

Skriptsprachen: Python

CGI

Jan Krüger, Alexander Sczyrba

Technische Fakultät
Universität Bielefeld

7. Oktober 2016



Universität Bielefeld

Python

–

CGI

Übersicht

- WWW, Web-Server
- CGI-Skripte
- Parameterübergabe
- Web-Formulare
- `cgi`, `cgitb`

Datenaustausch im WWW

- Adressierung durch *Uniform Resource Locator* (URL)
- Anfrage und Antwort werden per HTTP transportiert
- HTTP-Nachricht besteht aus *header* und *body*
- *body* kann leer sein
- Antwort kann dynamisch generiert werden

ein kleiner Web-Server

- Übungsverzeichnis verwenden
- Web-Server starten:

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

- Ein Verzeichnis Python2016_WWW wird im HOME angelegt
- Server ist unter `http://localhost:8000` erreichbar
- Zum Beenden des Servers..

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

CGI-Skripte

- informeller Standard: *Common Gateway Interface*
- Web-Server führt externes Programm aus
- Ausgabe des Programms wird als Antwort zurückgeschickt
- enormes Gefahrenpotential:
 - Ausführung von (fehlerhaftem?) Code
 - Ausführung von beliebigem Code
- eigenen Code sorgfältig prüfen

CGI-Skripte

- informeller Standard: *Common Gateway Interface*
- Web-Server führt externes Programm aus
- Ausgabe des Programms wird als Antwort zurückgeschickt
- enormes Gefahrenpotential:
 - Ausführung von (fehlerhaftem?) Code
 - Ausführung von beliebigem Code
- eigenen Code sorgfältig prüfen
- **Keine schnellen Hacks!!**

CGI-Skripte, cont.

- Aufbau einer Antwort: *header*, Leerzeile, *body*

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

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

- Python-Skript `helloworld.cgi`:

```
#!/usr/bin/python3  
print "Content-Type: \u0020text/plain\u0020\u0026amp;#x000A;";  
print "Hello \u0020World!\u0020\u0026amp;#x000A;";
```

- `http://localhost:8000/cgi-bin/helloworld.cgi`
- `besser` : Python-Interpreter direkt angeben, nicht mit `/usr/bin/env`

Aufgabe

Schreibe ein CGI-Skript, das die aktuelle Uhrzeit zurückliefert.

Hinweis: Benutze das Python Modul time um die Uhrzeit zu ermitteln!

Parameterübergabe

- Anfrage kann Daten enthalten (Google, Amazon, Wikipedia, ...)
- Schlüssel-Wert-Paare
- Datentransfer vom Client zum Web-Server:
 - GET: Parameter werden in der URL kodiert
`http://google.de/search?q=boggit&ie=UTF-8&oe=UTF-8&...`
 - POST: Daten werden im body transportiert
- Datentransfer vom Web-Server zum CGI-Skript:
 - Environment
 - STDIN

Einschub: Environment

- Environment-Variablen
- mit Shell-Kommando `env` ansehen
- Zugriff in Python:

```
import os

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

- schreibender Zugriff möglich
- Environment an Prozess gebunden

Aufgabe

- Schreibe ein CGI-Skript `showenv.cgi`, das das gesamte Environment des CGI-Prozesses anzeigt. Wie sieht das Environment aus, wenn Du `showenv.cgi` auf der Kommandozeile aufrufst?
- Hänge an den URL Parameter an:
`.../showenv.cgi?query=boggit` Wie verändert sich das Environment?
- Gib mehrere Parameter an. Wie könntest Du diese Parameterliste mit Python weiterverarbeiten?

Web-Formulare

- Erstelle ein Datei `form.html` und öffne diese mit dem Web-Browser

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

- Daten werden an `showenv.cgi` übergeben
- Methode GET, d.h. Parameter erscheinen im URL
- Eingabe testen: „Ernie & Bert“
- ersetze in `form.html` Methode GET durch POST

Auswertung von Web-Formularen

- immer wiederkehrende Aufgaben/Probleme:
 - Daten können auf verschiedene Arten übertragen werden
 - spezielle Zeichen müssen (de)kodiert werden
 - Antwort muß korrekten header und body enthalten
 - Eingaben müssen „vorsichtig“ ausgewertet werden
- universeller Helfer: cgi, cgitb

Parameterübergabe

- Skript `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)
```

- probiere URL `.../showparams.cgi?query=boggit&answers=10`
- trage `showparams.cgi` als action in `form.html` ein
- probiere GET und POST als method