

```

1 <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
2 <HTML>
3   <HEAD>
4     <TITLE>WebKompodium: HTML mit Ruby generieren</TITLE>
5     <STYLE type="text/css">@import url("/webkompodium/css/webkompodium.css");</STYLE>
6   </HEAD>
7   <BODY>
8     <HR><H2>Kompodium der Web-Programmierung<BR>Ruby-cgioutput</H2><HR>
9   </BODY>
10  </HTML>

```

Abbildung 13.34:
HTML-Sourcecode der Antwort

13.5.3 Der umgekehrte Weg: Ruby in HTML einbinden

Wir haben nun den prinzipiellen Weg gesehen, wie durch Ruby HTML-Code erzeugt wird – der CGI-typische Weg, welcher entweder durch Here-Documents oder durch HTML-Code generierende Funktionen erleichtert werden kann. Dennoch zeigt die Verbreitung etwa von PHP oder von Java Server Pages, dass der umgekehrte Weg von größerem Erfolg ist: nicht in Programmcode wie Ruby oder Perl die HTML-Anteile zu integrieren, sondern umgekehrt in ein HTML-Dokument die aktiven Programmteile einzubinden.

Diesen Erfolg sehen natürlich auch die Entwickler von Ruby, und es gibt inzwischen Ansätze, auch mit Ruby den umgekehrten Weg zu gehen. Den Kern dieser Ansätze bildet jeweils eine Erweiterung des Apache-Webservers um Ruby-Funktionalitäten:

- **mod_ruby** ist ein typisches Apache-Modul für Ruby; es erhöht die Performance wesentlich;

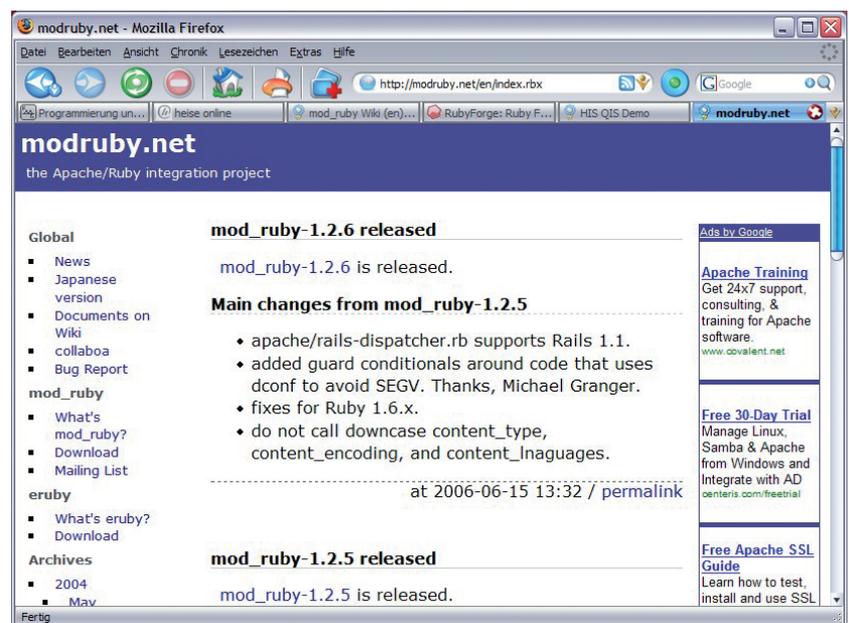


Abbildung 13.35:
mod_ruby im Netz

- **eRuby** ist eine auf CGI basierende Erweiterung, welche das Prinzip der Verwendung von Ruby im HTML-Code erlaubt:
eRuby stellt drei Arten neuer Tags bereit, die sich auf Ruby-Code beziehen:
 - durch das Tag `<% Ruby-Code %>`
wird der angegebene Ruby-Code ausgeführt;
 - durch `<%=Ruby-Ausdruck%>`
wird der Ruby-Ausdruck ausgewertet und durch seinen Wert ersetzt;
 - das Tag `<#Ruby-Code%>`
wird ignoriert (etwa für Debugging).

Der Einsatz von eRuby setzt voraus, dass der Apache-Server entsprechend konfiguriert wird; dies bedeutet etwa in der `httpd.conf` die Angabe folgender Direktiven:

```
AddType application/x-httpd-eruby .rhtml
Action application/x-httpd-eruby /cgi-bin/eruby

DirectoryIndex index.html index.shtml index.rhtml
```

Sowohl `mod_ruby` als auch eRuby sind, wenn überhaupt, dann nur schwierig auf Windows einsetzbar; hier sollte unbedingt Unix/Linux verwendet werden.



Sollen unter Windows komplexere Ruby-Funktionalitäten in einen Apache-Webserver eingebunden werden, kann das Sourceforge-Projekt „`rubyforapache`“^{web} sehr hilfreich sein.



Der Einsatz von fastCGI (vgl. Kapitel 19) ist prinzipiell auch für Ruby empfehlenswert.³

13.5.4 Ein HTTP-Server in Ruby

Zum Ruby-Paket gehört auch eine umfassende Netz-Bibliothek, welche alle gängigen Netz-Protokolle unterstützt. Wesentlich ist hier das Modul `socket`, welches Endpunkte von Netzwerkverbindungen verwaltet (Abbildung 13.36 zeigt die Struktur dieser Klassen).

Als Anwendung soll ein einfacher HTTP-Server mit Ruby realisiert werden, in Anlehnung an die Server-Implementierungen mit Java in Abschnitt 6.2. Hierzu wird in einer Ruby-Anwendung eine Instanz der Klasse `TCPServer` aus dem Modul `socket` auf einem bestimmten Port (hier 8086) erzeugt; die Methode `accept` dieser Klasse liefert eine konkrete Verbindung, zu welcher mittels `print` eine Ausgabe erfolgt. Das Ruby-Script `simplehttpd.rb` (Abbildung 13.37) zeigt diese exemplarische Anwendung.

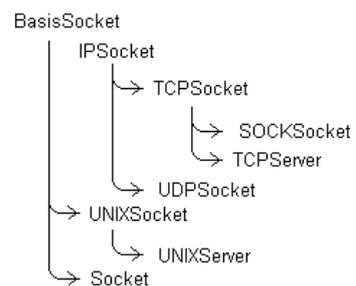


Abbildung 13.36:
Struktur der Socket-Klassen von Ruby

³insbesondere, weil der momentane Stand des Ruby-Interpreters keine Multi-Threads unterstützt.