

Ruby CGI方法

以下为CGI类的方法列表：

序号	方法描述
1	CGI::new([level="query"]) 创建 CGI 对象。query可以是以下值： <ul style="list-style-type: none">● query: 没有 HTML 生成输出● html3: HTML3.2● html4: HTML4.0 Strict● html4Tr: HTML4.0 Transitional● html4Fr: HTML4.0 Frameset
2	CGI::escape(str) 使用 URL 编码来转义字符串
3	CGI::unescape(str) 对通过 escape() 编码的字符串进行解码。
4	CGI::escapeHTML(str) 编码 HTML 特殊字符, 包括: & < >。
5	CGI::unescapeHTML(str) 解码 HTML 特殊字符, 包括: & < >。
6	CGI::escapeElement(str[, element...]) 在指定的 HTML 元素中编码 HTML 特殊字符。
7	CGI::unescapeElement(str, element[, element...]) 在指定的 HTML 元素中解码 HTML 特殊字符。
8	CGI::parse(query) 解析查询字符串, 并返回包含哈希的 键=》值 对。
9	CGI::pretty(string[, leader=" "]) 返回整齐的HTML格式。 如果指定了 <i>leader</i> , 它将写入到每一行的开头。 <i>leader</i> 默认值为两个空格。

10	CGI::rfc1123_date(time) 根据 RFC-1123 来格式化时间 (例如, Tue, 2 Jun 2008 00:00:00 GMT)。
----	---

CGI 实例化方法

以下实例中我们将 CGI::new 的对象赋值给 c 变量，方法列表如下：

序号	方法描述
1	c[name] 返回一个数组，包含了对应字段名为 <i>name</i> 的值。
2	c.checkbox(name[, value[, check=false]]) c.checkbox(options) 返回 HTML 字符串用于定义 checkbox 字段。标签的属性可以以一个哈希函数作为参数传递。
3	c.checkbox_group(name, value...) c.checkbox_group(options) >返回 HTML 字符串用于定义 checkbox 组。标签的属性可以以一个哈希函数作为参数传递。
4	c.file_field(name[, size=20[, max]]) c.file_field(options) 返回定义 file 字段的HTML字符串。
5	c.form([method="post"[, url]]) { ...} c.form(options) 返回定义 form 表单的HTML字符串。如果指定了代码块，将作为表单内容输出。标签的属性可以以一个哈希函数作为参数传递。
6	c.cookies 返回 CGI::Cookie 对象，包含了cookie 中的键值对。
7	c.header([header]) 返回 CGI 头部的信息。如果 header 参数是哈希值，其键 - 值对，用于创建头部信息。
8	c.hidden(name[, value]) c.hidden(options) 返回定义一个隐藏字段的HTML字符串。标签的属性可以以一个哈希函数作为参数传递。
9	c.image_button(url[, name[, alt]]) c.image_button(options) 返回定义一个图像按钮的HTML字符串。标签的属性可以以一个哈希函数作为参数传递。

10	c.keys 返回一个数组，包含了表单的字段名。
11	c.key?(name) c.has_key?(name) c.include?(name) 如果表单包含了指定的字段名返回 true。
12	c.multipart_form([url[, encode]]) { ...} c.multipart_form(options) { ...} 返回定义一个多媒体表单（ multipart ）的HTML字符串。标签的属性可以以一个哈希函数作为参数传递。
13	c.out([header]) { ...} 生成 HTML 并输出。使用由块的输出来创建页面的主体生成的字符串。
14	c.params 返回包含表单字段名称和值的哈希值。
15	c.params= hash 设置使用字段名和值。
16	c.password_field(name[, value[, size=40[, max]]]) c.password_field(options) 返回定义一个password字段的HTML字符串。标签的属性可以以一个哈希函数作为参数传递。
17	c.popup_menu(name, value...) c.popup_menu(options) c.scrolling_list(name, value...) c.scrolling_list(options) 返回定义一个弹出式菜单的HTML字符串。标签的属性可以以一个哈希函数作为参数传递。
18	c.radio_button(name[, value[, checked=false]]]) c.radio_button(options) 返回定义一个radio字段的HTML字符串。标签的属性可以以一个哈希函数作为参数传递。
19	c.radio_group(name, value...) c.radio_group(options) 返回定义一个radio按钮组的HTML字符串。标签的属性可以以一个哈希函数作为参数传递。
20	c.reset(name[, value]) c.reset(options)

	返回定义一个reset按钮的HTML字符串。 标签的属性可以以一个哈希函数作为参数传递
21	c.text_field(name[, value[, size=40[, max]]]) c.text_field(options) 返回定义一个text字段的HTML字符串。标签的属性可以以一个哈希函数作为参数传递。
22	c.textarea(name[, cols=70[, rows=10]] { ...} c.textarea(options) { ...} 返回定义一个textarea字段的HTML字符串。 如果指定了块，代码块输出的字符串将作为 textarea 的内容。 标签的属性可以以一个哈希函数作为参数传递。

HTML 生成方法

你可以再 CGI 实例中使用相应的 HTML 标签名来创建 HTML 标签，实例如下：

实例

```
#!/usr/bin/ruby
require "cgi"
cgi = CGI.new("html4")
cgi.out{
  cgi.html{
    cgi.head{ "\n"+cgi.title{"This Is a Test"} } +
    cgi.body{ "\n"+
    cgi.form{"\n"+
    cgi.hr +
    cgi.h1 { "A Form: " } + "\n"+
    cgi.textarea("get_text") +"\n"+
    cgi.br +
    cgi.submit
    }
  }
}
```

CGI 对象属性

你可以再 CGI 实例中使用以下属性：

属性	返回值
accept	可接受的 MIME 类型
accept_charset	可接受的字符集
accept_encoding	可接受的编码
accept_language	可接受的语言

auth_type	可接受的类型
raw_cookie	Cookie 数据 (原字符串)
content_length	内容长度 (Content length)
content_type	内容类型 (Content type)
From	Client e-mail 地址
gateway_interface	CGI 版本
path_info	路径
path_translated	转换后的路径
Query_string	查询字符串
referer	之前访问网址
remote_addr	客户端主机地址(IP)
remote_host	客户端主机名
remote_ident	客户端名
remote_user	经过身份验证的用户
request_method	请求方法(GET, POST, 等。)
script_name	参数名
server_name	服务器名
server_port	服务器端口
server_protocol	服务器协议
server_software	服务器软件
user_agent	用户代理 (User agent)

