



Grundlagen Internet-Technologien

INF3171

Serverseitige Web-Programmierung
mit CGI, Teil II: Perl im Web

Version 1.0

03.06.2013





aktuelles

2015 nähert sich der jährliche Internetverkehr dem Zettabyte-Schwelle

03.06.2012 11:16

2015 nähert sich der jährliche Internetverkehr dem Zettabyte-Schwellenwert

Die Internetnutzer werden in den kommenden fünf Jahren für das Vierfache des heutigen Datenverkehrs sorgen. 2015 sollen so rund 966 Petabyte (oder fast 1 Zettabyte) Daten durch die Leitungen gehen. Das **prognostiziert [1]** der Netzwerkausrüster **Cisco[2]**. Vor allem die Netzanbindung von Tablets, Smartphones, Fernsehern und anderen nicht-PCs sorgt für mehr Verkehr. 2015 sollen kabellos mit dem Netz verbundene Geräte für mehr Internetauslastung sorgen als per Kabel angeschlossene und die Zahl der mit dem Internet verbundenen Geräte soll dem doppelten der Weltbevölkerung entsprechen.

Cisco

Den Angaben zufolge ist das Datenaufkommen in den vergangenen fünf Jahren um das Achtfache gewachsen und wird 2015 ein Aufkommen von 7,3 Petabyte alle fünf Minuten generieren. Das entspräche dem Speichervolumen für alle jemals gedrehten Filme. In Westeuropa allein prognostiziert Cisco ein Datenaufkommen von 19 Exabytes pro Monat, was eine Wachstumsrate von 32 Prozent bedeutet. Den größten Zuwachs sieht Cisco allerdings für Lateinamerika voraus. Dort soll das Datenaufkommen um 50 Prozent zwischen 2010 und 2015 wachsen. Cisco erhebt die Daten im Rahmen seines **Visual Networking Index[3]** zum weltweiten Datenverkehr. (**ohu[4]**)

IP Traffic, 2010-2015	2010	2011	2012	2013	2014	2015	CAGR (2010-2015)
By Type (PB per Month)							
Fixed Internet	14,888	20,880	27,024	33,879	40,200	48,384	23%
Managed IP	4,588	6,835	9,214	11,592	15,168	18,888	24%
Mobile Data	237	545	1,183	2,188	3,806	6,254	85%
By Region (PB per Month)							
Continental Europe	8,227	10,100	12,168	14,460	16,971	19,788	24%
Rest of Europe	3,050	4,086	5,211	6,392	7,697	9,142	24%
By Geography (PB per Month)							
North America	6,888	8,947	11,278	13,718	16,348	19,174	24%
Western Europe	4,778	6,488	8,218	10,174	12,168	14,388	23%
Asia Pacific	5,368	7,217	9,247	11,541	14,160	17,100	24%
Japan	1,414	1,822	2,246	2,696	3,174	3,702	21%
Latin America	1,068	1,462	1,881	2,326	2,806	3,317	24%
Central and Eastern Europe	108	1,284	1,413	1,596	1,752	1,712	24%
Middle East and Africa	202	300	400	502	610	724	24%
Total IP per Month	25,101	34,225	42,863	51,627	61,207	71,886	23%

Source: Cisco VNI, 2011

Entwicklung des weltweiten Internetverkehrs von 2010 bis 2015
Bild:

URL dieses Artikels:
<http://www.heise.de/newsticker/meldung/2015-naehrt-sich-der-jaehrliche-Internetverkehr-dem-Zettabyte-Schwellenwert-1589635.html>

Links in diesem Artikel:
 [1] http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/VNI_Hyperconnectivity_WP.html
 [2] http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white_paper_c11-481360_ns827_Networking_Solutions_White_Paper.html
 [3] http://www.cisco.com/en/US/netsol/ns827/networking_solutions_sub_solution.html
 [4] <mailto:ohu@heise.de>



Situation

- bisher: Perl ist einfache und nützliche (Systemadministration) Programmiersprache
- nun: wir nutzen Perl als **CGI-Sprache**
- Fragen:
 - keine direkt ausführbaren Codes mit PERL
 - was ist alles zu tun für CGI mit PERL
 - **Formularverarbeitung** mit PERL
 - **Generation von XHTML** mit PERL





cgi Essentials

- cgi-Programm muss ausführbar sein
- liegt im Unterordner cgi-bin
- Ausgabe beginnt mit:

```
Content-type: text/html
\n
```

- Zugriff: URL beginnt mit „cgi-bin“





Wie wird Perl ausführbar?

- ...durch die Shebang-Zeile:

erste (!!!) Zeile im Scriptprogramm ist ein besonderer Kommentar:

```
#!<path_to_perl/perl -w
```

- zusätzlich auf Unix x-bit setzen:


```
chmod +x <perlscript>
```

 - und schon ist's ausführbar!
 - gilt für alle Scriptsprachen





erstes Beispiel

```

Perl - InternetTechnologien Perl/helloperl.cgi.pl - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
hello.pl printenv.pl liste.pl assArray.pl kreis.pl ifElse.pl schleifen.pl javacgi.pl helloperl.cgi.pl x2
1 #!"C:/Program Files/Perl/bin/perl.exe" -w
2 #
3 # Grundlagen Internet-Technologien
4 #
5 # Beispiel fuer einfachstes PERL-CGI
6
7 $phrase = "Hello World!<BR>I like PERL and<BR>\"Grundlagen Internet-Technologien\"";
8
9 print("Content-type: text/html\n\n<HTML><HEAD><TITLE>Internet-Technologien:Hello World als PERL-cgi</TITLE>");
10 print("<link rel=\"stylesheet\" type=\"text/css\" href=\"/css/webkompodium.css\">");
11 print("<link rel=\"shortcut icon\" href=\"/css/favicon.ico\"></HEAD>");
12 print("<BODY><HR><CENTER><H2>$phrase</H2><HR>");
13 print("<IMG src=\"/images/lcamel.gif\"><HR></CENTER></BODY></HTML>");
Writable Insert 1:1

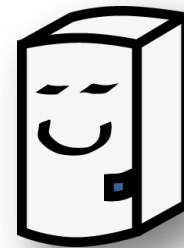
```





nächste Schritte

- Testen des Scripts lokal ("ohne Web")
 - Script muss auch ohne Web fehlerfrei laufen
- kopieren nach cgi-bin
- x-bit setzen (Unix/Linux)
- Aufruf aus Client-Browser







viel besser: Here-Documents

- direkt im Perl-Code "raw-formatierte" HTML-Ausgabe:

```
print <<endezeichen
HTML-Text
endezeichen
;
# weiter mit Perl
```

- Variablen werden ersetzt, wenn endezeichen oder "endezeichen", nicht bei 'endezeichen'





Perl - InternetTechnologien Perl/helloperlcgi2.pl - Eclipse

File Edit Source Refactor Navigate Search Project Run Window Help

hello.pl printenv.pl liste.pl assArray.pl kreis.pl javacgi.pl helloperlcgi.pl helloperlcgi2.pl

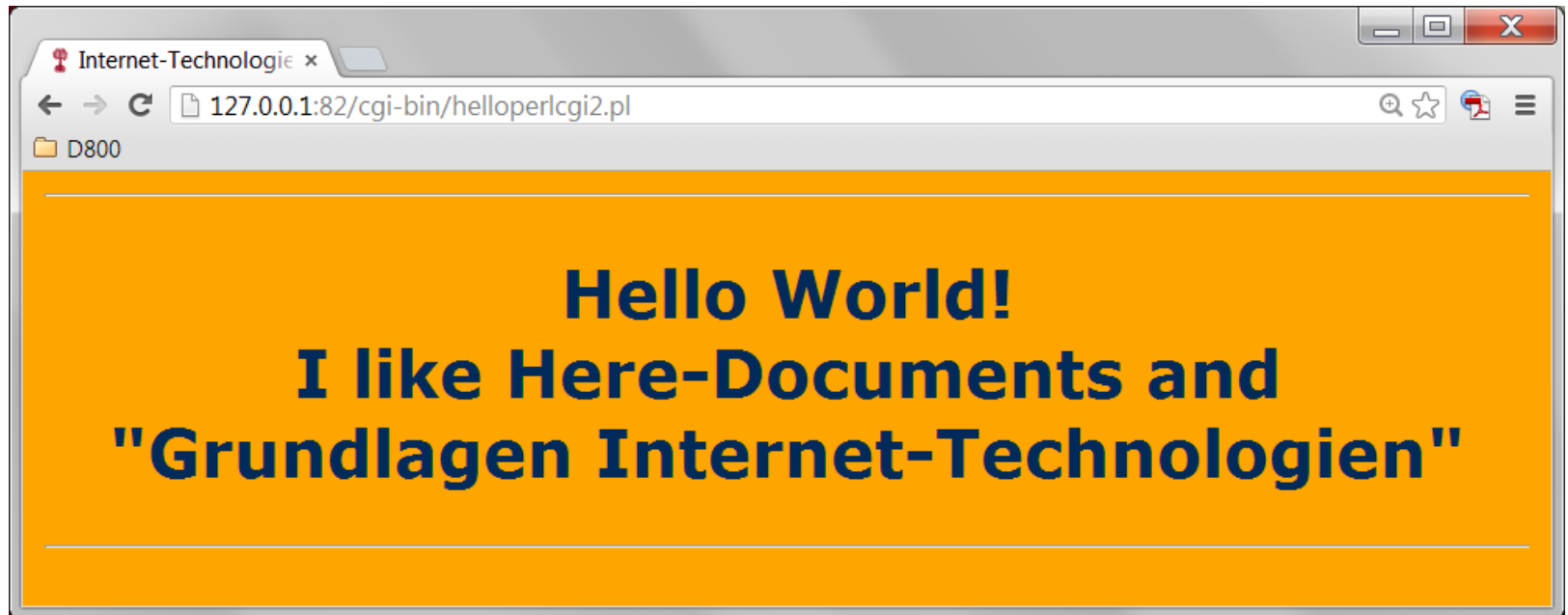
```

1  #!"C:/Program Files/Perl/bin/perl.exe" -w
2  #
3  # Grundlagen Internet-Technologien
4  #
5  # Beispiel fuer einfachstes PERL-CGI mit Here-Documnt
6
7  $phrase = "Hello World!<BR>I like Here-Documents and<BR>\\"Grundlagen Internet-Technologien\\"";
8
9  print <<ende
10 Content-type: text/html
11
12 <HTML>
13     <HEAD><TITLE>Internet-Technologien: Hello World als PERL-cgi</TITLE>
14         <link rel="stylesheet" type="text/css" href="/css/webkompodium.css">
15         <link rel="shortcut icon" href="/css/favicon.ico">
16     </HEAD>
17     <BODY>
18         <HR><CENTER><H2>$phrase</H2></CENTER><HR>
19     </BODY>
20 </HTML>
21 ende
22 ;

```

Writable Insert 1 : 1







weiteres Beispiel: Serverzeit

- Ausgabe der Zeit auf dem Server mittels Perl-
CGI-Script
 - Perl: `gmtime` liefert Array mit Zeitinformationen





```
Perl - InternetTechnologien Perl/zeit.pl - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
printenv.pl liste.pl kreis.pl helloperlcgi.pl helloperlcgi2.pl zeit.pl
1 #!"C:/Program Files/Perl/bin/perl.exe" -w
2 #
3 # Grundlagen Internet-Technologien
4 #
5 # PERL-CGI: Ausgabe der Serverzeit
6
7
8 @zeit = gmtime(time);      # Liste mittels Funktion gmtime
9 $month = $zeit[4]+1;
10 $year = $zeit[5]+1900;
11
12 ### here-Document mit Ersetzung
13 print <<"ENDE_des_textes"
14 Content-type: text/html
15
16 <HTML>
17   <HEAD>
18     <TITLE>Internet-Technologien: Zeit (GMT) auf dem Server</TITLE>
19     <link rel="stylesheet" type="text/css" href="/css/webkompendium.css">
20     <link rel="shortcut icon" href="/css/favicon.ico">
21   </HEAD>
22   <BODY>
23     <HR>
24     <H2>Grundlagen Internet-Technologien<BR>
25     Zeit (GMT) auf Server| $ENV{"SERVER_NAME"}</H2>
26     <H4>Es ist jetzt $zeit[2] Stunden, $zeit[1] Minuten
27     und $zeit[0] Sekunden nach Mitternacht<BR>
28     Wir haben den $zeit[3]. $month. $year<BR>
29     Anfragender Client: $ENV{"REMOTE_HOST"} von IP
30     $ENV{"REMOTE_ADDR"}</H4><HR>
31   </BODY>
32 </HTML>
33 ENDE_des_textes
34 ;
```



The screenshot shows a web browser window with the following details:

- Tab: Internet-Technologie x
- Address bar: 127.0.0.1:82/cgi-bin/zeit.pl
- Location: D800
- Content: A yellow background with blue text.

Grundlagen Internet-Technologien
Zeit (GMT) auf Server 127.0.0.1

Es ist jetzt 12 Stunden, 59 Minuten und 46 Sekunden nach Mitternacht
Wir haben den 31. 5. 2013
Anfragender Client: petitmouton von IP 127.0.0.1





Perl erweitern: Pakete und Module

- Erweiterungen von Perl werden in Paketen und Modulen gebündelt
- Importieren mittels
 - **do** importiert Perl-Datei
 - **require** wie do
 - **use** importiert Modul





Perl Module

- **CPAN:** Verzeichnis der Perl-Module
- hier findet man "alles"
- gute Suchfunktion, Dokumentation, ...
- Installation über Hilfsprogramm ppm
Perl Package Manager

starten mit ppm-shell (Shell) oder ppm
(GUI)


CPAN





The Comprehensive Perl Ar x
— □ ×

← → ↻ 🏠 🔍 cpan.perl.org
☆ 🔧



Comprehensive Perl Archive Network

OVER 15 YEARS OF PERL LIBRARIES AT YOUR FINGERTIPS

Home
Modules
Ports
Perl Source
FAQ
Mirrors

Search:

Welcome to CPAN

The Comprehensive Perl Archive Network (CPAN) currently has [107,203 Perl modules](#) in 24,952 distributions, written by 9,715 authors, [mirrored](#) on 276 servers.

The archive has been online since October 1995 and is constantly growing.

Search CPAN via

- [metacpan.org](#)
- [search.cpan.org](#)

Recent Uploads

- [Task-Belike-TOBYINK-2012.155](#)
- [Task-Belike-TOBYINK-Plack-2012.155](#)
- [Task-Belike-TOBYINK-Packaging-2012.155](#)
- [POE-Component-Client-HTTP-0.947](#)
- [Catalyst-TraitFor-Request-Plack-Session-0.001](#)
- [Validation-Class-7.39](#)
- [Astro-SpaceTrack-0.059](#)
- [Astro-App-Satpass2-0.004](#)
- [Test-Resub-2.01](#)
- [POE-Component-Resolver-0.920](#)
- [more...](#)


Getting Started

- [Installing Perl Modules](#)
- [Learn Perl](#)

Perl Resources

- [The Perl Programming language](#)
- [Perl Documentation](#)
- [Mailing Lists](#)
- [Perl FAQ](#)
- [Scripts Repository](#)

Yours Eclectically, The Self-Appointed Master Librarians (OOK!) of the CPAN.
 © 1995-2010 Jarkko Hietaniemi. © 2011 [Perl.org](#). All rights reserved. [Disclaimer](#).

Master mirror hosted by 



```

DOS Shell - ppm-shell
d:\users\thomas>ppm-shell
ppm 4.10 (ActivePerl 1007)
Copyright (C) 2010 ActiveState Software Inc. All rights reserved.

ppm> help
NAME
  ppm - Perl Package Manager, version 4.10

SYNOPSIS
  Invoke the graphical user interface:

  ppm
  ppm gui

  Install, upgrade and remove packages:

  ppm install [--area <area>] [--force] <pkg>
  ppm install [--area <area>] [--force] <module>
  ppm install [--area <area>] <url>
  ppm install [--area <area>] <file>.ppmx
  ppm install [--area <area>] <file>.ppd
  ppm install [--area <area>] <num>
  ppm upgrade [--install]
  ppm upgrade <pkg>
  ppm upgrade <module>
  ppm remove [--area <area>] [--force] <pkg>

  Manage and search install areas:

  ppm area list [--csv] [--no-header]
  ppm area sync
  ppm list [--fields <fieldnames>] [--csv]
  ppm list <area> [--fields <fieldnames>] [--csv]
  ppm files <pkg>
  ppm verify [<pkg>]

  Manage and search repositories:

  ppm repo list [--csv] [--no-header]
  
```

Perl Package Manager

File Edit View Action Help

Package N...	Area	Installed	Availa...	Abstract
DBD-ODBC	perl	1.23	1.23	ODBC DBD for Perl DBI
DBD-SQLite	perl	1.29	1.29	Self Contained SQLite RDBMS in a DBI Driver
DBI	perl	1.609	1.609	Database independent interface for Perl
Devel-PPPort	perl	3.19	3.19	Perl/Pollution/Portability
Digest	perl	1.16	1.16	Modules that calculate message digests
Digest-HMAC	perl	1.02	1.02	Keyed-Hashing for Message Authentication
Digest-MD2	perl	2.03	2.03	Perl interface to the MD2 Algorithm
Digest-MD4	perl	1.5-r1	1.5	Perl interface to the MD4 Algorithm
Digest-MD5	perl	2.39	2.39	Perl interface to the MD5 Algorithm
Digest-SHA	perl	5.48	5.48	Perl extension for SHA-1/224/256/384/512
Digest-SHA1	perl	2.12	2.12	Perl interface to the SHA-1 algorithm
Encode	perl	2.39	2.39	character encodings
encoding-war...	perl	0.11	0.11	Warn on implicit encoding conversions

Status Details

DBI
Database independent interface for Perl
 Version: 1.609
 Released: 2009-06-08
 Author: Tim Bunce <Tim.Bunce@pobox.com>
 CPAN: <http://search.cpan.org/dist/DBI-1.609/>

Installed files:
 C:/Program Files/Perl/bin/dbilogstrip
 C:/Program Files/Perl/bin/dbilogstrip.bat
 C:/Program Files/Perl/bin/dbiprof
 C:/Prooram Files/Perl/bin/dbiprof.bat

12836 packages, 204 listed | 204 installed, 0 to install, 0 to remove | Install Area:





Perl Package Manager

File Edit View Action Help

lwp

Package Name	Area	Installed	Availa...	Abstract
App-ZofCMS-Plugin-BasicLWP			0.0104	very basic "uri-to-content" style LWP plugin for ZofCMS.
Catalyst-Plugin-UserAgent			0.01	Add a singleton LWP::UserAgent to the context
Crypt-SSLLeay	perl	0.57	0.57	OpenSSL support for LWP
IO-All-LWP			0.14	Use HTTP and FTP URLs with IO::All
LWP	perl	5.834		The World-Wide Web library for Perl
LWP-attic			1.00	
LWP-Authen-Wsse			0.05	Library for enabling X-WSSE authentication in LWP
LWP-ConnCache-MaxKeepA...			0.32	A connection cache that enforces a max keep alive limit
LWP-ConnCache-Resolving			0.02	resolving connection cache.

Status Details

Syncing site PPM database with .packlists ... done
Synchronizing Database ... done
 DBD-mysql marked for install
 DBD-PgPP marked for install
Installing 2 packages ...
 Downloading DBD-PgPP-0.08 ... done
 Downloading DBD-mysql-4.011 ... done
 Unpacking DBD-PgPP-0.08 ... done
 Unpacking DBD-mysql-4.011 ... done
 Generating HTML for DBD-PgPP-0.08 ... done
 Generating HTML for DBD-mysql-4.011 ... done
 Updating files in site area ... done
Installing 2 packages done

12836 packages, 35 listed | 206 installed, 0 to install, 0 to remove | Install Area: **site**





das wichtigste Modul

- das für uns wichtigste Modul ist das Modul **CGI**
 - Autor Lincoln Stein
 - <http://search.cpan.org/~markstos/CGI.pm-3.63/lib/CGI.pm>
gehört inzwischen direkt zum Perl-Bundle dazu

- drei Hauptfunktionalitäten
 - Formularverarbeitung
 - Generierung von XHTML-Code
 - Nützliches für CGI





Internet-Technologie x CGI - search.cpan.org x

search.cpan.org/~markstos/CGI.pm-3.63/lib/CGI.pm

D800 M240


CPAN Home · Authors · Recent · News · Mirrors · FAQ · Feedback

in All

[Mark Stosberg](#) > [CGI.pm-3.63](#) > CGI [permlink](#)

Module Version: 3.63 [Source](#)

NAME
SYNOPSIS
DESCRIPTION
PROGRAMMING STYLE
CALLING CGI.PM ROUTINES
CREATING A NEW QUERY OBJECT (OBJECT-ORIENTED STYLE):
CREATING A NEW QUERY OBJECT FROM AN INPUT FILE
FETCHING A LIST OF KEYWORDS FROM THE QUERY:
FETCHING THE NAMES OF ALL THE PARAMETERS PASSED TO YOUR SCRIPT:
FETCHING THE VALUE OR VALUES OF A SINGLE NAMED PARAMETER:
SETTING THE VALUE(S) OF A NAMED PARAMETER:
APPENDING ADDITIONAL VALUES TO A NAMED PARAMETER:
IMPORTING ALL PARAMETERS INTO A NAMESPACE:
DELETING A PARAMETER COMPLETELY:
DELETING ALL PARAMETERS:
HANDLING NON-URLENCODED ARGUMENTS
DIRECT ACCESS TO THE PARAMETER LIST:
FETCHING THE PARAMETER LIST AS A HASH:
SAVING THE STATE OF THE SCRIPT TO A FILE:
RETRIEVING CGI ERRORS
USING THE FUNCTION-ORIENTED INTERFACE
PRAGMAS
SPECIAL FORMS FOR IMPORTING HTML-TAG FUNCTIONS
GENERATING DYNAMIC DOCUMENTS


Download:
[CGI.pm-3.63.tar.gz](#)
[Dependencies](#)
[Annotate this POD \(51\)](#)

CPAN RT
Open 13
Stalled 2
[View/Report Bugs](#)





Ausgangslage: Das Formular

- HTML-Formular `<FORM ...>`
 - wird an Perl-Script gesendet
 - get oder post zunächst egal
 - Eingabefeld mit Bezeichner, z.B.:
`<INPUT name="feld">`

 - `<FORM method="get" action="/cgi-bin/script.pl">`
`<INPUT name="feld">`
 - ...
 - `</FORM>`





Zugriff auf ein einzelnes Formularfeld

- im Perl-Script:

```
use CGI;           # Import des CGI-
Moduls

                   # von Lincoln Stein

$cgi = new CGI;
$wert = $cgi->param("feld");
```





Zugriff auf ganzes Formular

- im Perl-Script:

```
use CGI;      # Import des CGI-Moduls
              # von Lincoln Stein
```

```
$cgi = new CGI;
$formular = $cgi->Vars();
```

```
$wert = $formular->{"feld"};
```

- `$formular` ist Referenz auf Hash mit den Key-Value-Paaren des Formulars, Dereferenzieren durch `->`





```

Perl - GIT/auswert_1.pl - Eclipse SDK
File Edit Source Refactor Navigate Search Project Run Window Help
hello.pl assArray.pl kreis.pl ifElse.pl schleifen.pl subroutine.pl globallokal.pl auswert_1.pl x 2
1 #!"C:/Program Files/Perl/bin/perl.exe" -w
2 #
3 # Grundlagen Internet-Technologien
4 # verwendet CGI.pm, speichert CGI-Eingaben in Hash ab
5
6 # Verwendung des PERL-Moduls CGI.pm
7 ###use CGI (':standard', ':cgi-lib'); ### prozedural
8
9 use CGI;                ### 00
10
11 ###$eingabe = Vars();    ### prozedural
12
13 # Referenz auf Hash der CGI-Eingaben
14 $query = new CGI;       ### 00
15 $eingabe=$query->Vars(); ### 00
16
17 print <<ENDE
18 Content-type: text/html
19
20 <HTML>
21 <!-- Internet-Technologien: Antwort von auswert_1.pl -->
22 <HEAD>
23 <TITLE>Ihre Meldung</TITLE>
24 <link rel="stylesheet" type="text/css" href="/css/webkompodium.css">
25 </HEAD>
26 <BODY>
27 <CENTER>
28 <HR><H2>Grundlagen Internet-Technologien</H2>
29 <H3>Anmeldung Bachelorarbeit</H3>
30 </CENTER>
31 <P><HR></P>
32 <H4><CENTER>
33     $eingabe->{"vorname"} $eingabe->{"nachname"},
34     vielen Dank für Ihre Anmeldung<BR>zu dem Thema
35     "$eingabe->{"thema"}"
36     <P>Sie bekommen weitere Informationen unter der Mailadresse $eingabe->{"email"}</P>
37 </CENTER></H4>
38 </BODY>
39 </HTML>
40 ENDE
41 ;

```



Grundlagen Internet-Technologien

Anmeldung zur Bachelorarbeit

Nachname:	<input type="text" value="Zuse"/>
Vorname:	<input type="text" value="Konrad"/>
Email:	<input type="text" value="konrad@zuse.de"/>
Thema:	<input type="text" value="Die Z3 im Web"/>
Kurze Projektbeschreibung:	<input type="text" value="Was die Z3 alles im Web schon gekonnt hätte!"/>

Ihre Meldung

127.0.0.1:82/cgi-bin/auswert_1.pl

D800 M240

Grundlagen Internet-Technologien

Anmeldung Bachelorarbeit

**Konrad Zuse, vielen Dank für Ihre Anmeldung
zu dem Thema "Die Z3 im Web"**

**Sie bekommen weitere Informationen unter der Mailadresse
konrad@zuse.de**





Erstellung von XHTML-Code

- das Modul CGI erlaubt es problemlos, **XHTML zu generieren**
- Beispielsweise:
 - `header` erzeugt HTTP-Header
 - `start_html` Anfang des XHTML-Dokuments
 - `h1` Argument in H1-Tags
 - `startform` Anfang Formular
 - `endform` Ende Formular
 - `end_html` Ende XHTML-Dokument





Perl - InternetTechnologien Perl/CgiOutput.pl - Eclipse

File Edit Source Refactor Navigate Search Project Run Window Help

Perl Java EE

zeit.pl auswert_1.pl CgiOutput.pl CgiVaria.pl CgiFormular.pl »14

```

1  #!"C:/Program Files/Perl/bin/perl.exe" -w
2  #
3  # Grundlagen Internet-Technologien
4  #
5  # HTML-Ausgabe durch Modul CGI.pm
6
7  use CGI;
8  use WebKompendium;
9
10 $object = CGI->new();
11
12 ### "Prolog"
13 print $object->header(-type=>'text/html');
14
15 print $object->start_html(
16     -title    => "Internet-Technologien: HTML-Output mit CGI.pm",
17     -style=>{'src'=>'/css/webkompendium.css'}
18 );
19
20 $title = WebKompendium::title();
21
22 print $object->hr();
23 print $object->h2({-align=>"center"},
24 "$title<BR
25 print $obje
26
27 print $obje

```

Internet-Technologie x CGI - search.cpan.org x

127.0.0.1:82/cgi-bin/cgioutput.pl

D800 M240

**Kompendium der Web-Programmierung
HTML-Ausgabe generiert mit CGI.pm und CSS**



```

1 <!DOCTYPE html
2     PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
3     "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
4 <html xmlns="http://www.w3.org/1999/xhtml" lang="en-US" xml:lang="en-US">
5 <head>
6 <title>Internet-Technologien: HTML-Output mit CGI.pm</title>
7 <link rel="stylesheet" type="text/css" href="/css/webkompendium.css" />
8 <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
9 </head>
10 <body>
11 <hr /><h2 align="center">Kompendium der Web-Programmierung
12 <BR />HTML-Ausgabe generiert mit CGI.pm und CSS</h2><hr />
13 </body>
14 </html>

```





weitere Hilfen für CGI

- das Modul CGI stellt eine Vielzahl von "CGI-Hilfen" bereit, u.a. einfacher Zugriff auf CGI-Umgebungsvariablen
 - `request_method`
 - `server_software`
 - `server_name`
 - ...





```

Perl - InternetTechnologien Perl/CgiVaria.pl - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
zeit.pl auswert_1.pl auswert_2.pl auswert_3.pl CgiOutput.pl CgiVaria.pl CgiFormular.pl »12
1  #!"C:/Program Files/Perl/bin/perl.exe" -w
2  #
3  # Grundlagen Internet-Technologien
4  #
5  # weitere Funktionen des Moduls CGI.pm
6
7  use CGI;
8
9  $object = new CGI;
10
11  ### "Prolog"
12  print $object->header(-type=>'text/html');
13
14  print $object->start_html(
15      -title    => "Internet-Technologien: Weiteres mit CGI.pm",
16      -style=>{'src'=>'/css/webkompendium.css'}
17  );
18
19  my $title = "Grundlagen Internet-Technologien";
20
21  print $object->hr ();
22  print $object->h3({-align=>"center"}, "$title<BR />CGI-Umgebungsvariablen mittels CGI.pm");
23  print $object->hr ();
24  print $object->h4("<CODE>SERVER_NAME: ", $object->server_name(), "</CODE>");
25  print $object->h4("<CODE>SERVER_SOFTWARE: ", $object->server_software(), "</CODE>");
26  print $object->h4("<CODE>REQUEST_METHOD: ", $object->request_method(), "</CODE>");
27  print $object->h4("<CODE>REMOTE_HOST: ", $object->remote_host(), "</CODE>");
28  print $object->h4("<CODE>REMOTE_ADDRESS: ", $object->remote_addr(), "</CODE>");
29  print $object->hr ();
30
31  print $object->end_html ();
    
```



Internet-Technologie x CGI - search.cpan.org x
127.0.0.1:82/cgi-bin/cgivariable.pl
D800 M240

Grundlagen Internet-Technologien CGI-Umgebungsvariablen mittels CGI.pm

```
SERVER_NAME: 127.0.0.1  
SERVER_SOFTWARE: Apache/2.2.22 (Win32) PHP/5.4.4  
REQUEST_METHOD: GET  
REMOTE_HOST: petitmouton  
REMOTE_ADDRESS: 127.0.0.1
```

technologi... x CGI - search.cpan.org x +
http://134.2.2.38/~zrvwa01/cgi-bin/Cc

Grundlagen Internet-Technologien CGI-Umgebungsvariablen mittels CGI.pm

```
SERVER_NAME: 134.2.2.38  
SERVER_SOFTWARE: Apache/2.2.9 (Debian)  
REQUEST_METHOD: GET  
REMOTE_HOST: 134.2.6.25  
REMOTE_ADDRESS: 134.2.6.25
```




Perl und Datenbanken

- Perl bietet ein besonderes Modul DBI

"The DBI is the standard database interface module for Perl. It defines a set of methods, variables and conventions that provide a consistent database interface **independent of the actual database being used.**"

- Tim Bunce

- siehe dbi.perl.org






Perl DBI - dbi.perl.org

dbi.perl.org

HOME ABOUT DBI DOCUMENTATION CONTRIBUTING SUPPORT

Perl DBI

Perl's Database Interface



About DBI

The DBI is the standard database interface module for Perl. It defines a set of methods, variables and conventions that provide a consistent database interface independent of the actual database being used.

DBI Releases

- [Perldoc](#)
- [Changes](#)
- [Full content](#)
- [Old versions](#)

DBI Drivers

- [Find latest DBD modules](#)
- [Summaries \(1999\)](#)

DBI Extensions

- [Find latest DBIx modules](#)

Support

There is both community and commercial support available

Online Documentation

- [Manual pages](#)
- [FAQs](#)

Contributing

Please contribute, Time, Patches, Sponsorship or Donations.


Perl object relational mappers (ORMs)

ORMs convert results from a database to objects and back again; so you don't have to write SQL. Perl has several ORM's which use DBI: [DBIx::Class](#) and [Rose::DB::Object](#) are two of the most commonly used.

Find out more

- [Download latest](#)
- [Perldoc](#)
- [Latest DBD modules](#)
- [FAQ](#)


DBI Book



[Programming the Perl DBI](#)

Tim Bunce, Alligat...

[Best Price \\$0.01](#)
or Buy New \$29.29


Buy from  [amazon.com](#)

[Privacy Information](#)


Tip

Is there an ORM? See [DBIx::Class](#).

Sprache auswählen

Powered by  [Google Übersetzer](#)

When you need *Perl* think **Perl.org**: [www](#) | [blogs](#) | [jobs](#) | [learn](#) | [dev](#)

 © 2002-2012 Perl.org | [Site Info](#)



DBI

- Abstraktionsschicht zum Zugriff auf verschiedene DBMS
- Ansatz, der sich allgemein durchgesetzt hat (jdbc, Ruby, Python, PHP-PEAR, ...)
- Details zu Datenbanken behandeln wir bei PHP





das Perl-Modul LWP

- Perl erlaubt es auch einfach, HTTP-Clients zu programmieren
 - bisher: Perl ist nicht direkt "HTTP-fähig"
 - Lösung: das Perl-Modul LWP





Literatur

- Sean M. Buke: Perl & LWP, O'Reilly
- allgemeine PERL-Literatur

O'REILLY®

Perl & LWP





Installation

- LWP ist *nicht* Bestandteil des Standard-PERL-Paketes, sondern muß zusätzlich installiert werden
- dazu nutzen wir wieder ppm

O'REILLY®

Perl & LWP





die LWP-Bibliothek

- bisher: PERL standalone und serverseitig (CGI)
- nun: „Webclient“ mit PERL, etwa
 - Suchmaschine (“Crawler”)
 - Link-Checker
 - Proxy
 - Browser (! - mit GUI)
- Problem: Wir können noch keine Webseiten mit PERL „lesen“
 - genauer: *unser PERL kann noch kein HTTP*





LWP

- LWP: Library for WWW access in Perl
 - besteht aus *mehreren* Modulen
- Autoren: Gisle Aas und Martijn Koster
- aktuelle Version: 6.04
- Abfrage, welches LWP installiert ist:
 - `perl -MLWP -le "print(LWP->VERSION) "`

```

DOS Shell
C:\Program Files\Perl64\bin>
C:\Program Files\Perl64\bin>perl -MLWP -le "print(LWP->VERSION)"
6.04
    
```




Architektur von LWP

- LWP ist PERL-Bibliothek bestehend aus vielen Einzelmodulen

wir betrachten nur die wichtigsten





LWP::Simple

- Basis-Modul aus LWP-Bibliothek
- Implementiert elementarstes HTTP
 - stellt verschiedene Grundfunktionen zur Verfügung, etwa:
 - `get($url) ;`
 - `getprint($url) ;`
 - `getstore($url, $file) ;`





```

D:\users\thomas\Eberhardina\InformationsDienste\webst1\Perl\lwpsimple.pl - Notepad++
Datei Bearbeiten Suchen Ansicht Kodierung Sprachen Einstellungen Makro Ausführen TextFX Erweiterungen Fenster ?
zki_tagung_infos.bt zweiteAuflage.bt webst.bib informatik.bib java.bib rz.bib lwpsimple.pl
1 # Grundlagen IInternet-Technologien
2 #
3 # einfache Anwendung des Moduls LWP
4
5 use LWP::Simple;
6 use WebKompndium;
7
8 $url = "http://www.uni-tuebingen.de";
9
10 WebKompndium::printTitle("Anwendung von PERL-LWP\n");
11
12 ### Seitentext in Variable speichern
13 $doc = get($url);
14 print $doc;
15
16 print "\n\n\n zweite Ausgabe: \n\n\n";
17
18 ### Seitentext ausdrucken
19 $rc = getprint($url);
20
21 ### Seitentext in Datei schreiben
22 $rc = getstore($url, "index.html");
23
24 ### Einsatz von head
25 @head = head($url);
26 print "\n\n Content-length = $head[1] Bytes\n Server: $head[4]\n";

```

© Perl source file 517 chars 542 bytes 26 lines Ln: 1 Col: 1 Sel: 0 (0 bytes) in 0 ranges UNIX ANSI INS



Perl - GIT/lwpsimple.pl - Eclipse SDK

File Edit Source Refactor Navigate Search Project Run Window Help

Debug Quantum ... PHP Perl Resource

Console x

```
<terminated> lwpsimple.pl [Perl Local] Perl Interpreter
Kompendium der Web-Programmierung
Anwendung von PERL-LWP

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transit
<html xml:lang="de" lang="de" xmlns="http://www.w3.org/1999/xhtml"><head><meta http-equiv="Content-Type" conte
    SlimboxOptions.resizeSpeed = 400;
    SlimboxOptions.overlayOpacity = 0.8;
    SlimboxOptions.loop = true;
    SlimboxOptions.allowSave = false;
    SlimboxOptions.slideshowAutoplay = false;
    SlimboxOptions.slideshowInterval = 5000;
    SlimboxOptions.slideshowAutoclose = true;
    SlimboxOptions.counterText = 'Bild ##x## von ##y##';</script></head><body>
<div id="outer">
    <div id="top">
        <div id="uni-t-logo"><a href="http://www.uni-tuebingen.de/" target="_top">
            <div id="languages"><span class="language"><span class="de selected" title="Deutsche Version"></span><
            <div id="search"><form action="http://www.google.com/cse" id="csesearchbox" name="csesearchbox" encryp
```





einige Funktionen von LWP::Simple

- **head(\$url)** : Liste mit den 5 Elementen
 - content-type
 - document-length
 - modified-time
 - expires
 - server
- **\$get(\$url)** : Lädt \$url und gibt diese als Zeichenkette zurück
- **getprint(\$url)** : Lädt \$url und gibt es auf Handle <STDOUT> aus
- **mirror(\$url, \$localfile)** : Lädt \$url nur, falls neuer als \$localfile





Beispiel: Einfaches Script zum „WLAN-keepalive“

- WLAN-Verbindung wird nach ca. 5 Minuten idle-Zeit häufig beendet
- Idee:
 - wir Fragen eine WWW-Site (Beispiel ceres) ab
 - kleine Pause
 - wenn es eine Antwort gab erneut abfragen
 - wenn Schleife beendet, dann ist auch WLAN beendet





```

D:\users\thomas\Eberhardina\InformationsDienste\webst1\Perl\wlan.pl - Notepad++
Datei Bearbeiten Suchen Ansicht Kodierung Sprachen Einstellungen Makro Ausführen TextFX Erweiterungen Fenster ? X
zki_tagung_infos.bt zweiseiteAufgabe.bt webst.bib informatik.bib java.bib rz.bib lwpsimple.pl wlan.pl
1 # Grundlagen IONternet-Technologien
2 #
3 # Script, um WLAN-Verbindung aufrecht zu halten
4 # verwendet LWP
5 #
6
7 use LWP::Simple;
8 use WebKompodium;
9
10 $| = 1; # STDOUT unbuffered
11
12 WebKompodium::printTitle("Anwendung von PERL-LWP\n");
13
14 # beliebige URL, etwa www.ceres.uni-tuebingen.de
15 $url = "http://134.2.2.25";
16
17 print "\n WLAN aktiv ";
18
19 do {
20     ## kleine Pause
21     sleep 5;
22     ## Abfrage der Site
23     @head = head($url);
24     print ".";
25 } while ($head[0]);
26
27 print "\n WLAN nicht verfuegbar\n";
Perl source file 450 chars 502 bytes 27 lines Ln: 1 Col: 1 Sel: 0 (0 bytes) in 0 ranges Dos\Windo ANSI INS

```



LWP::UserAgent

- mehr Funktionalität bietet LWP::UserAgent
 - UserAgent verwaltet „Aufträge“ für Netzwerke
 - UserAgent nutzt die zwei zentralen Klassen

- **HTTP::Request**
- **HTTP::Response**

dies sind Behälter für Aufträge und Ergebnisse

- nützlich:

HTTP::Status





```

D:\users\thomas\Eberhardina\InformationsDienste\webst1\Perl\lwpua.pl - Notepad++
Datei Bearbeiten Suchen Ansicht Kodierung Sprachen Einstellungen Makro Ausführen TextFX Erweiterungen Fenster ?
zki_lagung_infos.bt zweiteAuflage.bt webst.bib informatik.bib java.bib rz.bib lwpsimple.pl wlan.pl lwpua.pl

1 # Grundlagen IInternet-Technologien
2 #
3 # Beispiel fuer LWP: UserAgent, Request, Response
4
5 use LWP;
6 use WebKompndium;
7
8 WebKompndium::printTitle("Anwendung von PERL-LWP\n");
9
10 ### Erzeugung eines UserAgents
11 $ua = LWP::UserAgent->new();
12
13 ### Browsertyp definieren
14 $ua->agent("Name des Browsers: WebKompndiumBrowser 0.07");
15
16 $url = "http://134.2.2.25";
17
18 ### Request-Objekt
19 $request = HTTP::Request->new('GET', $url);
20
21 ### Response-Objekt
22 $response = $ua->request($request);
23
24 if ( $response->is_success() ) {
25     print $response->content();
26 }
27 else {
28     print "ERROR Code: ", $response->code(), " Message ", $response->message(),
29         "\n\n";
30 }

Perl source file
603 chars 661 bytes 30 lines Ln: 1 Col: 1 Sel: 0 (0 bytes) in 0 ranges Dos\Windo ANSI INS

```





```

Perl - GIT/lwpua.pl - Eclipse SDK
File Edit Source Refactor Navigate Search Project Run Window Help
Debug Quantum ... PHP Perl Resource
Console
<terminated> lwpua.pl [Perl Local] Perl Interpreter
Kompendium der Web-Programmierung
Anwendung von PERL-LWP

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transit
<html xml:lang="de" lang="de" xmlns="http://www.w3.org/1999/xhtml"><head><meta http-equiv="Content-Type" conte
    SlimboxOptions.resizeSpeed = 400;
    SlimboxOptions.overlayOpacity = 0.8;
    SlimboxOptions.loop = true;
    SlimboxOptions.allowSave = false;
    SlimboxOptions.slideshowAutoplay = false;
    SlimboxOptions.slideshowInterval = 5000;
    SlimboxOptions.slideshowAutoclose = true;
    SlimboxOptions.counterText = 'Bild ###x### von ###y###';</script></head><body>
<div id="outer">
    <div id="top">
        <div id="uni-t-logo"><a href="http://www.uni-tuebingen.de/" target="_top">
            <div id="languages"><span class="language"><span class="de selected" title="Deutsche Version"></span><
            <div id="search"><form action="http://www.google.com/cse" id="csesearchbox" name="csesearchbox" encryp
    
```





zum Beispiel „uasimple.pl“

- `$ua` ist Referenz auf Objekt der Klasse `UserAgent`
- `$request` ist Referenz auf Objekt der Klasse `HTTP::Request`
- `$response` ist Referenz auf Objekt der Klasse `HTTP::Response`
 - Methode `is_success` liefert im Fehlerfall 0, die Methoden `code()` und `message()` geben Details
 - Methode `content()` gibt Inhalt des Response-Objektes zurück





Anwendungen

- es gibt sehr zahlreiche und wichtige Anwendungen der LWP-Bibliothek
- etwa: **Performance-Tests** einer Webapplikation durch Simulation zahlreicher Zugriffe über ein PERL-Script mittels LWP
- **Robots: Such-Robots, Crawler, ...**
 - ideal mit den leistungsstarken regexpr von PERL
- **Link-Checker**





strict

- **strict** ist ein Perl-Paradigma, das zu besserem Programmierstiel zwingt

– "Einschalten":

```
#!/usr/bin/perl
use strict;
```





strict

- Bedeutung von strict
 - Variablen müssen vor Verwendung deklariert werden:
 - ```
my $var;
```
  - reduziert Fehlerhäufigkeit





# Aktuelle Versionen

- 14.05.2011: Perl 5.14
  - `use 5.12.0;` (beinhaltet `use strict`)
  - Variabel \$,
  - Unicode
- 02.06.2013: Perl 5.16.3 und Perl 18.0
  - <http://search.cpan.org/~rjbs/perl-5.16.0/pod/perldelta.pod>
  - Deklaration von globalen Variablen mittels `our`
  - `use no strict;`





# ...und nun...

haben wir die Möglichkeiten von Perl als cgi-Sprache und den Einsatz mit LWP kennen gelernt

- dabei auch die Erweiterung von Perl durch Module
- und speziell das Perl-Modul CGI

als nächstes

- noch etwas mehr Perl, auch auf dem Client
- der Apache-Webserver
- **Web-Programmierung mit PHP**

