

Web Programming Basics

Software Languages Team
University of Koblenz-Landau
Ralf Lämmel and Andrei Varanovich

Good old times

\$ telnet www.microsoft.com 80

Trying 65.55.57.27...

Connected to lbl.www.ms.akadns.net.

Escape character is '^]'

HEAD / HTTP/1.0

HTTP/1.1 200 OK

Cache-Control: no-cache

Content-Length: 1020

Content-Type: text/html

Last-Modified: Mon, 16 Mar 2009 20:35:26 GMT

Accept-Ranges: bytes

ETag: "67991fbd76a6c91:0"

Server: Microsoft-IIS/7.5

X-Powered-By: ASP.NET

Date: Fri, 07 Sep 2012 04:45:28 GMT

Connection: close

Connection closed by foreign host.

User input
is in bold

Telnet

Telnet is a [network protocol](#) used on the [Internet](#) or [local area networks](#) to provide a **bidirectional interactive text-oriented communications** facility using a virtual [terminal](#) connection. User data is interspersed [in-band](#) with Telnet control information in an 8-bit [byte oriented](#) data connection over the [Transmission Control Protocol \(TCP\)](#).

<http://en.wikipedia.org/wiki/Telnet>

29 June 2012

Hypertext Transfer Protocol

http://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol

- **GET** Request representation for resource
- **HEAD** Like GET but without response body
- **PUT** Upload representation for resource
- **POST** Submit data for resource
- **OPTIONS** Query for available methods
- **CONNECT** Facilitate SSL-encrypted communication
- **DELETE** Delete specified resource
- **TRACE** Return request as it arrived at server
- **PATCH** Partial modification of resource

Hypertext Transfer Protocol

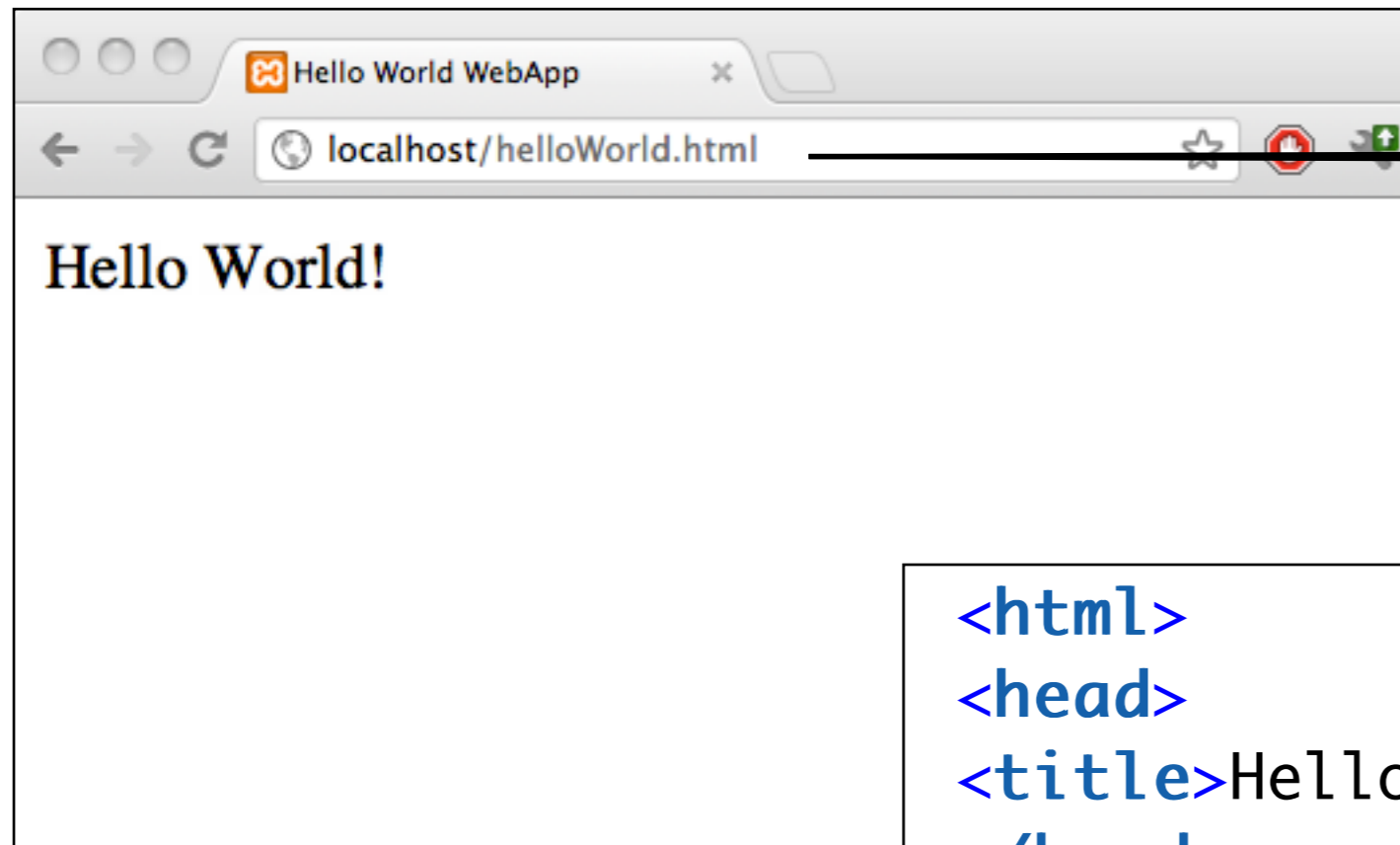
- Safe: HEAD, GET, OPTIONS, TRACE
- Unsafe: POST, PUT, DELETE
- Idempotent: PUT, DELETE + safe methods
- HTTP is a stateless protocol.

We skip the underlying TCP/IP.

More on HTTP,
when we talk
about 'REST'.

Static web pages

http://en.wikipedia.org/wiki/Static_web_page



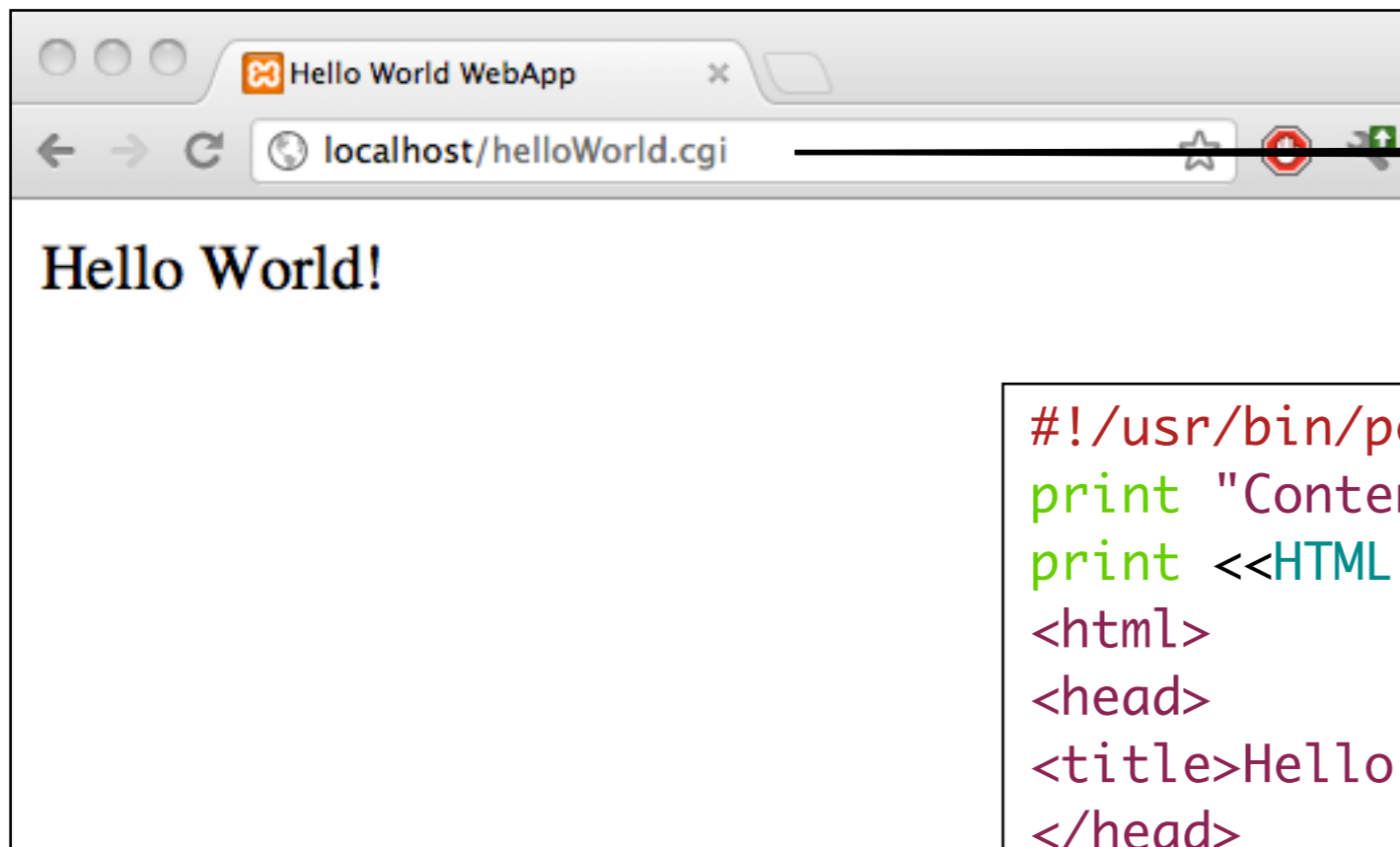
localhost/
helloWorld.html

HTML markup

```
<html>  
<head>  
<title>Hello World WebApp</title>  
</head>  
<body>  
<p>Hello World!</p>  
</body>  
</html>
```

Dynamic web pages

http://en.wikipedia.org/wiki/Dynamic_web_page



localhost/
helloWorld.cgi

HTML markup

```
#!/usr/bin/perl
print "Content-type: text/html\n\n";
print <<HTML;
<html>
<head>
<title>Hello World WebApp</title>
</head>
<body>
<p>Hello World!</p>
</body>
</html>
HTML
exit;
```

<http://en.wikipedia.org/wiki/Perl>

Perl

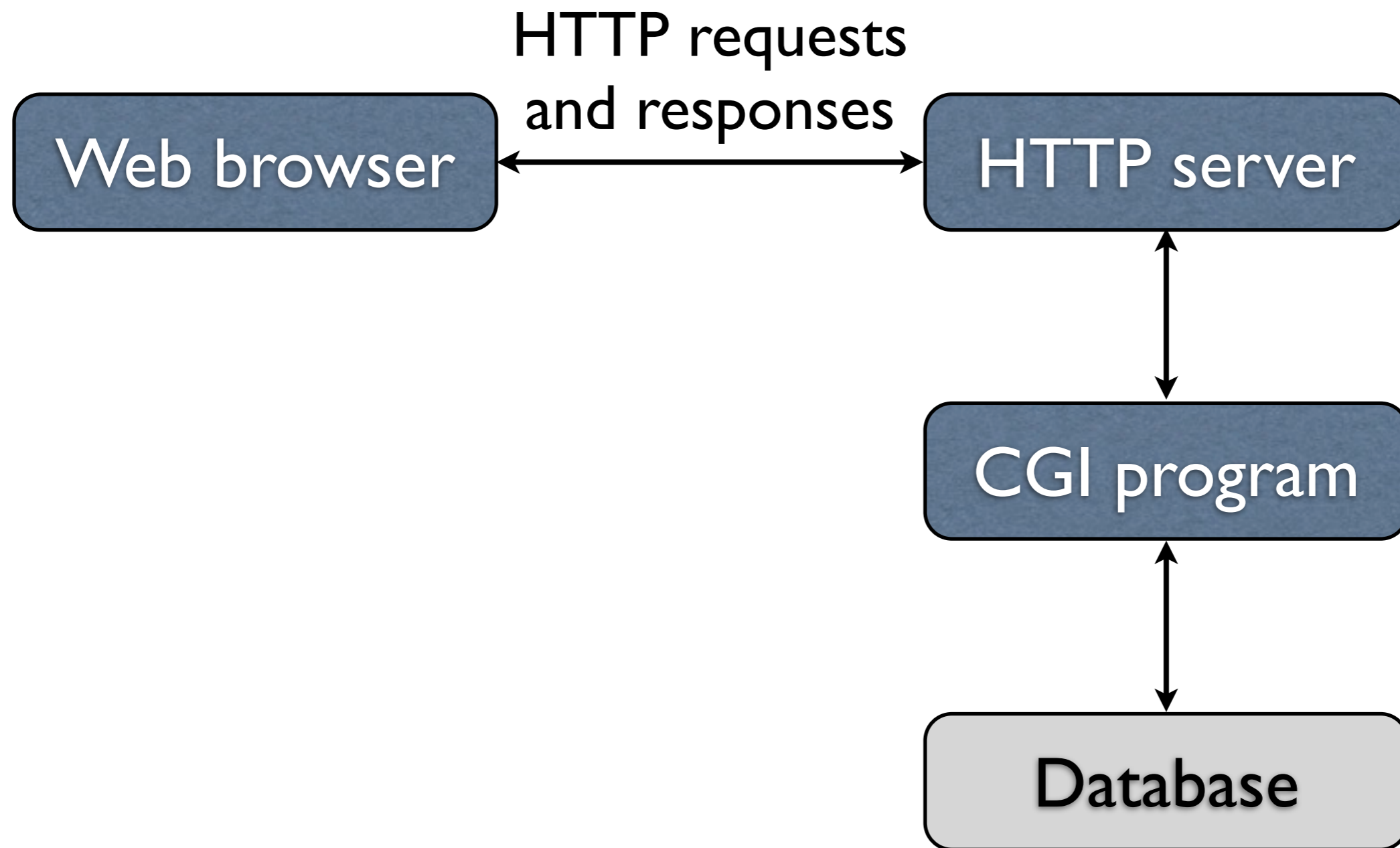


Paradigm(s)	multi-paradigm: functional, imperative, object-oriented (class-based), reflective, procedural, generic
Appeared in	1987
Designed by	Larry Wall
Developer	Larry Wall
Stable release	5.16.1 ^[1] (August 8, 2012; 29 days ago)
Preview release	5.17.2 ^[2] (July 20, 2012; 48 days ago)
Typing discipline	Dynamic

Influenced by	AWK , Smalltalk 80 , Lisp , C , C++ , sed , Unix shell , Pascal
Influenced	Python , PHP , Ruby , ECMAScript , LPC , Windows PowerShell , JavaScript , Falcon , Perl 6 , Qore
Implementation language	C
OS	Cross-platform
License	GNU General Public License or Artistic License ^[3]
Usual filename extensions	.pl .pm .t
Website	www.perl.org
	Perl Programming at Wikibooks

CGI - Common Gateway Interface

http://en.wikipedia.org/wiki/Common_Gateway_Interface



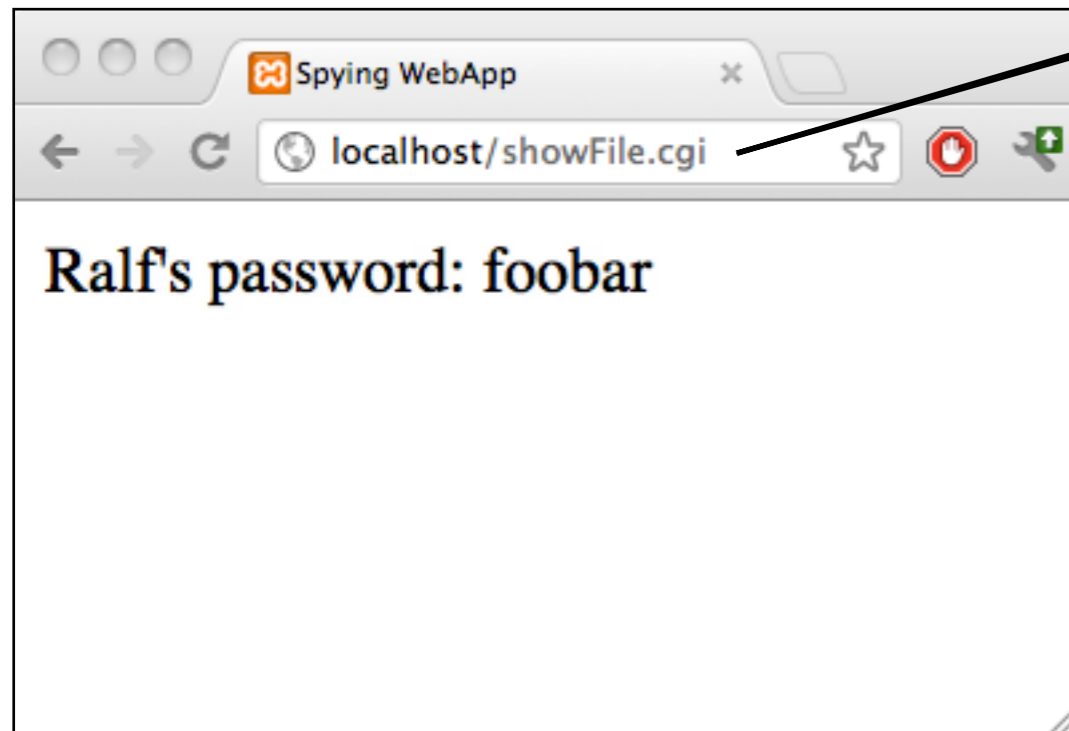
The Common Gateway Interface (CGI) is a standard (see RFC 3875: CGI Version 1.1) method for web server software to delegate the generation of web content to executable files. Such files are known as CGI scripts; they are programs, often stand-alone applications, usually written in a scripting language.

http://en.wikipedia.org/wiki/Common_Gateway_Interface

We do not go into detail here.
There is, for example, also the
advancement *FastCGI*.

Access to server-side resources

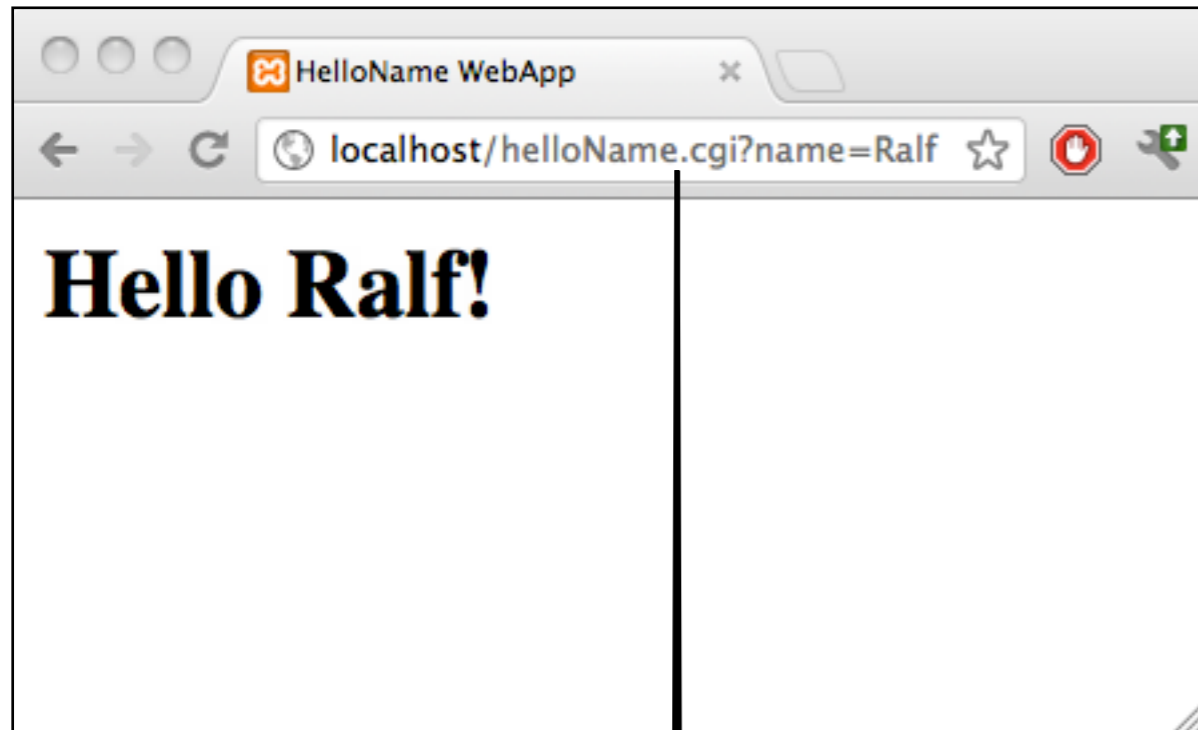
localhost/
showFile.cgi



```
#!/usr/bin/perl
open FILE, "demo.txt";
my $lines = <FILE>;
print "Content-type: text/html\n\n";
print <<HTML;
<html>
<head>
<title>Spying WebApp</title>
</head>
<body>
<p>
HTML
print $lines;
print <<HTML;
</p>
</body>
</html>
HTML
exit;
```

Perl code

HTTP request parameters



localhost/
helloName.cgi?
name=Ralf

Perl code

```
#!/usr/bin/perl

local ($buffer, @pairs, $pair, $name, $value, %FORM);
# Read in text
$ENV{'REQUEST_METHOD'} =~ tr/a-z/A-Z/;
if ($ENV{'REQUEST_METHOD'} eq "GET")
{
    $buffer = $ENV{'QUERY_STRING'};
}
# Split information into name/value pairs
@pairs = split(/&/, $buffer);
foreach $pair (@pairs)
{
    ($key, $val) = split(/=/, $pair);
    $val =~ tr/+//;
    $val =~ s/%(..)/pack("C", hex($1))/eg;
    $FORM{$key} = $val;
}
$name = $FORM{name};

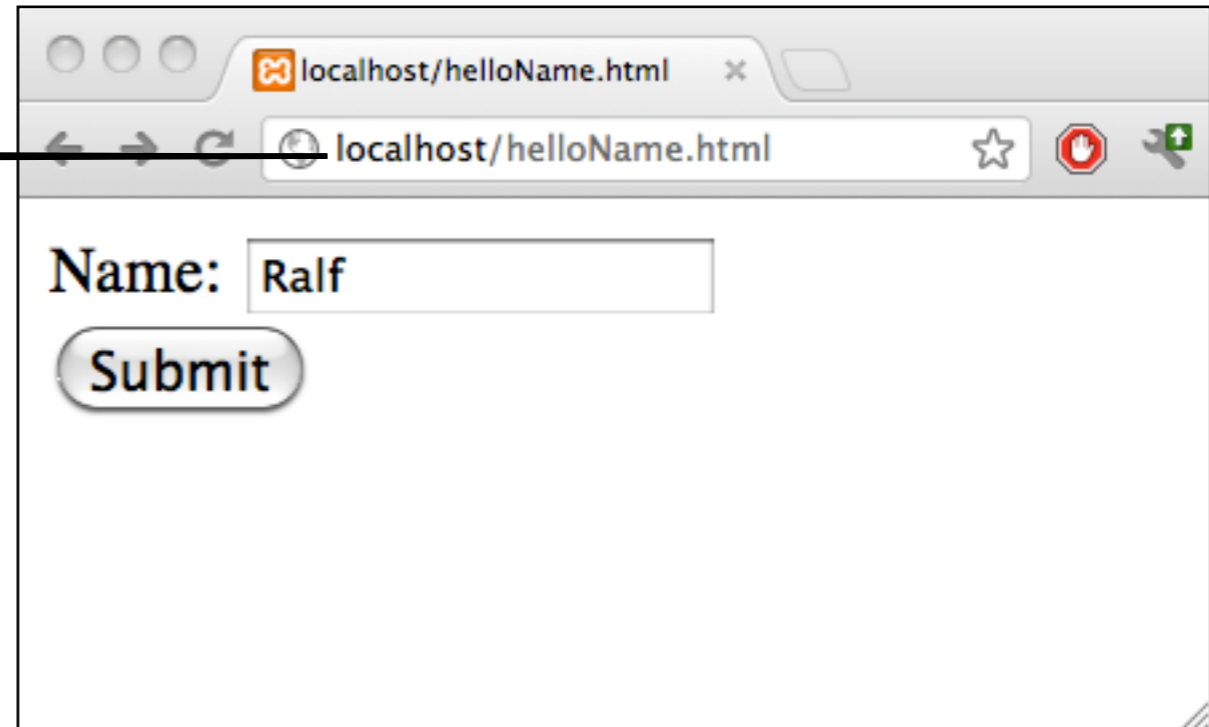
print "Content-type:text/html\r\n\r\n";
print "<html>";
print "<head>";
print "<title>HelloName WebApp</title>";
print "</head>";
print "<body>";
print "<h2>Hello $name!</h2>";
print "</body>";
print "</html>";

1;
```

HTML forms

[http://en.wikipedia.org/wiki/Form_\(web\)](http://en.wikipedia.org/wiki/Form_(web))

localhost/
helloName.html

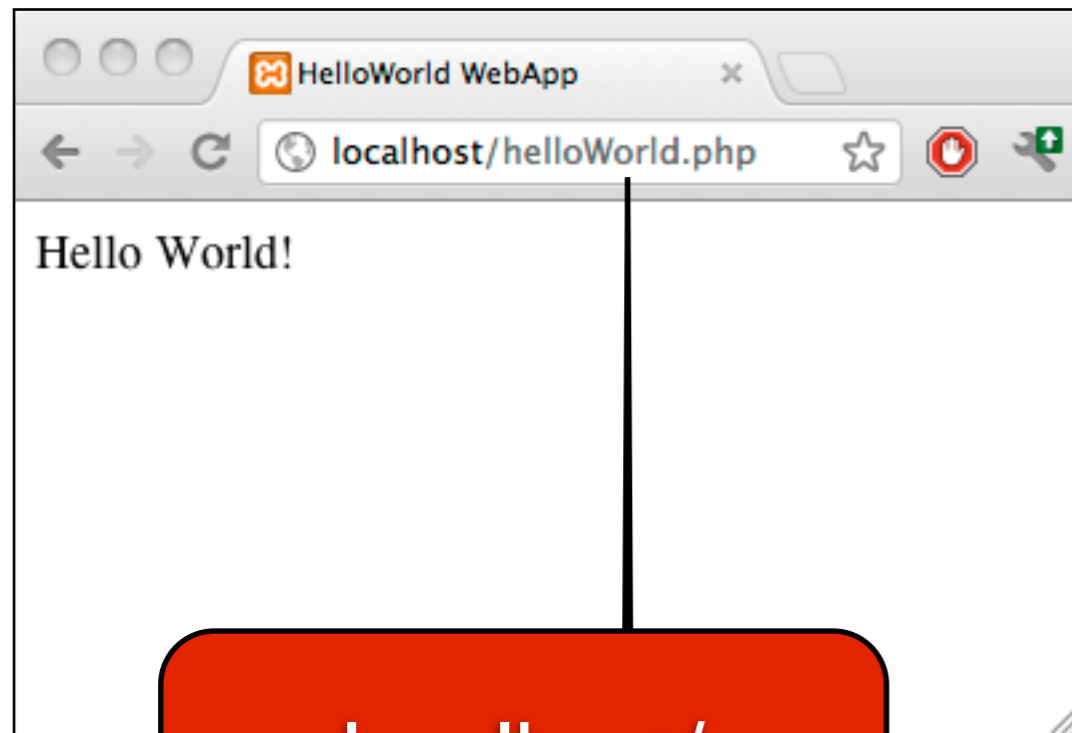


A screenshot of a web browser window. The address bar shows 'localhost/helloName.html'. The page content displays a form with the label 'Name:' followed by a text input field containing the text 'Ralf'. Below the input field is a button labeled 'Submit'.

HTML FORM

```
<FORM action="helloName.cgi" method="GET">  
Name: <input type="text" name="name"> <br>  
<input type="submit" value="Submit">  
</FORM>
```

Code embedded into HTML



localhost/
helloWorld.php

```
<html>  
<head>  
  <title>HelloWorld WebApp</title>  
</head>  
<body>  
<?php echo '<p>Hello World!</p>'; ?>  
</body>  
</html>
```

HTML with PHP

PHP - The PHP Hypertext Processor

<http://en.wikipedia.org/wiki/PHP>

PHP



PHP: Hypertext Preprocessor

Paradigm(s)	imperative, object-oriented, procedural, reflective
Appeared in	1995; 17 years ago ^[1]
Designed by	Rasmus Lerdorf
Developer	The PHP Group
Stable release	5.4.6 (August 15, 2012; 23 days ago)
Typing discipline	Dynamic, weak

Major implementations	Zend Engine, Phalanger, Quercus, Project Zero, HipHop
Influenced by	C, C++, Java, Perl, Tcl ^[1]
Implementation language	C
OS	Cross-platform
License	PHP License
Usual filename extensions	Common extensions .php Older, now uncommon extensions .phtml, .php4 .php3, .php5, .phps
Website	www.php.net 

 [PHP Programming at Wikibooks](#)

Summary

You learned about ...

- the difference between static and dynamic pages,
- the use of HTML forms,
- the architecture of CGI,
- basic techniques of writing CGI scripts,
- environment variables in CGI programs,
- and some bits of Perl and PHP.

Resources

- <http://www.citycat.ru/doc/CGI/overview/overview.html>