )
131     except Exception as e:
132         print(e)
```



```
.....  
210 def sendBack_sell(event, backdata): # 處理 Postback  
211     try:  
212         message = TextSendMessage( # 傳送文字  
213             text = '點選的是賣 ' + backdata.get('item')  
214         )  
215         line_bot_api.reply_message(event.reply_token, message)  
216     except:  
217         line_bot_api.reply_message(event.reply_token,  
            TextSendMessage(text='發生錯誤！'))  
.....
```

使用者按圖文選單中轉盤樣板的執行結果：



6.1.4 圖片轉盤樣板(ImageCarousel Template)

圖片轉盤樣板的基本語法為：

```
訊息變數 = TemplateSendMessage(  
    alt_text= 取代文字 ,  
    template=ImageCarouselTemplate(  
        columns=[  
            ImageCarouselColumn(  
                thumbnail_image_url= 圖片網址一  
                action= 按鈕元件一  
            ),  
            ImageCarouselColumn(  
                thumbnail_image_url= 圖片網址二  
                action= 按鈕元件二  
            ),  
            .....  
        ]  
    )  
)
```

使用者按圖文選單中轉盤樣板的執行結果：



6.2 圖片地圖及日期時間

6.2.1 圖片地圖

建立圖片地圖的語法為：

```
訊息變數 = ImagemapSendMessage(  
    base_url= 圖片網址 ,  
    alt_text= 替代文字 ,  
    base_size=BaseSize(height= 圖片高度 , width= 圖片寬度 ) ,  
    actions=[  
        圖片地圖按鈕一 ,  
        圖片地圖按鈕二 ,  
        .....  
    ]  
)
```

文字圖片地圖按鈕(MessageImagemapAction)

使用者點選文字圖片地圖按鈕會回傳文字訊息，語法為：

```
MessageImagemapAction(  
    text= 回傳文字訊息內容 ,  
    area=ImagemapArea(  
        x= 矩形左上角 X 坐標 ,  
        y= 矩形左上角 Y 坐標 ,  
        width= 矩形寬度 ,  
        height= 矩形高度  
    )  
) ,
```

連結圖片地圖按鈕(UriImagemapAction)

使用者點選連結圖片地圖按鈕就會開啟指定的網頁，語法為：

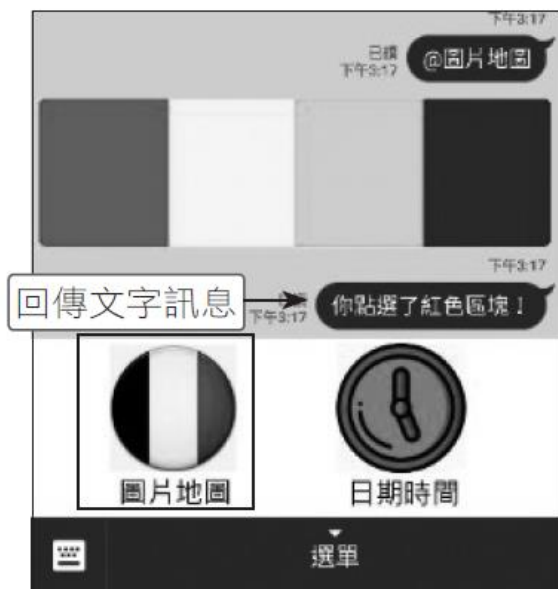
```
UriImagemapAction( # 開啟網頁  
    link_uri= 要開啟網頁的網址 ,  
    area=ImagemapArea(  
        x= 矩形左上角 X 坐標 ,  
        y= 矩形左上角 Y 坐標 ,  
        width= 矩形寬度 ,  
        height= 矩形高度  
    )  
) ,
```

程式碼：linebotFunc4.py

```
width=imgwidth), # 圖片寬及高
```

```
48         actions=[
49             MessageImagemapAction( # 顯示文字訊息
50                 text='你點選了紅色區塊!',
51                 area=ImagemapArea( # 設定圖片範圍：左方 1/4 區域
52                     x=0,
53                     y=0,
54                     width=imgwidth*0.25,
55                     height=imgheight
56                 )
57             ),
58             URIImagemapAction( # 開啟網頁
59                 link_uri='http://www.e-happy.com.tw',
60                 area=ImagemapArea( # 右方 1/4 區域 (藍色 1)
61                     x=imgwidth*0.75,
62                     y=0,
63                     width=imgwidth*0.25,
64                     height=imgheight
65                 )
66             ),
67         ]
68     )
69     line_bot_api.reply_message(event.reply_token, message)
70 except:
71     line_bot_api.reply_message(event.reply_token,
72                               TextSendMessage(text='發生錯誤!'))
73
74 .....
```

使用者按圖文選單中圖片地圖的執行結果：點選紅色區塊會回傳「你點選了紅色區塊！」，點選藍色區塊會開啟指定網頁，點選黃色或綠色區塊將沒有任何反應。



▲ 點選紅色區塊



▲ 點選藍色區塊

6.2.2 日期時間

建立日期時間按鈕的語法為：

```
DatetimePickerTemplateAction(  
    label= 按鈕文字 ,  
    data=Postback 資料 ,  
    mode= 模式  
    initial= 初始值  
    min= 最小值  
    max= 最大值  
)
```

本範例是以按鈕樣板示範三種日期時間模式的使用方法，關於日期時間的程式碼為：

程式碼：linebotFunc4.py (續)

```
.....  
9 import datetime  
.....  
30     elif mtext == '@日期時間':  
31         sendDatetime(event)  
32  
33 @handler.add(PostbackEvent)    #PostbackTemplateAction 觸發此事件
```

```
34 def handle_postback(event):
35     backdata = dict(parse_qs(event.postback.data)) # 取得 data 資料
36     if backdata.get('action') == 'sell':
37         sendData_sell(event, backdata)
38     .....
43 73 def sendDatetime(event): # 日期時間
74     try:
75         message = TemplateSendMessage(
76             alt_text='日期時間範例',
77             template=ButtonsTemplate(
78                 thumbnail_image_url='https://i.imgur.com/VxVB46z.jpg',
79                 title='日期時間示範',
80                 text='請選擇:',
81                 actions=[
82                     DatetimePickerTemplateAction(
83                         label="選取日期",
84                         data="action=sell&mode=date", # 觸發 postback 事件
85                         mode="date", # 選取日期
86                         initial="2020-10-01", # 顯示初始日期
87                         min="2020-10-01", # 最小日期
88                         max="2021-12-31" # 最大日期
89                     ),
```

```
113 def sendData_sell(event, backdata): #Postback, 顯示日期時間
114     try:
115         if backdata.get('mode') == 'date':
116             dt = '日期為：' +
117                 event.postback.params.get('date') # 讀取日期
118         elif backdata.get('mode') == 'time':
119             dt = '時間為：' + event.postback.params.get('time') # 讀取時間
120         elif backdata.get('mode') == 'datetime':
121             dt = datetime.datetime.strptime(event.postback.
122                 params.get('datetime'), '%Y-%m-%dT%H:%M')
123             # 讀取日期時間
124             dt = dt.strftime('{d}%Y-%m-%d, {t}%H:%M').
125                 format(d='日期為：', t='時間為：') # 轉為字串
126         message = TextSendMessage(
127             text=dt
128         )
129         line_bot_api.reply_message(event.reply_token,message)
```

使用者按圖文選單中日期時間的執行結果：點選 **選取日期** 會顯示日曆界面讓使用者選取日期，點選 **選取時間** 會顯示時鐘界面讓使用者選取時間，點選 **選取日期時間** 會先顯示日曆界面，再顯示時鐘界面。此處示範點選 **選取日期時間** 操作，下右圖為選取日期界面，選取完成後按 **下一步**。

