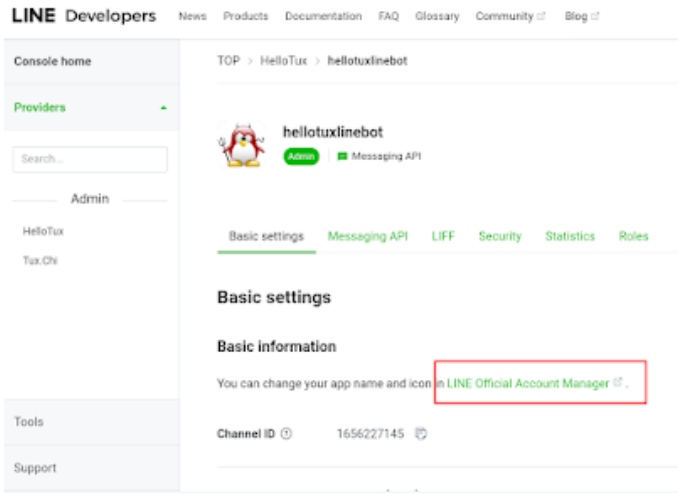
# 使用 Line Bot 圖文選單

**學習目標：**

- 利用 Line Bot 圖文選單，進行互動！

**使用 Line Developer 工具建立圖文選單**

1. 登入 Line Developer，選用 Line Official Account Manager ：




2. 在主頁選項，選擇「聊天室相關」的圖文選單：


3. 閱讀注意事項：


4. 建立圖文選單：

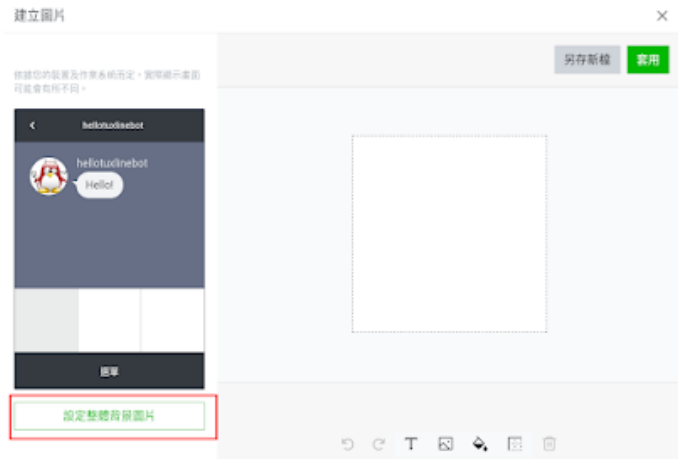

5. 輸入需要的項目內容：

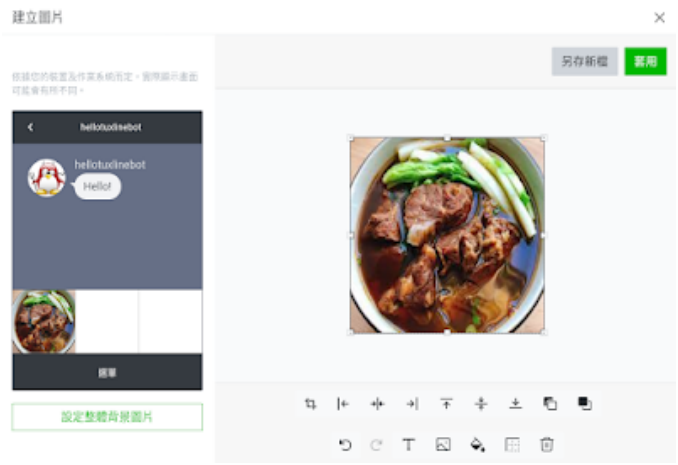

6. 在內容設定中，選擇版型：


7. 選擇需要的版型，按下「確定」！


8. 按下「建立圖片」：


9. 為每個對達框選擇圖片檔案：






PS: 可按「Ctrl」+ 三個圖片框，同時上傳三個圖檔，進行設定！

10. 指定「A」區塊類型為：文字，並輸入文字內容：


11. 完成後，可以按下「儲存」，完成編輯！


12. 打開 line 聊天室，可看到該圖文框！

**試寫自己的訂購單**

1. 在 app/router.py 中，加入下列內容：

```
@app.route('/orderMenu/<nodes>')
def orders(nodes):
    return render_template("orderMenu.html",nodeslist=nodes)
```

```
@app.route('/orderMenu',methods = ['GET','POST'])
def addorders():
    if request.method == 'POST':
      print(request.form)
      data = ""
      for key,value in request.form.items():
          data += value
          data += ','
      data = data[:-1]
      connectDB.addOrders(data)
      return render_template("showOrders.html")
    else:
      return render_template("showOrders.html")

# 顯示訂單列表
@app.route("/showOrders")
def showOrders():
    orderslists = connectDB.showOrders()
    return render_template('showOrders.html',menulist=orderslists)
```

2. 在 app/templates/orderMenu.html

```
{% extends "base.html" %}
{% block title %}My Web Site{% endblock %}
{% block main %}
    <div class="container my-5 py-5">
        <div class="row">
            <div class="rol">
                <h1>{{ nodeslist }}訂購單</h1>
                <form action="/orderMenu" method="post">
                    <label for="nos">{{ nodeslist }}數量</label>
                    <input type="text" id="nos" placeholder="請輸入數量" name="nos">
                    <br>
                    <label for="cusphone">手機號碼</label>
                    <input type="text" id="cusphone" placeholder="請輸入手機號碼" name="cusphone">
                    <br>
                    <input type="hidden" name="nodes" value="{{ nodeslist }}">
                    <input type="submit" value="送出">
                </form>
            </div>
        </div>

            </div>
{% endblock %}
```

3. 新增訂單資料表：

```
C:\workspace\LineBot> heroku login -i
C:\workspace\LineBot> python
>>> import os
>>> import psycopg2
>>> heroku_pgCLI = 'heroku config:get DATABASE_URL -a 你的APP名稱'
>>> DATABASE_URL = os.popen(heroku_pgCLI).read()[:-1]
>>> print(DATABASE_URL)
>>> connection_db=psycopg2.connect(DATABASE_URL, sslmode='require')
>>> cursor = connection_db.cursor()
>>> SQL_cmd = '''CREATE TABLE ordermenu(
... order_id serial PRIMARY KEY,
... nodes VARCHAR (50) NOT NULL,
... nos INT NOT NULL,
... cusphone VARCHAR (20) NOT NULL
... );'''
>>> cursor.execute(SQL_cmd)
>>> connection_db.commit()
```

```
>>> cursor.close()
>>> connection_db.close()
>>>
```

4. 新增訂單列表：

```
{% extends "base.html" %}
{% block title %}大學麵店訂單系統{% endblock %}

{% block main %}

<!-- 宣告巨集 -->
{% macro show_row(data, tag) -%}
{% for menu in data %}
<div class="row">
    {% if menu[4] == 'Action' %}
        {% for item in menu %}
                <div class="col">
                    <{{ tag }}>{{ item }}</{{ tag }}>
                </div>
        {% endfor %}
    {% else %}
        {% for item in menu %}
            <div class="col">
                <{{ tag }}>{{ item }}</{{ tag }}>
            </div>
        {% endfor %}
        <div class="col">
            <button type="submit" value="{{ menu[0],menu[1] }}" name="deldata">刪除</button>
        </div>
    {% endif %}
</div>
{% endfor %}
{%- endmacro %}
<!-- 宣告結束 -->

<div class="container my-5 py-5">
    <div class="row">
        <div class="rol">
            <h1>訂單列表</h1>
        </div>
    </div>
    <!-- 顯示菜單用 -->
    {% set cols = (("項次","菜單","數量","手機號碼","Action"),)%}

    {{ show_row(cols, "h2") }}
    <hr class="my-4">
    <form action="/delads" method="post">
        {{ show_row(menulist,"p") }}
    </form>
    <form action="/addOrders" method="get">
        <input type="submit" value="新增訂單">
    </form>
</div>

{% endblock %}
```

5. 修改 app/dataSQL/connectDB：

```
(前面略過....)
def addOrders(text):
    DATABASE_URL = os.environ['DATABASE_URL']
    connection_db = psycopg2.connect(DATABASE_URL,sslmode='require')
    cursor = connection_db.cursor()
    data = text.split(',')
```

```
        table_columns = '(nos,cusphone,nodes)'
        SQL_cmd = f"""INSERT INTO ordermenu { table_columns } VALUES(%s,%s,%s);"""
        cursor.execute(SQL_cmd,(int(data[0]),str(data[1]),str(data[2])))
        connection_db.commit()
        cursor.close()
        connection_db.close()
        return text

    def showOrders():
        DATABASE_URL = os.environ['DATABASE_URL']
        connection_db = psycopg2.connect(DATABASE_URL,sslmode='require')
        cursor = connection_db.cursor()
        SQL_cmd = f"""SELECT * FROM ordermenu ORDER by order_id DESC limit 10;"""
        cursor.execute(SQL_cmd)
        connection_db.commit()
        raws = cursor.fetchall()
        data = []
        for raw in raws:
            data.append(raw)

        cursor.close()
        connection_db.close()
        return data
```

6. 將資料表送上 Heroku 主機，進行測試！