:mod:`cgi` --- Common Gateway Interface support

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)cgi.rst, line 1); *backlink*

Unknown interpreted text role "mod".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)cgi.rst, line 4)

Unknown directive type "module".

```
.. module:: cgi
   :synopsis: Helpers for running Python scripts via the Common Gateway Interface.
   :deprecated:
```

Source code: :source:`Lib/cgi.py`

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)cgi.rst, line 8); *backlink*

Unknown interpreted text role "source".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)cgi.rst, line 10)

Unknown directive type "index".

```
.. index::
   pair: WWW; server
   pair: CGI; protocol
   pair: HTTP; protocol
   pair: MIME; headers
   single: URL
   single: Common Gateway Interface
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)cgi.rst, line 18)

Unknown directive type "deprecated".

```
.. deprecated:: 3.11
   The :mod:`cgi` module is deprecated (see :pep:`594` for details).
```

Support module for Common Gateway Interface (CGI) scripts.

This module defines a number of utilities for use by CGI scripts written in Python.

The global variable `maxlen` can be set to an integer indicating the maximum size of a POST request. POST requests larger than this size will result in a :exc:`ValueError` being raised during parsing. The default value of this variable is `0`, meaning the request size is unlimited.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)cgi.rst, line 28); *backlink*

Unknown interpreted text role "exc".

Introduction

A CGI script is invoked by an HTTP server, usually to process user input submitted through an HTML `<FORM>` or `<ISINDEX>` element.

Most often, CGI scripts live in the server's special :file:`cgi-bin` directory. The HTTP server places all sorts of information about the request (such as the client's hostname, the requested URL, the query string, and lots of other goodies) in the script's shell environment, executes the script, and sends the script's output back to the client.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)cgi.rst, line 42); *backlink*

Unknown interpreted text role "file".

The script's input is connected to the client too, and sometimes the form data is read this way; at other times the form data is passed via the "query string" part of the URL. This module is intended to take care of the different cases and provide a simpler interface to

the Python script. It also provides a number of utilities that help in debugging scripts, and the latest addition is support for file uploads from a form (if your browser supports it).

The output of a CGI script should consist of two sections, separated by a blank line. The first section contains a number of headers, telling the client what kind of data is following. Python code to generate a minimal header section looks like this:

```
print("Content-Type: text/html")    # HTML is following
print()                             # blank line, end of headers
```

The second section is usually HTML, which allows the client software to display nicely formatted text with header, in-line images, etc. Here's Python code that prints a simple piece of HTML:

```
print("<TITLE>CGI script output</TITLE>")
print("<H1>This is my first CGI script</H1>")
print("Hello, world!")
```

Using the cgi module

Begin by writing `import cgi`.

When you write a new script, consider adding these lines:

```
import cgitb
cgitb.enable()
```

This activates a special exception handler that will display detailed reports in the web browser if any errors occur. If you'd rather not show the guts of your program to users of your script, you can have the reports saved to files instead, with code like this:

```
import cgitb
cgitb.enable(display=0, logdir="/path/to/logdir")
```

It's very helpful to use this feature during script development. The reports produced by :mod:`cgitb` provide information that can save you a lot of time in tracking down bugs. You can always remove the `cgitb` line later when you have tested your script and are confident that it works correctly.

> **System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)cgi.rst, line 92); *backlink*
>
> Unknown interpreted text role "mod".

To get at submitted form data, use the :class:`FieldStorage` class. If the form contains non-ASCII characters, use the *encoding* keyword parameter set to the value of the encoding defined for the document. It is usually contained in the META tag in the HEAD section of the HTML document or by the :mailheader:`Content-Type` header. This reads the form contents from the standard input or the environment (depending on the value of various environment variables set according to the CGI standard). Since it may consume standard input, it should be instantiated only once.

> **System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)cgi.rst, line 97); *backlink*
>
> Unknown interpreted text role "class".

> **System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)cgi.rst, line 97); *backlink*
>
> Unknown interpreted text role "mailheader".

The :class:`FieldStorage` instance can be indexed like a Python dictionary. It allows membership testing with the :keyword:`in` operator, and also supports the standard dictionary method :meth:`~dict.keys` and the built-in function :func:`len`. Form fields containing empty strings are ignored and do not appear in the dictionary; to keep such values, provide a true value for the optional *keep_blank_values* keyword parameter when creating the :class:`FieldStorage` instance.

> **System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)cgi.rst, line 106); *backlink*
>
> Unknown interpreted text role "class".

> **System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)cgi.rst, line 106); *backlink*
>
> Unknown interpreted text role "keyword".

> **System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)cgi.rst, line 106); *backlink*
>
> Unknown interpreted text role "meth".

> **System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)cgi.rst, line 106); *backlink*

Unknown interpreted text role "func".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)cgi.rst, line 106); backlink

Unknown interpreted text role "class".

For instance, the following code (which assumes that the :mailheader:`Content-Type` header and blank line have already been printed) checks that the fields `name` and `addr` are both set to a non-empty string:

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)cgi.rst, line 114); backlink

Unknown interpreted text role "mailheader".

```
form = cgi.FieldStorage()
if "name" not in form or "addr" not in form:
    print("<H1>Error</H1>")
    print("Please fill in the name and addr fields.")
    return
print("<p>name:", form["name"].value)
print("<p>addr:", form["addr"].value)
...further form processing here...
```

Here the fields, accessed through `form[key]`, are themselves instances of :class:`FieldStorage` (or :class:`MiniFieldStorage`, depending on the form encoding). The :attr:`~FieldStorage.value` attribute of the instance yields the string value of the field. The :meth:`~FieldStorage.getvalue` method returns this string value directly; it also accepts an optional second argument as a default to return if the requested key is not present.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)cgi.rst, line 128); backlink

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)cgi.rst, line 128); backlink

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)cgi.rst, line 128); backlink

Unknown interpreted text role "attr".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)cgi.rst, line 128); backlink

Unknown interpreted text role "meth".

If the submitted form data contains more than one field with the same name, the object retrieved by `form[key]` is not a :class:`FieldStorage` or :class:`MiniFieldStorage` instance but a list of such instances. Similarly, in this situation, `form.getvalue(key)` would return a list of strings. If you expect this possibility (when your HTML form contains multiple fields with the same name), use the :meth:`~FieldStorage.getlist` method, which always returns a list of values (so that you do not need to special-case the single item case). For example, this code concatenates any number of username fields, separated by commas:

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)cgi.rst, line 135); backlink

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)cgi.rst, line 135); backlink

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)cgi.rst, line 135); backlink

Unknown interpreted text role "meth".

```
value = form.getlist("username")
usernames = ",".join(value)
```

If a field represents an uploaded file, accessing the value via the :attr:`~FieldStorage.value` attribute or the :meth:`~FieldStorage.getvalue` method reads the entire file in memory as bytes. This may not be what you want. You can test for an

uploaded file by testing either the :attr:`~FieldStorage.filename` attribute or the :attr:`~FieldStorage.file` attribute. You can then read the data from the :attr:`!file` attribute before it is automatically closed as part of the garbage collection of the :class:`FieldStorage` instance (the :func:`~io.RawIOBase.read` and :func:`~io.IOBase.readline` methods will return bytes):

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)cgi.rst, line 148); backlink

Unknown interpreted text role "attr".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)cgi.rst, line 148); backlink

Unknown interpreted text role "meth".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)cgi.rst, line 148); backlink

Unknown interpreted text role "attr".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)cgi.rst, line 148); backlink

Unknown interpreted text role "attr".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)cgi.rst, line 148); backlink

Unknown interpreted text role "attr".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)cgi.rst, line 148); backlink

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)cgi.rst, line 148); backlink

Unknown interpreted text role "func".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)cgi.rst, line 148); backlink

Unknown interpreted text role "func".

```
fileitem = form["userfile"]
if fileitem.file:
    # It's an uploaded file; count lines
    linecount = 0
    while True:
        line = fileitem.file.readline()
        if not line: break
        linecount = linecount + 1
```

:class:`FieldStorage` objects also support being used in a :keyword:`with` statement, which will automatically close them when done.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)cgi.rst, line 168); backlink

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)cgi.rst, line 168); backlink

Unknown interpreted text role "keyword".

If an error is encountered when obtaining the contents of an uploaded file (for example, when the user interrupts the form submission by clicking on a Back or Cancel button) the :attr:`~FieldStorage.done` attribute of the object for the field will be set to the value -1.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)cgi.rst, line 171); backlink

Unknown interpreted text role "attr".

The file upload draft standard entertains the possibility of uploading multiple files from one field (using a recursive

:mimetype:`multipart/\*` encoding). When this occurs, the item will be a dictionary-like :class:`FieldStorage` item. This can be determined by testing its :attr:`!type` attribute, which should be :mimetype:`multipart/form-data` (or perhaps another MIME type matching :mimetype:`multipart/\*`). In this case, it can be iterated over recursively just like the top-level form object.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)cgi.rst, line 176); backlink

Unknown interpreted text role "mimetype".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)cgi.rst, line 176); backlink

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)cgi.rst, line 176); backlink

Unknown interpreted text role "attr".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)cgi.rst, line 176); backlink

Unknown interpreted text role "mimetype".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)cgi.rst, line 176); backlink

Unknown interpreted text role "mimetype".

When a form is submitted in the "old" format (as the query string or as a single data part of type :mimetype:`application/x-www-form-urlencoded`), the items will actually be instances of the class :class:`MiniFieldStorage`. In this case, the :attr:`!list`, :attr:`!file`, and :attr:`filename` attributes are always `None`.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)cgi.rst, line 184); backlink

Unknown interpreted text role "mimetype".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)cgi.rst, line 184); backlink

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)cgi.rst, line 184); backlink

Unknown interpreted text role "attr".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)cgi.rst, line 184); backlink

Unknown interpreted text role "attr".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)cgi.rst, line 184); backlink

Unknown interpreted text role "attr".

A form submitted via POST that also has a query string will contain both :class:`FieldStorage` and :class:`MiniFieldStorage` items.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)cgi.rst, line 189); backlink

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)cgi.rst, line 189); backlink

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)cgi.rst, line 192)

Unknown directive type "versionchanged".

```
.. versionchanged:: 3.4
   The :attr:`~FieldStorage.file` attribute is automatically closed upon the
   garbage collection of the creating :class:`FieldStorage` instance.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)cgi.rst, line 196)

Unknown directive type "versionchanged".

```
.. versionchanged:: 3.5
   Added support for the context management protocol to the
   :class:`FieldStorage` class.
```

Higher Level Interface

The previous section explains how to read CGI form data using the :class:`FieldStorage` class. This section describes a higher level interface which was added to this class to allow one to do it in a more readable and intuitive way. The interface doesn't make the techniques described in previous sections obsolete --- they are still useful to process file uploads efficiently, for example.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)cgi.rst, line 204); *backlink*

Unknown interpreted text role "class".

The interface consists of two simple methods. Using the methods you can process form data in a generic way, without the need to worry whether only one or more values were posted under one name.

In the previous section, you learned to write following code anytime you expected a user to post more than one value under one name:

```
item = form.getvalue("item")
if isinstance(item, list):
    # The user is requesting more than one item.
else:
    # The user is requesting only one item.
```

This situation is common for example when a form contains a group of multiple checkboxes with the same name:

```
<input type="checkbox" name="item" value="1" />
<input type="checkbox" name="item" value="2" />
```

In most situations, however, there's only one form control with a particular name in a form and then you expect and need only one value associated with this name. So you write a script containing for example this code:

```
user = form.getvalue("user").upper()
```

The problem with the code is that you should never expect that a client will provide valid input to your scripts. For example, if a curious user appends another `user=foo` pair to the query string, then the script would crash, because in this situation the `getvalue("user")` method call returns a list instead of a string. Calling the :meth:`~str.upper` method on a list is not valid (since lists do not have a method of this name) and results in an :exc:`AttributeError` exception.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)cgi.rst, line 238); *backlink*

Unknown interpreted text role "meth".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)cgi.rst, line 238); *backlink*

Unknown interpreted text role "exc".

Therefore, the appropriate way to read form data values was to always use the code which checks whether the obtained value is a single value or a list of values. That's annoying and leads to less readable scripts.

A more convenient approach is to use the methods :meth:`~FieldStorage.getfirst` and :meth:`~FieldStorage.getlist` provided by this higher level interface.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)cgi.rst, line 250); *backlink*

Unknown interpreted text role "meth".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)cgi.rst, line 250); *backlink*

Unknown interpreted text role "meth".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)cgi.rst, line 254)

Unknown directive type "method".

```
.. method:: FieldStorage.getfirst(name, default=None)

   This method always returns only one value associated with form field *name*.
   The method returns only the first value in case that more values were posted
   under such name.  Please note that the order in which the values are received
   may vary from browser to browser and should not be counted on. [#]_  If no such
   form field or value exists then the method returns the value specified by the
   optional parameter *default*.  This parameter defaults to ``None`` if not
   specified.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)cgi.rst, line 265)

Unknown directive type "method".

```
.. method:: FieldStorage.getlist(name)

   This method always returns a list of values associated with form field *name*.
   The method returns an empty list if no such form field or value exists for
   *name*.  It returns a list consisting of one item if only one such value exists.
```

Using these methods you can write nice compact code:

```
import cgi
form = cgi.FieldStorage()
user = form.getfirst("user", "").upper()    # This way it's safe.
for item in form.getlist("item"):
    do_something(item)
```

Functions

These are useful if you want more control, or if you want to employ some of the algorithms implemented in this module in other circumstances.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)cgi.rst, line 289)

Unknown directive type "function".

```
.. function:: parse(fp=None, environ=os.environ, keep_blank_values=False, strict_parsing=False, separato

   Parse a query in the environment or from a file (the file defaults to
   ``sys.stdin``).  The *keep_blank_values*, *strict_parsing* and *separator* parameters are
   passed to :func:`urllib.parse.parse_qs` unchanged.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)cgi.rst, line 296)

Unknown directive type "function".

```
.. function:: parse_multipart(fp, pdict, encoding="utf-8", errors="replace", separator="&")

   Parse input of type :mimetype:`multipart/form-data` (for  file uploads).
   Arguments are *fp* for the input file, *pdict* for a dictionary containing
   other parameters in the :mailheader:`Content-Type` header, and *encoding*,
   the request encoding.

   Returns a dictionary just like :func:`urllib.parse.parse_qs`: keys are the
   field names, each value is a list of values for that field. For non-file
   fields, the value is a list of strings.

   This is easy to use but not much good if you are expecting megabytes to be
   uploaded --- in that case, use the :class:`FieldStorage` class instead
   which is much more flexible.

   .. versionchanged:: 3.7
      Added the *encoding* and *errors* parameters.  For non-file fields, the
      value is now a list of strings, not bytes.

   .. versionchanged:: 3.10
      Added the *separator* parameter.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)cgi.rst, line 319)

Unknown directive type "function".

```
.. function:: parse_header(string)

   Parse a MIME header (such as :mailheader:`Content-Type`) into a main value and a
   dictionary of parameters.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)cgi.rst, line 325)

Unknown directive type "function".

```
.. function:: test()

   Robust test CGI script, usable as main program. Writes minimal HTTP headers and
   formats all information provided to the script in HTML format.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)cgi.rst, line 331)

Unknown directive type "function".

```
.. function:: print_environ()

   Format the shell environment in HTML.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)cgi.rst, line 336)

Unknown directive type "function".

```
.. function:: print_form(form)

   Format a form in HTML.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)cgi.rst, line 341)

Unknown directive type "function".

```
.. function:: print_directory()

   Format the current directory in HTML.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)cgi.rst, line 346)

Unknown directive type "function".

```
.. function:: print_environ_usage()

   Print a list of useful (used by CGI) environment variables in HTML.
```

Caring about security

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)cgi.rst, line 356)

Unknown directive type "index".

```
.. index:: pair: CGI; security
```

There's one important rule: if you invoke an external program (via :func:`os.system`, :func:`os.popen` or other functions with similar functionality), make very sure you don't pass arbitrary strings received from the client to the shell. This is a well-known security hole whereby clever hackers anywhere on the web can exploit a gullible CGI script to invoke arbitrary shell commands. Even parts of the URL or field names cannot be trusted, since the request doesn't have to come from your form!

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)cgi.rst, line 358); *backlink*

Unknown interpreted text role "func".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)cgi.rst, line 358); *backlink*

Unknown interpreted text role "func".

To be on the safe side, if you must pass a string gotten from a form to a shell command, you should make sure the string contains only alphanumeric characters, dashes, underscores, and periods.

Installing your CGI script on a Unix system

Read the documentation for your HTTP server and check with your local system administrator to find the directory where CGI scripts should be installed; usually this is in a directory :file:`cgi-bin` in the server tree.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)cgi.rst, line 374); *backlink*

Unknown interpreted text role "file".

Make sure that your script is readable and executable by "others"; the Unix file mode should be `0o755` octal (use `chmod 0755 filename`). Make sure that the first line of the script contains `#!` starting in column 1 followed by the pathname of the Python interpreter, for instance:

```
#!/usr/local/bin/python
```

Make sure the Python interpreter exists and is executable by "others".

Make sure that any files your script needs to read or write are readable or writable, respectively, by "others" --- their mode should be `0o644` for readable and `0o666` for writable. This is because, for security reasons, the HTTP server executes your script as user "nobody", without any special privileges. It can only read (write, execute) files that everybody can read (write, execute). The current directory at execution time is also different (it is usually the server's cgi-bin directory) and the set of environment variables is also different from what you get when you log in. In particular, don't count on the shell's search path for executables (:envvar:`PATH`) or the Python module search path (:envvar:`PYTHONPATH`) to be set to anything interesting.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)cgi.rst, line 387); *backlink*

Unknown interpreted text role "envvar".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)cgi.rst, line 387); *backlink*

Unknown interpreted text role "envvar".

If you need to load modules from a directory which is not on Python's default module search path, you can change the path in your script, before importing other modules. For example:

```
import sys
sys.path.insert(0, "/usr/home/joe/lib/python")
sys.path.insert(0, "/usr/local/lib/python")
```

(This way, the directory inserted last will be searched first!)

Instructions for non-Unix systems will vary; check your HTTP server's documentation (it will usually have a section on CGI scripts).

Testing your CGI script

Unfortunately, a CGI script will generally not run when you try it from the command line, and a script that works perfectly from the command line may fail mysteriously when run from the server. There's one reason why you should still test your script from the command line: if it contains a syntax error, the Python interpreter won't execute it at all, and the HTTP server will most likely send a cryptic error to the client.

Assuming your script has no syntax errors, yet it does not work, you have no choice but to read the next section.

Debugging CGI scripts

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)cgi.rst, line 429)

Unknown directive type "index".

```
.. index:: pair: CGI; debugging
```

First of all, check for trivial installation errors --- reading the section above on installing your CGI script carefully can save you a lot of time. If you wonder whether you have understood the installation procedure correctly, try installing a copy of this module file (:file:`cgi.py`) as a CGI script. When invoked as a script, the file will dump its environment and the contents of the form in HTML format. Give it the right mode etc., and send it a request. If it's installed in the standard :file:`cgi-bin` directory, it should be possible to send it a request by entering a URL into your browser of the form:

> **System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)cgi.rst, **line 431**); *backlink*
>
> Unknown interpreted text role "file".

> **System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)cgi.rst, **line 431**); *backlink*
>
> Unknown interpreted text role "file".

> **System Message: WARNING/2** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)cgi.rst, **line 440**)
>
> Cannot analyze code. No Pygments lexer found for "none".
>
> ```
> .. code-block:: none
>
>    http://yourhostname/cgi-bin/cgi.py?name=Joe+Blow&addr=At+Home
> ```

If this gives an error of type 404, the server cannot find the script -- perhaps you need to install it in a different directory. If it gives another error, there's an installation problem that you should fix before trying to go any further. If you get a nicely formatted listing of the environment and form content (in this example, the fields should be listed as "addr" with value "At Home" and "name" with value "Joe Blow"), the :file:`cgi.py` script has been installed correctly. If you follow the same procedure for your own script, you should now be able to debug it.

> **System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)cgi.rst, **line 444**); *backlink*
>
> Unknown interpreted text role "file".

The next step could be to call the :mod:`cgi` module's :func:`test` function from your script: replace its main code with the single statement

> **System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)cgi.rst, **line 453**); *backlink*
>
> Unknown interpreted text role "mod".

> **System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)cgi.rst, **line 453**); *backlink*
>
> Unknown interpreted text role "func".

```
cgi.test()
```

This should produce the same results as those gotten from installing the :file:`cgi.py` file itself.

> **System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)cgi.rst, **line 458**); *backlink*
>
> Unknown interpreted text role "file".

When an ordinary Python script raises an unhandled exception (for whatever reason: of a typo in a module name, a file that can't be opened, etc.), the Python interpreter prints a nice traceback and exits. While the Python interpreter will still do this when your CGI script raises an exception, most likely the traceback will end up in one of the HTTP server's log files, or be discarded altogether.

Fortunately, once you have managed to get your script to execute *some* code, you can easily send tracebacks to the web browser using the :mod:`cgitb` module. If you haven't done so already, just add the lines:

> **System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)cgi.rst, **line 468**); *backlink*
>
> Unknown interpreted text role "mod".

```
import cgitb
cgitb.enable()
```

to the top of your script. Then try running it again; when a problem occurs, you should see a detailed report that will likely make apparent the cause of the crash.

If you suspect that there may be a problem in importing the :mod:`cgitb` module, you can use an even more robust approach (which only uses built-in modules):

> **System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)cgi.rst, line 479); *backlink*
>
> Unknown interpreted text role "mod".

```
import sys
sys.stderr = sys.stdout
print("Content-Type: text/plain")
print()
...your code here...
```

This relies on the Python interpreter to print the traceback. The content type of the output is set to plain text, which disables all HTML processing. If your script works, the raw HTML will be displayed by your client. If it raises an exception, most likely after the first two lines have been printed, a traceback will be displayed. Because no HTML interpretation is going on, the traceback will be readable.

Common problems and solutions

- Most HTTP servers buffer the output from CGI scripts until the script is completed. This means that it is not possible to display a progress report on the client's display while the script is running.
- Check the installation instructions above.
- Check the HTTP server's log files. (`tail -f logfile` in a separate window may be useful!)
- Always check a script for syntax errors first, by doing something like `python script.py`.
- If your script does not have any syntax errors, try adding `import cgitb; cgitb.enable()` to the top of the script.
- When invoking external programs, make sure they can be found. Usually, this means using absolute path names --- :envvar:`PATH` is usually not set to a very useful value in a CGI script.

  > **System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)cgi.rst, line 514); *backlink*
  >
  > Unknown interpreted text role "envvar".

- When reading or writing external files, make sure they can be read or written by the userid under which your CGI script will be running: this is typically the userid under which the web server is running, or some explicitly specified userid for a web server's `suexec` feature.
- Don't try to give a CGI script a set-uid mode. This doesn't work on most systems, and is a security liability as well.

Footnotes

[1] Note that some recent versions of the HTML specification do state what order the field values should be supplied in, but knowing whether a request was received from a conforming browser, or even from a browser at all, is tedious and error-prone.