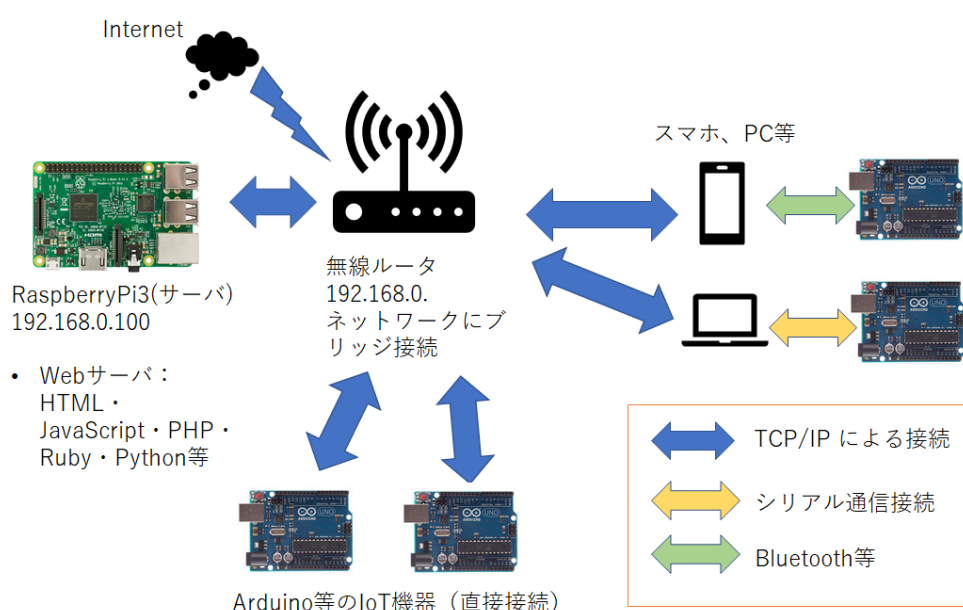


# IoT演習で用いるサーバについて

2017/10/17  
石山隼行

IoT演習向けに簡単な実験用サーバのサンプルとして、RaspberryPi3を利用したサーバ（Piサーバ）を作成しました。  
簡易的なものですが、外部のインターネット上にあるサーバ（Webサーバ）と同様の機能を持っていますので、ローカル環境でのIoT演習では自由に使ってください。

## 構成図



## 現在のセキュリティ上の制限について

この無線ルータは、（現在のところ）インターネットに接続してはいますが、インターネット側からは直接IPアドレスを指定できなくなっています。従って、インターネットのクラウド上のデータを取りに行ったり、クラウドにデータをアップすることはできませんが、クラウド側からこのPiサーバやArduinoのIPアドレスを直接指定して動作させることはできませんので注意してください。

※スマホやPCをこのルータにローカル環境で接続し、ローカルで完結する情報のやりとりならば問題ありません。Piサーバはローカル環境内の疑似クラウドとして動作します。

※完全なインターネット経由で情報のやりとりが必要な場合は、別途専用サーバを立てるか、LoRaによる通信経路を利用してください。（やり方は要相談）

※今後インターネット側からの接続も行えるように調整中です。

## 各機器のルータとの接続

- ルータ名(SSID) : Buffalo-G-7950
- WiFi (WPA2)暗号化キー : buff1212wpa2

上記は廃止。

## 現在はこちら

### 2.4GHz

- **SSID: IoTMLAB**
- **KEY: iotnishi2017**

WPA-Personal  
AES

### 5GHz

- **SSID: IoTMLAB\_5G**
- **KEY: iotnishi2017**

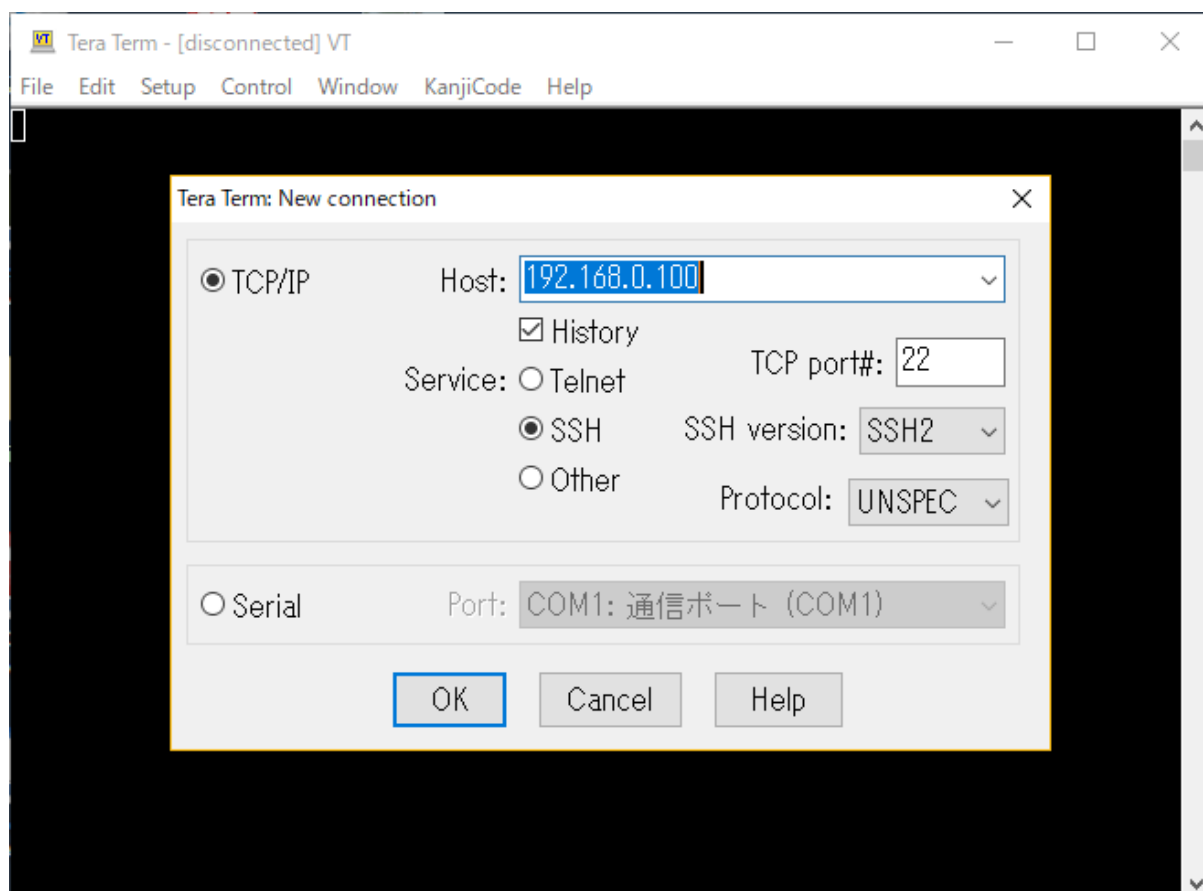
ルータとは、上記のWifi、またはイーサネットケーブルで接続します。

DHCPを有効にしていますので、接続したPC等は自動的にIPアドレスが割り振られます。  
接続時、自分のIPアドレスを確認するようにしてください。192.168.0.nnnとなっていれば正常です。

## サーバへのログイン

Piサーバはパソコンなどからは sshで接続できます。

WindowsならばTeraTermPro等、Macintoshからはターミナルアプリ等のターミナルソフトが必要です。



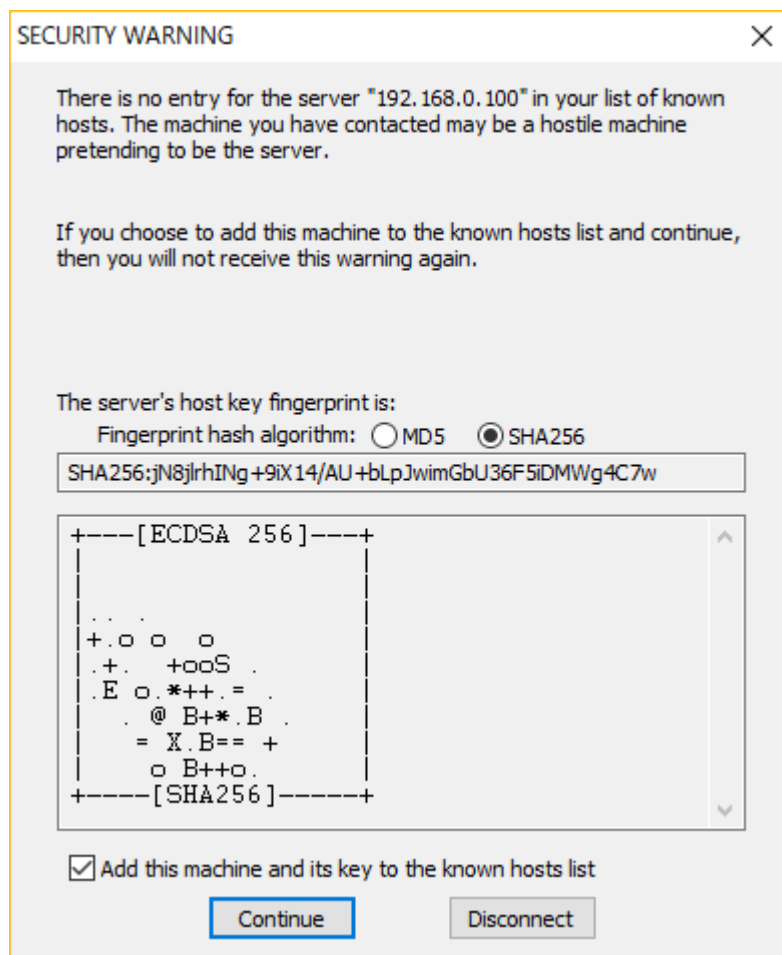
TeraTermPro(Windows)の場合

- IPアドレス : 192.168.0.100
- sshポート(22/tcp)

※2017/10/27現在、192.168.1.100 に変更されています

を指定して接続します。

初回接続時は

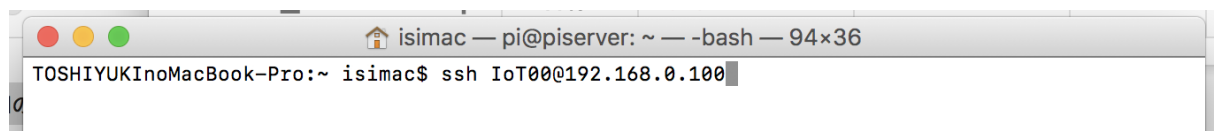


このようなセキュリティ警告がでます。Continueで続行してください。

アカウント（ログイン名）：IoT01 ～ IoT12 （数字は席番、IoTはoが小文字なので注意）  
デフォルトパスワードも上記と同様

※ 以下の例ではサンプル用にIoT00 というアカウントを私用しています。

Macの場合は、ターミナルから

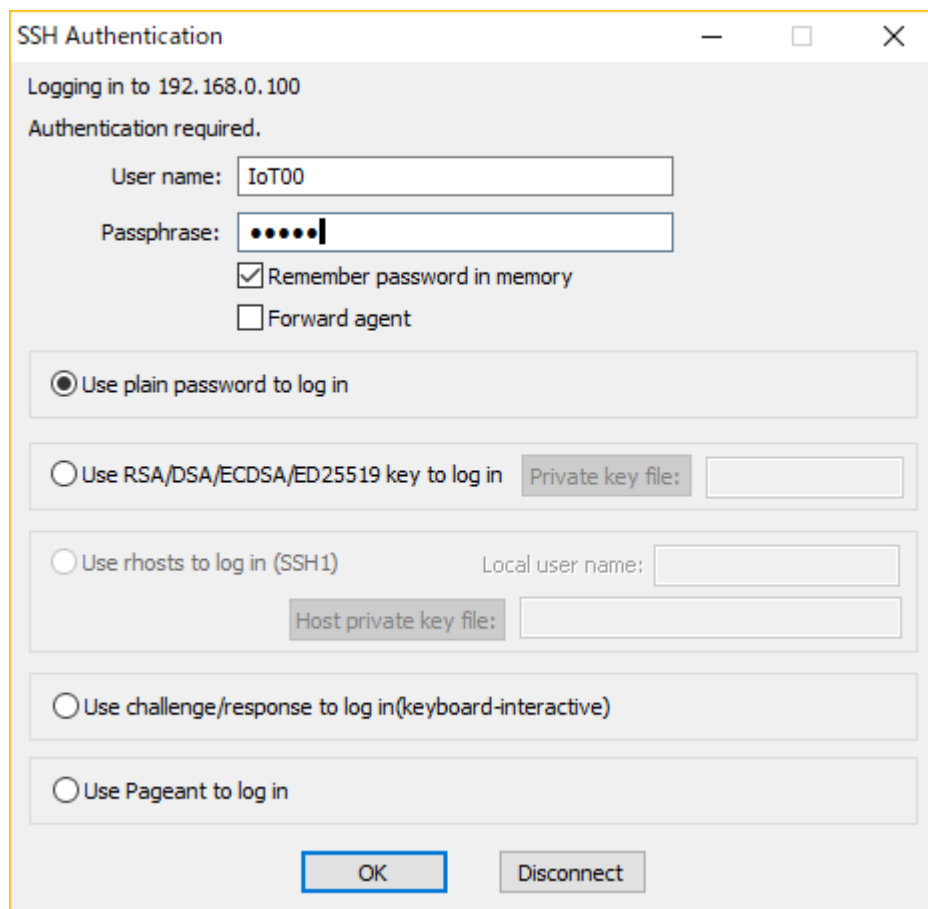


`$ ssh IoT00@192.168.0.100`

と入力して接続します。

この場合も初回接続時は警告がでますが、[Yes]と入力して許可をあたえ、パスワードを入力してください。

以下、再びWindowsの画面



The image shows a standard Linux SSH authentication dialog box. At the top, it says 'SSH Authentication' with window control buttons. Below that, it indicates 'Logging in to 192.168.0.100' and 'Authentication required.' The 'User name' field is filled with 'IoT00'. The 'Passphrase' field is masked with dots. There are two checkboxes: 'Remember password in memory' (checked) and 'Forward agent' (unchecked). Below these are five radio button options for authentication methods: 'Use plain password to log in' (selected), 'Use RSA/DSA/ECDSA/ED25519 key to log in' (with a 'Private key file:' field), 'Use rhosts to log in (SSH1)' (with 'Local user name:' and 'Host private key file:' fields), 'Use challenge/response to log in (keyboard-interactive)', and 'Use Pageant to log in'. At the bottom are 'OK' and 'Disconnect' buttons.

SSH Authentication

Logging in to 192.168.0.100

Authentication required.

User name: IoT00

Passphrase: .....

☒ Remember password in memory

☐ Forward agent

☒ Use plain password to log in

☐ Use RSA/DSA/ECDSA/ED25519 key to log in Private key file:

☐ Use rhosts to log in (SSH1) Local user name:

Host private key file:

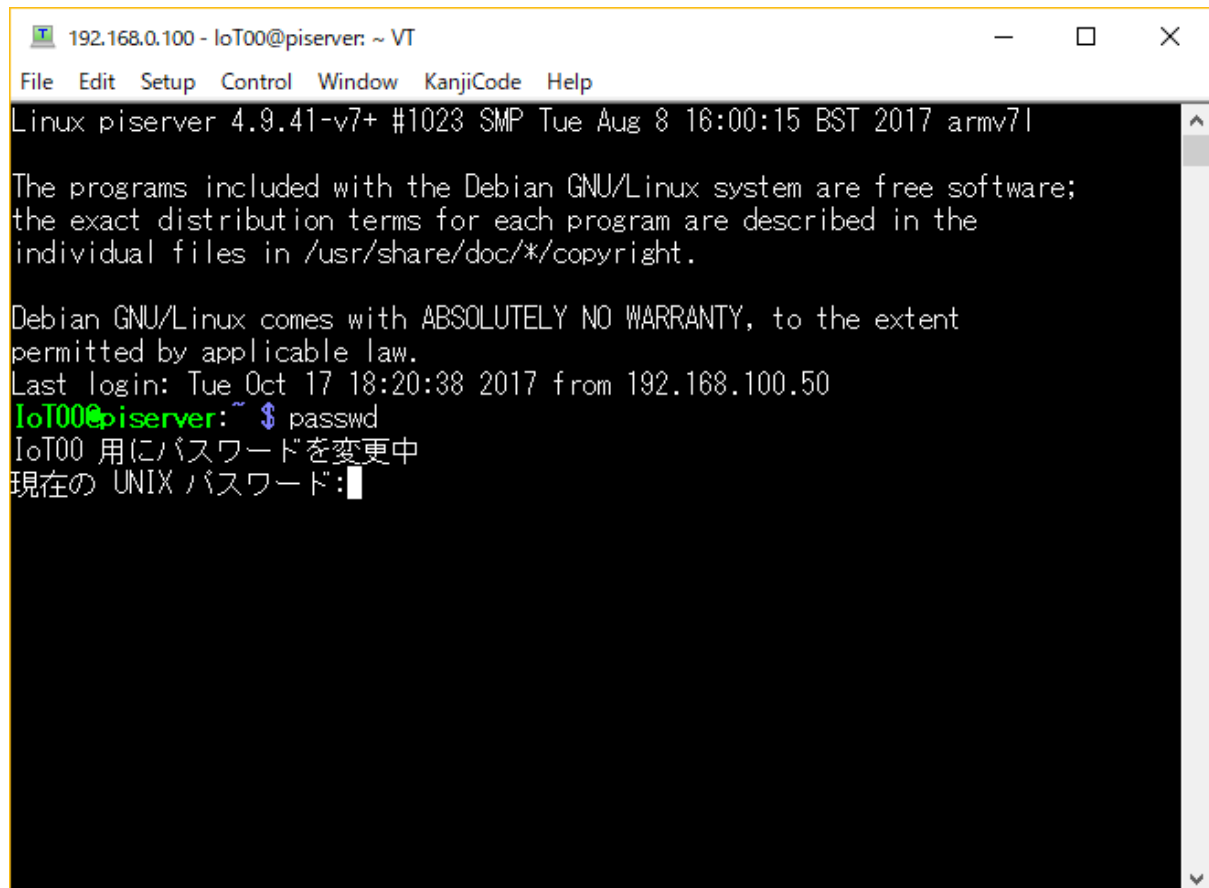
☐ Use challenge/response to log in (keyboard-interactive)

☐ Use Pageant to log in

OK Disconnect

ログインできたら、かならずパスワードを書き換えてください  
passwd コマンドで修正できます。

*\$ passwd*



```
192.168.0.100 - IoT00@piserver: ~ VT
File Edit Setup Control Window KanjiCode Help
Linux piserver 4.9.41-v7+ #1023 SMP Tue Aug 8 16:00:15 BST 2017 armv7l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Tue Oct 17 18:20:38 2017 from 192.168.100.50
IoT00@piserver:~$ passwd
IoT00 用にパスワードを変更中
現在の UNIX パスワード:
```

※旧パスワードを入れ、新しいパスワードをセットします。

※そのほかのUnixコマンドの基礎については別紙参照。

## ホームページの作成

ログインができ、パスワードも変更したら、次にホームページの作成準備をします。

Piサーバ上の各自のホームページは、ログインしたHomeディレクトリ上の `public_html` ディレクトリ内が表示されます。

```
$ mkdir public_html
```

でディレクトリを作成、HTMLファイルをその中に作成します。

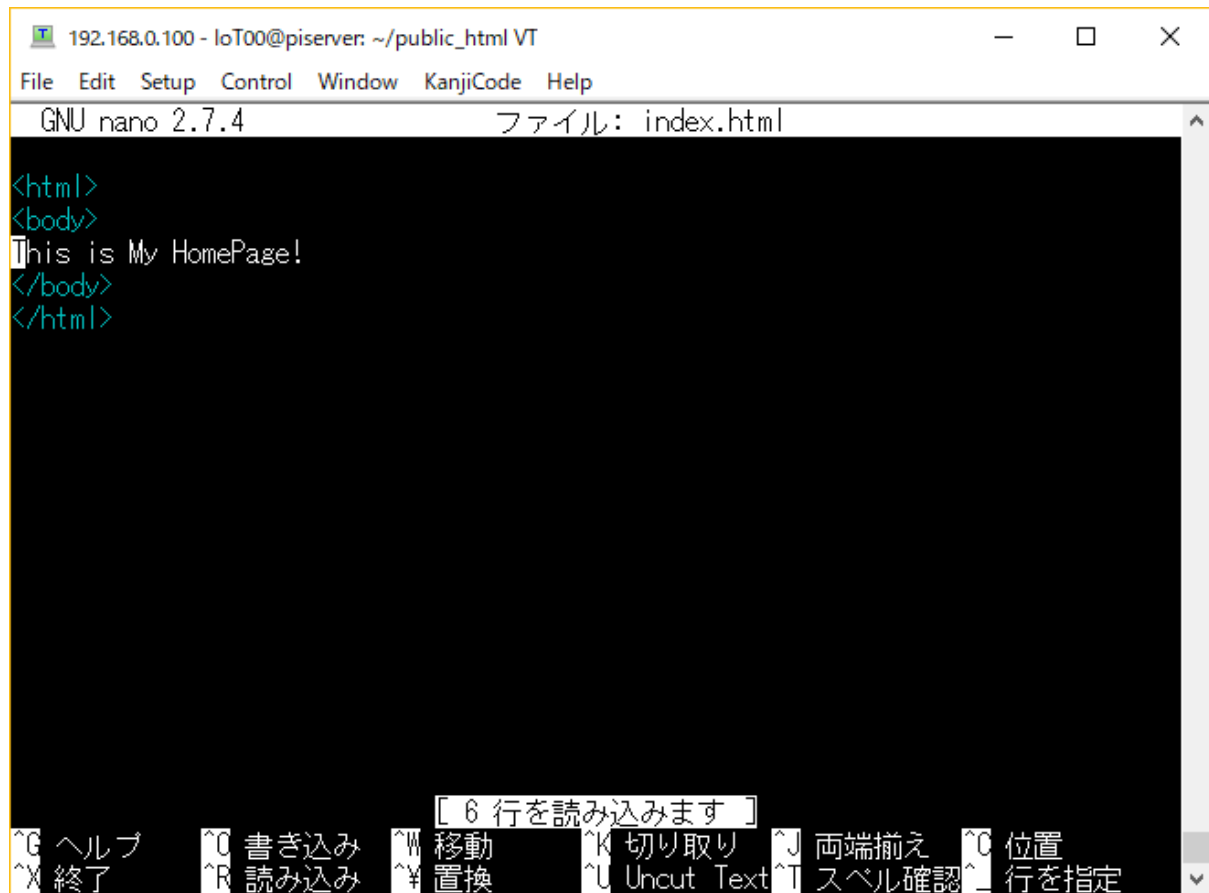
```
$ cd public_html
```

でディレクトリを移動

```
$ nano index.html
```

でindex.htmlファイルを作成します。

当初は、



```
192.168.0.100 - loT00@piserver: ~/public_html VT
File Edit Setup Control Window KanjiCode Help
GNU nano 2.7.4 ファイル: index.html

<html>
<body>
This is My HomePage!
</body>
</html>
```

ヘルプ 書き込み 移動 切り取り 両端揃え 位置  
終了 読み込み 置換 Uncut Text スペル確認 行を指定

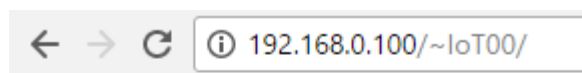
この程度の最低限の内容でよいでしょう。

入力ができたら、[<sup>^</sup>O] [<sup>^</sup>X]でファイルを保存、終了してください。

※日本語のページを作成したい場合は、冒頭の<html>を  
    <html lang="ja">  
    <meta charset="UTF-8" />  
と書き換えます。（もちろんこれは最低限の設定です）

## Piサーバ上のホームページについて

Piサーバ（192.168.0.100）には、各アカウントそれぞれにWebページが作成できます。  
ルータに接続したPC等のブラウザから、<http://192.168.0.100/~loT00/> を表示してみてください（IoTの左につく数字は自分の席番）



This is My Homepage!

このような表示が出たら成功です。

※サーバー内の /home/loT00/public\_html 以下の内容が、ブラウザの <http://192.168.0.100/~loT00/> に表示されるようになっていきます。loT01やloT02以降も同様。

## CGIの作成

動的に変化するWebページは、CGIという仕組みでプログラムを動作させます。

public\_htmlディレクトリ以下でCGIを動作させることができます。

public\_htmlに、以下の内容のテキストファイルを作成します。

\$ *nano test.cgi* でファイルを作成します。

*test.cgi*

```
#!/bin/bash

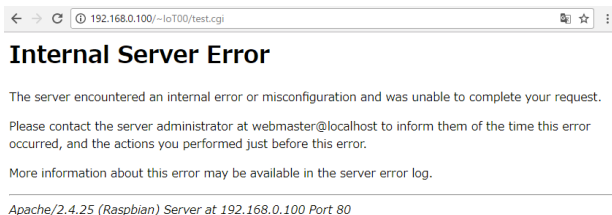
echo "Content-type:text/html"
echo ""

echo "Hello CGI World!"
```

ファイルを保存したら、作成したファイルに実行権限を与えます。

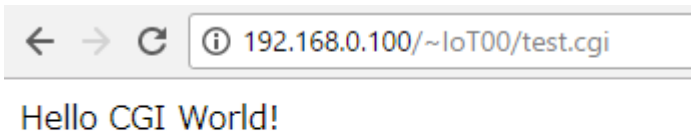
\$ *chmod a+x test.cgi*

※↑これを忘れると、



こういうエラーが出るので注意。

ブラウザで、 <http://192.168.0.100/~loT00/test.cgi> を開くと



このような表示が出たら成功。



このCGIはシェルスクリプトを実行していますが、同様にruby、perl、python等も実行することができるようになっています。

rubyの例

*test.rb*

```
#!/usr/bin/ruby  
  
print("content-type: text/html\n\n")  
print("Hello Ruby World\n")
```

Perlの例

*test.pl*

```
#!/usr/bin/perl  
print "Content-type: text/html\n\n";  
print "Hello Perl World.";
```

Pythonの例

*test.py*

```
#!/usr/bin/python  
  
print "Content-Type: text/plain"  
print  
print "hello Python world."
```

また、phpも動作するようにしてあります。

*test.php*

```
<?php  
echo 'Hello PHP!' . PHP_EOL;  
?>
```

※phpのみは通常の権限でOKです。

これらの言語で得意なものを使い、動的に変化するホームページを作成してください。

ー以上ー