

CGI-Grundlagen Teil 1

Perlen für Ihre Webseiten



Mit Perl und CGI erzeugen Sie dynamische Webanwendungen – vom Besucherzähler bis hin zu einem kompletten Online-Shop. Hier lernen Sie die Grundlagen

Haben Sie schon einmal Aktienkurse im Web abgerufen, einen Online-Shop besucht oder ein Webformular ausgefüllt? Dann hatten Sie bereits – ohne es zu bemerken – mit CGI (Common Gateway Interface) zu tun. Denn vielen Webanwendungen liegt CGI zu Grunde. Zudem sind dynamische Seiten wie *www.amazon.de* in reinem HTML nicht zu realisieren.

CGI-Skripts sind kleine Programme, die auf dem Webserver ablaufen und das Ergebnis an den Browser zurückliefern. Sie sind meist in der Skriptsprache Perl programmiert. In diesem Artikel lernen Sie wichtige Grundlagen und Beispiele von CGI kennen und erfahren, wie Sie Perl-Skripts zum Laufen bringen. Mit die-

sem Wissen ausgestattet richten Sie in Teil 2 des Workshops im nächsten Heft ein komplettes CGI-Projekt für Ihre Webseiten ein. Auf der Heft-CD finden Sie unter *HomeP@ge Magazin* alle Beispielskripts, die in diesem Artikel erwähnt sind.

GRUNDLAGEN VON CGI

Das Common Gateway Interface – die Schnittstelle zwischen Skript und Webserver – erlaubt es Ihnen, Programme auf einem Webserver auszuführen. Auf Anfrage des Browsers startet das CGI-Programm auf dem Server und liefert das Ergebnis zurück an den Browser. Im Gegensatz dazu läuft zum Beispiel ein JavaScript immer auf dem Rechner des Besuchers.

CGI-Skripts verarbeiten zum Beispiel Formulareingaben aus HTML-Dateien, speichern Daten auf dem Server-Rechner oder lesen Daten von dort aus. Auf diese Weise erhalten Webseiten komplexe dynamische Anwendungen, beispielsweise, um Waren elektro-

nisch zu bestellen, um Datenbanken abzufragen oder auch nur für ein Formular oder einen Besucherzähler. Reines HTML reicht dafür nicht.

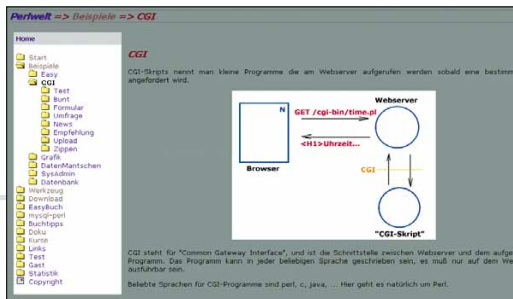
CGI-Programme können in verschiedenen Programmiersprachen geschrieben sein, etwa in C, Fortran oder gar AppleScript. Mit Abstand am beliebtesten ist jedoch Perl. Dieser zweiteilige Workshop beschäftigt sich daher vorwiegend mit Grundlagen und Beispielen dieser Skriptsprache.

Programme, die auf einem Webserver ablaufen, stellen stets ein Sicherheitsrisiko dar. Um die CGI-Schnittstelle nutzen zu können, benötigen Sie daher Zugriff auf ein besonders geschütztes Verzeichnis auf dem Server-Rechner. Meist hat dieses Verzeichnis den Namen *cgi-bin*. CGI-Programme oder CGI-Skripts laufen daher nur, wenn sie in diesem Verzeichnis liegen.

Beispiele, Werkzeuge und deutschsprachige Tutorials bietet <http://perlwelt.horus.at>

BLICK IN DIE GESCHICHTE

Perl erblickte am 18. Oktober 1987 das elektronische Licht der Welt, als der Autor Larry Wall es an die Usenet-Gruppe *comp.sources* sendete. Perl ist ein Akronym für Practical Extraction and Report Language. Seitdem erscheinen neue Versionen stets am 18. Oktober. Ein großer Meilenstein war Perl 5, das 1994 veröffentlicht wurde. Aktuell ist die Pearl-Version 5.8.0.



PERL IST DIE NUMMER 1

Perl ist nach wie vor die beliebteste Skriptsprache im Web. In einer Umfrage von Slashdot.org, an der sich über 22.000 Surfer beteiligt haben, belegte Perl mit 34 Prozent Platz 1 vor PHP mit 20 und Python mit 11 Prozent.

PERL ARBEITET DYNAMISCH

Perl funktioniert anders als HTML. Ein HTML-Dokument ist statisch, ein Perl-Skript hingegen kann Daten dynamisch erzeugen: Ruft ein Browser ein CGI-Programm auf, wird das Perl-Skript in den Speicher des Servers geladen und dort von einem Interpreter verarbeitet. Der Interpreter arbeitet wie ein Simultandolmetscher, der das Skript in Maschinen-Code übersetzt, den der Server versteht. Dieser gibt das Ergebnis des Programms an den Browser zurück. Das kann eine HTML-Seite sein, die aktuellen Aktienkurse oder der Warenkorb eines Shops.

Die Webseiten entstehen dynamisch: So nimmt etwa das Perl-Skript eine Aktienkennung entgegen, ermittelt den aktuellen Kurs und gibt diesen schließlich an den Browser in Form einer HTML-Seite zurück.

Grundsätzlich ist mit Perl die Programmiersprache gemeint und mit perl der Interpreter. Perl-Programme laufen auf jedem beliebigen Rechner und jedem Betriebssystem, sofern dort perl installiert ist.

Wie ist nun ein Perl-Skript aufgebaut? Die erste Zeile ist wichtig: Sie zeigt, wo der Interpreter zu finden ist, den der Server benötigt, um das Skript auszuführen. In der Regel lautet diese erste Zeile, die auch Shebang genannt wird, `#!/usr/bin/perl`. Das bedeutet: Der Interpreter liegt auf dem Webserver im Verzeichnis `/usr/bin/perl`. Gibt es damit Probleme, versuchen Sie es mit dem Shebang `#!/usr/local/bin/perl`, ansonsten fragen Sie bei Ihrem Provider nach.

Einen Perl-Befehl schließen Sie stets mit einem Semikolon ab. Kommentare

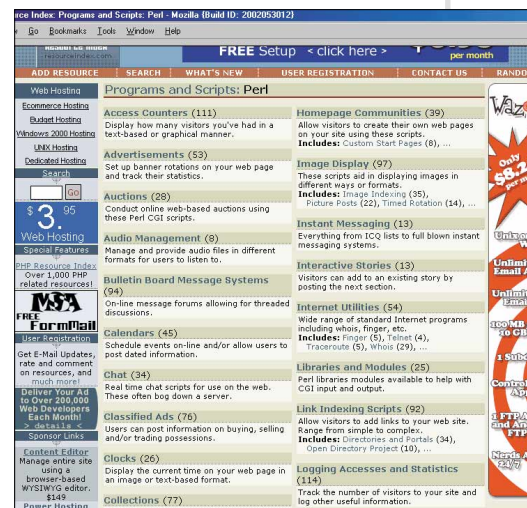
beginnen mit einem `#` und dienen dazu, das Skript verständlich zu gestalten oder den Benutzern zu erklären, wie sie es einsetzen. Der Interpreter ignoriert alle Kommentare.

IHR ERSTES CGI

Jetzt programmieren Sie Ihre erste kleine CGI-Anwendung. Geben Sie in einem Texteditor – etwa in das Notepad von Windows – folgende Zeilen ein:

```
#!/usr/bin/perl
print "Content-type:
text/html\n\n";
print "Mein erstes CGI!\n";
```

Die erste Zeile kennen Sie bereits. Sie teilt dem System mit, wo sich der Inter-



Webelastig, aber mit über 3000 Perl-Skripts die beste Fundgrube für CGI-Skripts: die Website <http://cgi.resourceindex.com>

preter befindet. Die zweite Zeile ist notwendig, damit der Rechner weiß, was er mit den folgenden Befehlen anfangen soll. Die ersten beiden Zeilen können Sie in jedem Skript so stehen lassen.

Der Kern des Skripts steht in der dritten Zeile. Der Befehl `print` ist eine Funktion, die in den Perl-Bibliotheken vordefiniert ist und dazu dient, Text auszugeben, in diesem Falle also *Mein erstes CGI!*. Das `\n` ist ein so genanntes Escape-Zeichen. So erzeugt `\n` ei- ►

nen Zeilenumbruch, während etwa `\a` einen Signalton erklingen lässt.

Speichern Sie die Datei unter dem Namen *hallo.pl* und kopieren Sie sie per FTP auf den Server in Ihr Root-

Serverinfos	Systeminfos	Perl-Debugger	Logfiles	Dead Letter	Webpace berec
Server - Infos:					
Betriebssystem:	linux (Kernel: 2.2.19-6.2.10)				
Perl Version:	5.00503				
Hostname:	www.meinedomain.de				
IP Adresse:	203.107.11.125				
IDs:	uid=99(nobody) gid=99(nobody) groups=99(nobody)				
Internet-Zeit:	@750				
Lokale Uhrzeit:	Mon Aug 19 17:00:47 2002				
GMT Uhrzeit:	Mon Aug 19 15:00:47 2002				
Server aktiv seit:	? Tage, ? Stunden, ? Minuten				
Eingeloggte User:					
Durchschnittliche Serverlast vor:					
01 Minuten:					
05 Minuten:					
15 Minuten:					
Allgemeine Pfade:					
Pfad zu Perl:	/usr/bin/perl				
Grep:	/bin/grep				
Date:	/bin/date				

Das Skript *sysinfo.cgi* schafft den gläsernen Server und verrät Ihnen etwa die Perl-Version und den Pfad zum Interpreter

Verzeichnis. Übertragen Sie die Datei unbedingt im *ASCII*-Modus und nicht als *binary*, sonst funktioniert das Skript wahrscheinlich nicht. Setzen Sie die Dateirechte auf *755* – wie Sie dabei vorgehen, erfahren Sie rechts im Kasten „Dateirechte ändern“. Rufen Sie die Datei in Ihrem Browser über *www.meinedomain.de/cgi-bin/hallo.pl* auf.

Jedoch erscheint jetzt lediglich der geschriebene Code – also die drei Zeilen, die Sie eben eingetippt haben. Was ist schiefgelaufen? Die Lösung: Das Skript läuft nur in dem *cgi-bin*-Verzeichnis. Kopieren Sie die Datei also in dieses Verzeichnis und rufen Sie sie dort erneut auf. Jetzt erscheint der Text *Mein erstes CGI!*. Schauen Sie in den Quelltext: Dort sehen Sie nichts mehr von Ihrem Listing, sondern nur das Ergebnis des Interpreters.

Für eine kleine Variante ändern Sie die dritte Zeile des Skripts in

```
print "<b>Mein  
zweites CGI!</b>\n";
```

Sie sehen, der Text erscheint nun im Browser fett. Sie können also in Perl auch HTML-Tags ausgeben, die der Browser dann korrekt darstellt.

SERVER-GEHEIMNISSE ERGRÜNDEN

Als Nächstes lernen Sie ein praktisches Beispiel für ein CGI-Skript kennen: Sie wollen mehr Informationen über Ihren Webserver, als Ihnen Ihr Provider verraten will? Mit dem kleinen Skript *sysinfo.cgi* erhalten Sie Informationen über den Webserver Ihres Providers, die vielleicht nicht einmal dieser selbst kennt. Zu wissen, wie der Webserver konfiguriert ist, hilft sehr bei Problemen mit CGI- und PHP-Skripts oder anderen Programmen.

Kopieren Sie die Datei *sysinfo.cgi* von der Heft-CD auf Ihre Festplatte und öffnen Sie sie mit einem Texteditor, etwa dem Notepad. In der ersten Zeile sehen Sie wie bereits erklärt den Verweis auf das perl-Verzeichnis, den Shebang: `#!/usr/bin/perl`.

Ersetzen Sie die Zahl *250* in der Zeile, die mit `$free = "50"`; beginnt, durch die Anzahl an MByte, die Sie als Webpace zur Verfügung haben. Das



Ein gutes Buch, das anschaulich die Grundlagen und Einsatzmöglichkeiten von Perl vermittelt – siehe Kasten oben

MEHR ÜBER CGI ERFAHREN

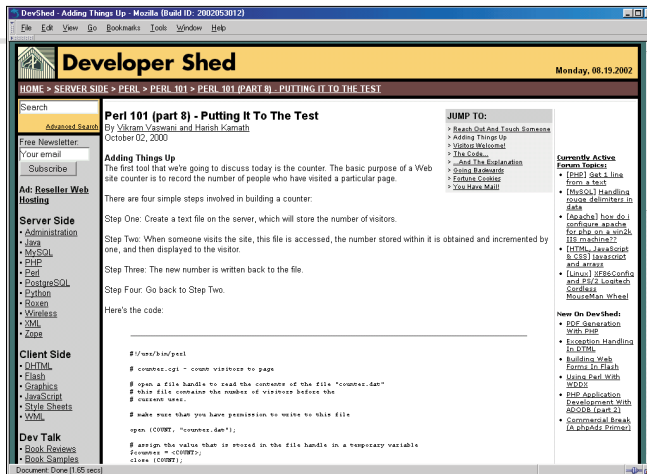
Wer tiefer in die Welt des CGI einsteigen möchte, sollte sich ein gutes Buch über Perl zulegen. Dazu empfiehlt sich etwa das Werk „Jetzt lerne ich Perl“ aus dem Verlag Markt und Technik, das mit 20 Euro zudem einen vernünftigen Preis hat. Es führt ausführlich in die Grundlagen der Skriptsprache ein und verdeutlicht Möglichkeiten anhand von Praxisbeispielen wie einem Gästebuch.

Skript errechnet daraus, wie viel Prozent davon belegt sind. Tauschen Sie in der Zeile, die mit `$password = "test"`; anfängt, das Wort *test* durch ein Passwort Ihrer Wahl, damit nicht jeder Besucher das Skript aufrufen kann. Speichern Sie das Skript und übertragen Sie es im *ASCII*-Modus auf Ihren Webserver in das Verzeichnis *cgi-bin*. Ändern Sie wiederum die Zugriffsrechte in *755*.

Rufen Sie das Skript im Browser über *www.meinedomain.de/cgi-bin/sysinfo.cgi* auf. Lehnen Sie sich zurück und staunen Sie: Sie sehen unter anderem, welche Hardware und welches Betriebssystem Ihr Provider verwendet, wie voll seine Festplatten sind und erfahren, wie der Server konfiguriert ist. Und unter *Webpace berechnen* verrät Ihnen das Skript, wie viel Prozent von Ihrem Webpace bereits belegt sind.

MAILEN MIT CGI

Mit dem folgenden Skript können Sie oder Ihre Besucher von Ihrer Webseite aus E-Mails verschicken. Das Skript erzeugt ein Formular, das E-Mail-Adresse, Betreff und die Mitteilung abfragt. Um es zu konfigurieren, öffnen Sie das Skript in einem Texteditor und suchen die Zeile, die mit `$from=` beginnt. Tragen Sie hier Ihre E-Mail-Adresse ein. Außerdem benötigen Sie die Pfadangabe zu *sendmail*, die Sie bei Ihrem Provider erfragen. *Sendmail* ist ein weit verbreiteter Mail-Transfer-Agent und



verwaltet einen großen Anteil aller im Internet von einem zum anderen Rechner verschickten E-Mails.

In den meisten Fällen funktioniert die Angabe `/usr/sbin/sendmail`, die bereits im Skript eingetragen ist. Schauen Sie sich ruhig ein wenig im Quellcode um, er ist kommentiert.

Das Skript prüft, ob in allen Feldern etwas eingetragen und ob eine der Form nach gültige Mail-Adresse angegeben ist. An diesem Beispiel lernen Sie eine weitere Besonderheit von Perl:

REGULÄRE AUSDRÜCKE

Eine der Stärken von Perl sind die so genannten regulären Ausdrücke. Damit lassen sich Zeichen in jedem Zu-

sammenhang finden und bearbeiten. Sie durchforsten damit Texte nach Textpassagen oder Zeichenmustern, zählen, wie oft diese vorkommen, oder ersetzen Textbausteine durch andere.

Der Code dazu sieht auf den ersten Blick abenteuerlich aus, ist aber schnell zu verstehen. Sehen Sie sich beispielsweise die Zeile

```
/^[w-].+@\[w-].+$/
```

aus dem E-Mail-Skript an. Dieser Ausdruck prüft, ob die eingegebene E-Mail-Adresse der Form nach korrekt ist, also in der Art `user@email.de`. Die Zeichen `/` am Anfang und Ende sind Begrenzer. Diese sagen dem Interpreter: Achtung, jetzt kommt ein regulärer Ausdruck. Die Zeichen `^` und `$` bezeichnen Anfang und Ende einer Zeichenkette. Also alles, was zwischen diesen beiden Zeichen steht, wird untersucht.

Die beiden `[w-]` bedeuten, dass vor und hinter dem `@` Buchstaben, Zahlen und Unterstriche sowie zusätzlich – und – vorkommen dürfen. Das `w` ist ein weiteres Escape-Zeichen wie das oben erwähnte `\n` und bedeutet Wort. Das Plus dahinter legt fest, dass das Wort mindestens einen Buchstaben lang sein muss. Da das `@` eine besondere Bedeutung in Perl-Skripts hat, teilen Sie über `\@` mit, dass Sie nicht diese Bedeutung, sondern Text meinen. Mehr über die regulären Ausdrücke in Perl

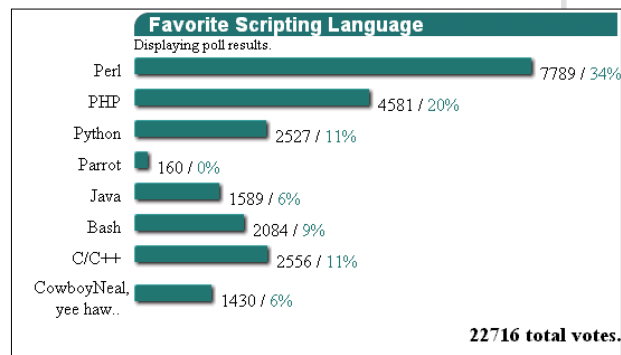
Unter www.devshed.com finden Einsteiger gute Perl-Tutorials und Fortgeschrittene einen Lehrgang in objektorientierter Perl-Programmierung – allerdings alles in Englisch

können Sie nachlesen unter www.rz.uni-hamburg.de/RRZ/W.Wiedl/Skripte/CGI-Perl/Regulaere_Ausdruecke.

DIE WELT DER VARIABLEN

Variablen sind das Herz jeder Programmiersprache. In den zuvor erwähnten Skripts haben Sie bereits Variablen kennengelernt. Elementar sind in Perl die skalaren Variablen, die je nach Kontext als Zahlen oder als Text interpretiert werden. Vor jedem Variablennamen steht ein `$`. Daran erkennt der Interpreter, dass eine Variable folgt. Der Befehl `$zahl=8`; etwa weist der Variablen namens `zahl` den Wert `8` zu. Schreiben Sie statt dessen `$zahl='8'`; ordnen Sie der Variablen einen Text und keine Zahl zu.

Perl unterscheidet zwischen Groß- und Kleinschreibung. `$zahl` ist also ei-



Nach einer Umfrage bei Slashdot.org ist Perl die beliebteste Skriptsprache im Web

ne andere Variable als `$zahl`. Sie können mit Variablen auch rechnen: `$zahl=3+5`; bedeutet etwa, addiere 3 und 5 und weise das Ergebnis der Variablen `zahl` zu. Bei dem Befehl `$a=$b`; macht Perl eine Kopie von `$b` und weist diese der Variablen `$a` zu. Um den Wert der Variablen auszugeben, verwenden Sie `print "$a";`.

Damit kennen Sie die Grundlagen, um im zweiten Teil des Workshops ein großes CGI-Projekt einzurichten. ■

Andreas Dumont

DATEIRECHTE ÄNDERN

Die meisten CGI-Programme benötigen als Zugriffsrecht 755. Viele Server setzen dieses jedoch standardmäßig auf 644. Sie müssen die Rechte also manuell ändern. Dazu benutzen Sie in Ihrem FTP-Programm oft den Befehl *Eigenschaften*. Bei SmartFTP, das Sie auf der Heft-CD finden, erledigen Sie dies unter *Properties*.

755 bedeutet, der Eigentümer hat alle Rechte, alle übrigen Nutzer haben Lese- und Ausführrechte, aber keine Schreibrechte. Eine ausführliche Beschreibung zur Bedeutung der Dateirechte finden Sie in Heft 7/02 auf Seite 109.