

## Perl CGI编程

### 什么是CGI

CGI 目前由NCSA维护，NCSA定义CGI如下：

CGI(Common Gateway Interface),通用网关接口,它是一段程序,运行在服务器上如：HTTP服务器，提供同客户端HTML页面的接口。

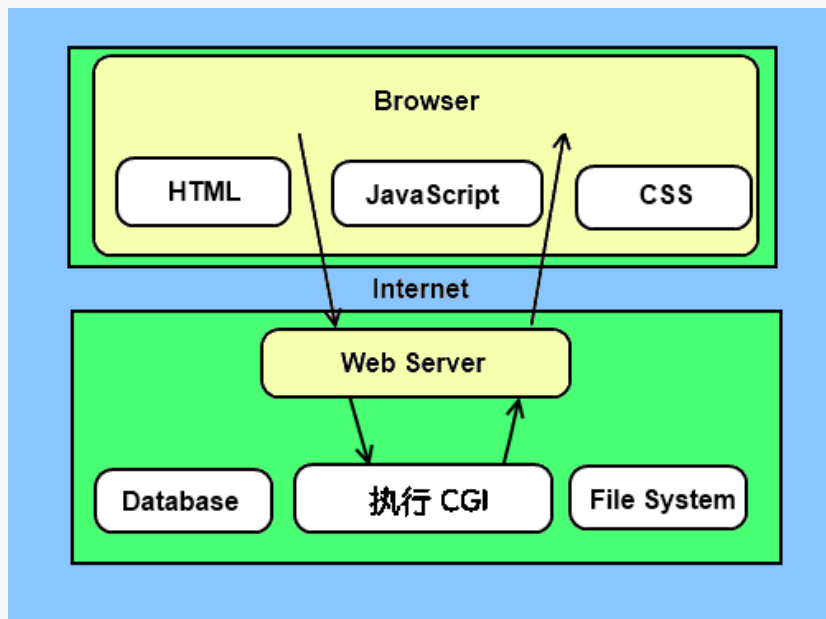
### 网页浏览

为了更好的了解CGI是如何工作的，我们可以从在网页上点击一个链接或URL的流程：

- 1、使用你的浏览器访问URL并连接到HTTP web 服务器。
- 2、Web服务器接收到请求信息后会解析URL，并查找访问的文件在服务器上是否存在，如果存在返回文件的内容，否则返回错误信息。
- 3、浏览器从服务器上接收信息，并显示接收的文件或者错误信息。

CGI程序可以是Python脚本，PERL脚本，SHELL脚本，C或者C++程序等。

### CGI架构图



### Web服务器支持及配置

在你进行CGI编程前，确保您的Web服务器支持CGI及已经配置了CGI的处理程序。

Apache 支持CGI 配置：

设置好CGI目录：

```
ScriptAlias /cgi-bin/ /var/www/cgi-bin/
```

所有的HTTP服务器执行CGI程序都保存在一个预先配置的目录。这个目录被称为CGI目录，并按照惯例，它被命名为/var/www/cgi-bin目录。

CGI文件的扩展名为.cgi，Perl 也可以使用.pl扩展名。

默认情况下，Linux服务器配置运行的cgi-bin目录中为/var/www。

如果你想指定其他运行CGI脚本的目录，可以修改httpd.conf配置文件，如下所示：

```
<Directory "/var/www/cgi-bin">
    AllowOverride None
    Options +ExecCGI
    Order allow,deny
    Allow from all
</Directory>
```

在 AddHandler 中添加 .pl 后缀，这样我们就可以访问 .pl 结尾的 Perl 脚本文件：

```
AddHandler cgi-script .cgi .pl .py
```

## 第一个 CGI 程序

下面我们创建一个 test.cgi 文件，代码如下所示：

### test.cgi 代码

```
#!/usr/bin/perl
print "Content-type:text/html\r\n\r\n";
print '<html>';
print '<head>';
print '<meta charset="utf-8">';
print '<title>菜鸟教程(runoob.com)</title>';
print '</head>';
print '<body>';
print '<h2>Hello Word! </h2>';
print '<p>来自菜鸟教程第一个 CGI 程序。</p>';
print '</body>';
print '</html>';
1;
```

然后通过浏览器打开 <http://localhost/cgi-bin/test.cgi>，输出结果如下：



脚本第一行的输出内容"Content-type:text/html\r\n\r\n"发送到浏览器并告知浏览器显示的内容类型为"text/html"。

## HTTP头部

test.cgi文件内容中的" Content-type:text/html"即为HTTP头部的一部分，它会发送给浏览器告诉浏览器文件的内容类型。  
HTTP头部的格式如下：

HTTP 字段名： 字段内容
----------------

例如：

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

以下表格介绍了CGI程序中HTTP头部经常使用的信息：

头	描述
Content-type:	请求的与实体对应的MIME信息。例如: Content-type:text/html
Expires: Date	响应过期的日期和时间
Location: URL	用来重定向接收方到非请求URL的位置来完成请求或标识新的资源
Last-modified: Date	请求资源的最后修改时间
Content-length: N	请求的内容长度
Set-Cookie: String	设置Http Cookie

## CGI环境变量

所有的CGI程序都接收以下的环境变量，这些变量在CGI程序中发挥了重要的作用：

变量名	描述
CONTENT_TYPE	这个环境变量的值指示所传递来的信息的MIME类型。目前，环境变量CONTENT_TYPE一般都是：application/x-www-form-urlencoded,他表示数据来自于HTML表单。

CONTENT_LENGTH	如果服务器与CGI程序信息的传递方式是POST，这个环境变量即使从标准输入STDIN中可以读到的有效数据的字节数。这个环境变量在读取所输入的数据时必须使用。
HTTP_COOKIE	客户机内的 COOKIE 内容。
HTTP_USER_AGENT	提供包含了版本数或其他专有数据的客户浏览器信息。
PATH_INFO	这个环境变量的值表示紧接在CGI程序名之后的其他路径信息。它常常作为CGI程序的参数出现。
QUERY_STRING	如果服务器与CGI程序信息的传递方式是GET，这个环境变量的值即使所传递的信息。这个信息经跟在CGI程序名的后面，两者中间用一个问号'?'分隔。
REMOTE_ADDR	这个环境变量的值是发送请求的客户机的IP地址，例如上面的192.168.1.67。这个值总是存在的。而且它是Web客户机需要提供给Web服务器的唯一标识，可以在CGI程序中用它来区分不同的Web客户机。
REMOTE_HOST	这个环境变量的值包含发送CGI请求的客户机的主机名。如果不支持你想查询，则无需定义此环境变量。
REQUEST_METHOD	提供脚本被调用的方法。对于使用 HTTP/1.0 协议的脚本，仅 GET 和 POST 有意义。
SCRIPT_FILENAME	CGI脚本的完整路径
SCRIPT_NAME	CGI脚本的名称
SERVER_NAME	这是你的 WEB 服务器的主机名、别名或IP地址。
SERVER_SOFTWARE	这个环境变量的值包含了调用CGI程序的HTTP服务器的名称和版本号。例如，上面的值为Apache/2.2.14(Unix)

以下是一个简单的CGI脚本输出CGI的环境变量：

实例

```
#!/usr/bin/perl
print "Content-type: text/html\n\n";
print '<meta charset="utf-8">';
print "<font size=+1>环境变量：</font>\n";
foreach (sort keys %ENV)
{
    print "<b>$_</b>: $ENV{$_}<br>\n";
}
1;
```

文件下载

如果我们想通过 Perl CGI 实现文件下载，需要设置不同的头部信息，如下所示：

### 实例

```
#!/usr/bin/perl
# HTTP Header
print "Content-Type:application/octet-stream; name=\"FileName\"\\r\\n";
print "Content-Disposition: attachment; filename=\"FileName\"\\r\\n\\n";
# Actual File Content will go hear.
open( FILE, "<FileName" );
while(read(FILE, $buffer, 100) )
{
    print("$buffer");
}
```

## 使用GET方法传输数据

GET方法发送编码后的用户信息到服务端，数据信息包含在请求页面的URL上，以"?"号分割, 如下所示：

```
http://www.test.com/cgi-bin/test.cgi?key1=value1&key2=value2
```

有关 GET 请求的其他一些注释：

- GET 请求可被缓存
- GET 请求保留在浏览器历史记录中
- GET 请求可被收藏为书签
- GET 请求不应在处理敏感数据时使用
- GET 请求有长度限制
- GET 请求只应当用于取回数据

### 简单的url实例：GET方法

以下是一个简单的URL，使用GET方法向test.cgi程序发送两个参数：

```
/cgi-bin/test.cgi?name=菜鸟教程&url=http://www.runoob.com
```

以下为test.cgi文件的代码：

### 实例

```
#!/usr/bin/perl
local ($buffer, @pairs, $pair, $name, $value, %FORM);
# 读取文本信息
$ENV{'REQUEST_METHOD'} =~ tr/a-z/A-Z/;
if ($ENV{'REQUEST_METHOD'} eq "GET")
{
    $buffer = $ENV{'QUERY_STRING'};
}
```

```
# 读取 name/value 对信息
@pairs = split(/&/, $buffer);
foreach $pair (@pairs)
{
    ($name, $value) = split(/=/, $pair);
    $value =~ tr/+// ;
    $value =~ s/%(..)/pack("C", hex($1))/eg;
    $FORM{$name} = $value;
}
$name = $FORM{name};
$url = $FORM{url};
print "Content-type:text/html\r\n\r\n";
print "<html>";
print "<head>";
print '<meta charset="utf-8">';
print '<title>菜鸟教程(runoob.com)</title>';
print "</head>";
print "<body>";
print "<h2>$name网址: $url</h2>";
print "</body>";
print "</html>";
1;
```

查看浏览器，输出结果如下：



## 简单的表单实例：GET方法

以下是一个通过HTML的表单使用GET方法向服务器发送两个数据，提交的服务器脚本同样是test.cgi文件，test.html 代码如下：

### test.html 文件代码

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>菜鸟教程(runoob.com)</title>
</head>
<body>
<form action="/cgi-bin/test.cgi" method="get">
  站点名称: <input type="text" name="name"> <br />
  站点 URL: <input type="text" name="url" />
  <input type="submit" value="提交" />
</form>
```

```
</body>
</html>
```

浏览器中，执行效果如下所示：



## 使用POST方法传递数据

使用POST方法向服务器传递数据是更安全可靠，像一些敏感信息如用户密码等需要使用POST传输数据。

以下同样是 test.cgi ，它也可以处理浏览器提交的POST表单数据：

### test.cgi 代码

```
#!/usr/bin/perl
local ($buffer, @pairs, $pair, $name, $value, %FORM);
# 读取文本信息
$ENV{'REQUEST_METHOD'} =~ tr/a-z/A-Z/;
if ($ENV{'REQUEST_METHOD'} eq "POST")
{
    read(STDIN, $buffer, $ENV{'CONTENT_LENGTH'});
}else {
    $buffer = $ENV{'QUERY_STRING'};
}
# 读取 name/value 对信息
@pairs = split(/&/, $buffer);
foreach $pair (@pairs)
{
    ($name, $value) = split(/=/, $pair);
    $value =~ tr/+/ /;
    $value =~ s/%(..)/pack("C", hex($1))/eg;
    $FORM{$name} = $value;
}
$name = $FORM{name};
$url = $FORM{url};
print "Content-type:text/html\r\n\r\n";
print "<html>";
print "<head>";
print '<meta charset="utf-8">';
print '<title>菜鸟教程(runoob.com)</title>';
print "</head>";
print "<body>";
print "<h2>$name网址: $url</h2>";
print "</body>";
```

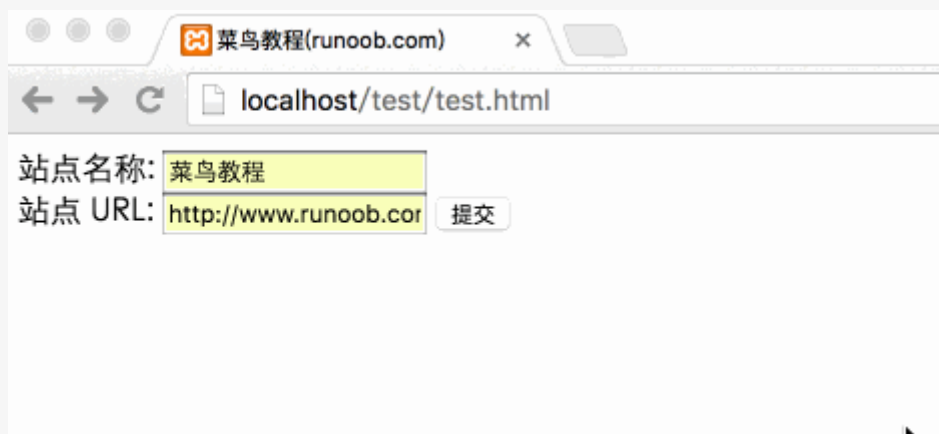
```
print "</html>";  
1;
```

以下是一个通过HTML的表单使用GET方法向服务器发送两个数据，提交的服务器脚本同样是test.cgi文件，test.html 代码如下：

#### test.html 代码

```
<!DOCTYPE html>  
<html>  
<head>  
<meta charset="utf-8">  
<title>菜鸟教程(runoob.com)</title>  
</head>  
<body>  
<form action="/cgi-bin/test.cgi" method="post">  
  站点名称: <input type="text" name="name"> <br />  
  站点 URL: <input type="text" name="url" />  
  <input type="submit" value="提交" />  
</form>  
</body>  
</html>
```

浏览器中，执行效果如下所示：



## 通过CGI程序传递checkboxbox数据

checkboxbox用于提交一个或者多个选项数据，test.html 代码如下：

#### test.html 代码

```
<!DOCTYPE html>  
<html>  
<head>  
<meta charset="utf-8">  
<title>菜鸟教程(runoob.com)</title>  
</head>  
<body>  
<form action="/cgi-bin/test.cgi" method="POST" target="_blank">  
  <input type="checkbox" name="runoob" value="on" /> 菜鸟教程  
  <input type="checkbox" name="google" value="on" /> Google  
  <input type="submit" value="选择站点" />  
</form>
```



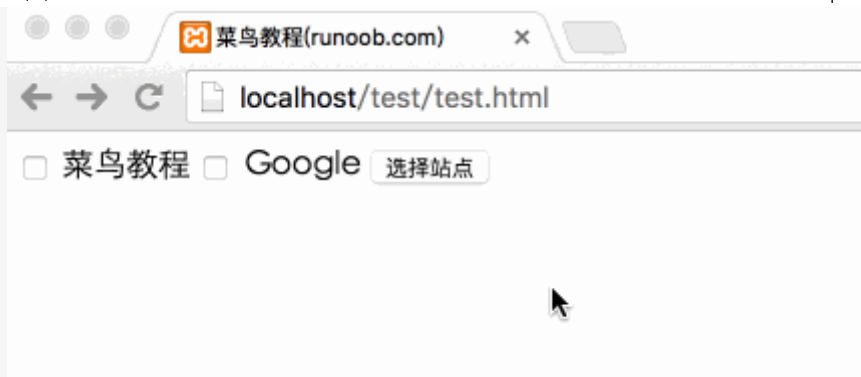
```
</form>
</body>
</html>
```

以下为 test.cgi 文件的代码：

### test.cgi 代码

```
#!/usr/bin/perl
local ($buffer, @pairs, $pair, $name, $value, %FORM);
# 读取信息
$ENV{'REQUEST_METHOD'} =~ tr/a-z/A-Z/;
if ($ENV{'REQUEST_METHOD'} eq "POST")
{
    read(STDIN, $buffer, $ENV{'CONTENT_LENGTH'});
}else {
    $buffer = $ENV{'QUERY_STRING'};
}
# 读取 name/value 对信息
@pairs = split(/&/, $buffer);
foreach $pair (@pairs)
{
    ($name, $value) = split(/=/, $pair);
    $value =~ tr/+//;
    $value =~ s/%(..)/pack("C", hex($1))/eg;
    $FORM{$name} = $value;
}
if( $FORM{runoob} ){
    $runoob_flag = "ON";
}else{
    $runoob_flag = "OFF";
}
if( $FORM{google} ){
    $google_flag = "ON";
}else{
    $google_flag = "OFF";
}
print "Content-type:text/html\r\n\r\n";
print "<html>";
print "<head>";
print '<meta charset="utf-8">';
print '<title>菜鸟教程(runoob.com)</title>';
print "</head>";
print "<body>";
print "<h2> 菜鸟教程选中状态 : $runoob_flag</h2>";
print "<h2> Google 选择状态 : $google_flag</h2>";
print "</body>";
print "</html>";
1;
```

浏览器中，执行效果如下所示：



## 通过CGI程序传递Radio数据

Radio 只向服务器传递一个数据，test.html 代码如下：

### test.html 代码

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>菜鸟教程(runoob.com)</title>
</head>
<body>
<form action="/cgi-bin/test.cgi" method="post" target="_blank">
<input type="radio" name="site" value="runoob" /> 菜鸟教程
<input type="radio" name="site" value="google" /> Google
<input type="submit" value="提交" />
</form>
</body>
</html>
```

test.cgi 脚本代码如下：

### test.cgi 代码

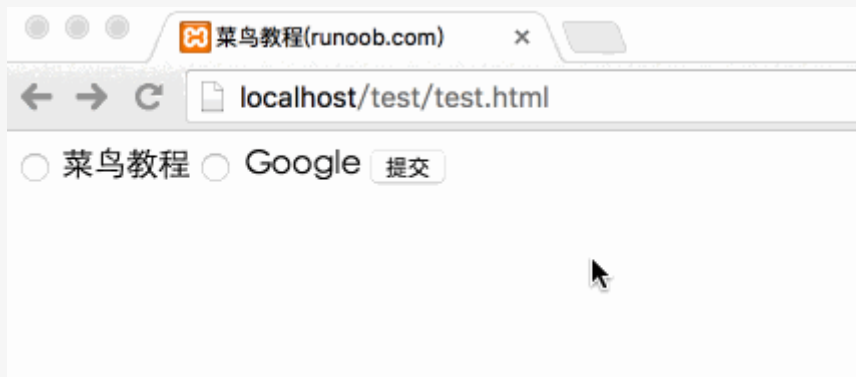
```
#!/usr/bin/perl
local ($buffer, @pairs, $pair, $name, $value, %FORM);
# 读取信息
$ENV{'REQUEST_METHOD'} =~ tr/a-z/A-Z/;
if ($ENV{'REQUEST_METHOD'} eq "POST")
{
read(STDIN, $buffer, $ENV{'CONTENT_LENGTH'});
}else {
$buffer = $ENV{'QUERY_STRING'};
}
# 读取 name/value 对信息
@pairs = split(/&/, $buffer);
foreach $pair (@pairs)
{
($name, $value) = split(/=/, $pair);
$value =~ tr/+//;
$value =~ s/%(..)/pack("C", hex($1))/eg;
$FORM{$name} = $value;
}
```

```

}
$site = $FORM{site};
print "Content-type:text/html\r\n\r\n";
print "<html>";
print "<head>";
print '<meta charset="utf-8">';
print '<title>菜鸟教程(runoob.com)</title>';
print "</head>";
print "<body>";
print "<h2> 选择的网站 $site</h2>";
print "</body>";
print "</html>";
1;

```

浏览器中，执行效果如下所示：



## 通过CGI程序传递 Textarea 数据

Textarea 向服务器传递多行数据，test.html 代码如下：

### test.html 代码

```

<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>菜鸟教程(runoob.com)</title>
</head>
<body>
<form action="/cgi-bin/test.cgi" method="post" target="_blank">
<textarea name="textcontent" cols="40" rows="4">
在这里输入内容...
</textarea>
<input type="submit" value="提交" />
</form>
</body>
</html>

```

test.cgi 脚本代码如下：

### test.cgi 代码

```

#!/usr/bin/perl
local ($buffer, @pairs, $pair, $name, $value, %FORM);

```

```
# 读取信息
$ENV{'REQUEST_METHOD'} =~ tr/a-z/A-Z/;
if ($ENV{'REQUEST_METHOD'} eq "POST")
{
    read(STDIN, $buffer, $ENV{'CONTENT_LENGTH'});
}else {
    $buffer = $ENV{'QUERY_STRING'};
}
# 读取 name/value 对信息
@pairs = split(/&/, $buffer);
foreach $pair (@pairs)
{
    ($name, $value) = split(/=/, $pair);
    $value =~ tr/+/ /;
    $value =~ s/%(..)/pack("C", hex($1))/eg;
    $FORM{$name} = $value;
}
$text_content = $FORM{textcontent};
print "Content-type:text/html\r\n\r\n";
print "<html>";
print "<head>";
print '<meta charset="utf-8">';
print '<title>菜鸟教程(runoob.com)</title>';
print "</head>";
print "<body>";
print "<h2>输入的文本内容为: $text_content</h2>";
print "</body>";
print "</html>";
1;
```

浏览器中，执行效果如下所示：



## 通过 CGI 程序传递下拉数据

HTML 下拉框代码如下：

### test.html 代码

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>菜鸟教程(runoob.com)</title>
```

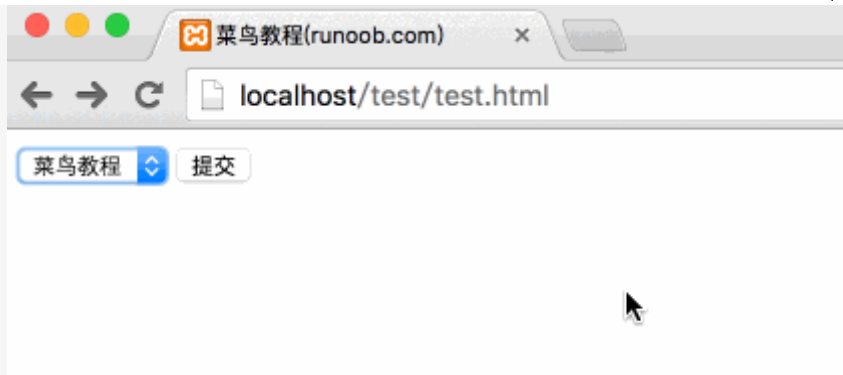
```
</head>
<body>
<form action="/cgi-bin/test.cgi" method="post" target="_blank">
<select name="dropdown">
<option value="runoob" selected>菜鸟教程</option>
<option value="google">Google</option>
</select>
<input type="submit" value="提交"/>
</form>
</body>
</html>
```

test.cgi 脚本代码如下所示：

### test.cgi 代码

```
#!/usr/bin/perl
local ($buffer, @pairs, $pair, $name, $value, %FORM);
# 读取信息
$ENV{'REQUEST_METHOD'} =~ tr/a-z/A-Z/;
if ($ENV{'REQUEST_METHOD'} eq "POST")
{
read(STDIN, $buffer, $ENV{'CONTENT_LENGTH'});
}else {
$buffer = $ENV{'QUERY_STRING'};
}
# 读取 name/value 对信息
@pairs = split(/&/, $buffer);
foreach $pair (@pairs)
{
($name, $value) = split(/=/, $pair);
$value =~ tr/+//;
$value =~ s/%(..)/pack("C", hex($1))/eg;
$FORM{$name} = $value;
}
$site = $FORM{dropdown};
print "Content-type:text/html\r\n\r\n";
print "<html>";
print "<head>";
print '<meta charset="utf-8">';
print '<title>菜鸟教程(runoob.com)</title>';
print "</head>";
print "<body>";
print "<h2>选择的网站是: $site</h2>";
print "</body>";
print "</html>";
1;
```

浏览器中，执行效果如下所示：



## CGI中使用Cookie

在 http 协议一个很大的缺点就是不对用户身份的的判断，这样给编程人员带来很大的不便，而 cookie 功能的出现弥补了这个不足。

cookie 就是在客户访问脚本的同时，通过客户的浏览器，在客户硬盘上写入纪录数据，当下次客户访问脚本时取回数据信息，从而达到身份判别的功能，cookie 常用在身份校验中。

### cookie的语法

http cookie的发送是通过http头部来实现的，他早于文件的传递，头部set-cookie的语法如下：

```
Set-cookie:name=name;expires=date;path=path;domain=domain;secure
```

- **name=name:** 需要设置cookie的值(name不能使用";"和";"),有多个name值时用 ";" 分隔，例如：**name1=name1;name2=name2;name3=name3**。
- **expires=date:** cookie的有效期限,格式： expires="Wdy,DD-Mon-YYYY HH:MM:SS"
- **path=path:** 设置cookie支持的路径,如果path是一个路径，则cookie对这个目录下的所有文件及子目录生效，例如： path ="/cgi-bin/"，如果path是一个文件，则cookie指对这个文件生效，例如：path="/cgi-bin/cookie.cgi"。
- **domain=domain:** 对cookie生效的域名，例如：domain="www.runoob.com"
- **secure:** 如果给出此标志，表示cookie只能通过SSL协议的https服务器来传递。
- cookie的接收是通过设置环境变量HTTP\_COOKIE来实现的，CGI程序可以通过检索该变量获取cookie信息。

## Cookie设置

Cookie的设置非常简单，cookie会在http头部单独发送。以下实例在cookie中设置了UserID、Password 和 expires：

### 实例

```
#!/usr/bin/perl
print "Set-Cookie:UserID=XYZ;\n";
print "Set-Cookie:Password=XYZ123;\n";
print "Set-Cookie:Expires=Tuesday, 31-Dec-2017 23:12:40 GMT;\n";
print "Set-Cookie:Domain=www.runoob.com;\n";
print "Set-Cookie:Path=/perl;\n";
```

```
print "Content-type:text/html\r\n\r\n";
.....其他 HTML 内容
```

## 查找 Cookie

Cookie信息检索页非常简单，Cookie信息存储在CGI的环境变量HTTP\_COOKIE中，存储格式如下：

### 实例

```
#!/usr/bin/perl
$rcvd_cookies = $ENV{'HTTP_COOKIE'};
@cookies = split /;/, $rcvd_cookies;
foreach $cookie ( @cookies ){
($key, $val) = split(/=/, $cookie); # splits on the first =.
$key =~ s/^\s+//;
$val =~ s/^\s+//;
$key =~ s/\s+$//;
$val =~ s/\s+$//;
if( $key eq "UserID" ){
$user_id = $val;
}elseif($key eq "Password"){
$password = $val;
}
}
print "User ID = $user_id\n";
print "Password = $password\n";
```

以上实例输出结果为：

```
User ID = XYZ
Password = XYZ123
```

## CGI 模块

Perl 提供了很多内置的 CGI 模块，常用以下两个：

- [CGI 模块](#)
- [Berkeley\\_cgi-lib.pl](#)

← Perl 数据库连接

Perl 包和模块 →

 点我分享笔记

