

Python - Flask (lokalny)

© Copyright by 3bird Projects 2023, <http://edukacja.3bird.pl>

Uwagi ogólne

Flask to zewnętrzny moduł Pythona. Jest serwerem WWW, który potrafi interpretować język Python i zamieniać go w znaczniki HTML. Umożliwia więc wyświetlanie stron WWW napisanych w języku Python (podobnie jak PHP wewnątrz HTML).

Aby wyświetlić taką stronę www, należy wpisać w przeglądarce internetowej następujący adres: `http://localhost:5000`

Uwaga! Wcięcia w kodzie, mają dla Pythona znaczenie (są konieczne w odpowiednich miejscach)! Python do pobrania (dla Windows): <https://www.python.org/downloads/windows/>

Uruchamianie skryptu:

```
C:\> python naszSkrypt.py
```

Uwaga: Nigdy nie wolno kopiować kodu z PDF-a, gdyż zawiera on niewidoczne znaki końca linii i tzw. twarde odstęp. Kod należy przepisać ze zrozumieniem.

Kod - wersja podstawowa

```
# Skrypt wymaga ręcznego zainstalowania modułu w „Wierszu poleceń”:
```

```
# pip install flask --user
```

```
# Flask to klasa w module flask:
```

```
from flask import Flask
```

```
# Tworzymy instancję klasy Flask:
```

```
serwerFlask = Flask(__name__)
```

```
# Powiązanie adresu URL z funkcją (trasa):
```

```
@serwerFlask.route("/")
```

```
def prostaStronaWWW(): # Nazwa funkcji może być dowolna
```

```
    dzialanieMatematyczne = 2+5 # To jest interpretowane jako integer
```

```
    # Na stronie internetowej mogą być prezentowane jedynie stringi:
```

```
    wynik = str(dzialanieMatematyczne)
```

```
    tresp = "<!DOCTYPE html>"
```

```
    tresp += "<html>"
```

```
    tresp += "<head>"
```

```
    tresp += '<meta charset="utf-8">'
```

```
    tresp += "<title>Strona we Flask</title>"
```

```
    tresp += "</head>"
```

```
    tresp += '<body style="text-align: center;">'
```

```
    tresp += "<h1>Testowanie Flaska</h1>"
```

```
    tresp += '<p>Ulubiona liczba Adama to: ' + wynik + ' (to jest wynik dodawania liczb).</p>'
```

```
    tresp += '<p>Przykładowy odnośnik: <a href="http://3bird.pl">3bird Projects</a>.</p>'
```

```
    tresp += "</body>"
```

```
tresc += "</html>"
return tresc # Domyślnym typem zwracanej zawartości jest HTML
```

serwerFlask.run()

Kod - wersja rozbudowana

```
#!/usr/bin/env python
# Powyższa linia, tylko dla systemu Linux.

# Skrypt wymaga ręcznego zainstalowania modułu (Linux / Windows) na koncie zwykłego
# użytkownika z ograniczeniami:
# pip install flask --user

# Flask to klasa w module flask, natomiast url_for potrzebne do favicon.png
from flask import Flask, url_for

# Tworzymy instancję klasy Flask:
serwerFlask = Flask(__name__)
# Info: W tym przypadku, __name__ oznacza nazwę bieżącego pliku (bez rozszerzenia).
# Potrzebne to jest do określenia lokalizacji głównego pliku aplikacji, aby później zlokalizować
# względem niego także inne składniki aplikacji (obrazy, szablony, itp).

# Powiązanie adresu URL z funkcją (trasa), czyli: Jaka ścieżka adresu w przeglądarce wyświetli
# poniższą funkcję. Na przykład, jeśli chcemy, aby strona była dostępna pod adresem
# http://localhost:5000/edukacja/ - wpisujemy "/edukacja/"
@serwerFlask.route("/")
# Info: W adresie strony możemy również umieszczać zmienne za pomocą <nazwaZmiennej>.
# Drugim argumentem route może być: methods=['POST', 'GET'].

def prostaStronaWWW(): # Nazwa może być dowolna
    dzialanieMatematyczne = 2+5 # To jest interpretowane jako integer
    # Na stronie internetowej mogą być prezentowane jedynie stringi:
    wynik = str(dzialanieMatematyczne)
    # Znak nowej linii tylko w celach estetycznych (żeby kod źródłowy wyglądał ładnie):
    tresc = "<!DOCTYPE html>\n"
    tresc += "<html>\n"
    tresc += "<head>\n"
    tresc += '<meta charset="utf-8">\n'
    tresc += '<link rel="icon" href="grafika/favicon.png">\n'
    tresc += "<title>Strona we Flask</title>\n"
    tresc += "</head>\n\n"
    tresc += '<body style="text-align: center;">\n'
    tresc += "<h1>Testowanie Flaska</h1>\n"
    tresc += '<p>Ulubiona liczba Adama to: <span style="color: red; font-size: larger; font-weight: bold;">' + wynik + '</span>(to jest wynik dodawania liczb).</p>'
    tresc += '<p>Przykładowy odnośnik: <a href="http://3bird.pl">3bird Projects</a>.</p>'
    tresc += "</body>\n"
```

```

trecsc += "</html>"
return trecsc # Domyślnym typem zwracanej zawartości jest HTML

# Aby wyświetlić zwykłą zewnętrzną statyczną stronę www:
from flask import render_template

@serwerFlask.route("/edukacja/")
def zewnetrznaStronaWWW():
    # Wyświetli zwykłą stronę internetową umieszczoną w podkatalogu "templates" (dokładnie
    # taka nazwa!):
    return render_template('zewnetrzna.html')

# Serwer Flask żąda na stronie favicon.ico. Problem pojawia się wtedy, gdy ikona znajduje się w
# innej lokalizacji lub w innym formacie. Dlatego właśnie definiujemy ścieżki zrozumiałe dla
# Flaska. Dla strony głównej:
@serwerFlask.route('/grafika/favicon.png') # Jeśli serwer Flask zażąda tego pliku, znajdzie go
# w poniższej (zrozumiałej dla siebie) ścieżce:
def favicon():
    return url_for('static', filename='/grafika/favicon.png')

# Dla zewnętrznej witryny:
@serwerFlask.route('/edukacja/grafika/favicon.png') # Jeśli serwer Flask zażąda tego pliku,
# znajdzie go w poniższej (zrozumiałej dla siebie) ścieżce:
def faviconEdukacja():
    return url_for('static', filename='/edukacja/grafika/favicon.png')

# Wykonaj, jeśli obecny skrypt został uruchomiony przez obecny plik (a nie z zewnętrznego
# pliku / modułu):
if __name__ == "__main__":
    serwerFlask.run(host='0.0.0.0', port='5001') # Argumentem funkcji może być:
    # run(host='0.0.0.0', port='5001'). IP 0.0.0.0 - oznacza KAŻDE. Domyślny port to: 5000.
    # Uważać na firewall!

```

Plik „zewnetrzny.html”

Info: plik należy zapisać do podfolderu „**templates**”. Należy także utworzyć podfoldery „**grafika**” oraz „**templates/grafika**” i umieścić w nich pliki „*favicon.png*” (dowiedz się, czym jest *favicon*).

```

<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <link rel="icon" href="grafika/favicon.png">
    <title>Testowanie Flaska</title>
</head>

<body>
    <h1>Zewnętrzna strona WWW</h1>

```

```
<p>Witaj na mojej witrynie!</p>  
</body>  
</html>
```

Ostatnia aktualizacja: 13 stycznia 2023.