



Σχολή Πολιτικών Μηχανικών

Μέθοδοι Επίλυσης με Η/Υ

Καθ. Νίκος Λαγαρός (nlagaros@central.ntua.gr)

Δρ. Αθανάσιος Στάμος ΕΔΙΠ (stamthan@central.ntua.gr)

Μάρκος Καραμπάτσης ΕΔΙΠ (marka@mail.ntua.gr)

Δρ. Χριστόδουλος Φραγκουδάκης ΕΔΙΠ
(chfrag@mail.ntua.gr)

Προγραμματισμός και Θέματα Τεχνητής Νοημοσύνης

```
index.ts x .env.local x _app.tsx x
1 import { useEffect } from 'react'; 8.23
2 import Head from 'next/head';
3 import type { AppProps } from 'next/app'
4 import { ApolloProvider } from '@apollo
5 import { ThemeProvider } from '@material
6 import CssBaseline from '@material-ui/
7 import { Container } from '@material-u
8 import { useApollo } from '../graphql
9
10 import { lightTheme, darkTheme } fro
11 import useLocalStorage from '../hoo
12
13 import NavBar from '../components/
14
15 function App({ Component, pagePro
16   const [currentTheme, setCurrent
17   const apolloClient = useApollo
18
19   useEffect( effect: () => {
20     const jssStyles = document
21     if (jssStyles) {
22       jssStyles.parentElement
23     }
24   }, [deps: []]);
25
26   return (
27     <>
28     <Head>
29       <title>ECU-DEV</t
30       <meta name="view
31     </Head>
32     <ThemeProvider th
33     <ApolloProvider
34     <CssBaseline
35     <Container>
```

Python Web Programming

Η Python από την 1η έκδοση υποστηρίζει δικτυακές εφαρμογές. Στην πρότυπη βιβλιοθήκη υπάρχουν κλάσεις που υλοποιούν διακομιστή (web server):

```
from http.server import HTTPServer, SimpleHTTPRequestHandler
server_address = ('', 8080)
myserver = HTTPServer(server_address, SimpleHTTPRequestHandler)
myserver.serve_forever()
```

Ο διακομιστής που δημιουργείται δείχνει τα αρχεία που βρίσκονται στον τρέχοντα φάκελο.

Η διεύθυνση είναι **127.0.0.1 (localhost)** και η θύρα (port number) είναι **8080**. Μπορούμε να δούμε την ιστοσελίδα στη διεύθυνση:

<http://127.0.0.1:8080> (ή <http://localhost:8080>)

Python Web Programming

Το πρόγραμμα Hello world με την πρότυπη βιβλιοθήκη

```
from http.server import BaseHTTPRequestHandler, HTTPServer

class HelloWorldHandler(BaseHTTPRequestHandler):

    def do_GET(self):
        self.send_response(200) # Send HTTP status code 200 (OK)
        self.send_header('Content-type', 'text/html') # Send headers
        self.end_headers()
        self.wfile.write(b"Hello, World!") # Send the response body

# Server settings
host = "localhost"
port = 8000

# Create and start the server
with HTTPServer((host, port), HelloWorldHandler) as server:
    print(f"Serving on http://{host}:{port}")
    server.serve_forever()
```

Python Web Programming

- Η πρότυπη βιβλιοθήκη είναι λίγο δυσχερής στη χρήση
- Υπάρχουν εξωτερικές βιβλιοθήκες:
 - bottle: χρησιμοποιεί την πρότυπη βιβλιοθήκη, είναι απλή και γρήγορη, για μικρές εφαρμογές ιστού
 - flask: πιο προχωρημένη και αξιόπιστη, για μεσαίες εφαρμογές ιστού
 - django: πολύ προχωρημένη και αξιόπιστη, για μεγαλύτερες εφαρμογές ιστού
- Και πολλές άλλες..
- Για εκπαιδευτικούς λόγους θα χρησιμοποιήσουμε την bottle

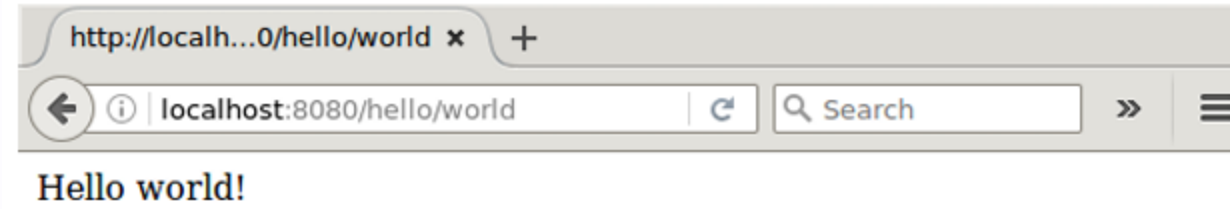
Python Web Programming

Το πρόγραμμα Hello world

```
from bottle import route, run

@route('/hello/world')
def index():
    return 'Hello world!'

run(host='localhost', port=8080)
```



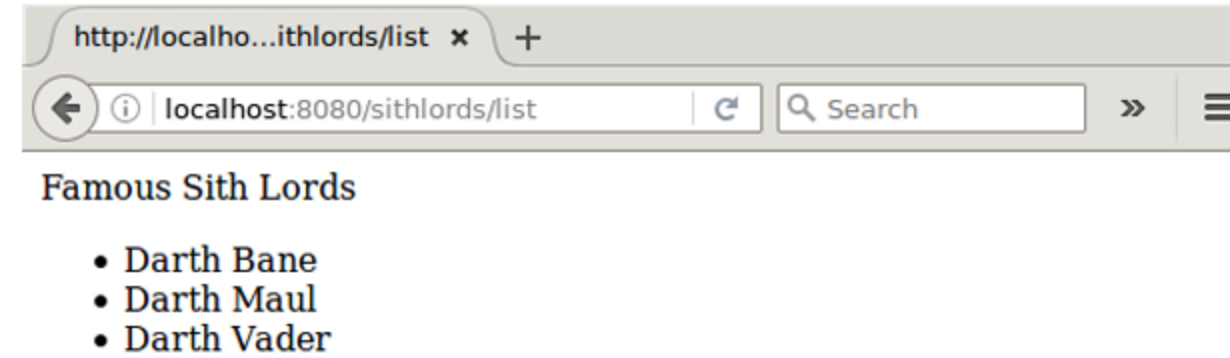
Python Web Programming

```
from bottle import route, run

sithlords = ['Darth Bane', 'Darth Maul', 'Darth Vader']

@route('/sithlords/list')
def index():
    listoflords = ['<li>' + lord + '</li>' for lord in sithlords ]
    html = 'Famous Sith Lords <br> <ul>' + ''.join(listoflords) + '</ul>'
    return html

run(host='localhost', port=8080)
```



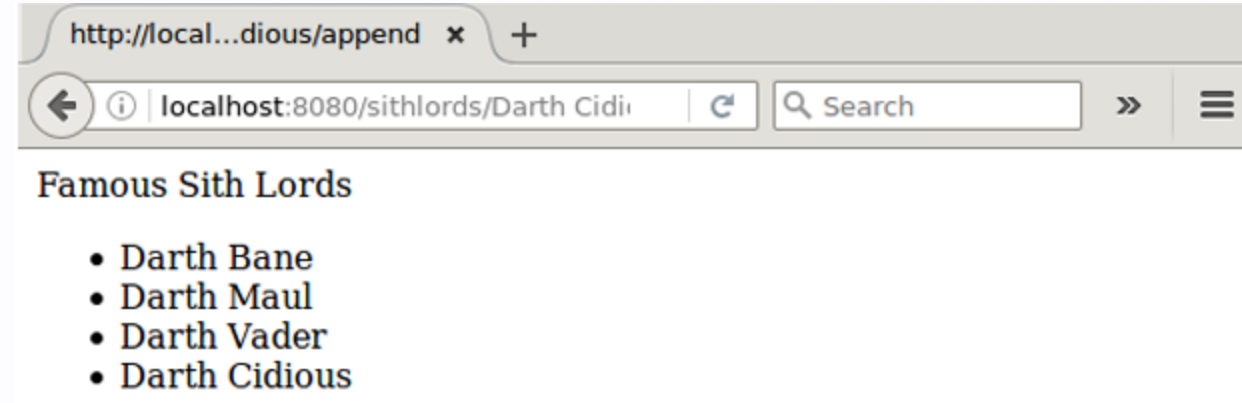
Python Web Programming

```
from bottle import route, run

sithlords = ['Darth Bane', 'Darth Maul', 'Darth Vader']

@route('/sithlords/<lord>/append')
def index(lord):
    sithlords.append(lord)
    listoflords = ['<li>' + lord + '</li>' for lord in sithlords ]
    html = 'Famous Sith Lords <br> <ul>' + ''.join(listoflords) + '</ul>'
    return html

run(host='localhost', port=8080)
```



Python Web Programming

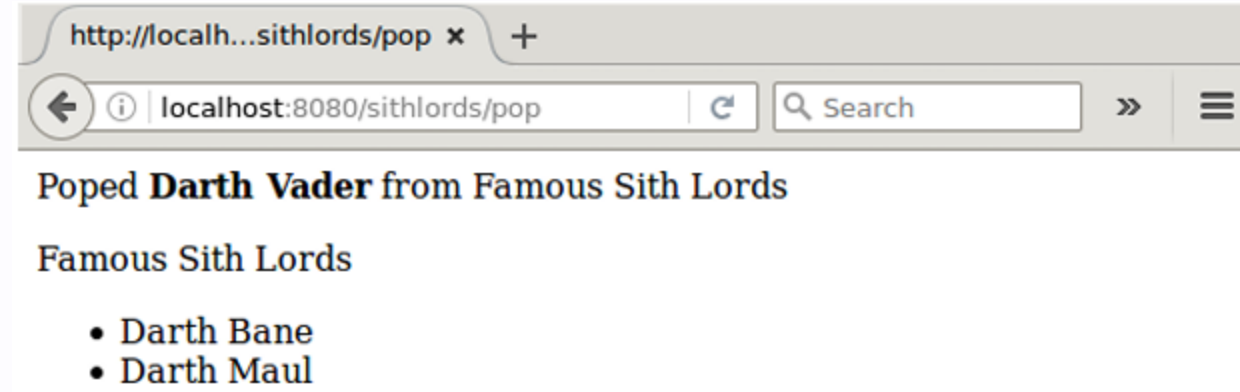
```
from bottle import route, run

sithlords = ['Darth Bane', 'Darth Maul', 'Darth Vader']

@route('/sithlords/pop')

def index():
    lord = sithlords.pop()
    html = '<p>Poped <b>' + lord + '</b> from Famous Sith Lords</p>'
    listoflords = ['<li>' + lord + '</li>' for lord in sithlords ]
    html += 'Famous Sith Lords <br> <ul>' + ''.join(listoflords) + '</ul>'
    return html

run(host='localhost', port=8080)
```

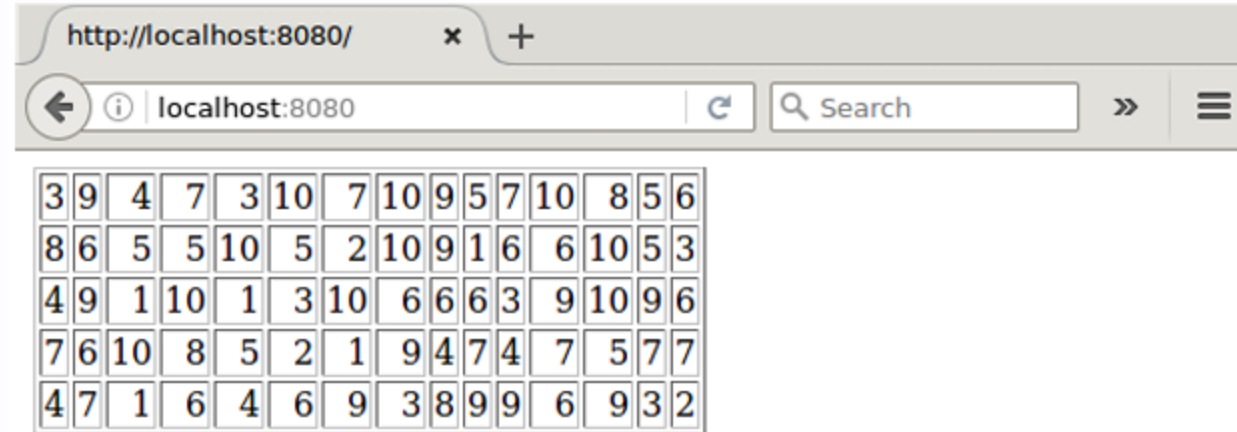


Python Web Programming

```
from bottle import route, run
from random import random

@route('/')
def index():
    array = [[int(random()*10+1) for i in range(15)] for j in range(5)]
    html = '<table border=1>'
    for row in array:
        html += '<tr>'
        for element in row:
            html += '<td align=right>' + str(element) + '</td>'
        html += '</tr>'
    return html

run(host='localhost', port=8080)
```



A screenshot of a web browser window showing the output of the Python code. The browser's address bar displays 'http://localhost:8080/'. The main content area shows a 5x15 grid of numbers, each enclosed in a small box with a border. The numbers are generated randomly and are right-aligned within their boxes. The grid contains the following values:

3	9	4	7	3	10	7	10	9	5	7	10	8	5	6
8	6	5	5	10	5	2	10	9	1	6	6	10	5	3
4	9	1	10	1	3	10	6	6	6	3	9	10	9	6
7	6	10	8	5	2	1	9	4	7	4	7	5	7	7
4	7	1	6	4	6	9	3	8	9	9	6	9	3	2

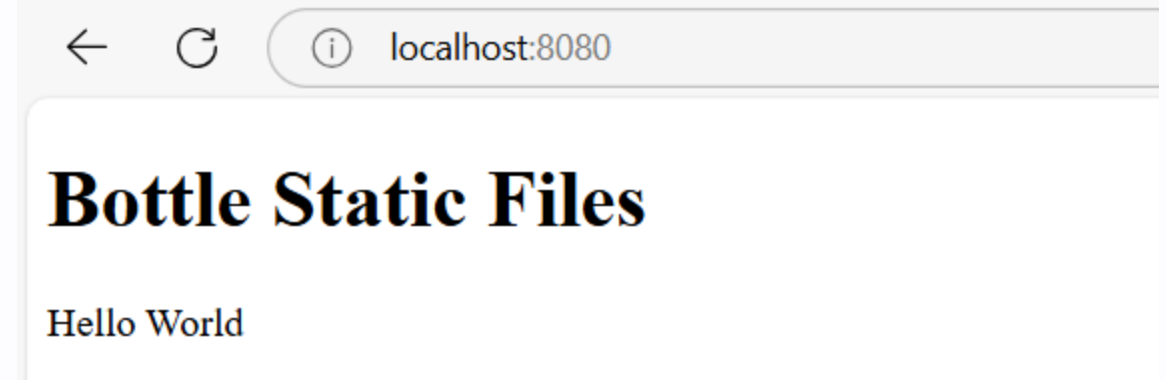
Python Web Programming

```
from bottle import get, run, template, static_file, debug

@get('/')
def index(name='World'):
    # return "I/O I/O it's off to hack we go"
    return template('test.html')

@get('/static/<filename:path>')
def get_static(filename):
    print(filename)
    return static_file(filename, root='static')

debug(mode=True)
run(host='localhost', port=8080, reloader=True)
```



Python Web Programming

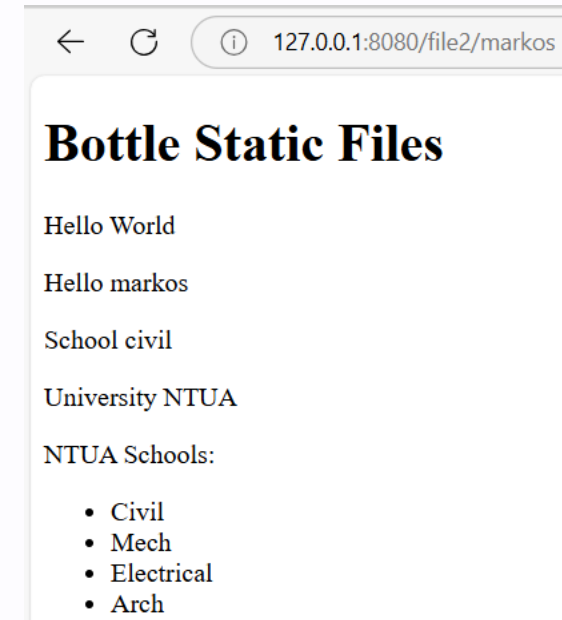
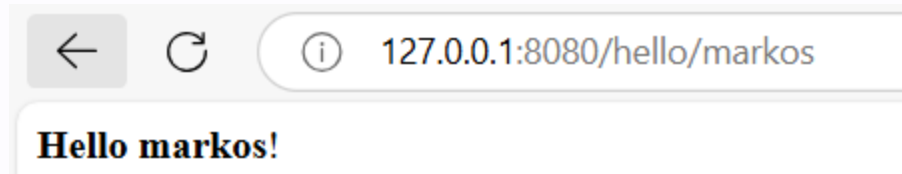
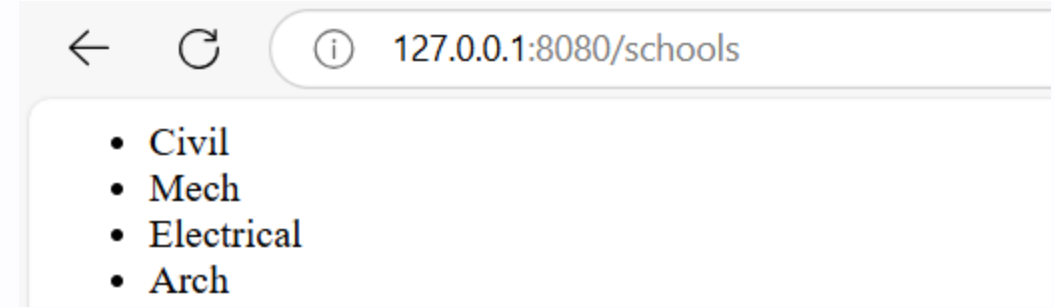
```
from bottle import route, run, template

@route('/hello/<name>')
def get_hello(name):
    return template('<b>Hello {{name}}</b>!', name=name)
# http://127.0.0.1:8080/hello/markos

@route('/schools')
def get_schools():
    schools = ['Civil', 'Mech', 'Electrical', 'Arch']
    html = '<ul>'
    for school in schools:
        html += '<li>' + school + '</li>'
    html += '</ul>'
    return html

@route('/file2/<name>')
def show_template(name):
    university = 'NTUA'
    schools = ['Civil', 'Mech', 'Electrical', 'Arch']
    return template('file2.html', name=name, school='civil', university=university, schools=schools)
# http://127.0.0.1:8080/file2/markos

run(host='localhost', port=8080)
```



Python Example 1

Υπολογισμός οπλισμού δοκού

Το μηχανικό ποσοστό οπλισμού ω δοκού δίνεται σε σχέση με την ανηγμένη ροπή

$$\mu_{sd}$$

από τη σχέση

$$\omega = 0.84 \left(1 - \sqrt{1 - 2.4\mu_{sd}} \right)$$

Να συνταχθεί πρόγραμμα σε python το οποίο να εμφανίζει σε ιστοσελίδα για $\omega=0, 0.01, \dots, 0.35$:

Πίνακα οπλισμού

$$\omega / \mu_{sd}$$

Προγραμματισμός και Θέματα Τεχνητής Νοημοσύνης
και το αντίστοιχο διάγραμμα

Python Example 1

Λύση

Αρχικά θα υπολογιστούν οι αριθμητικές τιμές των ω και μ_{sd} . Στη συνέχεια σε ένα πίνακα με τρείς στήλες θα τοποθετηθεί στην πρώτη στήλη η **Ανηγμένη ροπή** στη δεύτερη στήλη το **Ποσοστό οπλισμού** και στη τρίτη στήλη θα γίνει συγχώνευση κελιών και θα εμφανιστεί το διάγραμμα. Το διάγραμμα θα αποθηκευτεί σε αρχείο temp.jpg.

Σημείωση: βιβλιοθήκη bottle επιτρέπει άνοιγμα αρχείων μόνο σε ένα φάκελλο που ορίζεται μέσω της συνάρτησης static_file().

Κώδικας Python

```
from pathlib import Path
import numpy as np
from matplotlib import pyplot as plt
import matplotlib
from bottle import route, run, static_file, template

@route('/static/<filename>')
def server_static(filename):
    return static_file(filename, root='./static')

@route('/')
def show_template():
    mu = np.arange(0, 0.36, 0.01)
    w = 0.84*(1-np.sqrt(1-2.4*mu))
    plt.figure()
    plt.plot(mu, w)
    plt.xlabel("Ανηγμένη ροπή")
    plt.ylabel("Ποσοστό οπλισμού")
    plt.savefig("static/temp.jpg")
    return template('example1.html', muData=mu, wData=w)

run(host='localhost', port=8080)
```

Template, αρχείο HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Οπλισμός δοκού</title>
  <style>
    table, td, th{
      border:1px solid black;
      text-align: left;
    }

    table {
      border-collapse: collapse;
      width: 50%;
    }

    th{
      text-align: center;
    }
  </style>
</head>
```

```
<body>
  <table>
    <thead>
      <tr>
        <th>Ανηγμένη ροπή</th>
        <th>Ποσοστό οπλισμού</th>
        <th>Διάγραμμα</th>
      </tr>
    </thead>
    <tbody>
      % for i in range(len(muData)):
        % if i==0:
          <tr>
            <td>{{ '{0:.3f}'.format(muData[i]) }}</td>
            <td>{{ '{0:.3f}'.format(wData[i]) }}</td>
            <td rowspan="{{ len(muData) }}">
              
            </td>
          </tr>
        % else:
          <tr>
            <td>{{ '{0:.3f}'.format(muData[i]) }}</td>
            <td>{{ '{0:.3f}'.format(wData[i]) }}</td>
          </tr>
        %end
      % end
    </tbody>
  </table>
</body>
```