

- ホットスポット化した raspberrypi 上で HTTP サーバーを動作させます
- 各自の PC 上でも HTTP サーバーを動作させます (python3 -m http.server)
- 各自の PC 上に自分の WEB ページを作ります (雛形を用意してそれを編集)

※HTML を扱うアプリケーションが必要

Visual Studio Code とか軽くて良いと思います

(Atom, Sublime でも良いでしょう・・・)

Download Visual Studio Code (Mac, Linux, Windows)

<https://code.visualstudio.com/download>

kompoZer というシンプルな HTML のエディタ

Download kompoZer (Mac, Linux, Windows)

<https://ja.osdn.net/projects/kompozer/>

パソコン側のページの例

```
<!DOCTYPE html>
<html lang="en"><head><meta charset="utf-8">
<title>私のページ</title></head>
<body>
<h1>私のページ</h1>
<br>ようこそ！<br><br>
<hr><a href="http://172.24.1.1:8000">サーバーのゲストブックに移動する</a>
</body></html>
```



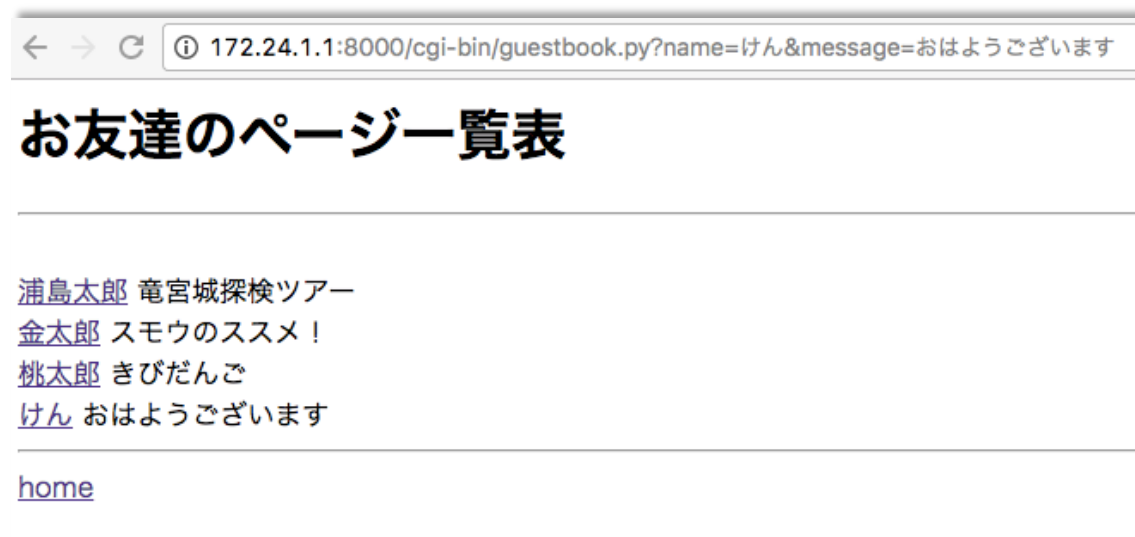
自分のページを表示させて下部のリンクからサーバーに移動します。

サーバー上ではゲストブックが以下のように表示されます。



A screenshot of a web browser window. The address bar shows the URL `172.24.1.1:8000`. The page title is "ゲストブック" (Guestbook). The content includes a greeting "ようこそ！" (Welcome!), a label "お名前:" (Name) followed by a text input field containing the name "けん" (Ken), a label "メッセージ" (Message) followed by a text area containing the message "おはようございます" (Good morning), and two buttons at the bottom: "とうろく" (Register) and "とりけす" (Cancel).

名前とメッセージを入力して「とうろく」ボタンをおすと
「お友達のページ一覧表」の画面に移ります。



A screenshot of a web browser window. The address bar shows the URL `172.24.1.1:8000/cgi-bin/guestbook.py?name=けん&message=おはようございます`. The page title is "お友達のページ一覧表" (List of friends' pages). The content displays a list of entries, each with a name and a message, all underlined as links: "浦島太郎 竜宮城探検ツアー", "金太郎 スモウのススメ！", "桃太郎 きびだんご", and "けん おはようございます". At the bottom, there is a link labeled "home".

名前（アクセス元の IP にリンクしたアンカー）とメッセージの一覧表が表示されて
子供達の PC 上で動作させた HTTP サーバ経由で、それぞれが作った自分のページに相互に
見せあう、といった具合です。

サーバーが返す「お友達のページ一覧表」

```
<HTML>
<HEAD><meta charset='utf-8' />
<TITLE>ゲストブック</TITLE>
</HEAD>
<BODY BGCOLOR="#ffffff">
<h1>お友達のページ一覧表</h1>
<HR>
<BR><A HREF="http://172.24.1.1:8000">浦島太郎</A> 竜宮城探検ツアー
<BR><A HREF="http://172.24.1.1:8000">金太郎</A> スモウのススメ！
<BR><A HREF="http://172.24.1.1:8000">桃太郎</A> きびだんご
<BR><A HREF="http://172.24.1.123:8000">けん</A> おはようございます
<br><hr><a href="http://172.24.1.1:8000">home</a>
</BODY>
</HTML>
```

【準備】など

サーバーとする raspberry pi は以下のようにになっています。

home/pi

```
├── pycgi
│   ├── UQ.txt
│   ├── cgi-bin
│   │   ├── guestbook.py
│   │   └── penguinUtil.py
│   ├── httpy.sh
│   └── index.htm
```

●rasberry pi 上で HTTP サーバーを起動

pycgi 直下の httpy.sh で起動

httpy.sh の内容

```
#!/bin/bash
```

```
python3 -m http.server 8000 --cgi --bind 172.24.1.1
```

サーバー側のトップ画面のソース

index.htm の内容

```
<!DOCTYPE html><html lang="en">
<head><meta charset="utf-8">
<title>ゲストブック</title></head>
<body><h1>ゲストブック</h1>ようこそ！
<form action="cgi-bin/guestbook.py" method="get">
おなまえ: <input name="name" size="25" type="text">
  <p>メッセージ <br>
  <textarea rows="4" cols="60" name="message"></textarea></p>
  <p><input value="とうろく" type="submit"> <input value="とりけす" type="reset">
</p>
</form>
</body></html>
```

フォームから呼び出される CGI のソース

guestbook.py の内容

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-
import cgi
import sys
import os
import penguinUtil
map = penguinUtil.file2map("UQ.txt")
ip = ""
if 'REMOTE_ADDR' in os.environ :
    ip = os.environ['REMOTE_ADDR']
form = cgi.FieldStorage()
message = '... 考え中です'
if 'message' in form :
    message = cgi.escape(form['message'].value)
```

```

if 'name' in form :
    name    = cgi.escape(form['name'].value)
    map[name]=[ip,message]
    penguinUtil.map2file("UQ.txt",map)

print ("Content-type: text/html")
print ("")
print ("<HTML>¥n<HEAD><meta charset='utf-8' />")
print ('<TITLE>ゲストブック</TITLE>')
print ('</HEAD>')
print ('<BODY BGCOLOR="#ffffff">')
print ('<h1>お友達のページ一覧表</h1>')
sys.stdout.write("<HR>¥n")
indexlist = penguinUtil.indexList(map)
for element in indexlist:
    print(element)
print ('<br><hr><a href="http://172.24.1.1:8000">home</a>')
print ("</BODY>¥n</HTML>")

```

サブルーチン

penguinUtil.py

```

def file2map(iPath):
    map={}
    f=open(iPath,"r")
    for line in f:
        array=line.split()
        if len(array) >= 3 :
            map[array[0]] = [array[1],array[2]]
    f.close()
    return map

def enumMap(map):
    keyList=map.keys()
    for key in keyList:

```

```

        print("%s => %s" % (key, map[key]))

def indexList(map):
    index=[]
    keyList=map.keys()
    for name in keyList:
        valList = map[name]
        ip = valList[0]
        message = valList[1]
        val = '<BR><A HREF="http://%s:8000">%s</A> %s¥n' % (ip, name, message)
        #print(val)
        index.append(val)
    return index

def map2file(oPath, map):
    f=open(oPath, "w")
    keyList=map.keys()
    for key in keyList:
        valList = map[key]
        ip = valList[0]
        message = valList[1]
        f.write(' %s¥t%s¥t%s¥n' % (key, ip, message))
    f.close()

def test():
    map = file2map("name_ip.txt")
    enumMap(map)
    map2file("UQ.txt", map)

```

UQ.txt の内容

浦島太郎	172. 24. 1. 1	竜宮城探検ツアー
金太郎	172. 24. 1. 1	スモウのススメ！
桃太郎	172. 24. 1. 1	きびだんご

途中にファイル I/O があるので、アクセスが集中したらどうなるんだろう、とは思いますが
まあ大丈夫でしょう

今の所ユーザーサイドで HTTP サーバーを起動させる時の IP 指定が問題です

```
└── forPc
    ├── cgi-bin
    ├── httpy. sh
    └── index. htm
```

ここはもうちょっと考えておきます

i p を調べないで起動できるようにしたほうがよいと考えています。