

Script language: Python

CGI



Cédric Saule

Technische Fakultät
Universität Bielefeld

13. Februar 2016

Python

—

CGI

An overview

- WWW, Web-Server
- Script CGI
- Parameters delivery
- Web Form
- `cgi`, `cgitb`

Data exchange in WWW

- Addressing by *Uniform Resource Locator* (URL).
- Requests and answers are transported via HTTP.
- HTTP message consists in *header* and *body*.
- *body* can be empty.
- Answers can be dynamically generated.

A small web-server

- Use a tutorial directory.
- Web-Server start:

```
/vol/lehre/python/2016/run_mini_httpd.sh
```

- Caution when performing a directory called Python2016_WWW created in your HOME
- Server and port number are outputs from the script.
- To stop the servers.

```
kill `cat $HOME/Python2016_WWW/mini_http.pid`
```

Script CGI

- Informal Standard: *Common Gateway Interface*
- The web server executes an external programs.
- The program outputs are returned in response.
- Extrem potential danger :
 - Execution of (false ?) code
 - Execution of arbitrary code.
- Examin your own code carefully.

Script CGI

- Informal Standard: *Common Gateway Interface*
- The web server executes an external programs.
- The program outputs are returned in response.
- Extrem potential danger :
 - Execution of (false ?) code
 - Execution of arbitrary code.
- Examin your own code carefully.
- **No quick hacks!!!**

Script CGI, cont.

- Building an answer: *header*, blank line, *body*

```
Content-Type: text/html
```

```
<html>
  <head>
  ...
```

- Script Python helloworld.cgi:

```
#!/usr/bin/env python
print "Content-Type: \text/plain\n";
print "Hello \World!\n";
```

- `http://localhost:XXXXX/helloworld.cgi`
- Better : The Python interpreter does not directly indicate `/usr/bin/env`

Task

Write a CGI script that returns the current time back.

Note: Use the Python module time to determine the time!

Redirections

- The script refers to another ressource.
- A new URL is admitted in the header to return
`print "Location: http://www.Uni-Bielefeld.DE\n";`
- Body can be empty.
- Redirect pursue:

```
GET -S http://localhost:XXXX/redirect.cgi | more
GET http://localhost::XXXX/redirect.cgi --> 302 Found
GET http://www.Uni-Bielefeld.DE --> 200 OK
<!DOCTYPE html PUBLIC ...
```

- 302 and 200 HTTP status codes.

Passing parameters

- Requests can contain data (Google, Amazon, Wikipedia, ...).
- Key-value pairs.
- Data transfers from client to web-server:
 - GET: Parameters will be coded in URL
`http://google.de/search?q=boggit&ie=UTF-8&oe=UTF-8&...`
 - POST: Data will be carried in the body.
- Data transfers from web transfer to a CGI script :
 - Environment
 - STDIN

Insertion: Environment

- Environment variables.
- View to a shell command env.
- Access in Python:

```
import os

print os.environ
print "User: %s" % os.environ['USER'];
print "Search path: %s" % os.environ['PATH'];
```

- Write possible access.
- Environment bounds to process.

Task

- Write a CGI script `showenv.cgi` that indicates the entire CGI process environment. What is the environment, if `showenv.cgi` uses the command line?
- Overhang the URL parameters:
`.../showenv.cgi?query=boggit` What happens to the environment?
- Given several parameters. How could you list these parameters for a further processing with Python?

Web forms

- Create a file form.html and open it with the web browser.

```
<html>
  <head><title>Form</title></head>
  <body>
    <form action="showenv.cgi" method="GET">
      Query: <input type="TEXT" name="query"> <br/>
      <input type="SUBMIT"/><input type="RESET"/>
    </form>
  </body>
</html>
```

- Data to be passed to showenv.cgi.
- Method GET, i.e. parameters appear in the URL.
- Test input: „Ernie & Bert“.
- Replace in form.html method GET with POST.

Evaluation of web forms

- Recurring tasks/problems:
 - Data can be transferred in different ways.
 - Special characters must be (de)coded
 - Answers must correct the contain of header and body.
 - Entries must be carefully evaluated.
- Universal helper: `cgi`, `cgitb`.

Parameters delivery

- Script `showparams.cgi`:

```
#!/usr/bin/env python
import cgi, cgitb
cgitb.enable()

print "Content-Type: \text/plain\n"
params = cgi.FieldStorage()

for name in params.keys():
    print "%s: %s" % (name, params[name].value)
```

- Test URL `.../showparams.cgi?query=boggit&answers=10`
- Enter `showparams.cgi` as an action in `form.html`
- Tests GET and POST as method.