

1. 基本

buturi.heteml.jp/Python/python1.py

python1.py

```
#!/usr/local/bin/python36

print ("Content-Type:text/html")
print ()      # 空行 HTTP の規約 : response header と body は 1 行の空行で区切る
print ('<!DOCTYPE html>')
print ('<html>')
print ('<head>')
print ('<meta charset="utf-8">')
print ('<title>Hello Word Python</title>')
print ('</head>')
print ('<body>')
print ('<h1>Hello Word! Python</h1>')
print ('</body>')
print ('</html>')
```

http request

```
GET / HTTP/1.1
Accept: image/gif, image/jpeg, */*
Accept-Language: ja
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (Compatible; MSIE 6.0;
Windows NT 5.1;)
Host: www.xxx.zzz
Connection: Keep-Alive
```

http response

```
HTTP/1.1 200 OK
Date: Sun, 11 Jan 2004 16:06:23 GMT
Server: Apache/1.3.22 (Unix) (Red-Hat/Linux)
Last-Modified: Sun, 07 Dec 2003 12:34:18 GMT
ETag: "1dba6-131b-3fd31e4a"
Accept-Ranges: bytes
Content-Length: 4891
Keep-Alive: timeout=15, max=100
Connection: Keep-Alive
Content-Type: text/html

<!DOCTYPE html>
<html>
:
</html>
```

status line

response header

response body

参考

<http://www.w3big.com/ja/python3/python3-cgi-programming.html>

2. Python の文字コードの指定

(1) スクリプトの文字コード指定

Python3 ではソースファイルの文字コードはデフォルトで UTF-8

Python2 ではデフォルトは ASCII なので、次の 1 行が必要

```
# -*- coding: utf-8 -*-
```

または単純に以下でも可

```
# coding= utf-8
```

なお `-*-` は Unix のエディタ `emacs` に文字コードを認識させるためのもの

(2) Heteml サーバの Python 標準入出力 (`stdin`, `stdout`, `stderr`) の文字コード確認

Heteml の Python3 の環境は、デフォルトは `ANSI_X3.4-1968` であった(2018年6月17日)。検索すると「This is another name for USAS X3.4-1968, a revision of ASCII that is distinguished by being:」とあるので、ASCII の亜種らしい。

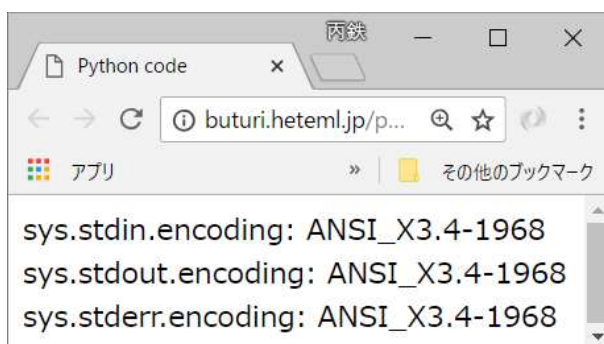
参照 : buturi.heteml.jp/python/python2.py

buturi.heteml.jp/Python/python2.py

`python2.py`

```
#!/usr/local/bin/python36
import sys

print("Content-Type: text/html")
print()
print('<!DOCTYPE html>')
print('<html><head>')
print('<meta charset="utf-8">')
print('<title>Python character code</title>')
print('</head><body>')
# 文字コード
print('sys.stdin.encoding:', sys.stdin.encoding)
print('<br>')
print('sys.stdout.encoding:', sys.stdout.encoding)
print('<br>')
print('sys.stderr.encoding:', sys.stderr.encoding)
print('<br>')
print('</body></html>')
```



(3) 日本語を Utf-8 で入出力する場合

```
import sys, io          #標準入出力を utf-8 に指定
sys.stdin = io.TextIOWrapper(sys.stdin.buffer, encoding='utf-8')
sys.stdout = io.TextIOWrapper(sys.stdout.buffer, encoding='utf-8')
sys.stderr = io.TextIOWrapper(sys.stderr.buffer, encoding='utf-8')
```

buturi.heteml.jp/Python/python3.py

python3.py

```
#!/usr/local/bin/python36
#標準入出力を utf-8 に変更
import sys, io
sys.stdin = io.TextIOWrapper(sys.stdin.buffer, encoding='utf-8')
sys.stdout = io.TextIOWrapper(sys.stdout.buffer, encoding='utf-8')
sys.stderr = io.TextIOWrapper(sys.stderr.buffer, encoding='utf-8')
print ('Content-Type: text/html')
print ()
print('<!DOCTYPE html>')
print('<html><head>')
print ('<meta charset="utf-8">')
print('<title>Python character code</title>')
print('</head><body>')
print ('<h1>標準入出力文字コード</h1>')
print('sys.stdin.encoding:', sys.stdin.encoding)
print('<br>')
print('sys.stdout.encoding:', sys.stdout.encoding)
print('<br>')
print('sys.stderr.encoding:', sys.stderr.encoding)
print('<br>')

print ('<h2>CGI 環境変数</h2>')
print ('<ul>')
import os
for key in os.environ.keys():
    print ('<li>%30s: %s </li>' % (key, os.environ[key]))
print ('</ul>')
print ('</body></html>')
```

3. Post されたデータを受け取る

buturi.heteml.jp/Python/post.html

```
<!DOCTYPE html>
<html><head><meta charset="utf-8">
<title>Post テスト</title></head><body>
<h1>Post テスト</h1>
<p>action="./post.py"</p>
<form action="./post.py" method="POST">
<input type="text" name="sendText" value="ハロー" />
<input type="submit" name="submit" />
</form>
</body></html>
```

post.py

```
#!/usr/local/bin/python36
#標準入出力を utf-8 に変更
import sys, io
sys.stdin = io.TextIOWrapper(sys.stdin.buffer, encoding='utf-8')
sys.stdout = io.TextIOWrapper(sys.stdout.buffer, encoding='utf-8')
sys.stderr = io.TextIOWrapper(sys.stderr.buffer, encoding='utf-8')
#cgi モジュール
import cgi
#例外処理表示 (本番では外す)
import cgitb
cgitb.enable()
#form データの取得
form = cgi.FieldStorage()
text = form.getvalue('sendText', '')
print ('Content-Type: text/html')
print ()
print('<!DOCTYPE html>')
print('<html><head><meta charset="utf-8">')
print ('<title>POST データを受け取る</title>')
print('</head><body>')
print ('<h1>POST データ</h1>')
print ('<p> %s </p>' % text)
print ('</body></html>')
```