<https://www.tutorialspoint.com/python/python_cgi_programming.htm>

## Resumo

A Common Gateway Interface, ou CGI, é um conjunto de padrões que definem como as informações são trocadas entre o servidor da web e um script personalizado. As especificações CGI são mantidas atualmente pela NCSA.

## O que é CGI?

* A Common Gateway Interface, ou CGI, é um padrão para programas de gateway externo fazerem interface com servidores de informações, como servidores HTTP.
* A versão atual é CGI / 1.1 e CGI / 1.2 está em andamento.

## Navegação na web

Para entender o conceito de CGI, vamos ver o que acontece quando clicamos em um hiperlink para navegar em uma determinada página da web ou URL.

* Seu navegador entra em contato com o servidor da web HTTP e exige a URL, ou seja, o nome do arquivo.
* O servidor da Web analisa o URL e procura o nome do arquivo. Se ele encontrar esse arquivo, o enviará de volta ao navegador; caso contrário, enviará uma mensagem de erro indicando que você solicitou um arquivo errado.
* O navegador da Web obtém a resposta do servidor da Web e exibe o arquivo recebido ou a mensagem de erro.

No entanto, é possível configurar o servidor HTTP de forma que sempre que um arquivo em um determinado diretório for solicitado, esse arquivo não seja enviado de volta; em vez disso, ele é executado como um programa e tudo o que esse programa produz é enviado de volta para o seu navegador exibir. Esta função é chamada de Interface de Gateway Comum ou CGI e os programas são chamados de scripts CGI. Esses programas CGI podem ser Python Script, PERL Script, Shell Script, programa C ou C ++, etc.

## Diagrama de Arquitetura CGI



## Suporte e configuração do servidor web

Antes de prosseguir com a Programação CGI, certifique-se de que seu servidor da Web suporta CGI e está configurado para lidar com programas CGI. Todos os programas CGI a serem executados pelo servidor HTTP são mantidos em um diretório pré-configurado. Este diretório é denominado Diretório CGI e por convenção é denominado / var / www / cgi-bin. Por convenção, os arquivos CGI têm extensão como. cgi, mas você também pode manter seus arquivos com a extensão python .py .

Por padrão, o servidor Linux é configurado para executar apenas os scripts no diretório cgi-bin em / var / www. Se você deseja especificar qualquer outro diretório para executar seus scripts CGI, comente as seguintes linhas no arquivo httpd.conf -

<Directory "/var/www/cgi-bin">

AllowOverride None

Options ExecCGI

Order allow,deny

Allow from all

</Directory>

<Directory "/var/www/cgi-bin">

Options All

</Directory>

Aqui, presumimos que você tenha o servidor da Web instalado e funcionando com êxito e seja capaz de executar qualquer outro programa CGI, como Perl ou Shell, etc.

## Primeiro Programa CGI

Aqui está um link simples, que está vinculado a um script CGI chamado [hello.py](https://www.tutorialspoint.com/cgi-bin/hello.py) . Este arquivo é mantido no diretório / var / www / cgi-bin e possui o seguinte conteúdo. Antes de executar seu programa CGI, certifique-se de alterar o modo de arquivo usando o comando chmod 755 hello.py UNIX para tornar o arquivo executável.

#!/usr/bin/python

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

print '<html>'

print '<head>'

print '<title>Hello World - First CGI Program</title>'

print '</head>'

print '<body>'

print '<h2>Hello World! This is my first CGI program</h2>'

print '</body>'

print '</html>'

Se você clicar em hello.py, isso produzirá a seguinte saída -

|  |
| --- |
| Olá Mundo! Este é o meu primeiro programa CGI |

Este script hello.py é um script Python simples, que grava sua saída no arquivo STDOUT, ou seja, na tela. Há um recurso importante e extra disponível que é a primeira linha a ser impressa Tipo de conteúdo: texto / html \ r \ n \ r \ n . Essa linha é enviada de volta ao navegador e especifica o tipo de conteúdo a ser exibido na tela do navegador.

Agora você deve ter entendido o conceito básico de CGI e pode escrever muitos programas CGI complicados usando Python. Este script pode interagir com qualquer outro sistema externo também para trocar informações como RDBMS.

## Cabeçalho HTTP

A linha Content-type: text / html \ r \ n \ r \ n faz parte do cabeçalho HTTP que é enviado ao navegador para entender o conteúdo. Todo o cabeçalho HTTP estará no seguinte formato -

HTTP Field Name: Field Content

For Example

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

Existem alguns outros cabeçalhos HTTP importantes, que você usará com freqüência em sua programação CGI.

|  |  |
| --- | --- |
| **Sr. Não.** | **Cabeçalho e Descrição** |
| 1 | Tipo de conteúdo:  Uma string MIME que define o formato do arquivo que está sendo retornado. O exemplo é o tipo de conteúdo: text / html |
| 2 | Expira em: Data  A data em que a informação se torna inválida. É usado pelo navegador para decidir quando uma página precisa ser atualizada. Uma string de data válida está no formato 01 de janeiro de 1998 12:00:00 GMT. |
| 3 | Localização: URL  O URL que é retornado em vez do URL solicitado. Você pode usar este campo para redirecionar uma solicitação para qualquer arquivo. |
| 4 | Última modificação: Data  A data da última modificação do recurso. |
| 5 | Comprimento do conteúdo: N  O comprimento, em bytes, dos dados que estão sendo retornados. O navegador usa esse valor para relatar o tempo estimado de download de um arquivo. |
| 6 | Set-Cookie: String  Defina o cookie passado pela *string* |

## Variáveis ​​de ambiente CGI

Todos os programas CGI têm acesso às seguintes variáveis ​​de ambiente. Essas variáveis ​​desempenham um papel importante ao escrever qualquer programa CGI.

|  |  |
| --- | --- |
| **Sr. Não.** | **Nome e descrição da variável** |
| 1 | TIPO DE CONTEÚDO  O tipo de dados do conteúdo. Usado quando o cliente está enviando conteúdo anexado ao servidor. Por exemplo, upload de arquivo. |
| 2 | COMPRIMENTO DO CONTEÚDO  O comprimento das informações da consulta. Ele está disponível apenas para solicitações POST. |
| 3 | HTTP\_COOKIE  Retorna os cookies definidos na forma de par de chave e valor. |
| 4 | HTTP\_USER\_AGENT  O campo de cabeçalho de solicitação do Agente do Usuário contém informações sobre o agente do usuário que originou a solicitação. É o nome do navegador da web. |
| 5 | PATH\_INFO  O caminho para o script CGI. |
| 6 | QUERY\_STRING  As informações codificadas por URL que são enviadas com a solicitação do método GET. |
| 7 | REMOTE\_ADDR  O endereço IP do host remoto que está fazendo a solicitação. Este é um registro útil ou para autenticação. |
| 8 | HOSPEDEIRO REMOTO  O nome totalmente qualificado do host que está fazendo a solicitação. Se esta informação não estiver disponível, REMOTE\_ADDR pode ser usado para obter o endereço de IR. |
| 9 | REQUEST\_METHOD  O método usado para fazer a solicitação. Os métodos mais comuns são GET e POST. |
| 10 | SCRIPT\_FILENAME  O caminho completo para o script CGI. |
| 11 | SCRIPT\_NAME  O nome do script CGI. |
| 12 | NOME DO SERVIDOR  O nome de host ou endereço IP do servidor |
| 13 | SERVER\_SOFTWARE  O nome e a versão do software que o servidor está executando. |

Aqui está um pequeno programa CGI para listar todas as variáveis ​​CGI. Clique neste link para ver o resultado [Get Environment](http://www.tutorialspoint.com/cgi-bin/get_env.py)

#!/usr/bin/python

import os

print "Content-type: text/html\r\n\r\n";

print "<font size=+1>Environment</font><\br>";

for param in os.environ.keys():

print "<b>%20s</b>: %s<\br>" % (param, os.environ[param])

## Métodos GET e POST

Você deve ter se deparado com muitas situações em que precisa passar algumas informações de seu navegador para o servidor da web e, finalmente, para o seu programa CGI. Mais freqüentemente, o navegador usa dois métodos: dois para passar essas informações para o servidor da web. Esses métodos são o Método GET e o Método POST.

## Passando informações usando o método GET

O método GET envia as informações codificadas do usuário anexadas à solicitação de página. A página e as informações codificadas são separadas por? personagem da seguinte forma -

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

O método GET é o método padrão para passar informações do navegador para o servidor web e produz uma longa string que aparece na caixa Location: do seu navegador. Nunca use o método GET se você tiver uma senha ou outras informações confidenciais para passar para o servidor. O método GET tem limitação de tamanho: apenas 1024 caracteres podem ser enviados em uma string de solicitação. O método GET envia informações usando o cabeçalho QUERY\_STRING e estará acessível em seu Programa CGI através da variável de ambiente QUERY\_STRING.

Você pode passar informações simplesmente concatenando pares de chave e valor junto com qualquer URL ou pode usar tags HTML <FORM> para passar informações usando o método GET.

## Exemplo de URL simples: Método Get

Aqui está uma URL simples, que passa dois valores para o programa hello\_get.py usando o método GET.

[/cgi-bin/hello\_get.py?first\_name=ZARA&last\_name=ALI](https://www.tutorialspoint.com/cgi-bin/hello_get.py?first_name=ZARA&last_name=ALI)

Abaixo está o script hello\_get.py para lidar com a entrada fornecida pelo navegador da web. Vamos usar o módulo cgi , o que torna muito fácil acessar as informações passadas -

#!/usr/bin/python

# Import modules for CGI handling

import cgi, cgitb

# Create instance of FieldStorage

form = cgi.FieldStorage()

# Get data from fields

first\_name = form.getvalue('first\_name')

last\_name = form.getvalue('last\_name')

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

print "<html>"

print "<head>"

print "<title>Hello - Second CGI Program</title>"

print "</head>"

print "<body>"

print "<h2>Hello %s %s</h2>" % (first\_name, last\_name)

print "</body>"

print "</html>"

Isso geraria o seguinte resultado -

|  |
| --- |
| Olá ZARA ALI |

## Exemplo de FORM Simples: Método GET

Este exemplo passa dois valores usando HTML FORM e botão de envio. Usamos o mesmo script CGI hello\_get.py para lidar com essa entrada.

<form action = "/cgi-bin/hello\_get.py" method = "get">

First Name: <input type = "text" name = "first\_name"> <br />

Last Name: <input type = "text" name = "last\_name" />

<input type = "submit" value = "Submit" />

</form>

Aqui está o resultado real do formulário acima, você insere o nome e o sobrenome e, a seguir, clica no botão enviar para ver o resultado.

Primeiro nome:

Último nome:

## Passando informações usando o método POST

Um método geralmente mais confiável de passar informações para um programa CGI é o método POST. Isso empacota as informações exatamente da mesma maneira que os métodos GET, mas em vez de enviá-las como uma string de texto após um? no URL, ele o envia como uma mensagem separada. Esta mensagem chega ao script CGI na forma de entrada padrão.

Abaixo está o mesmo script hello\_get.py que lida com GET e também com o método POST.

#!/usr/bin/python

# Import modules for CGI handling

import cgi, cgitb

# Create instance of FieldStorage

form = cgi.FieldStorage()

# Get data from fields

first\_name = form.getvalue('first\_name')

last\_name = form.getvalue('last\_name')

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

print "<html>"

print "<head>"

print "<title>Hello - Second CGI Program</title>"

print "</head>"

print "<body>"

print "<h2>Hello %s %s</h2>" % (first\_name, last\_name)

print "</body>"

print "</html>"

Tomemos novamente o mesmo exemplo acima, que passa dois valores usando HTML FORM e o botão submit. Usamos o mesmo script CGI hello\_get.py para lidar com essa entrada.

<form action = "/cgi-bin/hello\_get.py" method = "post">

First Name: <input type = "text" name = "first\_name"><br />

Last Name: <input type = "text" name = "last\_name" />

<input type = "submit" value = "Submit" />

</form>

Aqui está a saída real do formulário acima. Você insere o nome e o sobrenome e, em seguida, clica no botão enviar para ver o resultado.

Primeiro nome:

Último nome:

## Passando os dados da caixa de seleção para o programa CGI

As caixas de seleção são usadas quando mais de uma opção deve ser selecionada.

Aqui está um exemplo de código HTML para um formulário com duas caixas de seleção -

<form action = "/cgi-bin/checkbox.cgi" method = "POST" target = "\_blank">

<input type = "checkbox" name = "maths" value = "on" /> Maths

<input type = "checkbox" name = "physics" value = "on" /> Physics

<input type = "submit" value = "Select Subject" />

</form>

O resultado deste código é o seguinte formato -

Matemáticas Física

Abaixo está o script checkbox.cgi para lidar com a entrada fornecida pelo navegador da web para o botão de caixa de seleção.

#!/usr/bin/python

# Import modules for CGI handling

import cgi, cgitb

# Create instance of FieldStorage

form = cgi.FieldStorage()

# Get data from fields

if form.getvalue('maths'):

math\_flag = "ON"

else:

math\_flag = "OFF"

if form.getvalue('physics'):

physics\_flag = "ON"

else:

physics\_flag = "OFF"

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

print "<html>"

print "<head>"

print "<title>Checkbox - Third CGI Program</title>"

print "</head>"

print "<body>"

print "<h2> CheckBox Maths is : %s</h2>" % math\_flag

print "<h2> CheckBox Physics is : %s</h2>" % physics\_flag

print "</body>"

print "</html>"

## Passando Dados do Botão de Rádio para o Programa CGI

Os botões de rádio são usados ​​quando apenas uma opção deve ser selecionada.

Aqui está um exemplo de código HTML para um formulário com dois botões de opção -

<form action = "/cgi-bin/radiobutton.py" method = "post" target = "\_blank">

<input type = "radio" name = "subject" value = "maths" /> Maths

<input type = "radio" name = "subject" value = "physics" /> Physics

<input type = "submit" value = "Select Subject" />

</form>

O resultado deste código é o seguinte formato -

Matemáticas Física

Abaixo está o script radiobutton.py para lidar com a entrada fornecida pelo navegador da web para o botão de rádio -

#!/usr/bin/python

# Import modules for CGI handling

import cgi, cgitb

# Create instance of FieldStorage

form = cgi.FieldStorage()

# Get data from fields

if form.getvalue('subject'):

subject = form.getvalue('subject')

else:

subject = "Not set"

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

print "<html>"

print "<head>"

print "<title>Radio - Fourth CGI Program</title>"

print "</head>"

print "<body>"

print "<h2> Selected Subject is %s</h2>" % subject

print "</body>"

print "</html>"

## Passando dados da área de texto para o programa CGI

O elemento TEXTAREA é usado quando o texto de várias linhas deve ser passado para o programa CGI.

Aqui está um exemplo de código HTML para um formulário com uma caixa TEXTAREA -

<form action = "/cgi-bin/textarea.py" method = "post" target = "\_blank">

<textarea name = "textcontent" cols = "40" rows = "4">

Type your text here...

</textarea>

<input type = "submit" value = "Submit" />

</form>

O resultado deste código é o seguinte formato -

Abaixo está o script textarea.cgi para lidar com a entrada fornecida pelo navegador da web -

#!/usr/bin/python

# Import modules for CGI handling

import cgi, cgitb

# Create instance of FieldStorage

form = cgi.FieldStorage()

# Get data from fields

if form.getvalue('textcontent'):

text\_content = form.getvalue('textcontent')

else:

text\_content = "Not entered"

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

print "<html>"

print "<head>";

print "<title>Text Area - Fifth CGI Program</title>"

print "</head>"

print "<body>"

print "<h2> Entered Text Content is %s</h2>" % text\_content

print "</body>"

## Passando os dados da caixa suspensa para o programa CGI

A caixa suspensa é usada quando temos muitas opções disponíveis, mas apenas uma ou duas serão selecionadas.

Aqui está um exemplo de código HTML para um formulário com uma caixa suspensa -

<form action = "/cgi-bin/dropdown.py" method = "post" target = "\_blank">

<select name = "dropdown">

<option value = "Maths" selected>Maths</option>

<option value = "Physics">Physics</option>

</select>

<input type = "submit" value = "Submit"/>

</form>

O resultado deste código é o seguinte formato -

Matemáticas Física

Abaixo está o script dropdown.py para lidar com a entrada fornecida pelo navegador da web.

#!/usr/bin/python

# Import modules for CGI handling

import cgi, cgitb

# Create instance of FieldStorage

form = cgi.FieldStorage()

# Get data from fields

if form.getvalue('dropdown'):

subject = form.getvalue('dropdown')

else:

subject = "Not entered"

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

print "<html>"

print "<head>"

print "<title>Dropdown Box - Sixth CGI Program</title>"

print "</head>"

print "<body>"

print "<h2> Selected Subject is %s</h2>" % subject

print "</body>"

print "</html>"

## Usando Cookies em CGI

O protocolo HTTP é um protocolo sem estado. Para um site comercial, é necessário manter as informações da sessão entre as diferentes páginas. Por exemplo, o registro de um usuário termina após completar muitas páginas. Como manter as informações da sessão do usuário em todas as páginas da web?

Em muitas situações, o uso de cookies é o método mais eficiente de lembrar e rastrear preferências, compras, comissões e outras informações necessárias para uma melhor experiência do visitante ou estatísticas do site.

## Como funciona?

Seu servidor envia alguns dados para o navegador do visitante na forma de um cookie. O navegador pode aceitar o cookie. Em caso afirmativo, ele é armazenado como um registro de texto simples no disco rígido do visitante. Agora, quando o visitante chega a outra página do seu site, o cookie fica disponível para recuperação. Uma vez recuperado, seu servidor sabe / lembra o que foi armazenado.

Cookies são um registro de dados de texto simples de 5 campos de comprimento variável -

* Expira - A data em que o cookie irá expirar. Se estiver em branco, o cookie irá expirar quando o visitante sair do navegador.
* Domínio - o nome de domínio do seu site.
* Caminho - o caminho para o diretório ou página da web que define o cookie. Isso pode ficar em branco se você quiser recuperar o cookie de qualquer diretório ou página.
* Seguro - se este campo contiver a palavra "seguro", o cookie só pode ser recuperado com um servidor seguro. Se este campo estiver em branco, não existe tal restrição.
* Nome = valor - os cookies são definidos e recuperados na forma de pares de chave e valor.

## Configurando Cookies

É muito fácil enviar cookies para o navegador. Esses cookies são enviados junto com o cabeçalho HTTP antes do campo do tipo de conteúdo. Supondo que você deseja definir a ID do usuário e a senha como cookies. A configuração dos cookies é feita da seguinte forma -

#!/usr/bin/python

print "Set-Cookie:UserID = XYZ;\r\n"

print "Set-Cookie:Password = XYZ123;\r\n"

print "Set-Cookie:Expires = Tuesday, 31-Dec-2007 23:12:40 GMT";\r\n"

print "Set-Cookie:Domain = www.tutorialspoint.com;\r\n"

print "Set-Cookie:Path = /perl;\n"

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

...........Rest of the HTML Content....

A partir deste exemplo, você deve ter entendido como definir cookies. Usamos o cabeçalho HTTP Set-Cookie para definir cookies.

É opcional definir atributos de cookies como Expires, Domain e Path. É notável que os cookies são definidos antes de enviar a linha mágica "Tipo de conteúdo: texto / html \ r \ n \ r \ n .

## Recuperando Cookies

É muito fácil recuperar todos os cookies configurados. Os cookies são armazenados na variável de ambiente CGI HTTP\_COOKIE e terão a seguinte forma -

key1 = value1;key2 = value2;key3 = value3....

Aqui está um exemplo de como recuperar cookies.

#!/usr/bin/python

# Import modules for CGI handling

from os import environ

import cgi, cgitb

if environ.has\_key('HTTP\_COOKIE'):

for cookie in map(strip, split(environ['HTTP\_COOKIE'], ';')):

(key, value ) = split(cookie, '=');

if key == "UserID":

user\_id = value

if key == "Password":

password = value

print "User ID = %s" % user\_id

print "Password = %s" % password

Isso produz o seguinte resultado para os cookies definidos pelo script acima -

User ID = XYZ

Password = XYZ123

## Exemplo de upload de arquivo

Para fazer upload de um arquivo, o formulário HTML deve ter o atributo enctype definido como multipart / form-data . A tag de entrada com o tipo de arquivo cria um botão "Navegar".

<html>

<body>

<form enctype = "multipart/form-data"

action = "save\_file.py" method = "post">

<p>File: <input type = "file" name = "filename" /></p>

<p><input type = "submit" value = "Upload" /></p>

</form>

</body>

</html>

O resultado deste código é o seguinte formato -

Arquivo:

O exemplo acima foi desabilitado intencionalmente para salvar as pessoas fazendo upload de arquivos em nosso servidor, mas você pode tentar o código acima com seu servidor.

Aqui está o script save\_file.py para lidar com o upload do arquivo -

#!/usr/bin/python

import cgi, os

import cgitb; cgitb.enable()

form = cgi.FieldStorage()

# Get filename here.

fileitem = form['filename']

# Test if the file was uploaded

if fileitem.filename:

# strip leading path from file name to avoid

# directory traversal attacks

fn = os.path.basename(fileitem.filename)

open('/tmp/' + fn, 'wb').write(fileitem.file.read())

message = 'The file "' + fn + '" was uploaded successfully'

else:

message = 'No file was uploaded'

print """\

Content-Type: text/html\n

<html>

<body>

<p>%s</p>

</body>

</html>

""" % (message,)

Se você rodar o script acima em Unix / Linux, então você precisa tomar o cuidado de substituir o separador de arquivo como segue, caso contrário, em sua máquina Windows, a instrução open () deve funcionar bem.

fn = os.path.basename(fileitem.filename.replace("\\", "/" ))

## Como abrir uma caixa de diálogo "Download de arquivo"?

Às vezes, é desejável que você dê a opção de um usuário clicar em um link e abrir uma caixa de diálogo "Download de arquivo" para o usuário em vez de exibir o conteúdo real. Isso é muito fácil e pode ser feito por meio do cabeçalho HTTP. Este cabeçalho HTTP é diferente do cabeçalho mencionado na seção anterior.

Por exemplo, se você quiser que um arquivo FileName possa ser baixado de um determinado link, sua sintaxe é a seguinte -

#!/usr/bin/python

# 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 here.

fo = open("foo.txt", "rb")

str = fo.read();

print str

# Close opend file

fo.close()

Espero que tenha gostado desse tutorial. Em caso afirmativo, envie-me seus comentários em: [Fale conosco](https://www.tutorialspoint.com/about/contact_us.htm)