



# Discord Bot 開發



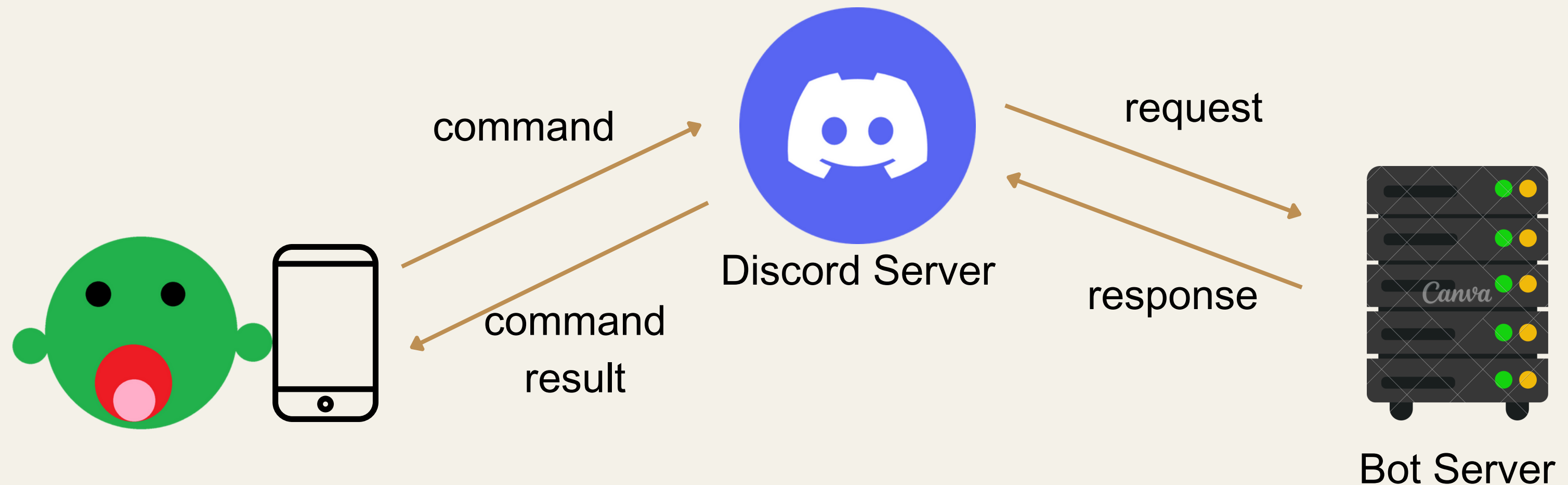


# 課程內容

- 建立 Discord bot 帳號
- 實作指令
- 實作Event
- 更多API查詢教學

# Discord Bot 是什麼？

其實概念很簡單，Discord 設計了 API 讓我們能夠透過程式自動處理一些事件或是用戶的回覆，Discord 的 Server 會和我們的 bot server 做連線，互相傳輸請求或回應



# 建立機器人帳號

---

首先我們需要建立機器人的帳號，取得 Token 後才能與 Discord API 連線

請先點選連結前往 DC 開發人員網站，並登入你的 Discord 帳號

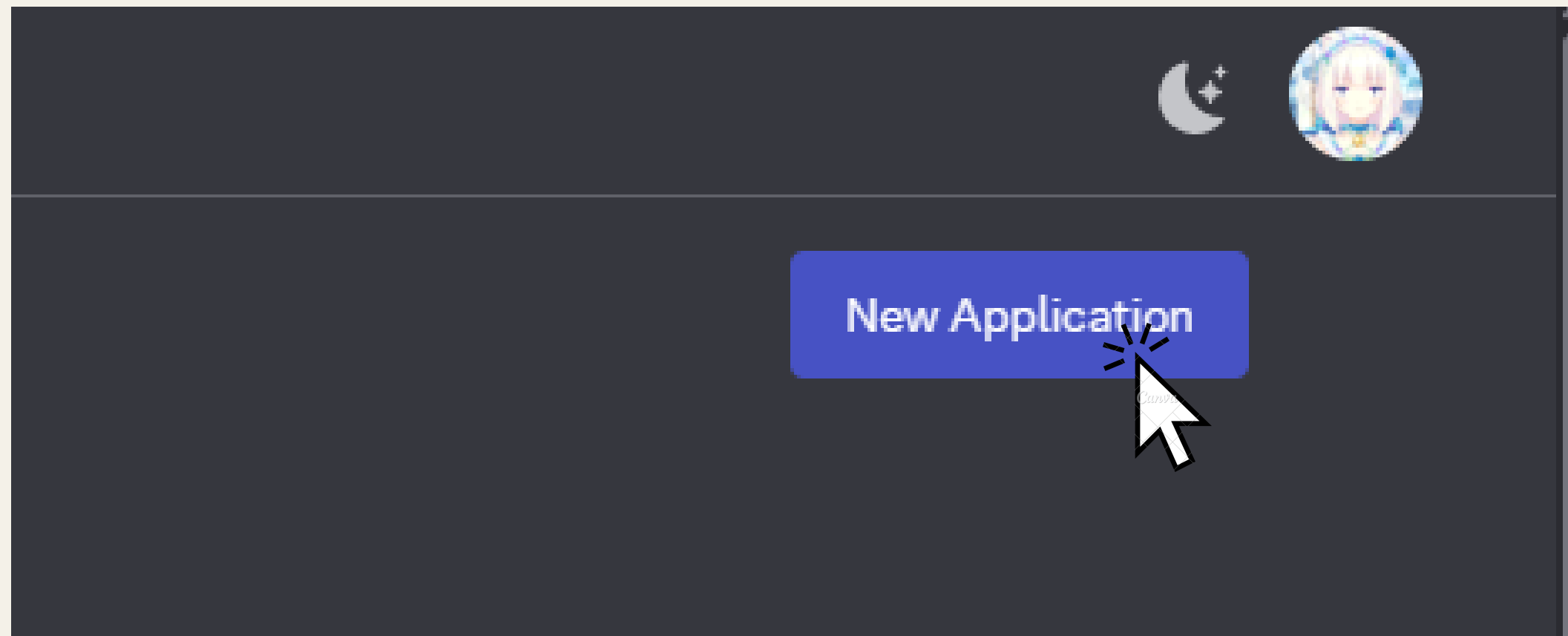
<https://discord.com/developers/applications>

如果你還沒有 Discord 帳號的話，請先去註冊一支帳號

# 建立機器人帳號

---

點這顆按鈕



# 建立機器人帳號

---

幫你的機器人取個名字

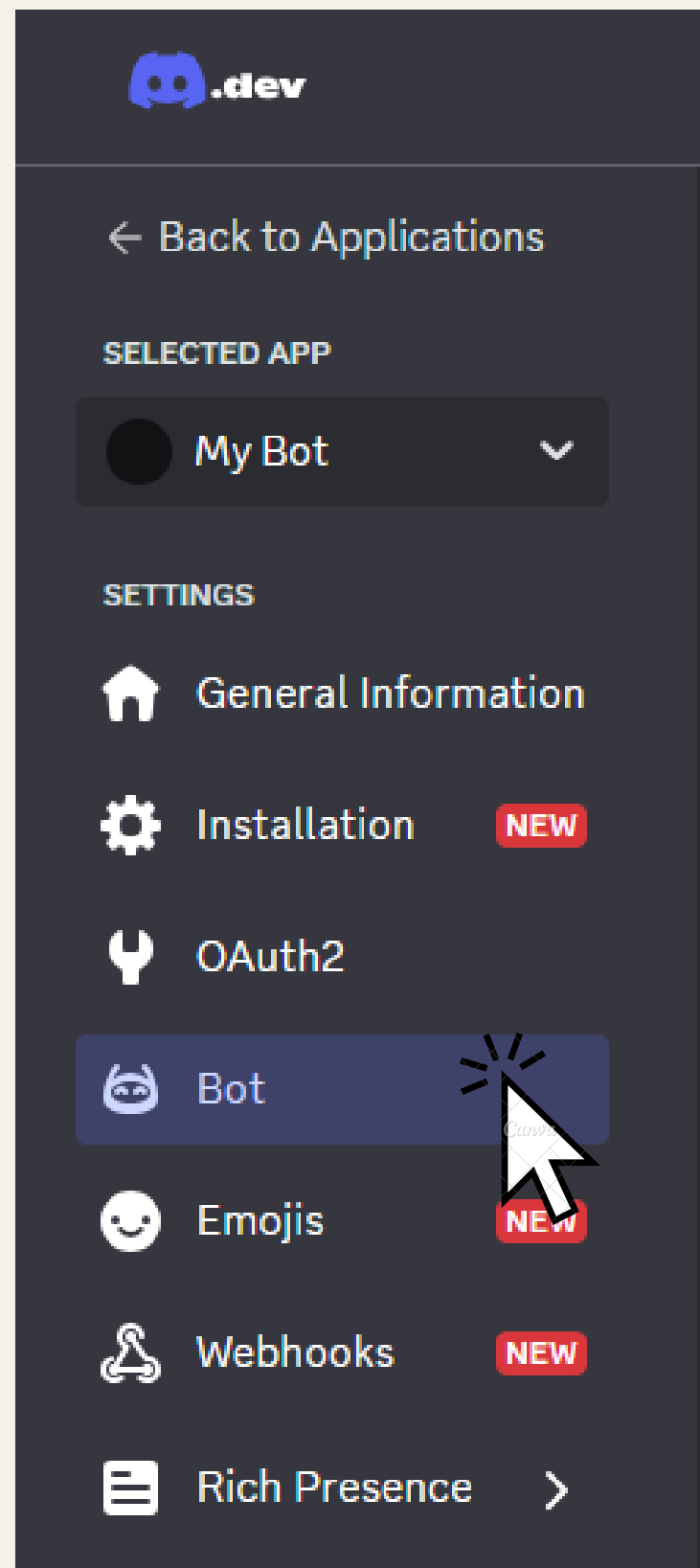
## CREATE AN APPLICATION

NAME \*

☒ By clicking Create, you agree to the [Discord Developer Terms of Service](#) and [Developer Policy](#).

CancelCreate

# 建立機器人帳號



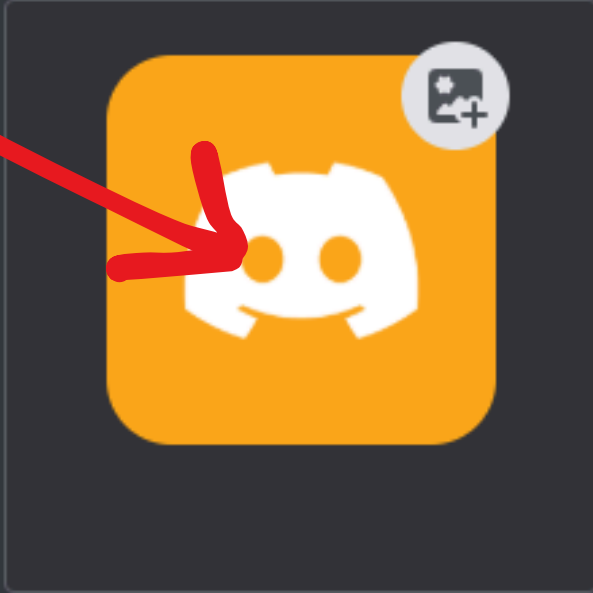
在左側找到 Bot

# 建立機器人帳號

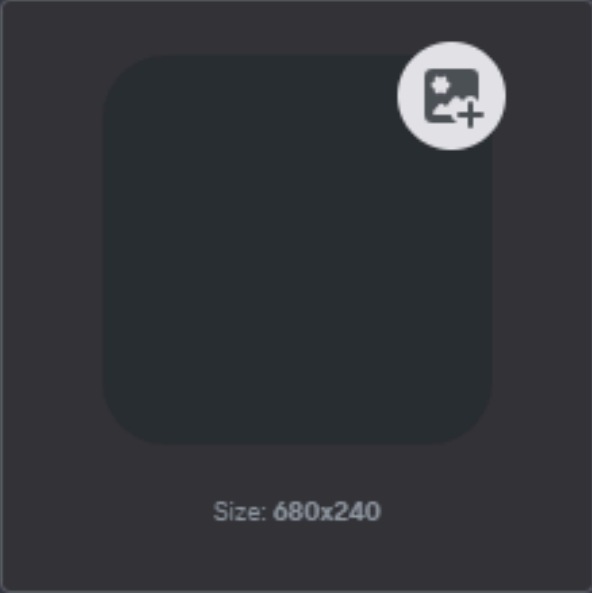
這邊可以設定機器人頭像

**Build-A-Bot**  
Bring your app to life by adding a bot user. This action is irreversible (because robots are too cool to be deleted).

**ICON**



**BANNER**



Size: 680x240

**USERNAME**

My Bot

**TOKEN**  
For security purposes, tokens can only be viewed once, when created. If you forgot or lost access to your token, please regenerate a new one.

Reset Token

機器人名字



# 建立機器人帳號

---

往下滑一點，把這三個選項打開

## Privileged Gateway Intents

Some [Gateway Intents](#) require approval if your bot is verified. If your bot is not verified, you can toggle those intents below to access them.

### PRESENCE INTENT

Required for your bot to receive [Presence Update](#) events.

**NOTE:** Once your bot reaches 100 or more servers, this will require verification and approval. [Read more here](#)



### SERVER MEMBERS INTENT

Required for your bot to receive events listed under [GUILD\\_MEMBERS](#).

**NOTE:** Once your bot reaches 100 or more servers, this will require verification and approval. [Read more here](#)



### MESSAGE CONTENT INTENT

Required for your bot to receive [message content](#) in most messages.

**NOTE:** Once your bot reaches 100 or more servers, this will require verification and approval. [Read more here](#)




# 建立機器人帳號

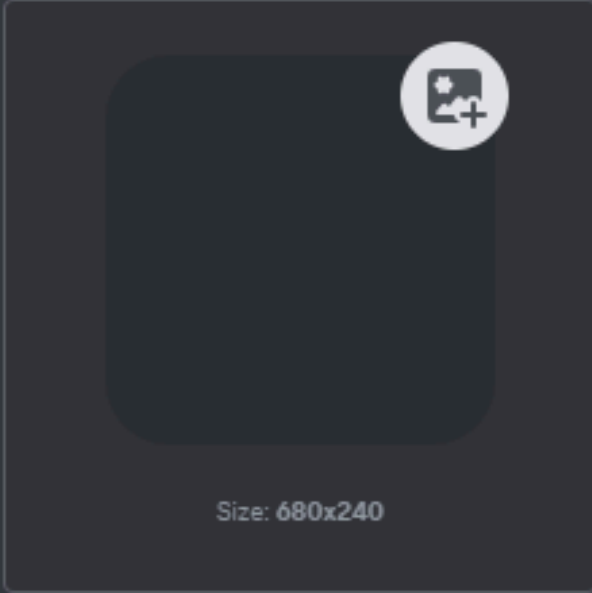
## Build-A-Bot

Bring your app to life by adding a bot user. This action is irreversible (because robots are too cool to die).

ICON



BANNER



USERNAME

TOKEN

For security purposes, tokens can only be viewed once, when created. If you forgot or lost access to your token, please regenerate a new one.

Reset Token

點選 Reset Token

# 建立機器人帳號

---

## TOKEN

For security purposes, tokens can only be viewed once, when created.

MTM4ODA3NjY2NjM4Mjc5Njk5OA.GEiDcw.uYTQU

Copy

Reset Token

點 Copy

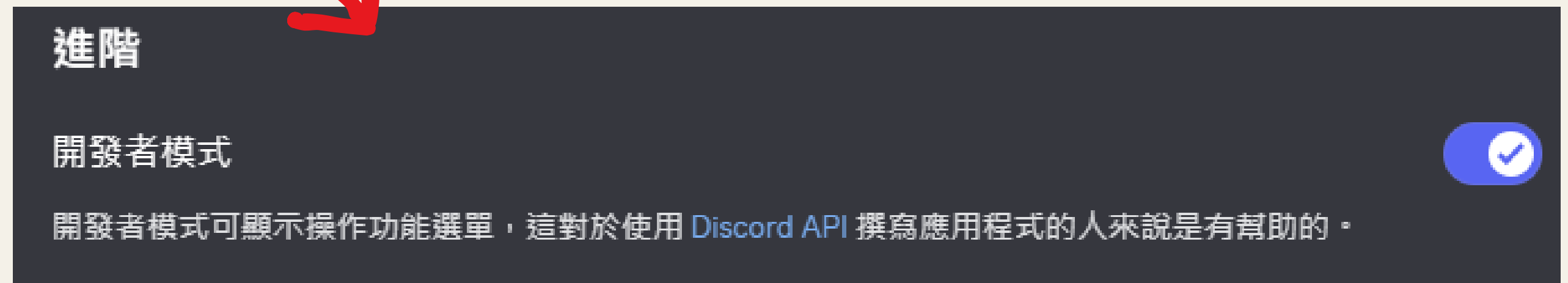
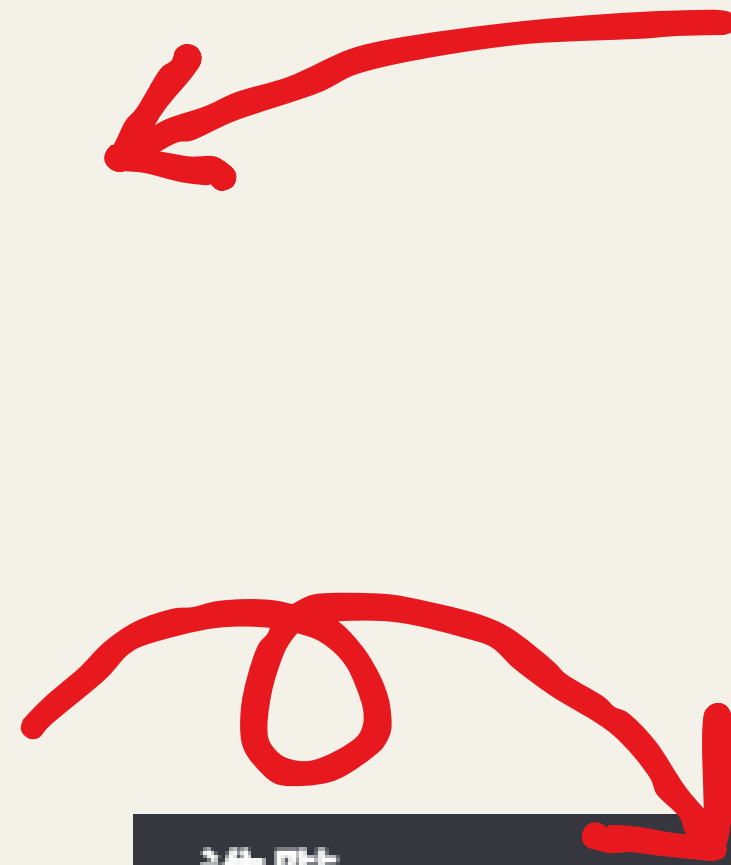
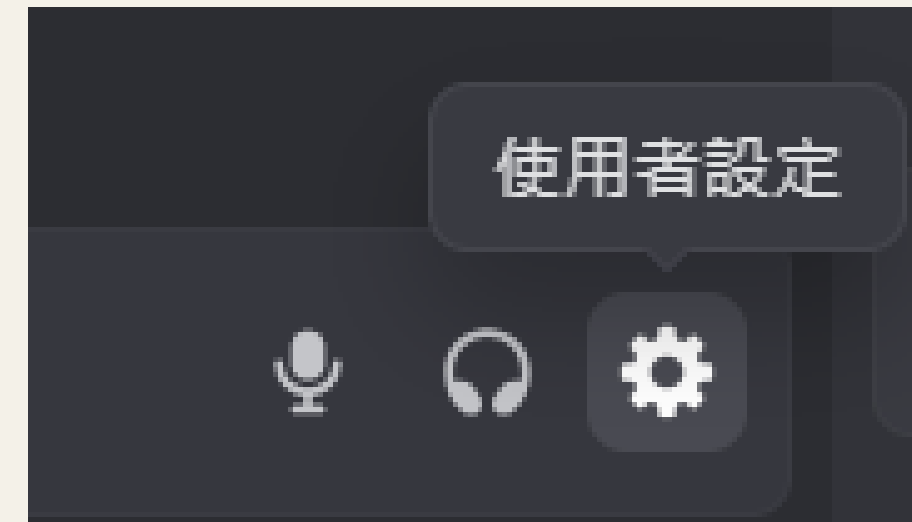
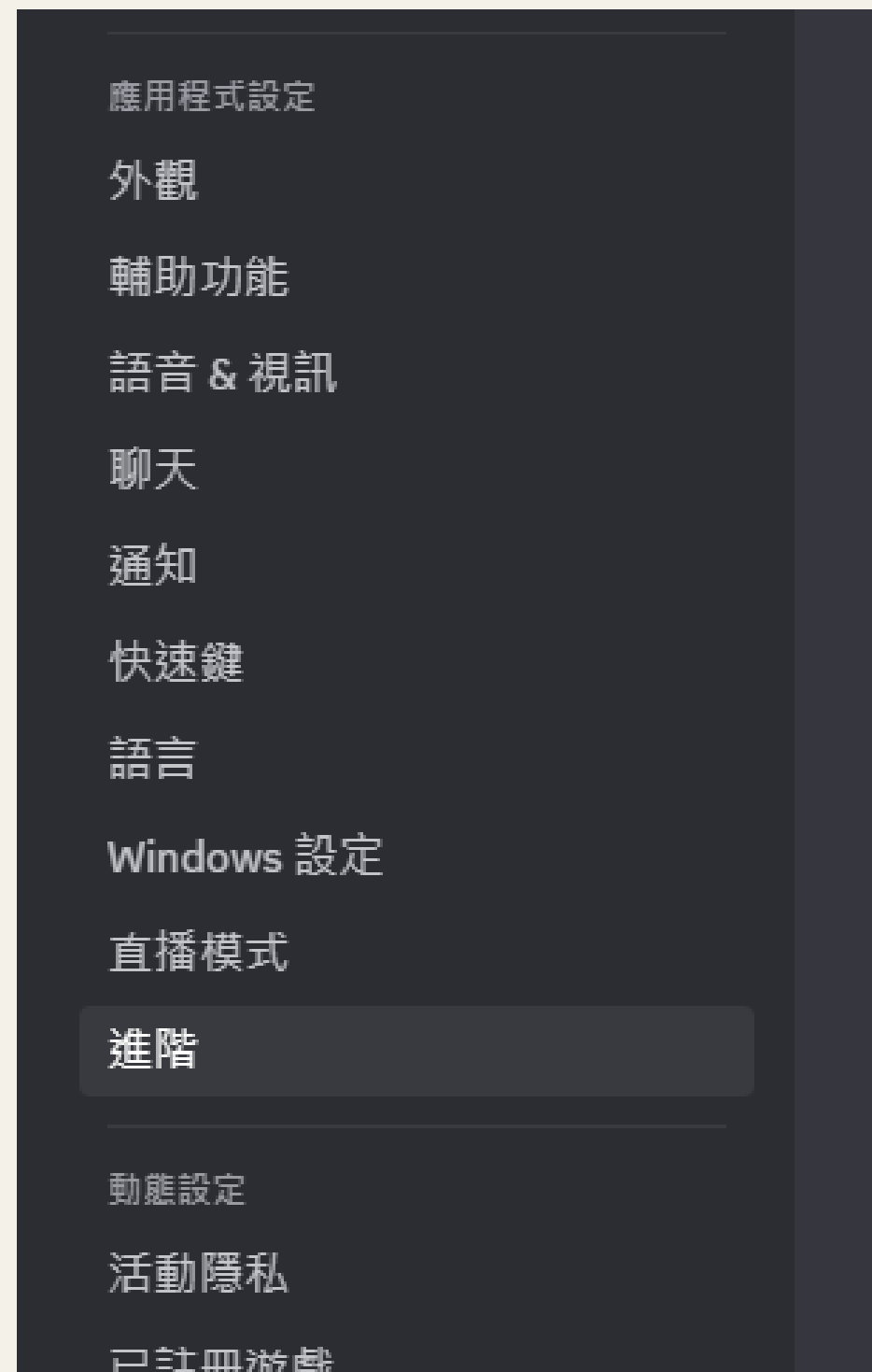
# 修改程式

---

將剛剛複製的 Token 貼到這邊

```
21  # 啟動機器人
22  async def main():
23      try:
24          await client.start('YOUR_BOT_TOKEN')
25      except Exception as e:
26          print(f'啟動失敗：{e}')
```

# 更改設定



# 複製伺服器ID

去你的伺服器  
對伺服器大標題右鍵->複製伺服器ID



# 修改程式

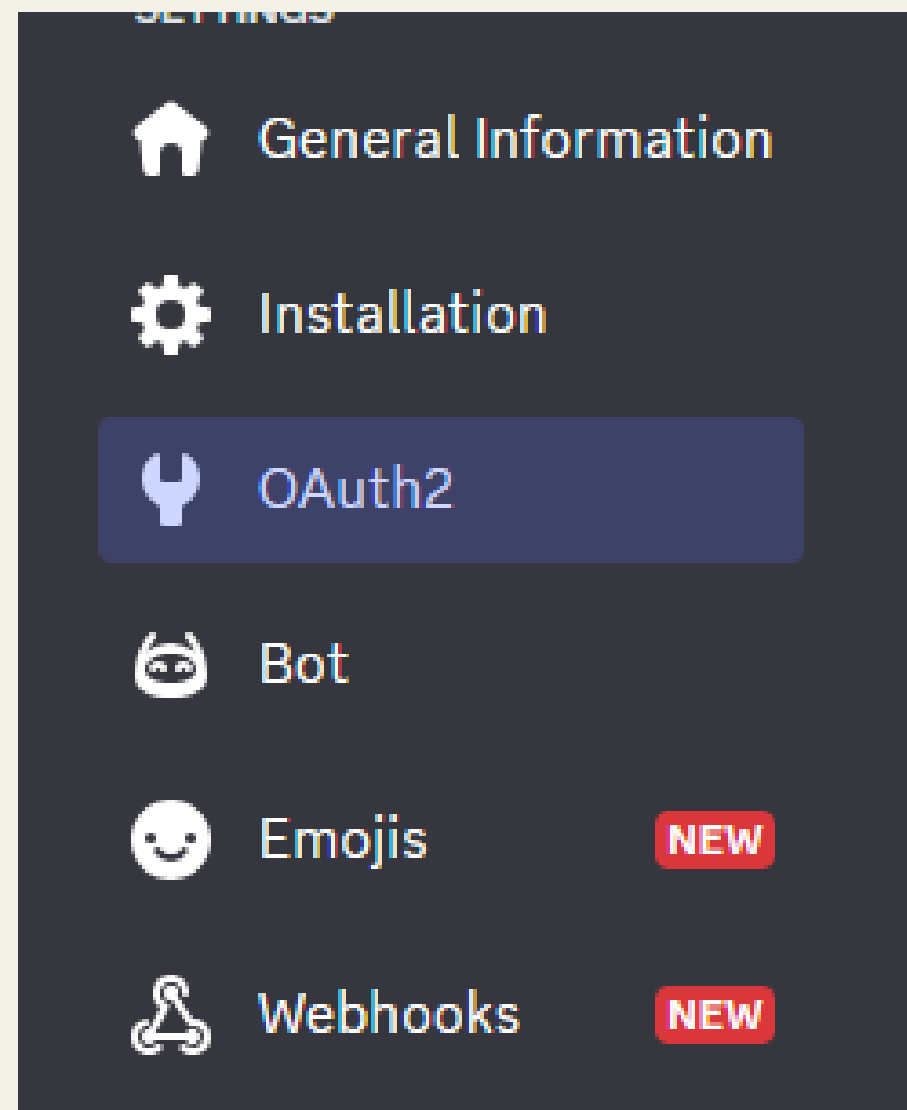
---

將剛剛複製的 ID 貼到這邊

```
9 | GUILD_ID = 000000000000000000000000
```

# 將機器人加入伺服器

---



在左側找到 OAuth2



# 將機器人加入伺服器

## OAuth2 URL Generator

Generate an invite link for your application by picking the scopes and permissions it needs to function. Then, share the URL to others!

### SCOPES

- |   |   |   |
|---|---|---|
| <input type="checkbox"/> identify             | <input type="checkbox"/> email                  | <input type="checkbox"/> connect          |
| <input type="checkbox"/> guilds               | <input type="checkbox"/> guilds.join            | <input type="checkbox"/> guilds.manage    |
| <input type="checkbox"/> guilds.channels.read | <input type="checkbox"/> gdm.join               | <input checked="" type="checkbox"/> bot   |
| <input type="checkbox"/> rpc                  | <input type="checkbox"/> rpc.notifications.read | <input type="checkbox"/> rpc.voice        |
| <input type="checkbox"/> rpc.voice.write      | <input type="checkbox"/> rpc.video.read         | <input type="checkbox"/> rpc.video.upload |

把 bot 打勾

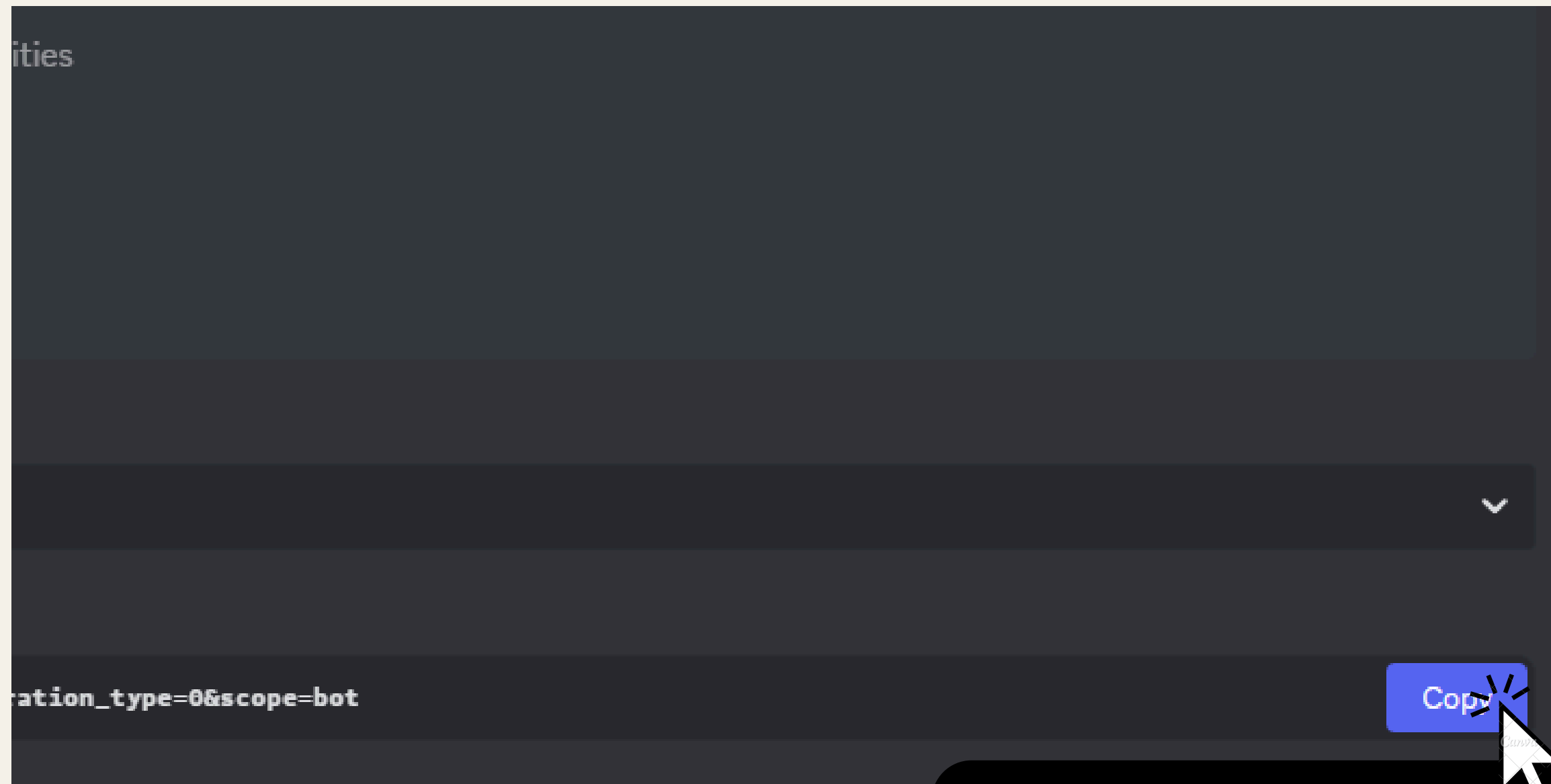
# 將機器人加入伺服器

---



# 將機器人加入伺服器

---



最下方右側有 Copy 按鈕

# 將機器人加入伺服器

---

接著把複製到的網址貼到瀏覽器，就可以把機器人加到你的伺服器了  
請注意你加入的伺服器是自己的，不要影響到別人喔

# 安裝 Python 虛擬環境

---

到 Terminal 輸入 `python -m venv .venv`

```
Bot>python -m venv .venv
```

等環境建立好後進入虛擬環境 `.venv\Scripts\activate`

```
ot>.venv\Scripts\activate
```

# 安裝 Python 套件

---

到 Terminal 輸入 `pip install discord.py`

```
ord Bot>pip install discord.py
```

# 啟動機器人

---

Terminal 輸入 main.py 執行後，你會發現你的機器人上線了！

```
Bot>main.py
```

線上 - 2



My Bot 應用



小湊sunny...  WYNN 

# 啟動機器人

---

試著輸入看看指令





# 啟動機器人

---

恭喜你成功啟動機器人！



# 實作指令

---

我們剛剛跑指令其實就是這一段程式定義好的指令

```
18 # 範例 Slash command: /hello
19 @tree.command(name="hello", description="回應一個問候訊息")
20 @app_commands.guilds(discord.Object(id = GUILD_ID))
21 ✓ async def hello(interaction: discord.Interaction):
22     |     await interaction.response.send_message(f'你好，{interaction.user.mention}!')
```

- name="指令名稱"
- description="指令描述"
- async def {指令函式名稱}(.....):
- 定義指令執行動作的python code區域

# 實作指令

---

指令也是可以定義輸入參數的

hint: 在參數後加上 “:type” 可以限制使用者輸入時的資料型態

```
24 @tree.command(name="add_number", description="輸出 a 和 b 相加的結果")
25 @app_commands.guilds(discord.Object(id = GUILD_ID))
26 async def add_number(interaction: discord.Interaction, a: int, b: int):
27     await interaction.response.send_message(f'{a} + {b} = {a+b}')
```

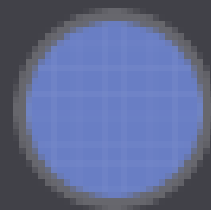
注意：更改指令或程式後都需要重新啟動機器人

# 實作指令

---

機器人重啟後應該可以看到新的指令  
如果沒有，請重開 discord

指令符合 /ADD



/add\_number **a** **b**

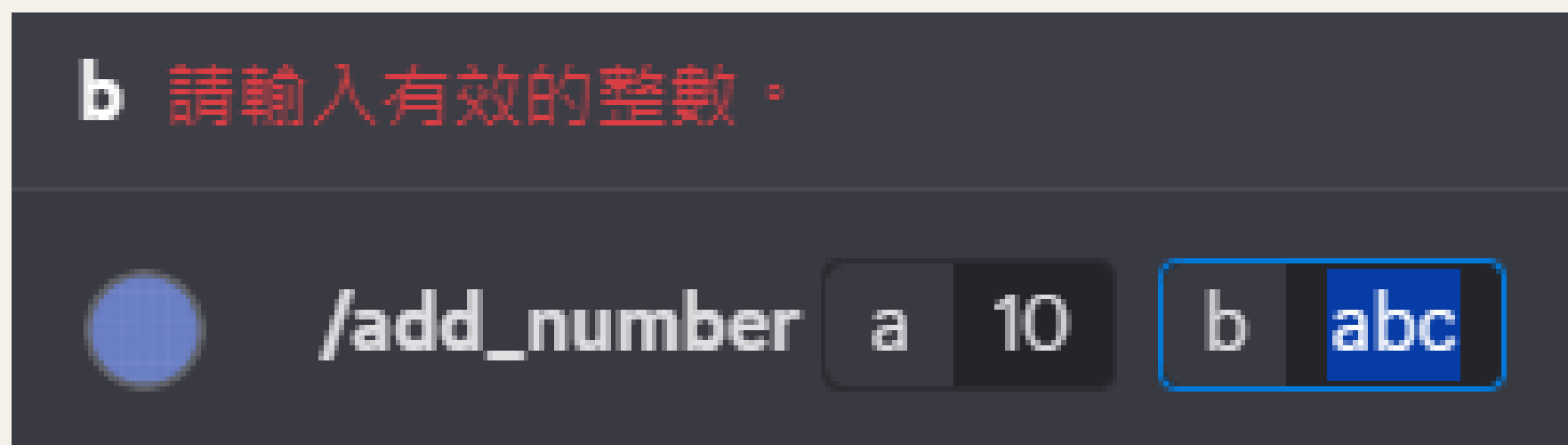
輸出 a 和 b 相加的結果



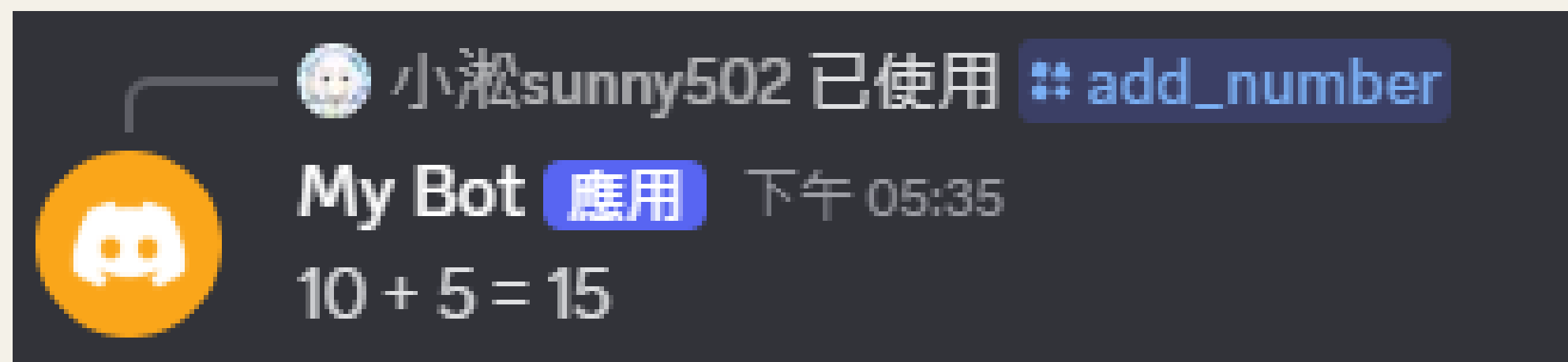
/add|

# 實作指令

可以看到我不能輸入 int 以外的資料型態



正確執行後的結果



# 實作指令

---

現在換你們練習實作指令囉～  
幫我做一個輸出 O 三角形的指令，輸入 n，輸出 n 層正直角三角形

## 範例結果



# 實作事件

---

除了使用者主動執行指令外，我們也可以設定事件  
事件就是在特定情況發生時，bot 會被动觸發到對應事件，並執行程式

以下為 on\_message Event 的範例，當有人發送訊息時會被觸發  
注意：加上這行 message.author.bot 判斷是為了避免機器人自己觸發到這個 Event

```
33 @client.event
34 async def on_message(message: discord.Message):
35     if message.author.bot == False:
36         await message.channel.send(f"{message.author.mention} send the message")
```

# 實作事件

---

## 範例結果



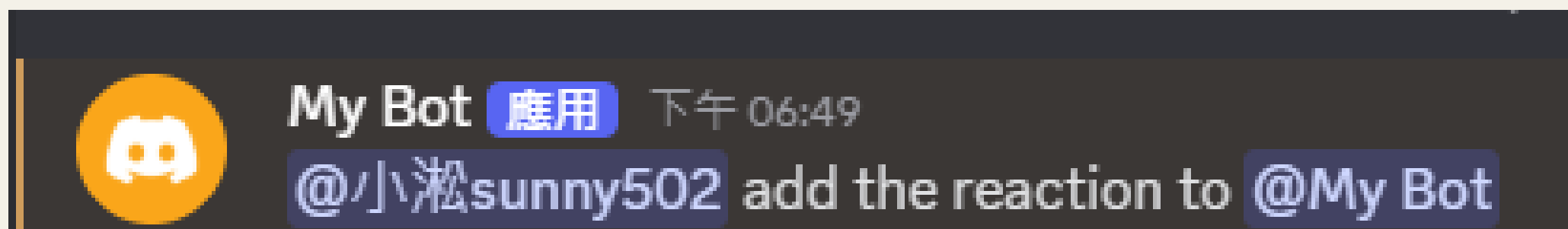


# 實作事件

---

現在要請你們練習實作事件  
當訊息被新增反應時，在特定頻道輸出紀錄

## 範例結果



Hint:

- 可以利用 `on_raw_reaction_add` 來偵測此事件 [點我查看API說明](#)

# 常見的功能

---

- `client.get_user(user_id)` → 取得使用者物件
- `client.get_guild(guild_id)` → 取得 guild 的物件
- `client.get_channel(channel_id)` → 取得頻道物件
- `msg_obj.author` → 取得傳送此訊息的使用者物件
- `user.mention` → 取得 `@user` 的字串格式
- `await channel_object.send(message)` → 對指定頻道傳送訊息
- `await user_obj.send(message)` → 對指定使用者私訊訊息
- `await interaction.response.send_message(message)` → 對指令回應傳送訊息

# API 查詢

---

<https://discordpy.readthedocs.io/en/stable/index.html#manuals>



感謝聆聽