Dynamic Web Pages

Forms in Python Scripts

- Python code to write HTML
- default input values

Web Interface for Determinant

- dynamic interactive forms
- asking for a matrix
- computing the determinant

Exercises

MCS 275 Lecture 20
Programming Tools and File Management
Jan Verschelde, 29 February 2008

Forms in Python Scripts
Python code to write HTML
default input values
Web Interface for Determinant
dynamic interactive forms
asking for a matrix
computing the determinant
Exercises

Dynamic Web Pages

Forms in Python Scripts
- Python code to write HTML
- default input values

Web Interface for Determinant
- dynamic interactive forms
- asking for a matrix
- computing the determinant

Exercises

Forms in Python Scripts

writing HTML

What we did before:

1. form element in HTML page triggers action, example: submit user name and password.
2. Action triggered by submit button is defined by a Python script.

Advantage/disadvantage:

+ Separate HTML layout and actions.
– Submit brings up a new page each time.

Forms in Python Scripts

writing HTML

What we did before:

1. form element in HTML page triggers action,
   example: submit user name and password.
2. Action triggered by submit button
   is defined by a Python script.

Advantage/disadvantage:

+ Separate HTML layout and actions.
– Submit brings up a new page each time.

Forms in Python Scripts

writing HTML

What we did before:

1. form element in HTML page triggers action,
   example: submit user name and password.
2. Action triggered by submit button
   is defined by a Python script.

Advantage/disadvantage:

+ Separate HTML layout and actions.
– Submit brings up a new page each time.

Forms in Python Scripts

writing HTML

What we did before:

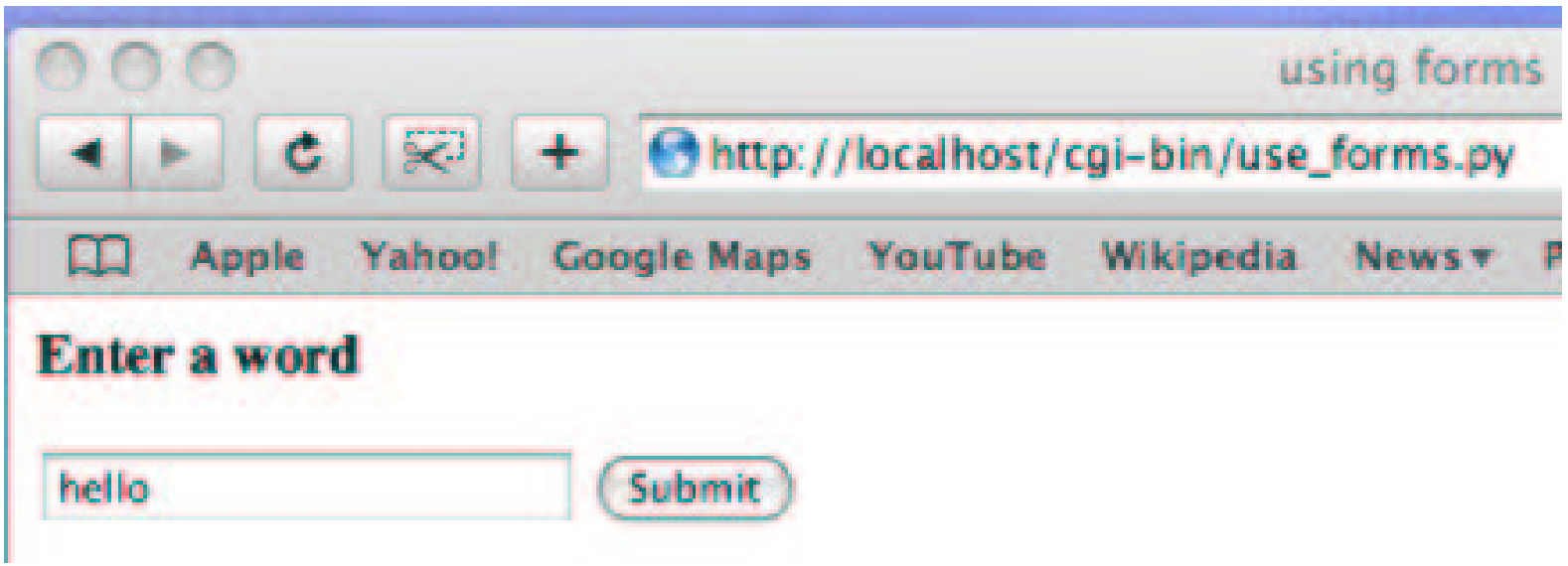
1. form element in HTML page triggers action,
   example: submit user name and password.
2. Action triggered by submit button
   is defined by a Python script.

Advantage/disadvantage:

- \+ Separate HTML layout and actions.
- – Submit brings up a new page each time.

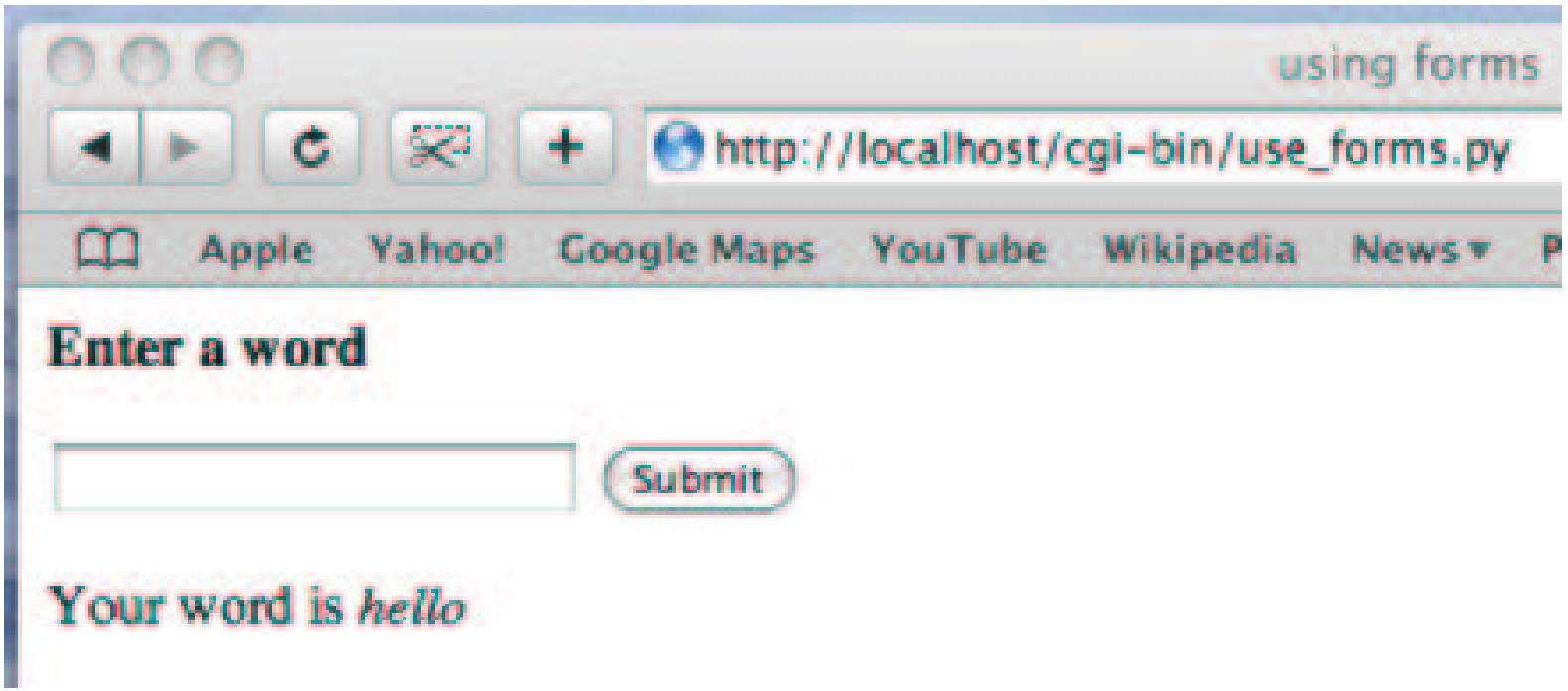
Asking for a Word

running `use_forms.py`



Confirming the Word

running `use_forms.py`



Printing a Header

start of `use_forms.py`

```
#!/Library/Frameworks/.../bin/python
import cgi

def PrintHeader(title):
   """
   writes title and header of page
   """
   print """Content-type: text/html

<html>
<head>
<title>%s</title>
</head>
<body>""" % title

PrintHeader("using forms")
```

Printing a Header

start of `use_forms.py`

```
#!/Library/Frameworks/.../bin/python
import cgi

def PrintHeader(title):
    """
    writes title and header of page
    """
    print """Content-type: text/html

<html>
<head>
<title>%s</title>
</head>
<body>""" % title

PrintHeader("using forms")
```

Printing a Header

start of `use_forms.py`

```
#!/Library/Frameworks/.../bin/python
import cgi

def PrintHeader(title):
    """
    writes title and header of page
    """
    print """Content-type: text/html

<html>
<head>
<title>%s</title>
</head>
<body>""" % title

PrintHeader("using forms")
```

Printing a Header

start of `use_forms.py`

```
#!/Library/Frameworks/.../bin/python
import cgi

def PrintHeader(title):
    """
    writes title and header of page
    """
    print """Content-type: text/html

<html>
<head>
<title>%s</title>
</head>
<body>""" % title

PrintHeader("using forms")
```

printing a form

script use_forms.py continued

```
print """<b>Enter a word</b>
   <form method = "post" action = "use_forms.py">
   <p>
     <input type = "text" name = "word">
     <input type = "submit">
   </p>
   </form>"""

form = cgi.FieldStorage()
if form.has_key("word"):
   print """<p>Your word is <em>%s</em></p>
   """ % form["word"].value

print "</body></html>\n"
```

printing a form

script use_forms.py continued

```
print """<b>Enter a word</b>
   <form method = "post" action = "use_forms.py">
   <p>
     <input type = "text" name = "word">
     <input type = "submit">
   </p>
   </form>"""

form = cgi.FieldStorage()
if form.has_key("word"):
   print """<p>Your word is <em>%s</em></p>
   """ % form["word"].value

print "</body></html>\n"
```


Forms in Python Scripts
Python code to write HTML
default input values
Web Interface for Determinant
dynamic interactive forms
asking for a matrix
computing the determinant
Exercises

printing a form

script use_forms.py continued

```
print """<b>Enter a word</b>
   <form method = "post" action = "use_forms.py">
   <p>
     <input type = "text" name = "word">
     <input type = "submit">
   </p>
   </form>"""

form = cgi.FieldStorage()
if form.has_key("word"):
   print """<p>Your word is <em>%s</em></p>
   """ % form["word"].value

print "</body></html>\n"
```


Forms in Python Scripts
Python code to write HTML
default input values
Web Interface for Determinant
dynamic interactive forms
asking for a matrix
computing the determinant
Exercises

Dynamic Web Pages

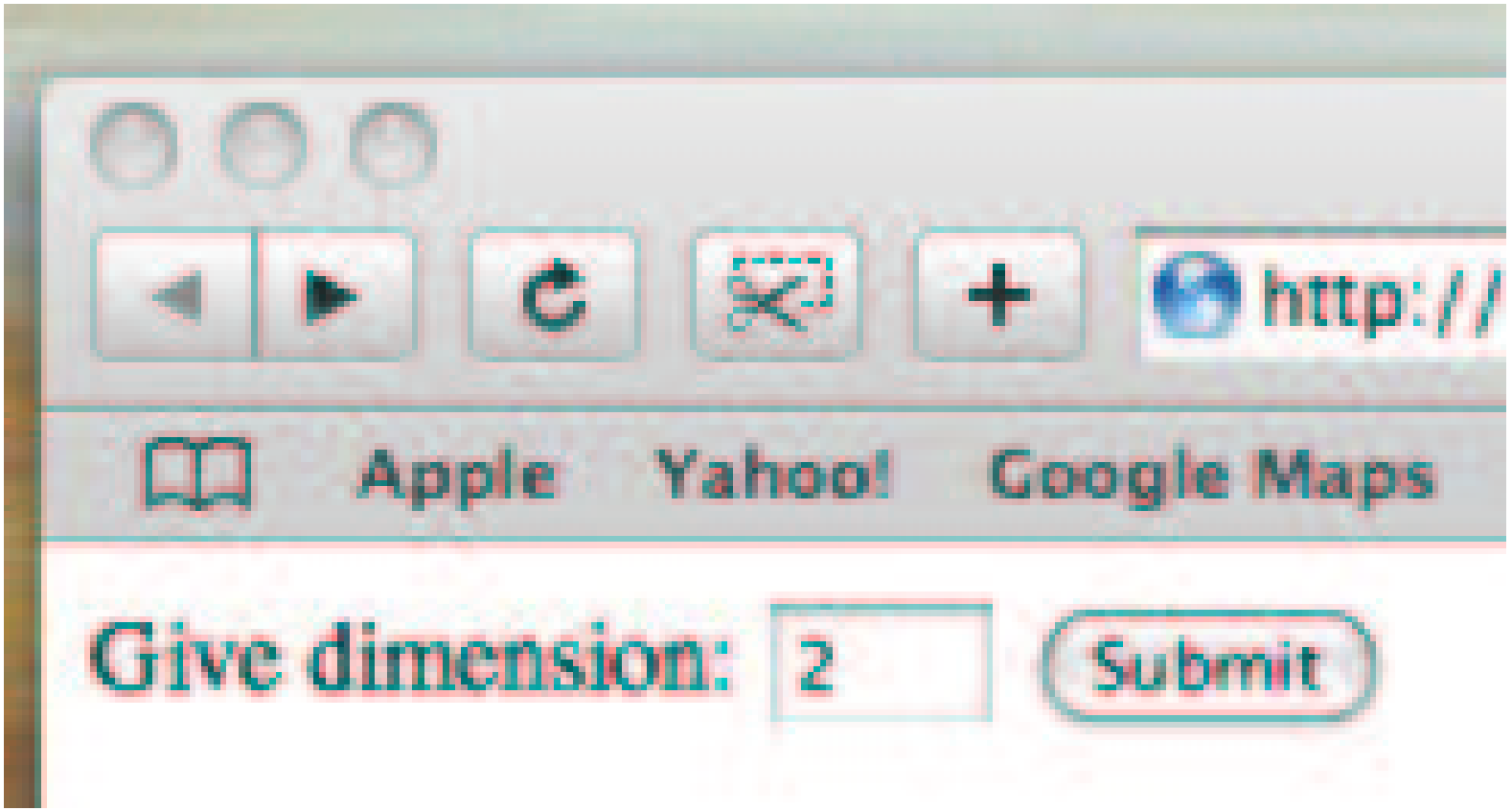
Forms in Python Scripts
- Python code to write HTML
- default input values

Web Interface for Determinant
- dynamic interactive forms
- asking for a matrix
- computing the determinant

Exercises

Default Values

in input fields



The "2" is filled in already, is default value.

Forms in Python Scripts
Python code to write HTML
default input values
Web Interface for Determinant
dynamic interactive forms
asking for a matrix
computing the determinant
Exercises

Forms with Defaults

in input fields

```
print """<form method = "post"
                 action = "web_det.py">
<p>
  Give dimension:
  <input type = "text" name = "dim"
         size = 4 value = 2>

  <input type = "submit">
</p>
</form>"""
```

Forms with Defaults

in input fields

```
print """<form method = "post"
                 action = "web_det.py">
<p>
  Give dimension:
  <input type = "text" name = "dim"
         size = 4 value = 2>

  <input type = "submit">
</p>
</form>"""
```

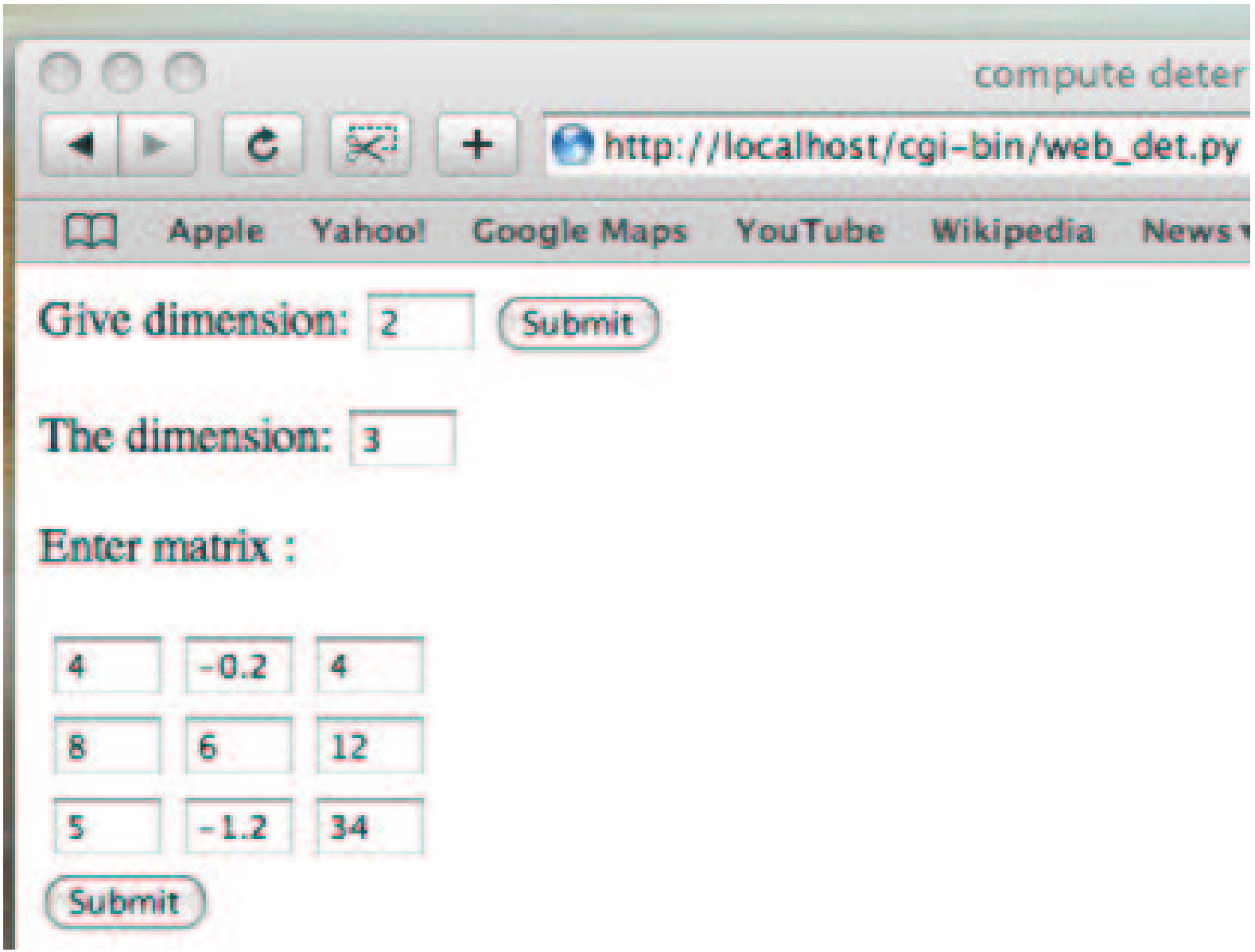
Dynamic Web Pages

Forms in Python Scripts
- Python code to write HTML
- default input values

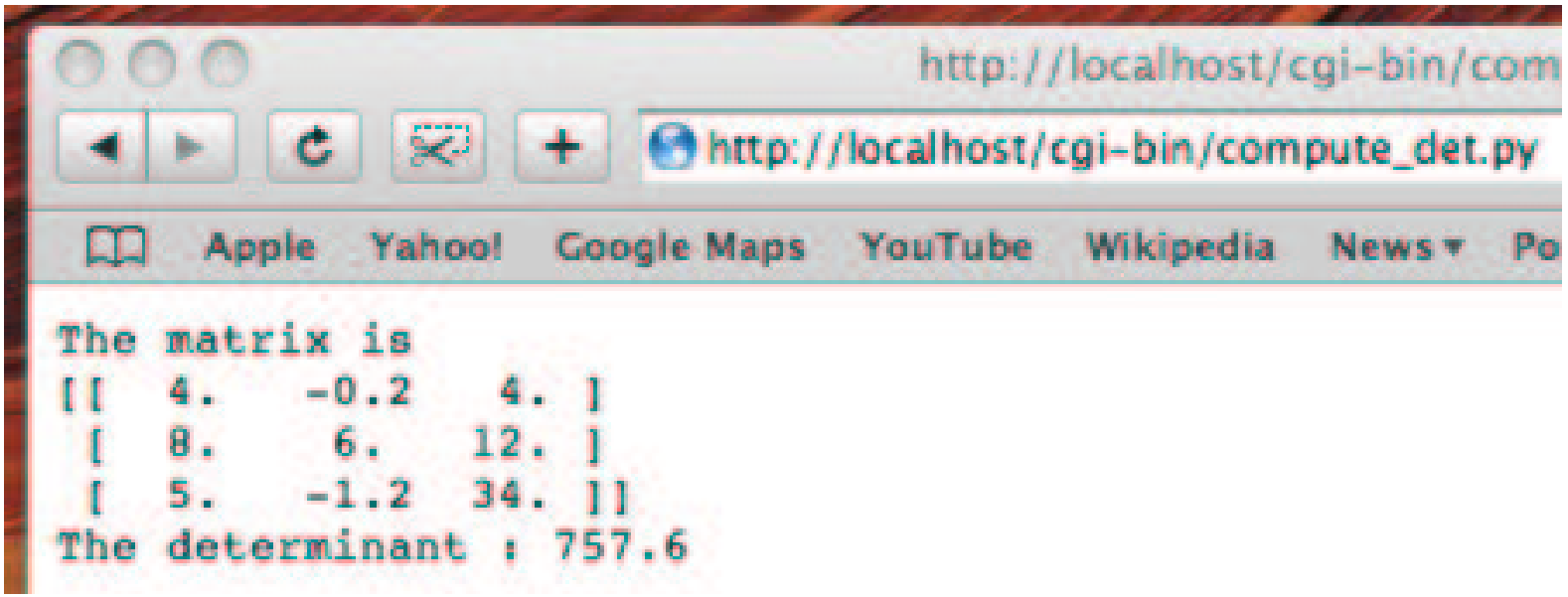
Web Interface for Determinant
- dynamic interactive forms
- asking for a matrix
- computing the determinant

Exercises

Asking for a Matrix



Computing the Determinant



Forms in Python Scripts
Python code to write HTML
default input values
Web Interface for Determinant
dynamic interactive forms
asking for a matrix
computing the determinant
Exercises

Script for Web Determinant

The script is a sequence of two forms:

1. we ask for *the dimension* of the matrix
   The dimension is then used to generate a table of input elements to give the entries of the matrix.

2. we ask for *the entries* of the matrix
   The second form is processed by the same script.
   The action of this form brings up a new page with the matrix and the determinant.

The dimension is passed from one form to the other ***through an input element***.

Script for Web Determinant

The script is a sequence of two forms:

1. we ask for *the dimension* of the matrix
   The dimension is then used to generate a table of input elements to give the entries of the matrix.

2. we ask for *the entries* of the matrix
   The second form is processed by the same script.
   The action of this form brings up a new page with the matrix and the determinant.

The dimension is passed from one form to the other ***through an input element***.

Script for Web Determinant

The script is a sequence of two forms:

1. we ask for *the dimension* of the matrix
   The dimension is then used to generate a table of input elements to give the entries of the matrix.

2. we ask for *the entries* of the matrix
   The second form is processed by the same script.
   The action of this form brings up a new page with the matrix and the determinant.

The dimension is passed from one form to the other ***through an input element***.

Script for Web Determinant

The script is a sequence of two forms:

1. we ask for *the dimension* of the matrix
   The dimension is then used to generate a table of input elements to give the entries of the matrix.

2. we ask for *the entries* of the matrix
   The second form is processed by the same script.
   The action of this form brings up a new page with the matrix and the determinant.

The dimension is passed from one form to the other ***through an input element***.

Script for Web Determinant

The script is a sequence of two forms:

1. we ask for *the dimension* of the matrix
   The dimension is then used to generate a table of input elements to give the entries of the matrix.

2. we ask for *the entries* of the matrix
   The second form is processed by the same script.
   The action of this form brings up a new page with the matrix and the determinant.

The dimension is passed from one form to the other ***through an input element***.

The Main Function

of `web_det.py`

```
def main():
    """
    web interface to compute a determinant
    """
    PrintHeader("compute determinant")
    AskDimension()
    form = cgi.FieldStorage()
    if form.has_key("dim"):
        n = int(form["dim"].value)
        AskMatrix(n)

    print "</body></html>\n"

main()
```

The Main Function

of `web_det.py`

```
def main():
    """
    web interface to compute a determinant
    """
    PrintHeader("compute determinant")
    AskDimension()
    form = cgi.FieldStorage()
    if form.has_key("dim"):
        n = int(form["dim"].value)
        AskMatrix(n)

    print "</body></html>\n"

main()
```

The Main Function

of `web_det.py`

```
def main():
    """
    web interface to compute a determinant
    """
    PrintHeader("compute determinant")
    AskDimension()
    form = cgi.FieldStorage()
    if form.has_key("dim"):
        n = int(form["dim"].value)
        AskMatrix(n)

    print "</body></html>\n"

main()
```

Dynamic Web Pages

Forms in Python Scripts
- Python code to write HTML
- default input values

Web Interface for Determinant
- dynamic interactive forms
- asking for a matrix
- computing the determinant

Exercises

The Function AskMatrix()

```
def AskMatrix(n):
    """
    form to enter an n-by-n  matrix
    """
    print """<form method = "post"
                    action = "compute_det.py">"""
    print """The dimension:
       <input type = "text" name = "dim"
              size = 4 value = %d>""" % n
```

Observe how the value of the dimension is set in the input element dim of the form.

The Function AskMatrix()

```
def AskMatrix(n):
    """
    form to enter an n-by-n  matrix
    """
    print """<form method = "post"
                    action = "compute_det.py">"""
    print """The dimension:
       <input type = "text" name = "dim"
              size = 4 value = %d>""" % n
```

Observe how the value of the dimension is set in the input element `dim` of the form.

AskMatrix() continued

```
print "<p>Enter matrix :</p>"
print "<table>"
for i in range(0,n):
   print "<tr>"
   for j in range(0,n):
      print """<td><input type = "text"
      name = %d,%d size = 4></td>""" % (i,j)
print "</table>"
print """<input type = "submit">"""
print "</form>"
```

The names of the entries are strings i, j.

AskMatrix() continued

```
print "<p>Enter matrix :</p>"
print "<table>"
for i in range(0,n):
   print "<tr>"
   for j in range(0,n):
      print """<td><input type = "text"
      name = %d,%d size = 4></td>""" % (i,j)
print "</table>"
print """<input type = "submit">"""
print "</form>"
```

The names of the entries are strings i, j.

AskMatrix() continued

```
print "<p>Enter matrix :</p>"
print "<table>"
for i in range(0,n):
   print "<tr>"
   for j in range(0,n):
      print """<td><input type = "text"
      name = %d,%d size = 4></td>""" % (i,j)
print "</table>"
print """<input type = "submit">"""
print "</form>"
```

The names of the entries are strings i, j.

Dynamic Web Pages

Forms in Python Scripts
- Python code to write HTML
- default input values

Web Interface for Determinant
- dynamic interactive forms
- asking for a matrix
- computing the determinant

Exercises

Beginning of `compute_det.py`

```
#!/Library/Frameworks/.../bin/python
import cgi

from numpy import matrix, zeros
from numpy.linalg import det
```

The Main Function

in `compute_det`

```
def main():
    """
    web interface to compute a determinant
    """
    print "Content-type: text/plain\n"
    form = cgi.FieldStorage()
    if form.has_key("dim"):
        n = int(form["dim"].value)
        d = Determinant(form,n)
        print "The determinant :", d
```

The Main Function

in `compute_det`

```
def main():
    """
    web interface to compute a determinant
    """
    print "Content-type: text/plain\n"
    form = cgi.FieldStorage()
    if form.has_key("dim"):
        n = int(form["dim"].value)
        d = Determinant(form,n)
        print "The determinant :", d
```

The Main Function

in `compute_det`

```
def main():
    """
    web interface to compute a determinant
    """
    print "Content-type: text/plain\n"
    form = cgi.FieldStorage()
    if form.has_key("dim"):
        n = int(form["dim"].value)
        d = Determinant(form,n)
        print "The determinant :", d
```

The Determinant() Function

```
def Determinant(form,n):
    """
    returns the determinant of the matrix
    available in the form
    """
    A = zeros((n,n),float)
    for i in range(0,n):
        for j in range(0,n):
            info = "%d,%d" % (i,j)
            if form.has_key(info):
                A[i,j] = float(form[info].value)
    print "The matrix is"
    print A
    return det(A)
```

The Determinant() Function

```
def Determinant(form,n):
    """
    returns the determinant of the matrix
    available in the form
    """
    A = zeros((n,n),float)
    for i in range(0,n):
        for j in range(0,n):
            info = "%d,%d" % (i,j)
            if form.has_key(info):
                A[i,j] = float(form[info].value)
    print "The matrix is"
    print A
    return det(A)
```

The Determinant() Function

```
def Determinant(form,n):
    """
    returns the determinant of the matrix
    available in the form
    """
    A = zeros((n,n),float)
    for i in range(0,n):
        for j in range(0,n):
            info = "%d,%d" % (i,j)
            if form.has_key(info):
                A[i,j] = float(form[info].value)
    print "The matrix is"
    print A
    return det(A)
```

The Determinant() Function

```
def Determinant(form,n):
    """
    returns the determinant of the matrix
    available in the form
    """
    A = zeros((n,n),float)
    for i in range(0,n):
        for j in range(0,n):
            info = "%d,%d" % (i,j)
            if form.has_key(info):
                A[i,j] = float(form[info].value)
    print "The matrix is"
    print A
    return det(A)
```

Exercises

1. Modify the `use_forms.py` so that it places by default one word in the input element, *with all vowels removed*. Ask the user to supply the missing vowels. In reaction to the submit, compare the word of the user with the original word and give feedback.
2. Write a Python script `web_sum.py` to sum n numbers. Like in `web_det.py`, first the user is asked to supply n and then n input elements have to be filled out before another script `compute_sum.py` can compute the sum.
3. The current version of `web_det.py` takes undefined entries to equal zero. Change it so the user is asked to return and provide the missing entries.
4. Make a recursive form. The input element asks the user for a number. With each submit, the number is doubled.