使用SymPy 符號運算套件 與 Flask建構微分網站

SymPy

About

SymPy is a Python library for symbolic mathematics. It aims to become a full-featured computer algebra system (CAS) while keeping the code as simple as possible in order to be comprehensible and easily extensible. SymPy is written entirely in Python.

Get started with the tutorial | Download Now

Why SymPy

SymPy is...

- **Free:** Licensed under BSD, SymPy is free both as in speech and as in beer.
- **Python-based:** SymPy is written entirely in Python and uses Python for its language.
- **Lightweight:** SymPy only depends on mpmath, a pure Python library for arbitrary floating point arithmetic, making it easy to use.
- **A library:** Beyond use as an interactive tool, SymPy can be embedded in other

```
Python Console
>>> import sympy as sp
... from sympy import *
```

■ 使用 sympy 套件

- import sympy ：使用 sympy 套件內定義的名稱之前需加上套件名稱，即 sympy.
- from sympy import * ：直接使用 sympy 套件內定義的所有名稱

```
>>> x, y = symbols("x,y")
>>> var("x y")
(x, y)
```

var():直接設定符號變數

```
>>> x + 2*y -3*x
-2*x + 2*y
```

```
>>> var("x:3")
(x0, x1, x2)
>>> var("x:z")
(x, y, z)
>>> var("x:z:2")
(x0, x1, y0, y1, z0, z1)
```

設定多個符號變數

$(x+2y)^2 = x^2 + 4xy + 4y^2$ $(x+2y)^3 = x^3 + 6x^2y + 12xy^2 + 8y^3$

```
>>> ((x+2*y)**2).expand()
x**2 + 4*x*y + 4*y**2
```

expand()、subs():展開式子、取代式子

```
>>> ((x-y)**2).subs(x,1)
(1 - y)**2
```

subs() :x 用 1 取代

```
>>> ((x+y)**2).subs(x,-2*y)
y**2
```

x 用 -2y 取代

```
>>> import sympy as sp
>>> from sympy import *
>>> x, y = symbols("x,y")
>>> ((x+2*y)**2).expand()
x**2 + 4*x*y + 4*y**2
```

因式分解

$$f(x) = x^3 - 4x - 1$$

$$g(x) = -2x^2 + x + 5$$

```
>>> ((x**2-2*x+1)).factor()
(x - 1)**2
```

$$f - g = x^3 + 2x^2 - 5x - 6$$

$$= (x^3 + x^2) + (x^2 + x) - 6(x + 1)$$

$$= x^2(x + 1) + x(x + 1) - 6(x + 1)$$

$$(x + 1)(x^2 + x - 6) = (x + 1)(x + 3)(x - 2)$$

```
>>> fn = x**3 - 4*x - 1
>>> gn = -2*x**2 + x + 5
>>> factor(fn-gn)
(x - 2)*(x + 1)*(x + 3)
```

定義 fn gn 兩式子

因式分解

$$\frac{x^2-1}{x-1}$$

```
>>> (x**2-1)/(x-1)
(x**2 - 1)/(x - 1)
>>> ((x**2-1)/(x-1)).simplify()
x + 1
```

沒有做約分

簡化運算

```
>>> 1/3
0.3333333333333333
>>> Rational(2,6)
1/3
```

$$\frac{x^2-1}{x+1}, \text{ x is set to -1}$$

```
>>> ((x**2-1)/(x+1)).subs(x,-1)
nan
>>> limit( (x**2-1)/(x+1) , x , -1 )
-2
>>> ((x**2-1)/(x+1)).limit(x,-1)
-2
```

```
# 數值不存在

# lim_{x→-1} (x^2 - 1)/(x + 1)
```

單邊極限：

```
>>> limit( floor(x)/x , x , 3 )                    # lim_{x→3} ⌊x⌋/3
1

>>> limit( floor(x)/x , x , 3 , '+' )               # 同上 lim_{x→3⁺} ⌊x⌋/3
1

>>> limit( floor(x)/x , x , 3 , '-' )               # lim_{x→3⁻} ⌊x⌋/3
2/3
```

無窮極限：

```
>>> limit( (x**3-4)/(2*abs(x)**3), x, oo )    # lim_{x→∞} (x^3 - 4)/(2|x|^3)
1/2
>>> limit( (x**3-4)/(2*abs(x)**3), x, -oo )   # lim_{x→-∞} (x^3 - 4)/(2|x|^3)
-1/2
```

```
>>> limit( sin(1/x), x, oo )                  # lim_{x→∞} sin(1/x)
0
>>> fn = x*sin(1/x)                           # fn = x sin(1/x)
>>> limit( fn, x, oo )                        # lim_{x→∞} x sin(1/x)
1
```

使用sympy求兩個多項式函數的加、減、乘、除

```python
import sympy as sp
x = sp.symbols('x')
f1 = sp.Function('f1')
f2 = sp.Function('f2')
f1 = x**2-x+1
f2 = x**2+1-x
print(f1-f2)
```

兩個函數相減

```
0

Process finished with exit code 0
```

```python
import sympy as sp
x = sp.symbols('x')
f1 = sp.Function('f1')
f2 = sp.Function('f2')
f1 = x**2-x+1
f2 = x**2+1-x
print(f1 - f2)
print(f1 + f2)
```

在個函數相加

```
0
2*x**2 - 2*x + 2

Process finished with exit code 0
```

```python
import sympy as sp
x = sp.symbols('x')
f1 = sp.Function('f1')
f2 = sp.Function('f2')
f1 = x**2-x+1
f2 = x**2+1-x
print(f1 - f2)
print(f1 + f2)
print(f1 * f2)
```

兩個函數相乘

```python
import sympy as sp
x = sp.symbols('x')
f1 = sp.Function('f1')
f2 = sp.Function('f2')
f1 = 2*x**2-x-1
f2 = 1-x
print(f1 - f2)
print(f1 + f2)
print(f1 * f2)
```

$2x^2 - x - 1$

$1 - x$

兩個函數相乘

```
2*x**2 - 2
2*x**2 - 2*x
(1 - x)*(2*x**2 - x - 1)
```

```python
import sympy as sp
x = sp.symbols('x')
f1 = sp.Function('f1')
f2 = sp.Function('f2')
f1 = 2*x**2-x-1
f2 = 1-x
print(f1 - f2)
print(f1 + f2)
print(f1 * f2)
print(f1 / f2)
```

```
2*x**2 - 2
2*x**2 - 2*x
(1 - x)*(2*x**2 - x - 1)
(2*x**2 - x - 1)/(1 - x)
```

```
import sympy as sp
from sympy import *
x= sp.symbols('x')
fn = x**3 -4*x-1
gn = -2*x**2+x+5
factor(fn-gn)
```

因式分解

```
(x - 2)*(x + 1)*(x + 3)
```

```
import sympy as sp
from sympy import *
x= sp.symbols('x')
fn = x**3 -4*x-1
print(fn.subs(x,3))
```

帶入特定變數值，求函數值

```
14
```

使用sympy求函數微分

```
import sympy as sp
from sympy import *
x= sp.symbols('x')
fn = x**3 -4*x-1
ans = diff(fn)
print(ans)
```

$$fn = x^3 - 4x - 1$$

$$fn' = 3x^2 - 4$$

求函數微分

```
3*x**2 - 4
```

```
import sympy as sp
from sympy import *
x= sp.symbols('x')
ss = "x**3 -4*x-1"
eval("diff("+ss+")")
```

1. 將代表函數的字串轉為微分指令字串
2.使用eval，求微分

```
3*x**2 - 4
```

```
import sympy as sp
from sympy import *
x= sp.symbols('x')
ss = "x**3 -4*x-1"
tt = "diff("+ss+")"
ans = eval(tt)
print(ans)
```

函數字串

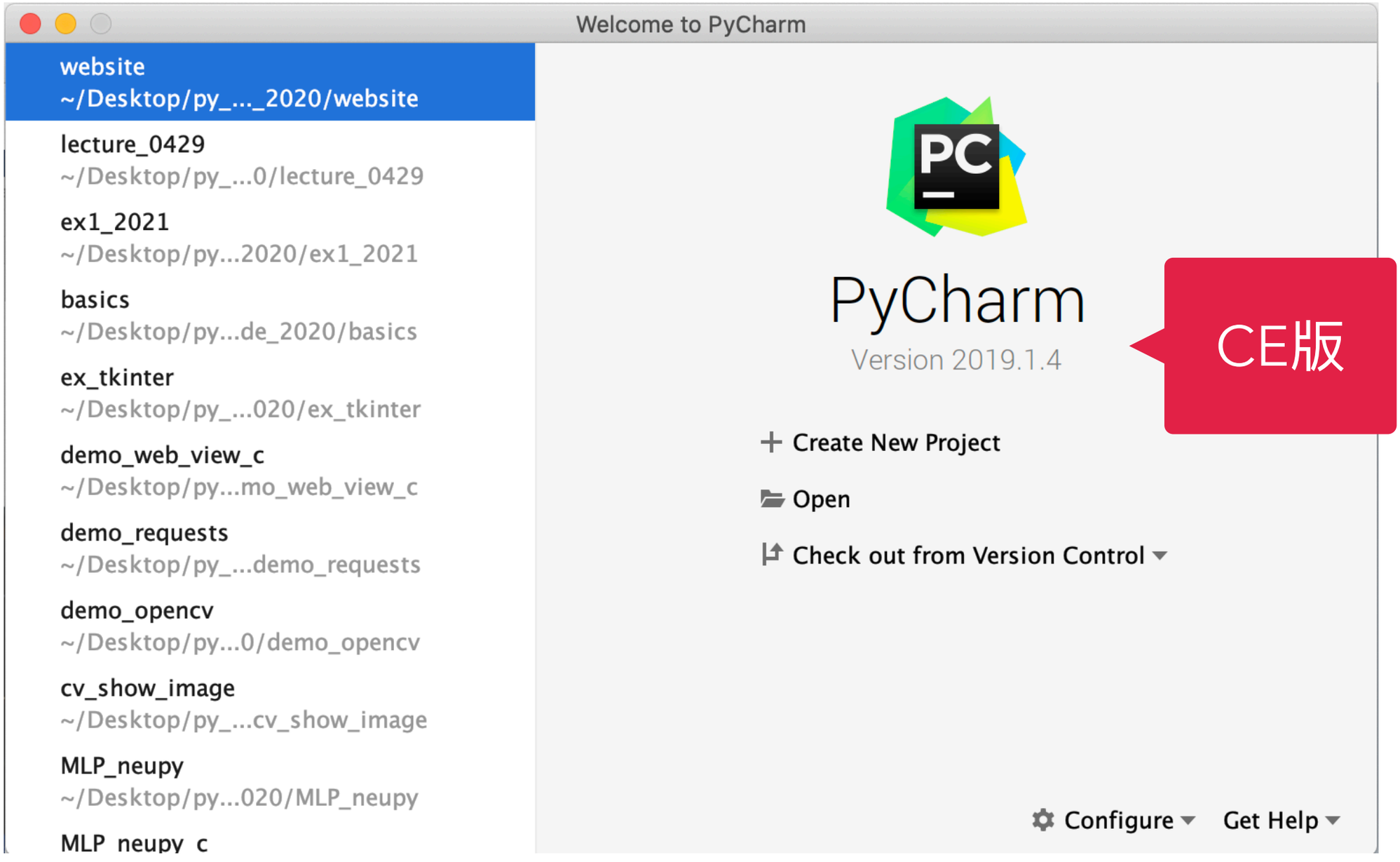
微分指令字串

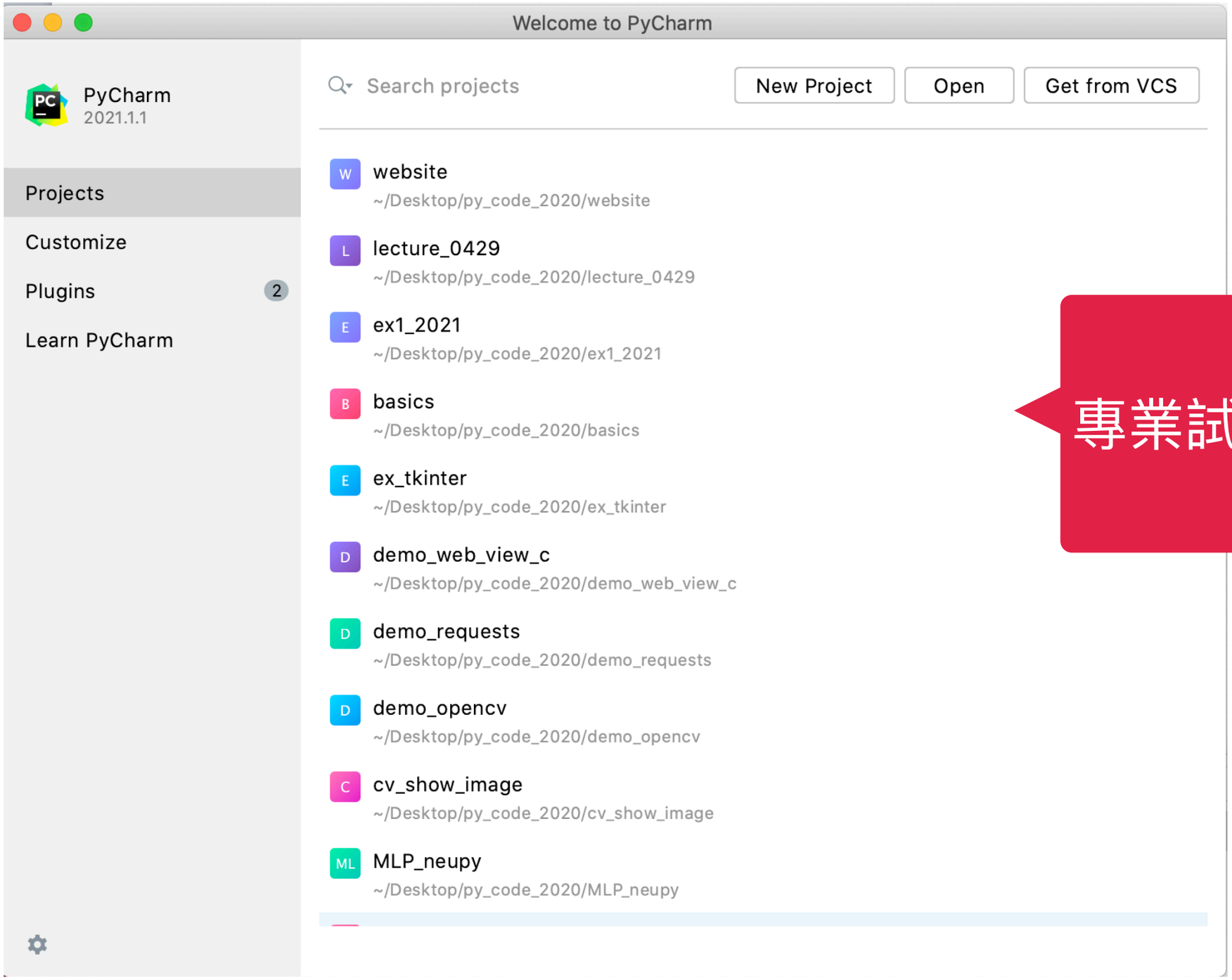
使用eval，求微分

```
3*x**2 - 4
```

Website : POST, GET

Flask and Pycharm


Welcome to PyCharm
website
~/Desktop/py_..._2020/website
lecture_0429
~/Desktop/py_...0/lecture_0429
ex1_2021
~/Desktop/py...2020/ex1_2021
basics
~/Desktop/py...de_2020/basics
ex_tkinter
~/Desktop/py_...020/ex_tkinter
demo_web_view_c
~/Desktop/py...mo_web_view_c
demo_requests
~/Desktop/py_...demo_requests
demo_opencv
~/Desktop/py...0/demo_opencv
cv_show_image
~/Desktop/py_...cv_show_image
MLP_neupy
~/Desktop/py...020/MLP_neupy
MLP_neupy_c
PyCharm
Version 2019.1.4
CE版
Create New Project
Open
Check out from Version Control
Configure
Get Help


Welcome to PyCharm
PyCharm
2021.1.1
Projects
Customize
Plugins
2
Learn PyCharm
Search projects
New Project
Open
Get from VCS
website
~/Desktop/py_code_2020/website
lecture_0429
~/Desktop/py_code_2020/lecture_0429
ex1_2021
~/Desktop/py_code_2020/ex1_2021
basics
~/Desktop/py_code_2020/basics
ex_tkinter
~/Desktop/py_code_2020/ex_tkinter
demo_web_view_c
~/Desktop/py_code_2020/demo_web_view_c
demo_requests
~/Desktop/py_code_2020/demo_requests
demo_opencv
~/Desktop/py_code_2020/demo_opencv
cv_show_image
~/Desktop/py_code_2020/cv_show_image
MLP_neupy
~/Desktop/py_code_2020/MLP_neupy
專業試用版

Project Links

Donate to Pallets
Flask Website
PyPI releases
Source Code
Issue Tracker

Contents

Welcome to Flask
- User's Guide
- API Reference
- Additional Notes

Quick search

Go

Welcome to Flask's documentation. Get started with Installation and then get an overview with the Quickstart. There is also a more detailed Tutorial that shows how to create a small but complete application with Flask. Common patterns are described in the Patterns for Flask section. The rest of the docs describe each component of Flask in detail, with a full reference in the API section.

Flask depends on the Jinja template engine and the Werkzeug WSGI toolkit. The documentation for these libraries can be found at:

- Jinja documentation
- Werkzeug documentation

User's Guide

This part of the documentation, which is mostly prose, begins with some background information about Flask, then focuses on step-by-step instructions for web development with Flask.

匯入模組

flask

website 〉 app.py

Project

website ~/Desktop/py_code_2020/websi
venv
app.py
External Libraries
Scratches and Consoles

app.py

```
1  from flask import Flask
2
3  app = Flask(__name__)
4
5  @app.route('/')
6  def hello_world():
7      return "Hello World!"
8
9  if __name__ == '__main__':
10     app.run(debug=True)
```

匯入類別 Flask

宣告變數 app的類別 為Flask

if __name__ == '__main__'

Run: app

```
   Use a production WSGI server instead.
 * Debug mode: on
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 634-232-453
127.0.0.1 - - [05/May/2021 21:56:59] "GET / HTTP/1.1" 200 -
```

1: Project
7: Structure
2: Favorites

website 〉 app.py

Project

website ~/Desktop/py_code_2020/websi
venv
app.py
External Libraries
Scratches and Consoles

app.py

```
1  from flask import Flask
2
3  app = Flask(__name__)
4
5  @app.route('/')
6  def hello_world():
7      return "Hello World!"
8
9  if __name__ == '__main__':
10     app.run(debug=True)
```

網頁根路徑下，執行函數hello_world()

如果是主程式，保留字__name__為__main__

if __name__ == '__main__'

Run: app

```
   Use a production WSGI server instead.
 * Debug mode: on
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 634-232-453
127.0.0.1 - - [05/May/2021 21:56:59] "GET / HTTP/1.1" 200 -
```

1: Project
Z: Structure
2: Favorites

website 〉 app.py

Project

website ~/Desktop/py_code_2020/websi
venv
app.py
External Libraries
Scratches and Consoles

app.py

```
1  from flask import Flask
2
3  app = Flask(__name__)
4
5  @app.route('/')
6  def hello_world():
7      return "Hello World!"
8
9  if __name__ == '__main__':
10     app.run(debug=True)
```

if __name__ == '__main__'

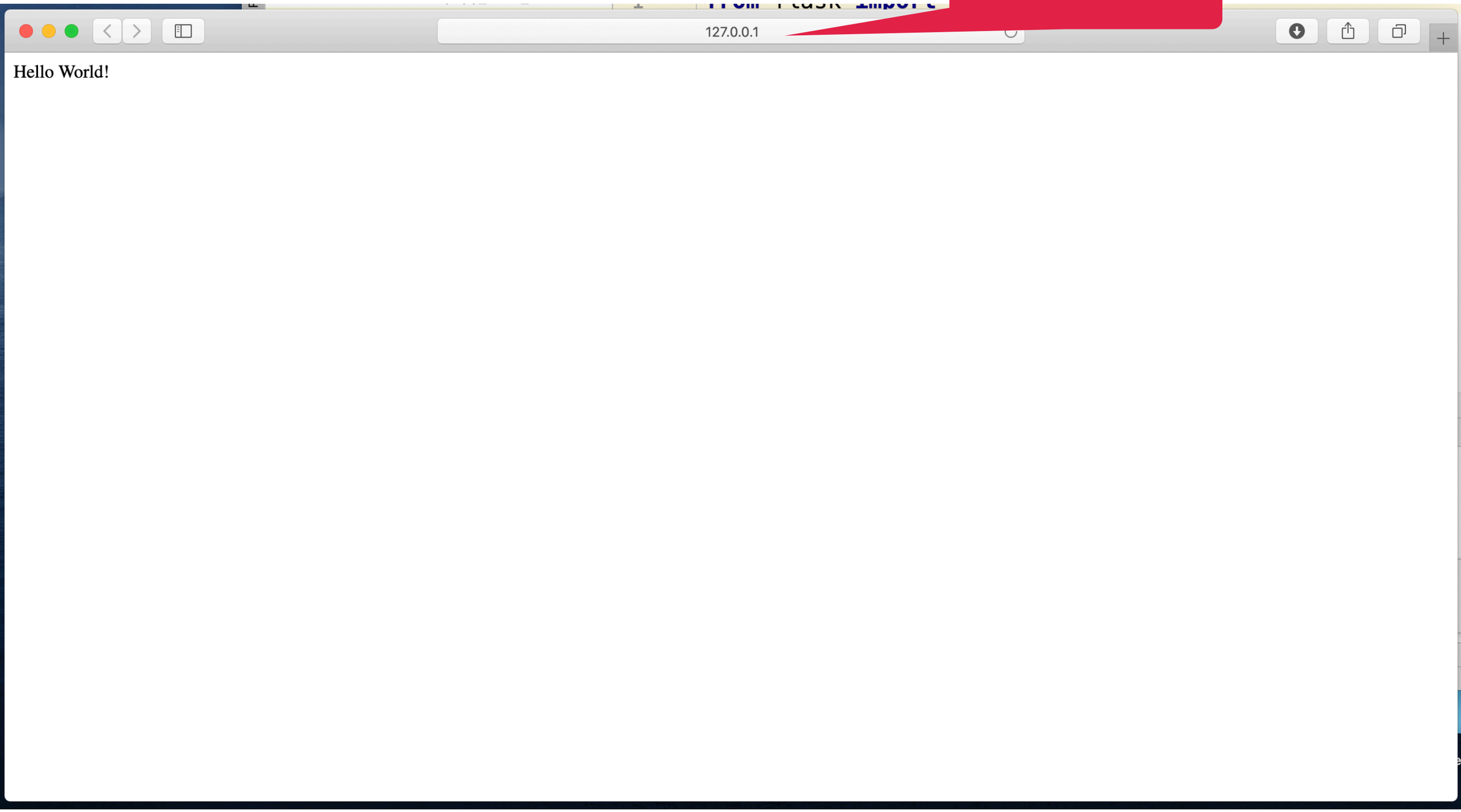
Run: app

```
   Use a production WSGI server instead.
 * Debug mode: on
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 634-232-453
127.0.0.1 - - [05/May/2021 21:56:59] "GET / HTTP/1.1" 200 -
```

網頁位址

1: Project
7: Structure
2: Favorites

網頁位址



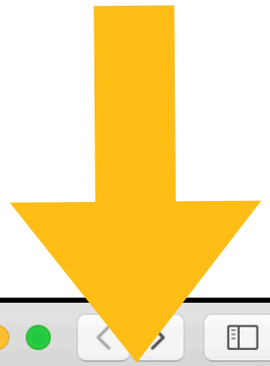
使用者從網頁上輸入

127.0.0.1

134.208.26.59/軟體實作 2021/軟體實作 2021.htm

Hello Page

Submit



127.0.0.1

134.208.26.59/軟體實作 2021/軟體實作 2021.htm

Hello Page

jiann-ming

Submit

127.0.0.1

134.208.26.59/軟體實作2021/軟體實作2021.htm

Hello Page

jiann-ming Submit

按下submit

127.0.0.1:5000

127.0.0.1

134.208.26.59/軟體實作2021/軟體實作2021.htm

127.0.0.1:5000

Hello jiann-ming

建立html檔

目錄名稱為
templates

website [~/Desktop/py_code_2020/website] - .../templates/post_submit.html [website]

website templat

Project

website ~/D sktop/py_code_2020/websi

templates

from_ex.html

post_submit.html

venv

app.py

gettextva

gettextva

External Libr

Scratches an

html檔

gettextvalues2.py gettextvalue.py post_submit.html

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>Hello Page</title>
6  </head>
7  <body>
8  <form method='post' action="/">
9      <input type='text' name='username'>
10     <button type='submit'>Submit</button>
11 </form>
12 </body>
13 </html>
```

網頁標題

website [~/Desktop/py_code_2020/website] - .../templates/post_submit.html [website]

website 〉 templates 〉 post_submit.html

Project

website ~/Desktop/py_code_2020/websi
templates
from_ex.html
post_submit.html
venv
app.py
gettextvalue.py
gettextvalues2.py
External Libraries

gettextvalues2.py | gettextvalue.py | post_submit.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Hello Page</title>
</head>
<body>
<form method='post' action="/">
    <input type='text' name='username'>
    <button type='submit'>Submit</button>
</form>
</body>
</html>
```

使用方法:POST
讓使用者貼上username

匯入類別Flask, request, render_template

website [~/Desktop/py_code_

website 〉 gettextvalues2.py

Project

website ~/Desktop/py_code_2020/websi

templates

from_ex.html

post_submit.html

venv

app.py

gettextvalue.py

gettextvalues2.py

External Librarie

Scratches a

py檔

gettextvalues2.py | gettextvalue.py | post_submit.ht l

```
1  from flask import Flask, request, render_template
2
3  app = Flask(__name__)
4
5  @app.route("/", methods=['GET', 'POST'])
6  def submit():
7      if request.method == 'POST':
8          return 'Hello ' + request.values.get('username')
9      return render_template('post_submit.html')
10
11
12 if __name__ == '__main__':
13     app.debug = True
14     app.run()
```

使用方法有兩種

website [~/Desktop/py_code_2020/website] - .../gettextvalues2.py [website]

website 〉 gettextvalues2.py

Project

website ~/Desktop/py_code_2020/websi
- templates
  - from_ex.html
  - post_submit.html
- venv
- app.py

gettextvalues2.py | gettextvalue.py | post_submit.html

```
1  from flask import Flask, request, render_template
2
3  app = Flask(__name__)
4
5  @app.route("/", methods=['GET', 'POST'])
   def submit():
       if request.method == 'POST':
           return 'Hello ' + request.values.get('username')
       return render_template('post_submit.html')


   if __name__ == '__main__':
       app.debug = True
       app.run()
```

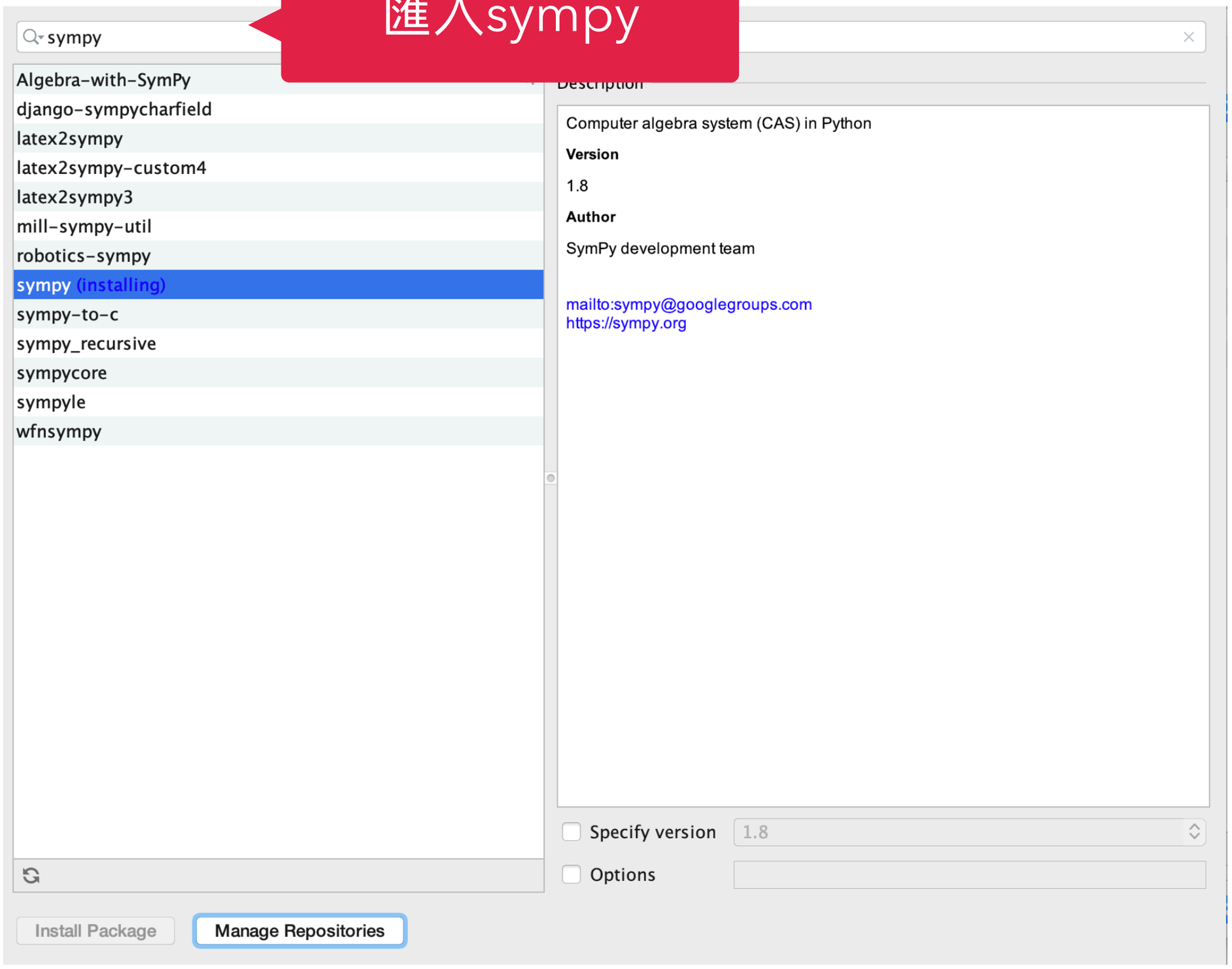
使用方法為'POST'將抓取username附加在'Hello '之後，貼在標題為'127.0.0.1:5000'的網頁上

使用方法為'GET'透過標題為'hello page'的網頁，送出username

可以建立微分與積分網站讓同學查詢解答

建立微分網站

匯入sympy


sympy
Algebra-with-SymPy
django-sympycharfield
latex2sympy
latex2sympy-custom4
latex2sympy3
mill-sympy-util
robotics-sympy
sympy (installing)
sympy-to-c
sympy_recursive
sympycore
sympyle
wfnsympy
Description
Computer algebra system (CAS) in Python
Version
1.8
Author
SymPy development team
mailto:sympy@googlegroups.com
https://sympy.org
Specify version
1.8
Options
Install Package
Manage Repositories

```python
from flask import Flask, request, render_template

app = Flask(__name__)

@app.route("/", methods=['GET', 'POST'])
def submit():
    if request.method == 'POST':
        return 'Hello ' + request.values.get('username')
    return render_template('post_submit.html')


if __name__ == '__main__':
    app.debug = True
    app.run()
```

```
import sympy as sp
from sympy import *
```

```python
from flask import Flask, request, render_template
import sympy as sp
from sympy import *


app = Flask(__name__)

@app.route("/", methods=['GET', 'POST'])
def submit():
    if request.method == 'POST':
        return 'Hello ' + request.values.get('username')
    return render_template('post_submit.html')


if __name__ == '__main__':
    app.debug = True
    app.run()
```

```python
x= sp.symbols('x')
ss = request.values.get('username')
tt = "diff("+ss+")"
ans = eval(tt)
```

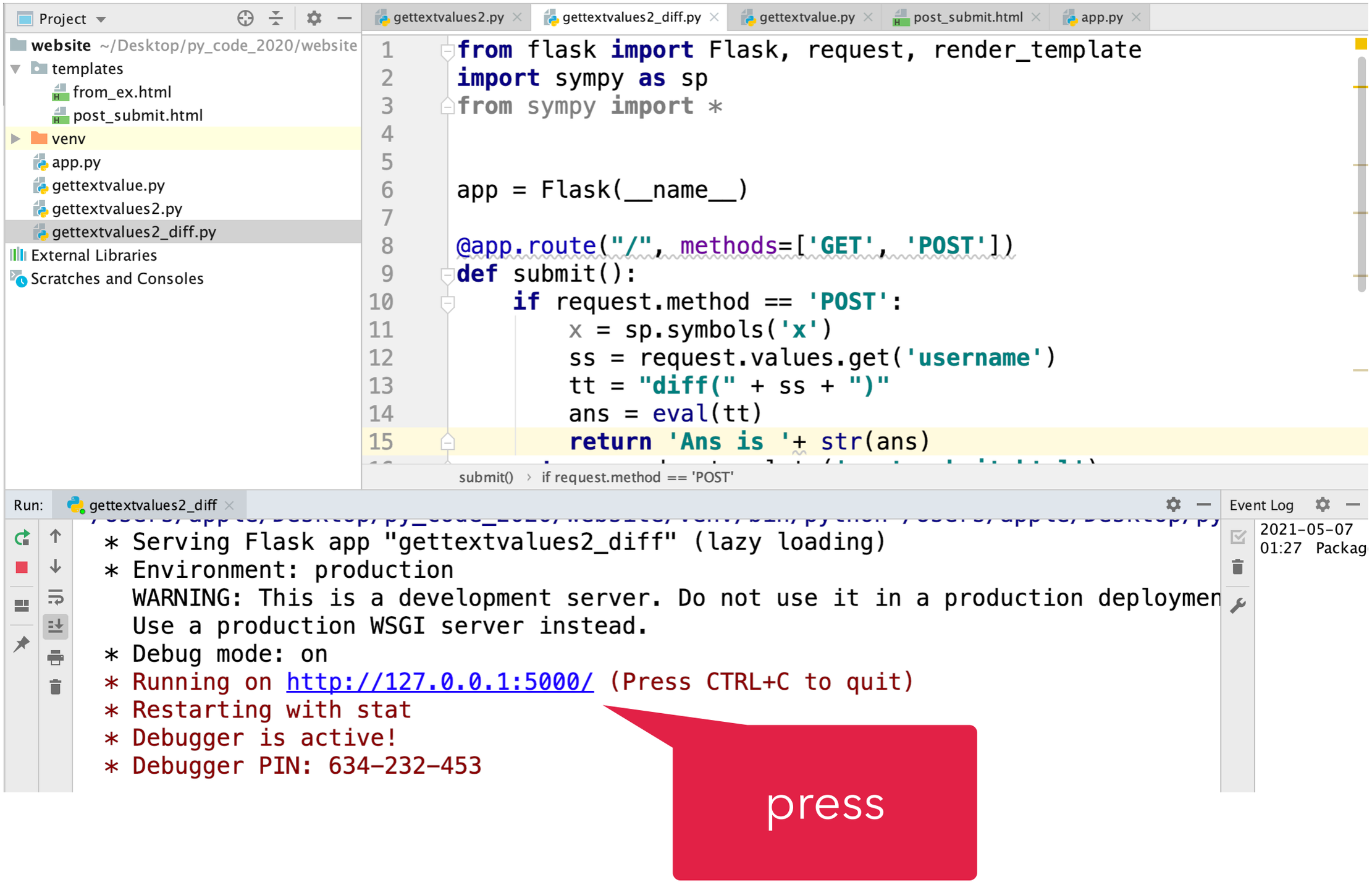
```python
from flask import Flask, request, render_template
import sympy as sp
from sympy import *


app = Flask(__name__)

@app.route("/", methods=['GET', 'POST'])
def submit():
    if request.method == 'POST':
        x = sp.symbols('x')
        ss = request.values.get('username')
        tt = "diff(" + ss + ")"
        ans = eval(tt)
        return 'Hello ' + request.values.get('username')
    return render_template('post_submit.html')


if __name__ == '__main__':
    app.debug = True
    app.run()
```

'Ans is '+str(ans)


Project
website ~/Desktop/py_code_2020/website
templates
from_ex.html
post_submit.html
venv
app.py
gettextvalue.py
gettextvalues2.py
gettextvalues2_diff.py
External Libraries
Scratches and Consoles
gettextvalues2.py
gettextvalues2_diff.py
gettextvalue.py
post_submit.html
app.py
1 from flask import Flask, request, render_template
2 import sympy as sp
3 from sympy import *
4
5
6 app = Flask(__name__)
7
8 @app.route("/", methods=['GET', 'POST'])
9 def submit():
10 if request.method == 'POST':
11 x = sp.symbols('x')
12 ss = request.values.get('username')
13 tt = "diff(" + ss + ")"
14 ans = eval(tt)
15 return 'Ans is '+ str(ans)
submit() › if request.method == 'POST'
Run:
gettextvalues2_diff
* Serving Flask app "gettextvalues2_diff" (lazy loading)
* Environment: production
WARNING: This is a development server. Do not use it in a production deploymen
Use a production WSGI server instead.
* Debug mode: on
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
* Restarting with stat
* Debugger is active!
* Debugger PIN: 634-232-453
Event Log
2021-05-07
01:27 Packag
press

127.0.0.1

Submit

127.0.0.1

x**2+2*x-1

Submit

127.0.0.1

Ans is 2*x + 2

HTML設計

函數微分

本網站使用符號運算sympy進行微分

微分運算，請輸入函數f(x)

範例，輸入函數:x**2+2*x-1，答案：2*x+2

範例，輸入函數:cos(x)，答案：-sin(x)

按鍵傳送

標題顯示
微分網頁

本網站使用符號運算sympy進行微分

微分運算，請輸入函數f(x)

範例，輸入函數:x**2+2*x-1，答案：2*x+2

範例，輸入函數:cos(x)，答案：-sin(x)

按鍵傳送

本網站使用符號運算sympy進行微分

微分運算，請輸入函數f(x)

範例，輸入函數:x**2+2*x-1，答案：2*x+2

範例，輸入函數:cos(x)，答案：-sin(x)

按鍵傳送

```
<head>
    <meta charset="UTF-8">
    <title>微分網頁</title>
</head>
```

標題顯示
微分網頁

使用<title></title>
設定在標題顯示的文字
為"微分網頁"

本網站使用符號運算sympy進行微分

微分運算，請輸入函數f(x)

範例，輸入函數:x**2+2*x-1，答案：2*x+2

範例，輸入函數:cos(x)，答案：-sin(x)

在網頁上顯示文字

按鍵傳送

本網站使用符號運算sympy進行微分

微分運算，請輸入函數f(x)

範例，輸入函數:x**2+2*x-1，答案：2*x+2

範例，輸入函數:cos(x)，答案：-sin(x)

按鍵傳送

在網頁上顯示文字

使用 顯示單行文字

```
<p>本網站使用符號運算sympy進行微分</p>
<p>微分運算，請輸入函數f(x)</p>
<p>範例，輸入函數:x**2+2*x-1，答案：2*x+2</p>
<p>範例，輸入函數:cos(x)，答案：-sin(x)</p><br>
```

使用
跳一行

本網站使用符號運算sympy進行微分

微分運算，請輸入函數f(x)

範例，輸入函數:x**2+2*x-1，答案：2*x+2

範例，輸入函數:cos(x)，答案：-sin(x)

按鍵傳送

在網頁上顯示文字輸入欄位，使用者輸入函數

本網站使用符號運算sympy進行微分

微分運算，請輸入函數f(x)

範例，輸入函數:x**2+2*x-1，答案：2*x+2

範例，輸入函數:cos(x)，答案：-sin(x)

在網頁上顯示文字輸入欄位，使用者輸入函數

text輸入欄位，變數名稱name設定為username

```
<input type='text' name='username'>
<button type='submit'>按鍵傳送</button>
```

本網站使用符號運算sympy進行微分

微分運算，請輸入函數f(x)

範例，輸入函數:x**2+2*x-1，答案：2*x+2

範例，輸入函數:cos(x)，答案：-sin(x)

按鍵傳送

在網頁上顯示按鍵，使用者可以用滑鼠按下按鍵，完成get 模式

本網站使用符號運算sympy進行微分

微分運算，請輸入函數f(x)

範例，輸入函數:x**2+2*x-1，答案：2*x+2

範例，輸入函數:cos(x)，答案：-sin(x)

在網頁上顯示按鍵，使用者可以用滑鼠按下按鍵，完成get 模式

Button按鍵，型態submit，顯示文字“按鍵傳送”

```
<input type='text' name='username'>
<button type='submit'>按鍵傳送</button>
```

目錄名稱為
templates

post_submit.
html檔

website ~/Desktop/py_code
templates
from_ex.html
post_submit.html
venv
app.p

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>微分網頁</title>
</head>
<body>
<form method='post' action="/">
    <p>本網站使用符號運算sympy進行微分</p>
    <p>微分運算，請輸入函數f(x)</p>
    <p>範例，輸入函數:x**2+2*x-1，答案：2*x+2</p>
    <p>範例，輸入函數:cos(x)，答案：-sin(x)</p><br>
    <input type='text' name='username'>
    <button type='submit'>按鍵傳送</button>
</form>
</body>
</html>
```

website 〉 templates 〉 post_submit.html

gettextvalues2_diff2

Project

website ~/Desktop/py_code

templates

from_ex.html

post_submit.html

venv

app.py

gettextvalue.py

gettextvalues2.py

gettextvalues2_diff.py

gettextvalues2_diff2.py

External Libraries

Scratches and Consoles

gettextvalues2.py | gettextvalue.py | gettextvalues2_diff.py | gettextvalues2_diff2.py | app.py | post_submit.html | from_ex.htm

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>微分網頁</title>
6  </head>
7  <body>
8  <form method='post' action="/">
9      <p>本網站使用符號運算sympy進行微分</p>
10     <p>微分運算，請輸入函數f(x)</p>
11     <p>範例，輸入函數:x**2+2*x-1，答案：2*x+2</p>
12     <p>範例，輸入函數:cos(x)，答案：-sin(x)</p><br>
13     <input type='text' name='username'>
14     <button type='submit'>按鍵傳送</button>
15 </form>
16 </body>
17 </html>
```

使用<title></title>設定在標題顯示的文字為"微分網頁"

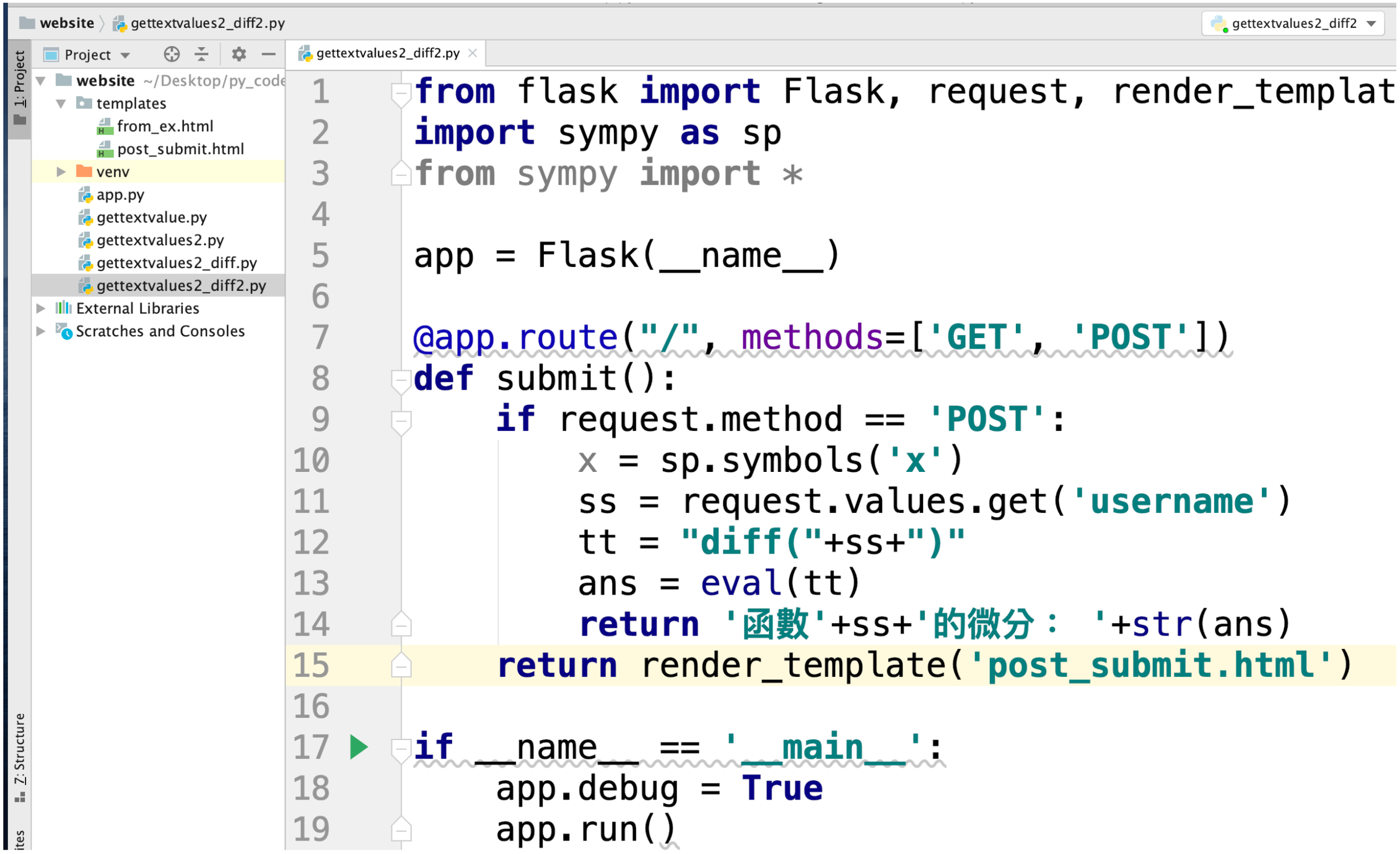
使用 顯示單行文字

text輸入欄位，變數名稱name設定為username

Button按鍵，型態submit，顯示文字"按鍵傳送"

1: Project

Z: Structure


website
gettextvalues2_diff2.py
gettextvalues2_diff2
Project
website ~/Desktop/py_code
templates
from_ex.html
post_submit.html
venv
app.py
gettextvalue.py
gettextvalues2.py
gettextvalues2_diff.py
gettextvalues2_diff2.py
External Libraries
Scratches and Consoles
1: Project
Z: Structure
from flask import Flask, request, render_templat
import sympy as sp
from sympy import *
app = Flask(__name__)
@app.route("/", methods=['GET', 'POST'])
def submit():
if request.method == 'POST':
x = sp.symbols('x')
ss = request.values.get('username')
tt = "diff("+ss+")"
ans = eval(tt)
return '函數'+ss+'的微分： '+str(ans)
return render_template('post_submit.html')
if __name__ == '__main__':
app.debug = True
app.run()

gettextvalues2_diff2

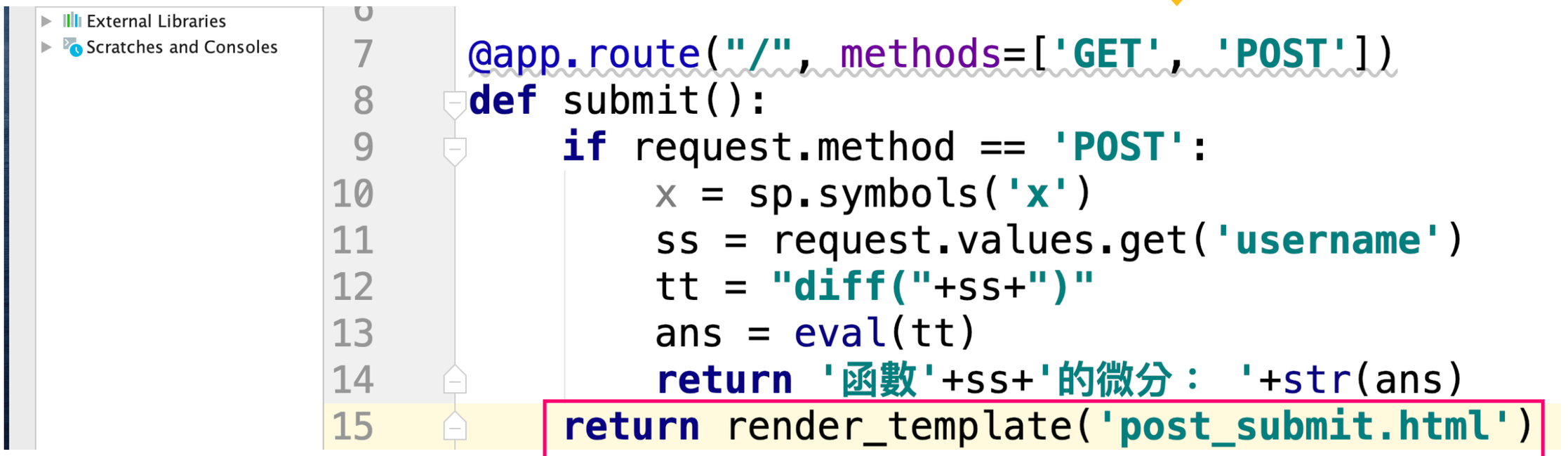
gettextvalues2_diff2.py

```
1  from flask import Flask, request, render_templat
2  import sympy as sp
3  from sympy import *
4
```

匯入套件sympy，以設計微分符號運算功能

request
GET方法:render_template，執行html檔，擷取輸入文字，完成submit
POST方法：在網頁上輸出運算結果

GET方法與POST方法



request.method為'GET'時（先執行GET方法），使用render_template，執行post_template.html檔，擷取輸入文字，完成submit

GET方法與POST方法

External Libraries
Scratches and Consoles

```
7   @app.route("/", methods=['GET', 'POST'])
8   def submit():
9       if request.method == 'POST':
10          x = sp.symbols('x')
11          ss = request.values.get('username')
12          tt = "diff("+ss+")"
13          ans = eval(tt)
14          return '函數'+ss+'的微分： '+str(ans)
15      return render_template('post_submit.html')
```

request.method為'POST'時，執行以下步驟

1. 宣告符號變數x
2. 將ss設定為，GET模式傳回的username內容
3. 將字串tt設定為微分指令
4. 使用eval執行tt所代表的微分指令，儲存在ans中
5. 回傳結果

GET方法

本網站使用符號運算sympy進行微分

微分運算，請輸入函數f(x)

範例，輸入函數:x**2+2*x-1，答案：2*x+2

範例，輸入函數:cos(x)，答案：-sin(x)

tanh(x) 按鍵傳送

submit，將GET方法擷取的資料傳送給python

POST方法：
運算與回傳結果

函數tanh(x)的微分： 1 - tanh(x)**2