

子どもIT未来塾 第7回

ネットワークについて学ぼう

清水 尚彦 先生

1 今回の授業

目標

- ネチケットを知ろう
- ネットワークについて学ぼう
- CGI を使いこなそう

注意点

- 授業の合間の休憩きゅうけいでは、遠くのものを眺めたりして目を休めましょう
- 水分補給ほきゅうをこまめにしましょう
- 先生が説明中は先生の話を聞きましょう
- わからないことがあったら TA の先生方にすぐ聞きましょう

教科書について

- 教科書には例題、それに似た問題があります
まずは、例題をよく読みながら試してみましょう
その後問題を解きましょう
問題の答えは一番最後のページにあります
- 例題、問題をクリアしたらシールラリーカードにシールを貼りましょう
- すべての問題、例題をクリアできたら TA の先生にシールラリーカードを見せて、
Complete(コンプリート) シールをもらおう
- 授業中に終わらなかった例題、問題は、できるだけ家でやって終わらせよう
- 授業中にわからないところがあったらすぐに TA の先生に聞こう
- 家でわからないことがあったら、すぐに質問フォームから質問しよう

教材を自分のフォルダに置く

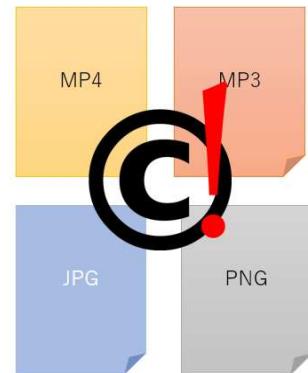
- /usr/local/share/ome にあるフォルダ 07 をホームディレクトリにコピーしましょう。
- やり方を忘ってしまった人は、「第 1 回 4.1 例題 1-18 教材をじぶんのフォルダに置こう」
を参考にしてみてください。

ネチケット

学校の先生たちから、「友達を傷つけてはいけない」、「敬語は正しく使おう」、などとマナーを教えてきたと思います。マナーは、ネットワークにも存在します。ネットワークのマナーを「ネチケット」と言います。ネットをするときに、何も考えずに利用していると気がつかず問題や犯罪に手を染める可能性があります。他人に迷惑をかけないことに加えて、ネチケットを守らない人から自分を守るために理解をしましょう。

誰かが作ったものを勝手に使わない

インターネットをしているとたくさんの画像や文章などのデータがあります。見て楽しむ程度には問題ないですが、誰かが作ったものや情報には著作権というものがあります。法律で勝手に使うことは禁止されているから気をつけましょう。ただし、著作権は個人で楽しむもの、出典を明らかにして、引用することは認められています。



人を傷つけることや、迷惑になることはやめる

インターネットは自分以外の人も使っています。だから、自分勝手なことや悪い言葉を使ってばかりいると人を傷つけたり、不快に思われて犯罪に巻き込まれたり、訴えられたりする可能性もあります。清らかに生きていきましょう。

他のひとたちが困らない文字情報を使う

ネットワークを使っている人たちはみんな同じものを使っているとは限りません。PCだったり、スマートフォンだったり、ラズベリーパイだったりします。文字の種類によっては見えなかったりすることが考えられます。

迷惑メールは見ない、送らない

悪い人が色々な都合で出してくる怪しいメールのことです。これには、変なアドレスがはられていたり、知らないソフトウェアが貼り付けられたりすることがあります。それを見るとウイルスという悪いものが入ってくることがあります。ウイルスはコンピュータが壊れたり、操作出来なくなったりするものもあります。犯罪者の協力をしたと思われることもあるので、怪しいメールは絶対に開かないようにしましょう。また、チェーンメール（呪いのメールなど）は絶対に信じず他の人に送ってはいけません。メール本文中に、『誰かに回して』『〇〇人に転送するように』などと書かれているメールを受け取ったのなら、それはどんな内容でもチェーンメールです。

自分の情報を外に漏らさない

自分の名前や、住所などをネットにあげるのはとても危ないから止めましょう。悪い人の目に入ってしまうと家族の身の危険にもつながり命の保証ができません。また、ネットに一度でもあげてしまうと消すことができなくなります。多くの人の目につくからデータが

広がってしまうこともあります。ネットにデータをあげるときは問題が無いか確認してからあげましょう。

教材を自分のフォルダに置こう

まずは、今回利用する教材をコピーしましょう。これまでと同じように、/usr/local/share/omeという場所にあるフォルダ 07 をコピーして、/home/ユーザー名 に貼り付けてください。

やり方を忘ってしまった人は、「第1回 4.1 例題 1-18 教材をじぶんのフォルダに置こう」を参考にしてみてください。

1 - 1 インターネットのキホン

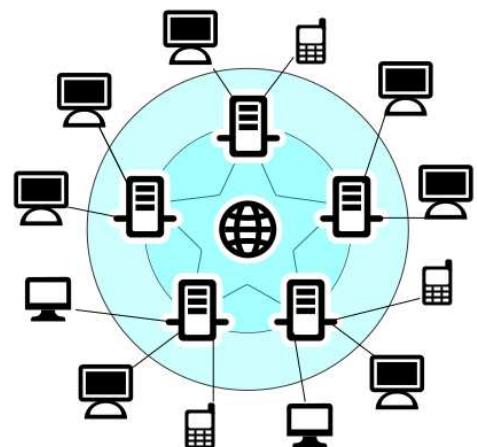
みんなが使っているパソコンやスマートフォンなどをたくさんつなぎ、おたがいに情報の交換ができるようにしたものを「ネットワーク」とよんでいます。

インターネットとは、そのネットワークが数えきれないほどのケーブルなどでおたがいにつながりあい、網の目のように世界中に広がっているそのもののことです。

みなさんはすでにインターネットを触っています。

第一回の講義でみなさんは web から画像を自分のラズベリーパイ (PC) に保存しましたね。あれもインターネットを触ったということになるのです。網の目のように世界中に広がっているネットワークから好きな画像を調べるということは自分のラズベリーパイでインターネットをしたということになります。

さらに、有名な動画サイト "YouTube" のコンピュータもインターネットにつながっています。そのおかげでインターネットにつながっているパソコンやスマートフォンであれば YouTube の動画を見るすることができます。もちろん、インターネットにつながっていなければ YouTube の動画や画像検索など行うことはできません。試しにインターネットを切斷してみても面白いかもしれません。



問題 7-1 みんなが使っているパソコンやスマートフォンなどをたくさんつなぎ、おたがいに情報の交換ができるようにしたものを何というでしょうか。

答え：

1 - 2 IP アドレスとは何だろう？

ネットワークでコンピュータ同士がつながっているときにどうやって通信相手のコンピュータの場所を探すのでしょうか？

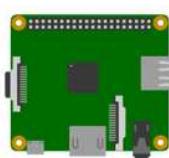
わたしたちのいる住んでいる場所をしめす住所（英語でアドレス）と同じ考え方を使います。インターネットの住所のことをIP(アイピー)アドレスと言います。わたしたち日本人は住所を東京都港区高輪2-3-23（東海大学高輪キャンパスの住所）のように表します。コンピュータの場合は**数字**で表します。IPアドレスの場合はxxx.xxx.xxx.xxxのように四つの数字を使って表します。3ヶタづつ「.」点（ドット）で区切ります。xxxは0～255の数字が入ります。

青梅市立第一小学校



住所
東京都青梅市本町223

みんなのラズベリーパイ



IPアドレス
192.168.1.199



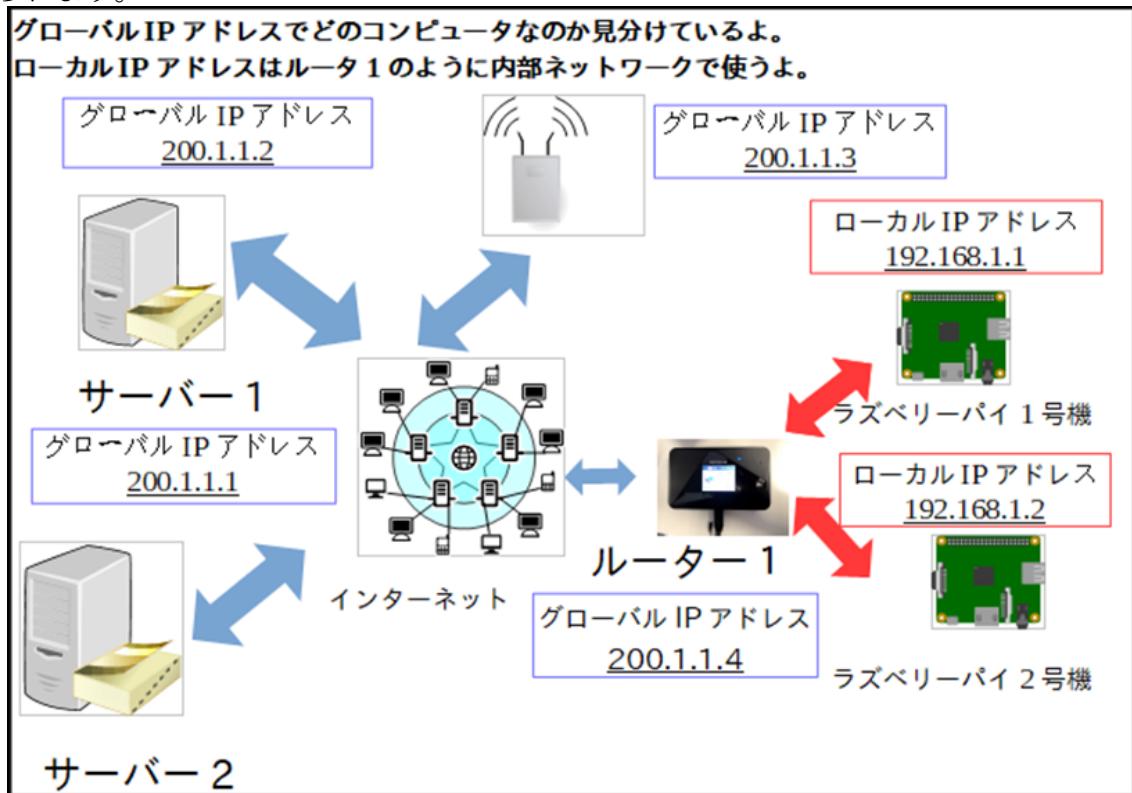
IPアドレスというのは
みんなのラズベリーパイにとって
ネットワーク上の住所だよ

IPアドレスは同じネットワーク上に同じもの
があってはいけないよ。自分のラズパイと友達のラズパイのIPアドレスを調べて確認して
見てね

IPアドレスはラズベリーパイのネットワーク上の住所に当たるものです。一つのネットワークではIPアドレスは同じものがあつてはいけません。例えば、配達する人は同じ住所が世界中にあればどこへ配達したらよいかわからなくなってしまいます。インターネットの場合も同じです。なのでIPアドレスは同じではありません。ですが、例外として他のローカルIPアドレスでは同じになってしまいます。

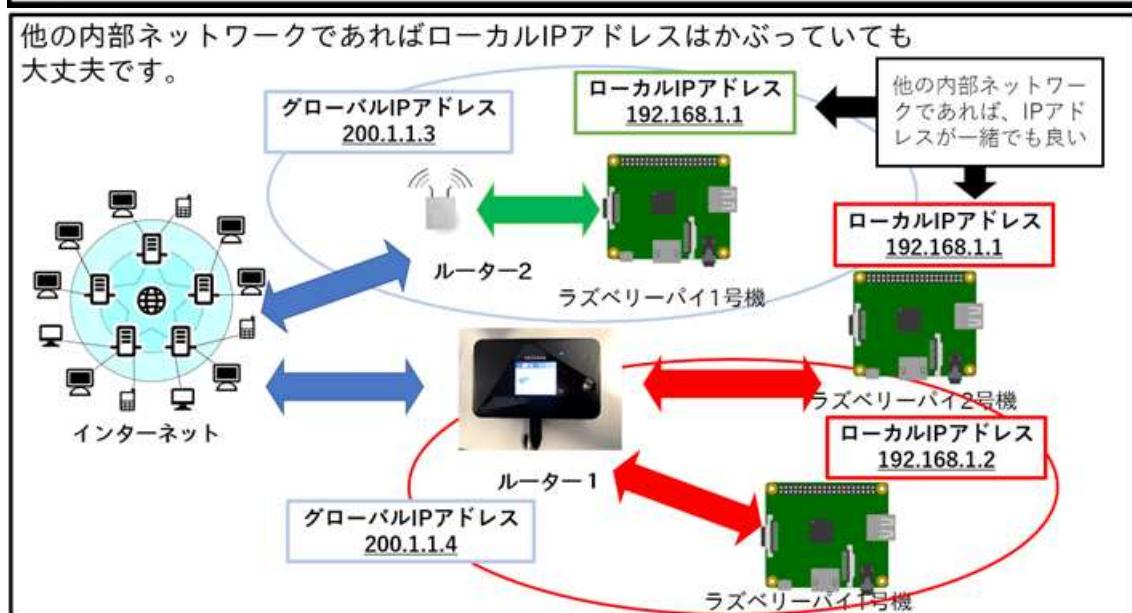
1 - 3 グローバルIPアドレスとローカルIPアドレス

IPアドレスにはグローバルIPアドレスとローカル（プライベート）IPアドレスという二つに分けられます。



サーバー2

他の内部ネットワークであればローカルIPアドレスはかぶっていても大丈夫です。



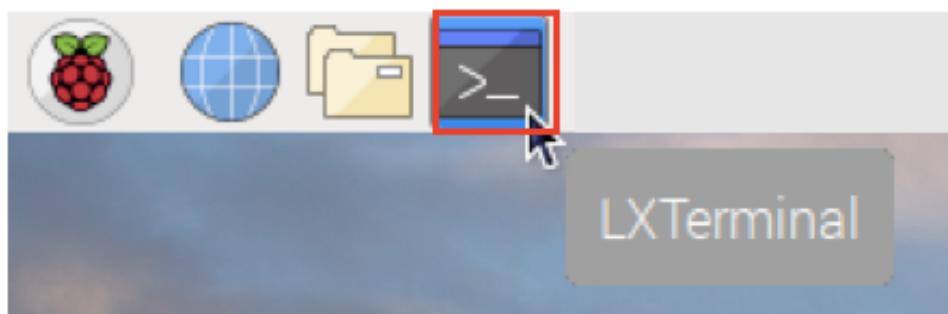
問題 7-2 ローカルIPアドレスがかぶっても良いときはどのときですか。

答え：

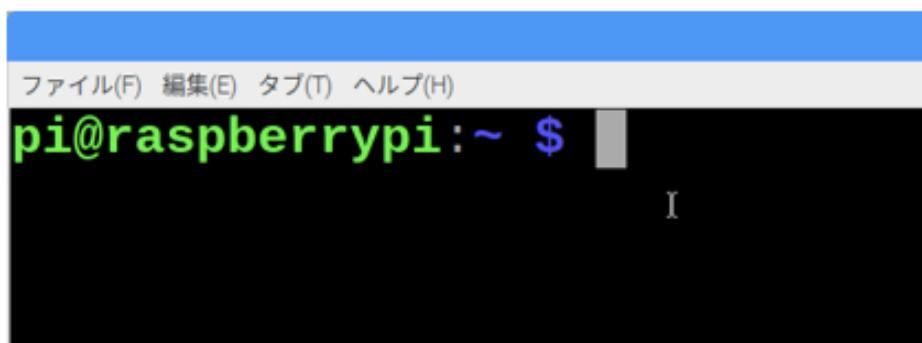
例題 7-1 IP アドレスを調べてみよう

ターミナルにコマンド ” hostname -I ” を入力し、自分のラズベリーパイのローカル IP アドレスを確認しよう

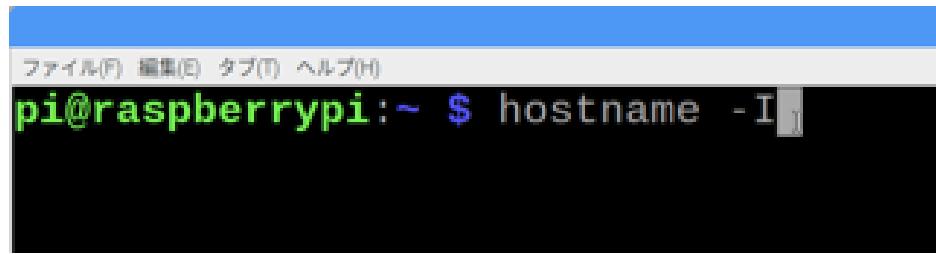
方法



1 赤枠で囲まれているアイコンをクリックし、ターミナルを開こう

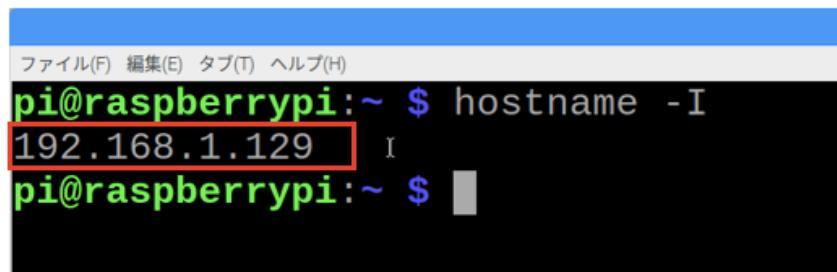


2 ターミナルが図のようになって開いているか確認してね



```
pi@raspberrypi:~ $ hostname -I
```

3 "hostname -I" とコマンドを入力し、
Enterを押そう



```
pi@raspberrypi:~ $ hostname -I  
192.168.1.129 I  
pi@raspberrypi:~ $
```

4 図のように表示されるので自分のラズベリーパイのIPアドレスを確認しよう

調べた自分のラズベリーパイの IP アドレスを書こう

グループの友達の IP アドレスも教えてもらって書こう

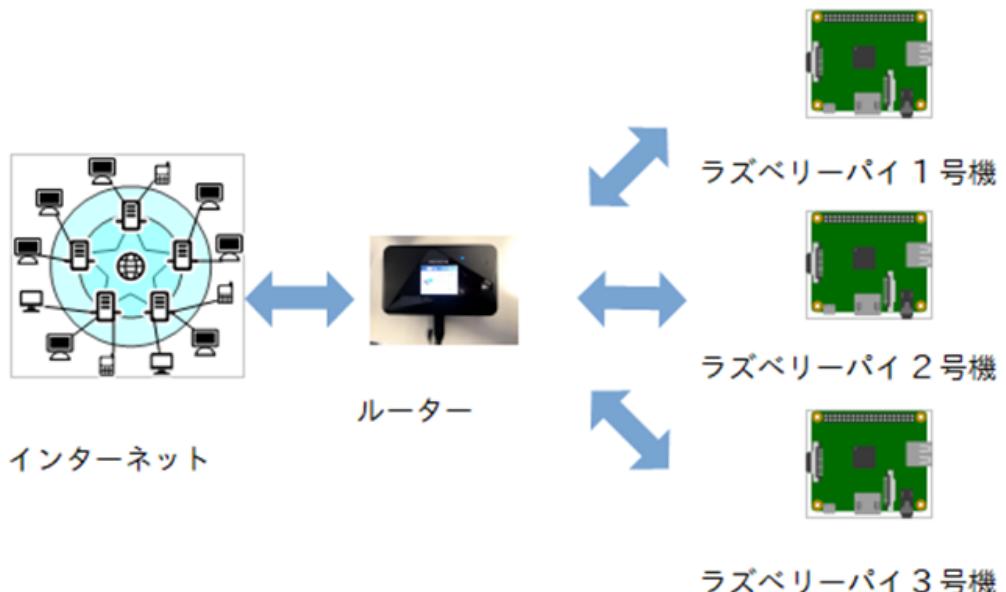
1 - 4 インターネットのつながりについて知ろう

教室内では無線通信を利用してインターネットを使用しています。第1回目の授業の最初に皆さんにはインターネット接続をしました。^{みな}実際にはどこにつながっているかというと下の図のルータです。

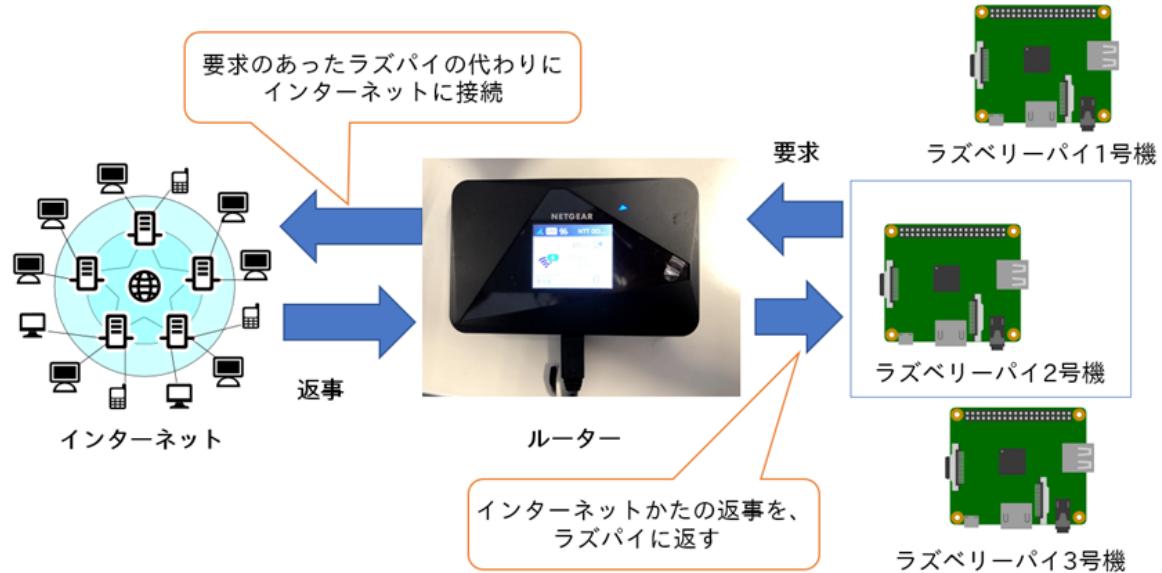


この機械のおかげで、みんなは教室内でインターネットをすることができているんだ。どういう原理で、みんなのラズベリーパイでインターネットを使用できているか説明していくよ。

みんなのラズベリーパイは直接インターネットにつながってはいません。一度、ルータを経由しています。このルータのおかげで複数のラズベリーパイやPCを同時にインターネットにアクセスさせることができるんだ。以下の図がイメージ図だよ。



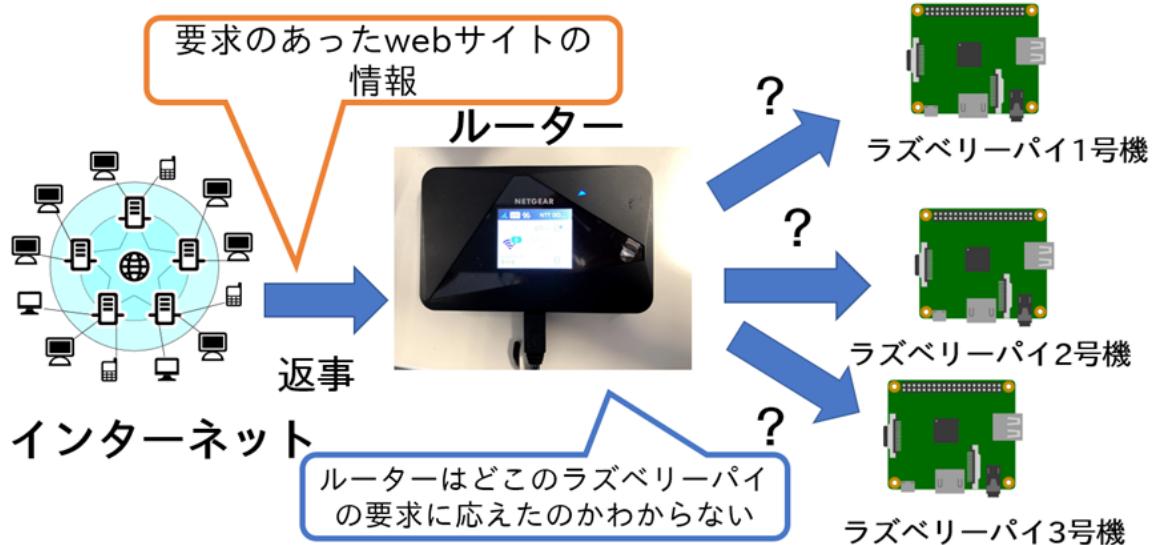
ではどのようにして複数台のラズベリーパイをインターネットに接続しているのでしょうか。それはルータにグローバルIPアドレスというものを割り当てて、ルータが他のラズベリーパイの代表として通信を行っています。どういうことかわかりづらいので、図でみていきましょう。



図ではラズベリーパイ 2号機がインターネットにアクセスし、何かしらの web サイトを観たいと要求したとします。その要求はまずルータのところにいき、ラズベリーパイ 2号機の代わりにインターネットに接続します。その接続で、ルータはインターネット先から返事をもらい、そのインターネットからの返事をラズパイに返しています。

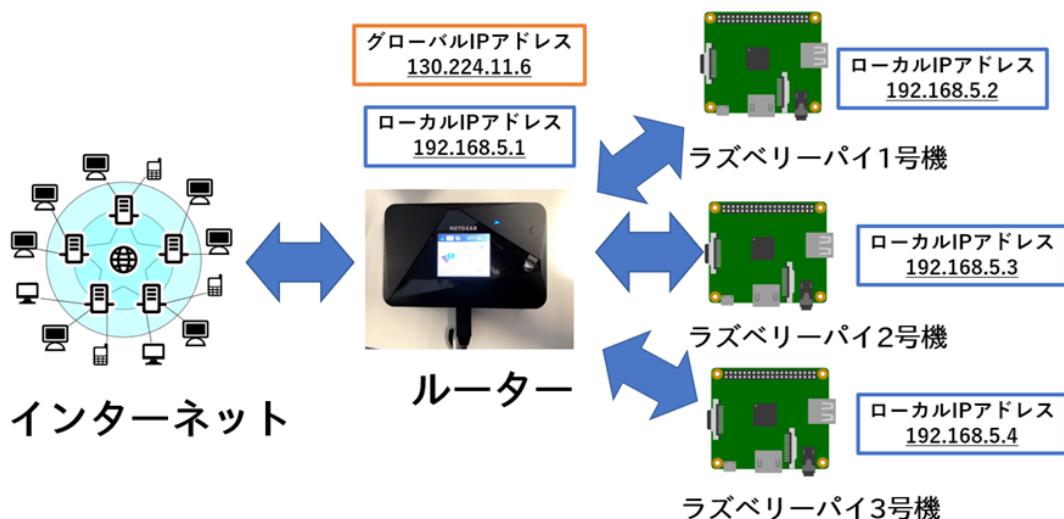
**みんなのラズパイは
基本、ルーターを経由してインターネット
に接続しているよ**

これだけではうまくいかないことがあります。下の図をみてください。



どれかしらのラズベリーパイから web サイトへの要求があり、ルータを経由してインターネットに接続します。そのあと、インターネットから要求のあった web サイトの情報がルータにいきます。ここで問題が発生します。それはいったいどのラズベリーパイの要求であったのかルータがわからないのです。僕らが、勝手にラズベリーパイ 1号機、ラズベリーパイ 2号機、ラズベリーパイ 3号機と名前をつけて判断していても、そのことはルータはわかりません。そこで IP アドレスというものを使います。

ルータは自分を含め、ローカル IP アドレスを割り当てています。このローカル IP アドレスでラズベリーパイを見分けています。



問題 7-3 ルータはラズベリーパイたちを見分けるために何を割り当てていますか。

答え：

例題 7-2 ポケット Wi-Fi ルータのグローバル IP アドレスを調べよう

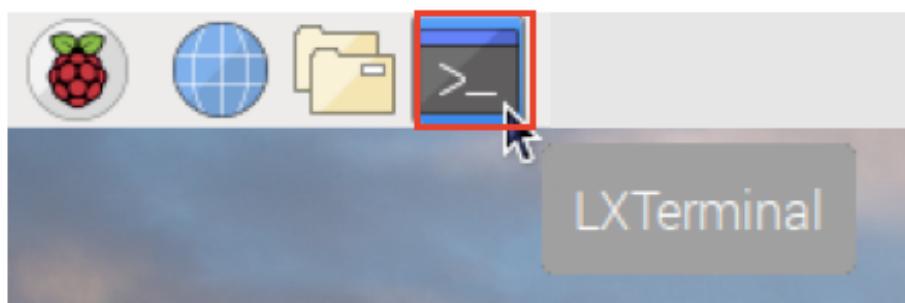
ターミナルを開き、”curl

inet-ip.info”とコマンドを入力して、自分のラズベリーパイが接続している wifi ルータのグローバル IP アドレスを確認しよう

curl コマンドはインターネットから情報を取ってくるときに使用します。今回は”inet-ip.info”というサイトからグローバル IP アドレスを調べてターミナルに表示させています。web ブラウザから”inet-ip.info”にアクセスしてみると右の図のような形でグローバル IP アドレスを確認することもできます。

IP address information	
✓ Your IP Address is 126.147.99.146.	
Field name	Value
ip	[126.147.99.146]
User Name	[pi@raspberrypi:~]
Country Code	[JP]
Country Name	[Japan]
User Agent	[Mozilla/5.0 (Macintosh; Intel Mac OS X 10_13_6) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/76.0.3809.132 Safari/537.36]
Accept	[text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/png,*/*;q=0.8,application/signed-exchange;v=b3]
Accept Language	[ja,en-US;q=0.9,en;q=0.8]
Accept Encoding	[gzip, deflate]
Via	[]
XForwardedFor	[126.147.99.146]
XForwardedPort	[]
XForwardedProto	[http]
Request URI	[/]

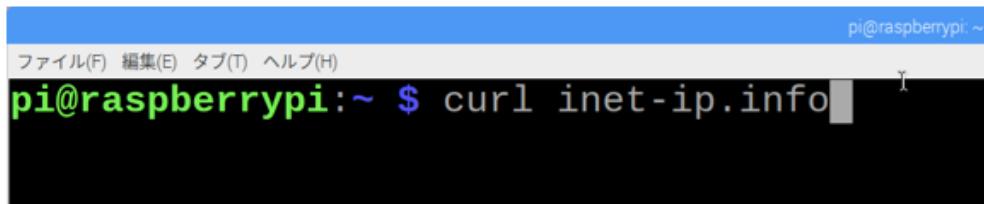
方法



1 赤枠で囲まれているアイコンをクリックし、ターミナルを開こう

A screenshot of an open terminal window titled "LXTerminal". The window has a blue header bar with menu options: ファイル(F), 編集(E), タブ(T), ヘルプ(H). Below the header is a green command line prompt: pi@raspberrypi:~ \$. The window is black with white text.

2 ターミナルが図のように開けているか確認してね

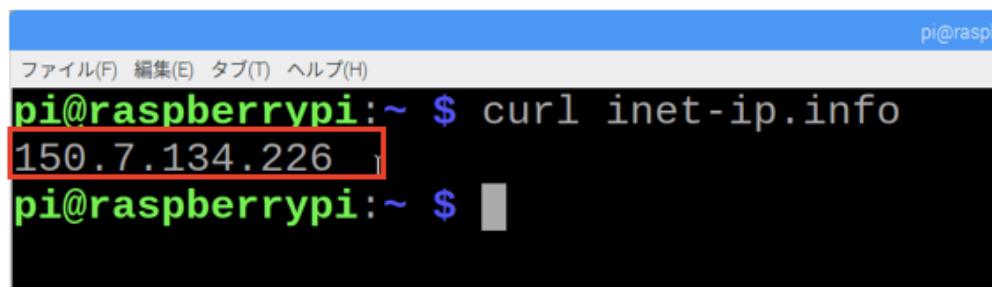


```
pi@raspberrypi: ~
```

ファイル(F) 編集(E) タブ(T) ヘルプ(H)

```
pi@raspberrypi:~ $ curl inet-ip.info
```

3 "curl inet-ip.info" とコマンドを入力し、
Enterを押そう



```
pi@raspberrypi: ~
```

ファイル(F) 編集(E) タブ(T) ヘルプ(H)

```
pi@raspberrypi:~ $ curl inet-ip.info
```

150.7.134.226

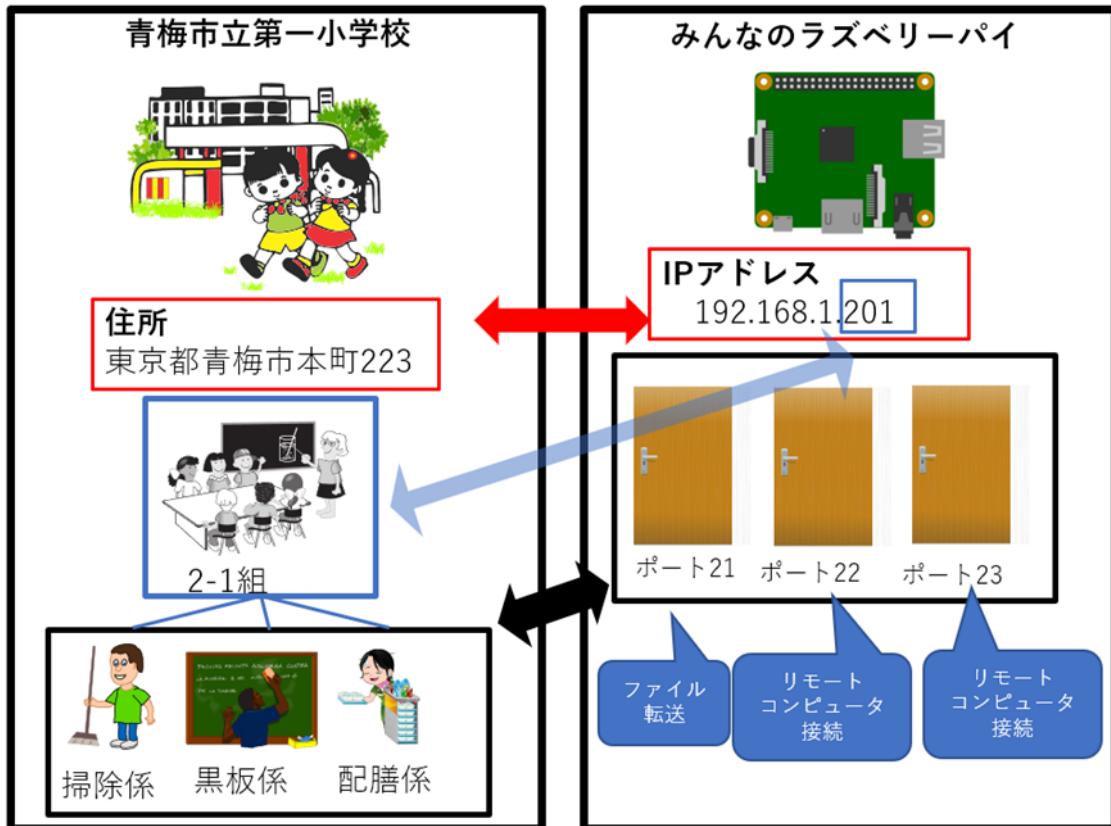
```
pi@raspberrypi:~ $
```

4 図のように表示されるポケットWi-FiのIP
アドレスを確認しよう

調べたグローバル IP アドレスを書こう

グループの友達がしらべたポケット Wi-Fi の IP アドレスも教えてもらって書こう

1 - 5 ポート番号とは



前のページではみんなはラズベリーパイがどのようにインターネットにつながっているのか理解してくれたと思います。ここではさらにポート番号について知識を深めましょう。IP アドレスでは通信相手となる別のネットワークのコンピュータを特定することができます。しかし、コンピュータでは通常、多くのプログラム（機能）が動作していて、IP アドレスを指定するだけでは、どのプログラムと接続するかを区別できません。そこでコンピュータ上の、どのプログラムと接続するかを指定するためにポート番号が用いられます。下の図は小学校を例にしています。

青梅市立第一小学校の住所がみんなのラズベリーパイの IP アドレスに対応しており、IP アドレスの末尾の 201（青で囲まれている）が小学校のクラスに対応しています。皆さんの学校でも生徒にはそれぞれ係という役割が与えられていますね。例えば、プリントなどの配布物を配る配り係、黒板を使い終わったらきれいにする黒板係、教室の窓をきれいにする窓ふき係などこれらの係に相当するものがラズベリーパイにもあり、それがポートです。ポート 21 はファイル転送用の働きをする担当で、ポート 22 と 23 は他のコンピュータに接続する担当です。ポートには役割がありインターネットの通信はファイル転送であったり、他のコンピュータに接続して遠隔操作したり、web ページにアクセスしたりといろいろあります。なので、その分のポートを用意しておく、役割を与え、対応している役割のときに頑張ってもらう仕組みになっています。みんなのラズベリーパイやパソコン、サーバにはポートは何個あるのか調べてみるのも面白いですね。

問題 7-4 ポート番号はどのようなことに使われますか。

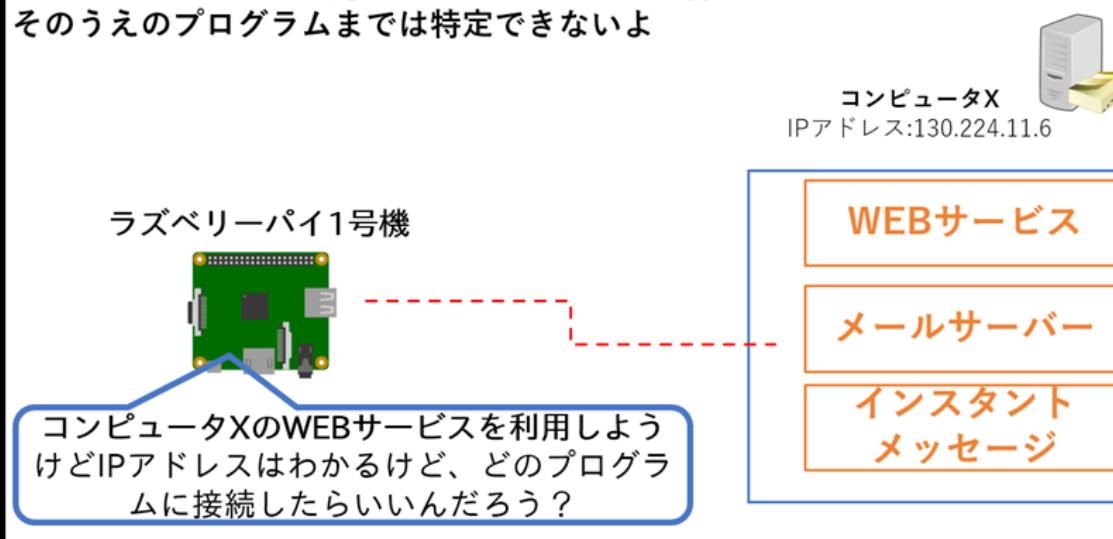
答え：

ここでは実際にポートはネットワーク上ではどのように使うのか、下の図を見てみましょう。

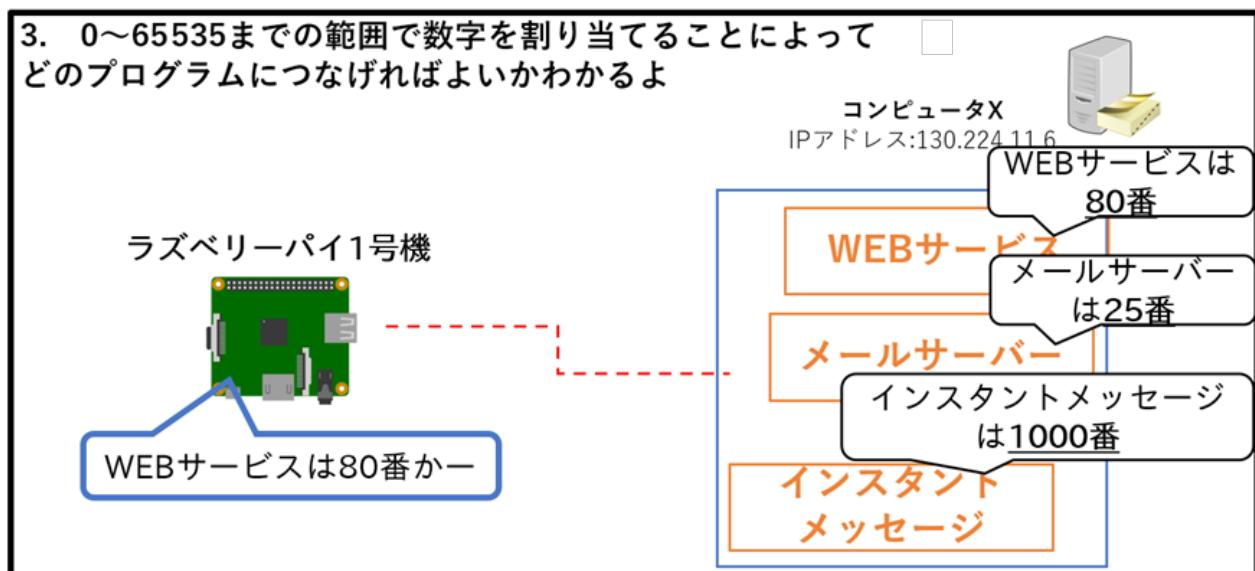
- 1.通常、コンピュータ上では複数のプログラムが動いているよ
(下の図ではWEBサービス、メールサーバ、インスタントメッセージ)



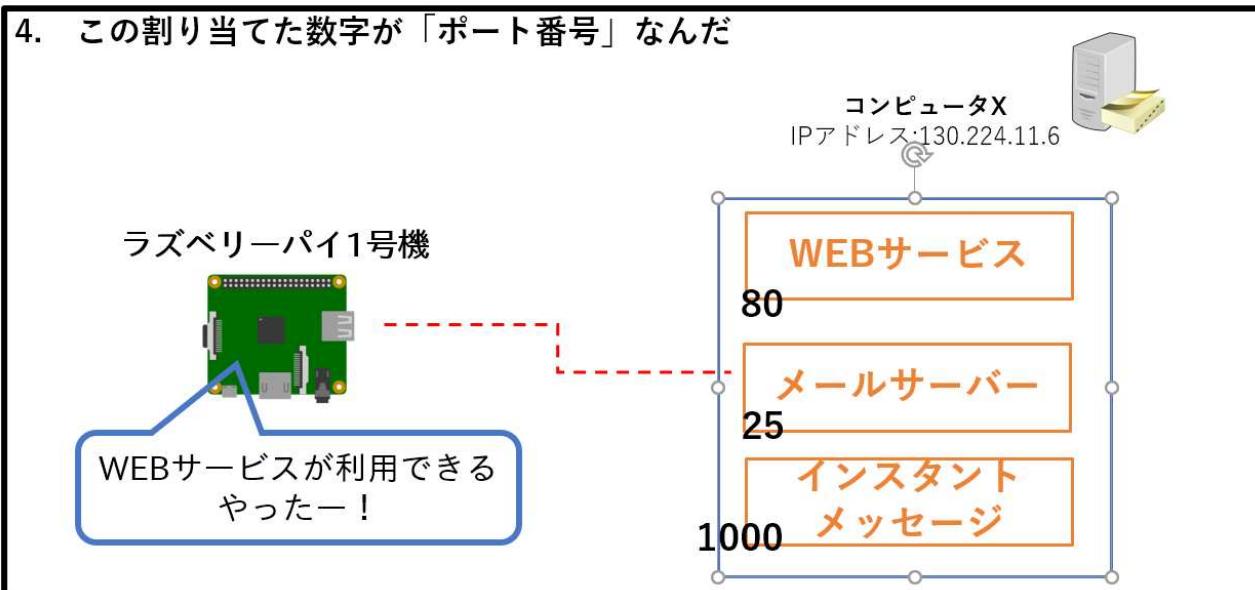
- 2.IPアドレスではあて先となるコンピュータは特定できても
そのうえのプログラムまでは特定できないよ



3. 0～65535までの範囲で数字を割り当てることによって
どのプログラムにつなげればよいかわかるよ



4. この割り当てた数字が「ポート番号」なんだ

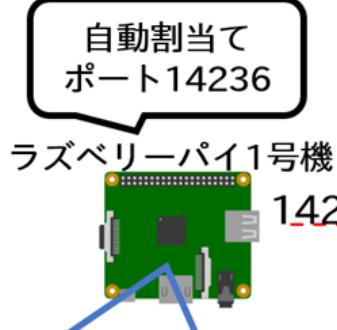


さらにラズベリーパイ側のポートはラズベリーパイが自動で割り当ており、そこからデータを受け取っているよ。

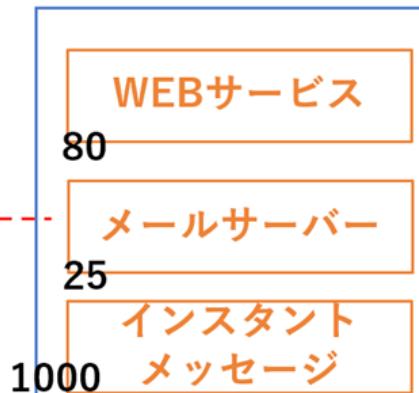
ラズベリーパイ側のポートは
ラズベリーパイが自動で割り当てているよ



コンピュータX
IPアドレス:130.224.11.6



ポート14236でWEBサービスの
データを受け取ろう



コンピュータXのようにネット上にサービスを提供するコンピュータをサーバといいます。
ラズベリーパイ1号機のようにサーバからサービスを受けるコンピュータをクライアント言います。

これらのポートが今現在どのように使われているのか例題7-3で確認してみましょう。

問題7-5 コンピュータ上のプログラムを判断するためになにが用いられていますか。

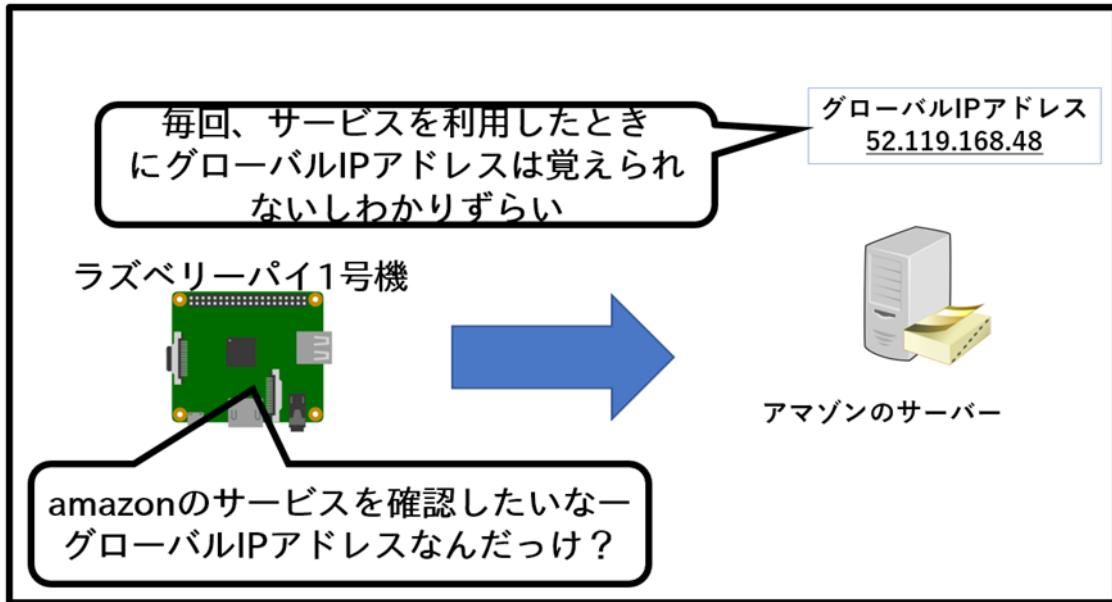
答え：

ネット上にサービスを提供するコンピュータとネット上のコンピュータからサービスを受けるコンピュータをそれぞれ何といいますか。

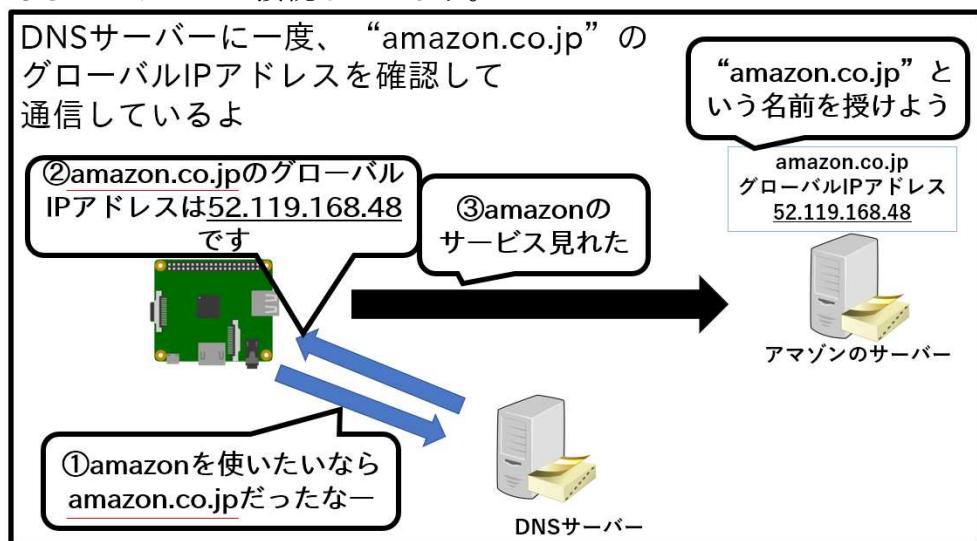
答え：

1 - 6 DNS (Domain Name System) について

IP アドレスは住所であり、それがわからないとサーバのサービスをうけることはできません。ですが以下の図のようなことが起こることがあります。



そこで、DNS サーバとと言うものを作り、一度そこに ”amazon.co.jp” のグローバル IP アドレスを教えてもらってサーバに接続しています。



dig コマンドを用いるとグローバルIPアドレスが確認できるよ

```
$ dig amazon.ac.jp
```

グローバルIPアドレス
(複数表示される可能性があります)

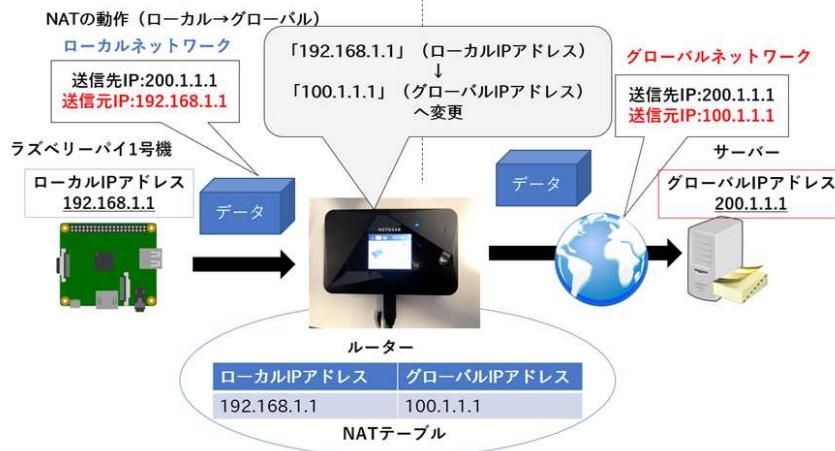
;;ANSWER SECTION
amazon.ac.jp 0 IN A 52.119.168.48
amazon.ac.jp 0 IN A 52.119.164.121
amazon.ac.jp 0 IN A 52.119.161.5

問題 7-6 ターミナルを開き、”dig amazon.co.jp” を実行してグローバル IP アドレスを確認しよう。

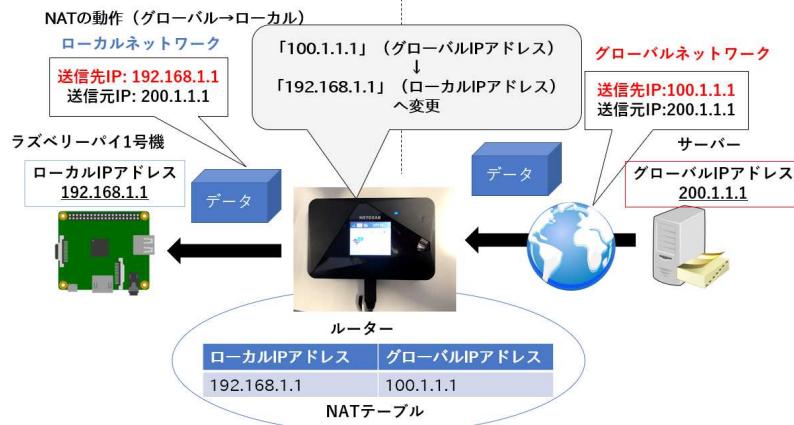
答え：

コラム NATについて

教科書の前のページでローカルIPアドレスとグローバルIPアドレスが存在することを皆さんには知りましたね。ここではもっと細かくIPアドレスを用いたインターネット接続を勉強しましょう。インターネットに接続するコンピュータは自分自身を示すIPアドレスとして世界で唯一のグローバルIPアドレスを使わなければならないのです。みなさんのラズベリーパイにはローカルIPアドレスしか割り当てられていませんので、このままではインターネットをすることはできません。そこでNAT(Network Address Translation)というものを用います。このNATはルータが行っていて、送信元のローカルIPアドレスをグローバルIPアドレスに変換して通信を行っています。具体的にはどのようにになっているのか下の図をみて確認してみよう。



ラズベリーパイ1号機からサーバにデータの送信先IPアドレスは200.1.1.1(グローバルIPアドレス)、送信元IPアドレスは192.168.1.1(ローカルIPアドレス)です。NATを行うルータは、プライベートアドレスとグローバルアドレスの境界に位置しています。ここでルータは、ラズベリーパイから送信されたデータの送信元IPアドレス192.168.1.1を、グローバルIPアドレスの「100.1.1.1」に変換して転送します。このときルータは、その変換情報をNATテーブルに記録しています。この記録が、「返ってくるデータ」つまりサーバーからラズベリーパイへ送られるデータの転送時に役立ちます。次の図を見てください。

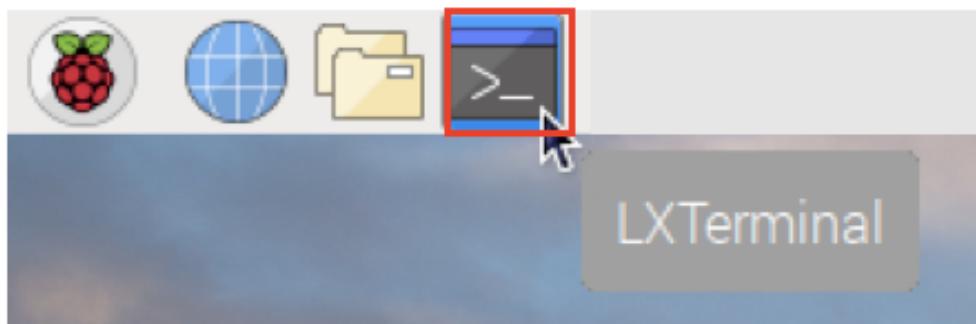


サーバーからラズベリーパイへの返事のデータの送信先IPアドレスは「100.1.1.1」、送信元IPアドレス「200.1.1.1」になりますね。データがルータへ転送されてくると、ルータはNAT

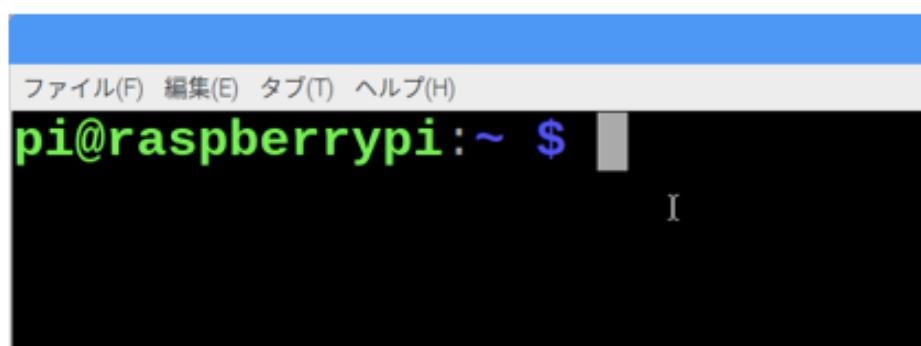
テーブルを確認して、送信元 IP アドレスを「100.1.1.1」から元の「192.168.1.1」へ変換します。これにより返りのデータの通信が可能になるのです。NAT によって送信元のローカル IP アドレスをグローバル IP アドレスに変換して通信をおこなっているおかげでみんなはインターネットを利用できているんだね。

例題 7-3 開いているポートを調べてみよう

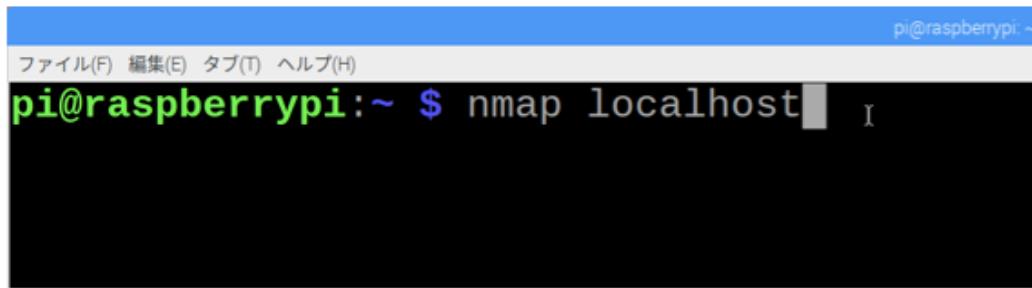
ターミナルを開き、” nmap localhost ” とコマンドを入力し、ポートについて知ろう
方法



1 赤枠で囲まれているアイコンを
クリックし、ターミナルを開こう

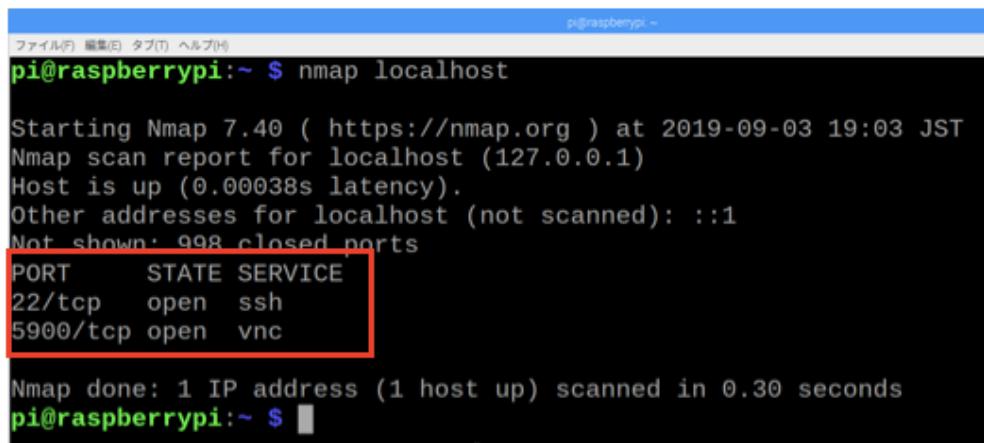


2 ターミナルが図のよう開けて
いるか確認してね



```
pi@raspberrypi:~ $ nmap localhost
```

3 “nmap localhost” とコマンドを入力し
Enterを押そう



```
pi@raspberrypi:~ $ nmap localhost

Starting Nmap 7.40 ( https://nmap.org ) at 2019-09-03 19:03 JST
Nmap scan report for localhost (127.0.0.1)
Host is up (0.00038s latency).
Other addresses for localhost (not scanned): ::1
Not shown: 998 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
5900/tcp  open  vnc

Nmap done: 1 IP address (1 host up) scanned in 0.30 seconds
pi@raspberrypi:~ $
```

4 表示されているものを確認しよう

自分のラズベリーパイに表示されたものを下の表に書こう

PORt	STATE	SERVICE

service に出てきた言葉をインターネットを使って調べよう。皆さんにとっては難しい言葉が多いと思います。それでも自分で調べてみましょう。それが知識を深める近道です。
ポートが開いていない場合、コマンドは何も表示せず、改行が起こるだけのこともあります。

2 - 1 HTMLについて

例題 7-4 HTMLの復習

考え方

じこしょうかい
第1回で作成した自己紹介ページのタグや機能を理解することでHTMLについて自分の思ったとおりに作れるようになります。

HTMLは、タグという一つの機能をつかい、ウェブページを作り出します。

タグというのは、<></>という形です。見覚えのある形ですね。

例えば、

<p>文章をここに書きます。</p>

のようなタグをpタグと呼びます。pタグはタグの間にある文章を表示します。

まずは、第1回目の授業でみささんが作成した自己紹介ページとその画像のファイルを第6回のフォルダにコピーし、それをmousepadでhtmlファイルを開いて見ましょう。

1. cp ~/01/self_intro.html ~/07/www/index.html

cpコマンドを使って、自己紹介ページのhtmlと
<!-- コメント -->
<html lang="ja">
 <head>
 <meta charset="utf-8">
 <!-- 例題 1-20 ホームページの中身をのぞいてみよう -->
 <!-- 自己紹介ページ から "自分の名前"の自己紹介ページ -->
 <title>自己紹介ページ</title>
 </head>
 <body>
 <!-- 例題 1-21 見出しを作ってみよう -->
 <!-- 見出しの〇〇〇を自分の名前(ニックネームでもよい)! -->
 <p>○○○の自己紹介ページ</p>
 <!-- 例題 1-22 自分の好きな画像をはってみよう -->

 <!-- 例題 1-23 自分のことを紹介しよう -->
```

開いたhtmlファイルの中に、実際にはウェブページに出てこない文章がちらほら見えます。

<!-- コメント -->

それは、コメントという機能でウェブページには表示されない、メモのような機能でプログラムを書くときにあると便利です。HTMLに限らずHSPや様々なプログラム言語、はたまた現実世界でもメモというの大切なので必要な情報はしっかりと書いておきましょう。

self\_intro.html内に使われたタグをまとめました。これを参考に自己紹介ページをより良くしていきましょう。

## HTMLまとめ

| タグ名             | 効果                                                               |
|-----------------|------------------------------------------------------------------|
| <title></title> | ウェブページのタイトルをつける                                                  |
| <h1></h1>       | 見出し(h1～h6)まで使える。h1が一番大きい                                         |
| <ul></ul>       | 点付のリスト。リストの項目はliを使って表す                                           |
| <li></li>       | リストの項目。ulもしくはolタグなどの中で使う                                         |
| <p></p>         | 段落を作るために使う。文字を表示したいときはこれを使う。改行は自動で行われるので自分のタイミングで行いたいときは<br>を使う  |
| <br>            | 改行。pの中でも使うことができる                                                 |
| <i></i>         | イタリック(少し傾いた感じの文字)                                                |
| <u></u>         | 文字の下に線を引くときに使う                                                   |
| <a href=""></a> | リンクを貼りたいときに使う。hrefにはクリックしたときに飛びたいURLを入れる。                        |
| <img src="">    | 画像を表示するときに使用する。srcには画像の場所を指定する。ラズベリーパイの中の画像を指定しても、URLとして指定しても良い。 |
| <ol></ol>       | 番号付きのリスト。liでリストの項目を決める。<br>番号はliの書いた順番通りになる                      |

HTMLで使いたい色を探すときにはここを使うと便利です。

<https://www.colordic.org/m/>

地下鉄のシンボルカラーとカラーコードで使いたい色がわかるはずです。

<https://www.colordic.org/>

色の名前とカラーコードがわかります。色がまんべんなくあります。

<https://tech-unlimited.com/color.html>

上級者向け、色の三原色というものから実際にHTML用のカラーコードに変えるサイトです。

また、HTMLにはボタンや文字を書き込む部分も作れます。

通常見るだけのウェブページだったものがデータの送受信を可能とするなど、可能性を広げられます。

一部のタグを下に記載します。

|                                                                      |                                                                                  |
|----------------------------------------------------------------------|----------------------------------------------------------------------------------|
| <form></form><br>(使用例)<br><form action=" cgi-bin/formmail.cgi " >    | 入力送信フォームを作成<br>form タグの中に<input>, <select>, <textarea>などをいれることで<br>ボタンや文字入力欄を作れる |
| <input></input><br>(使用例)<br><input type=" password " name=" pass " > | 文字の入力や、ボタンなどを作成するタグ。<br>input の次に type などの属性を入れることで<br>一行の文章やパスワードの入力などを決定する     |

それでは、formを使った例題を説いてみましょう。

#### 例題

自己紹介ページの一番下から2行上あたりにある</body>の上の行から<form>タグで文字の入力をし、送信ボタンを押せる機能を書きましょう。

#### 考え方

はじめに、formタグの間(<form> \* \*ここ\* \*</form>)に、inputタグでテキストボックスを作る。そして、同じようにinputタグを使って送信ボタンという名前のボタンを作成する。

1. <form></form>を書き込めるスペースを作ろう。

```
133
134
135
136
137
138 </body>
139 </html>
140
```

<ul>  
 <li>ディスプレイ</li>  
</ul>  
<hr>

```
135
136
137
138 <form>
139 <input type="text" name="text_box">
140 <input type="button" value="送信" name="submit_button">
141 </form>
142 </body>
143 </html>
```

2. <form>タグを書き足そう。

```
136
137
138
139
140
141
142 </body>
143 </html>
144
```

←

<html> や <body> タグと同じように、  
この間にいろいろなタグを入れることができます。  
これからはそれをていきましょう。

3. <form>タグの間に、行を作り「テキストボックス」という文字が出るようにしよう。

```
<form>
 <p>テキストボックス</p>
</form>
```

4. 次の行に<input>タグを入れよう。その際に、テキストを書き込めるようにしよう。

画像のようにするためには、

```
<form>
```

```
<p>テキストボックス</p>
<input type="text" size="10">
</form>
```



5. 同じようなやり方で、先にかいた<input>タグから改行して次の行に<input>タグで送信ボタンを作成しよう。

```
<input type="submit" value="送信">
```

6. 作成した HTML ファイルを保存し、実際にブラウザで作成されているか確認してみましょう。

このように、form タグ、input タグには便利な機能があります。

以下に、input タグの属性を一部記載しておきますので、[参考にして](#)問題を解いていきましょう。

## 問題 7-7

1. 最初に書いたテキストボックスの属性をパスワード入力ボックスに書き換えてみよう。
2. 食べ物の名前を<p>タグを使って3つ追加してその横にチェックボックスを置きましょう。
3. リセットボタンを送信ボタンの次の行に追加してみよう。

わからなかったら、先生に質問をすること。その間は一覧を見て考えてみよう。  
input タグの機能になるもの

type= "text"	1行のテキストボックスを作ります。 size の属性を指定することで大きさも決定で きる。
type= "password"	パスワード用のテキストボックスを作ります。  入力されたら＊のマークなどに置き換わりま す。送信する際には普通の文字で送られるの に注意。
type= "checkbox"	チェックボックスを作ります。複数選択可 能で、checked 属性を指定すると予めチェック が入ったかどうかを決めれる。
type= "radio"	ラジオボタンを作ります。複数の中から1つ しか選択できないのが特徴。 こちらも checked で状態を決めれる。
type= "submit"	送信ボタンを作ります。
type= "button"	普通のボタンを作ります。
type= "image"	画像のボタンを作ります。 使用する画像ファイルは、src 属性で指定し ましょう。また alt 属性が必須。
type= "reset"	リセットボタンを作ります。 このボタンを押すと、書いていたデータがな くなり最初の状態に戻ります。

## 2 - 2 例題 7-5 自分のホームページを公開しよう

### 考え方

クライアントとサーバーと呼ばれるコンピュータがあります。クライアントとはサービスを利用するコンピュータです。サーバーとはそれと逆にサービスを提供しているコンピュータです。クライアントとサーバーは同じコンピュータにあってもOKです。例えば、googleは検索するためのウェブサイトを提供しています。ウェブサイトを提供するサーバーをウェブサーバーと呼びます。自分のホームページを公開するためには、ウェブサーバーを動かす必要があります。

### 注意

今回のみんなのホームページはローカルIPの間だけです。つまり、インターネット上に公開はしません。

これを公開して、実際確認したりみんなで見せ合いをしてみましょう。

1. まず~/07/wwwのディレクトリに移りましょう。

ターミナルを立ち上げて次のコマンドを打ちましょう。

```
cd ~/07/www
```

2. サーバーを動かしてみましょう。

次のコマンドを打ちましょう。

```
./webserver.py
```

を実行するとウェブサーバが動き始めます。

webserver.pyを起動したディレクトリがドキュメントルートになります。

これはファイルシステムのルートディレクトリとは別のものでドキュメントルートはウェブサーバから見たルートディレクトリのことです。

つまりウェブサーバーしかそのディレクトリをルートディレクトリとは認識しません。

ディレクトリに関しては第3回の教科書3.3.2「ディレクトリの関係」を参考にしてください。

ファイル(F) 編集(E) タブ(T) ヘルプ(H)

```
pi@raspberrypi:~ $ cd ~/07/www/
pi@raspberrypi:~/07/www $./webserver.py
Serving HTTP on 0.0.0.0 port 3000 (http://0.0.0.0:3000/) ...
```

3. サーバーが動いているかどうかを確認してみましょう。

別のターミナルを立ち上げ、次のコマンドを打ちましょう。

```
nmap localhost
```

今回使っているポート番号は:3000です。STATE(状態)がopen(開く)になっています。

これでサーバーが動いていることが分かります。

```

ファイル(F) 編集(E) タブ(T) ヘルプ(H)
pi@raspberrypi:~ $ nmap localhost
Starting Nmap 7.40 (https://nmap.org) at 2019-09-20 00:54 JST
Nmap scan report for localhost (127.0.0.1)
Host is up (0.00065s latency).
Other addresses for localhost (not scanned): ::1
Not shown: 997 closed ports
PORT STATE SERVICE
22/tcp open ssh
3000/tcp open ppp
5900/tcp open vnc

Nmap done: 1 IP address (1 host up) scanned in 0.31 seconds
pi@raspberrypi:~ $

```

4. みんなが作ったウェブページを表示してみましょう。

みんなが作ったindex.htmlの場所は webserver.py を起動したディレクトリと同じでしたね。「2」で説明したドキュメントルートです。

これは webserver.py を起動したディレクトリ

~/07/www/

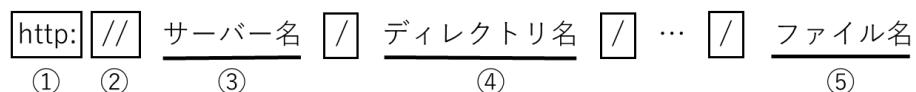
を指しています。

次にブラウザのアドレスバーに打つURLはどのように書けばいいのでしょうか。

URLの書き方は次のようになります。

なので自分のパソコンのサーバーを見るには

### URLの意味



① アクセスする方法

② これの後ろがサーバーの名前(ドメイン名)

③ サーバーの名前

IPアドレスを書ける

自分のパソコンは「localhost」と書ける

登録するとドメイン名を使うことができる

④ ドキュメントルートの下にあるディレクトリを指すときに使う

⑤ 見たいファイル名

<http://localhost:3000/>

というURLになります。

サーバー名の後の/（スラッシュ）はドキュメントルートを示しています。

ブラウザはファイル名を指定しないと index.html というファイルを開くように設定されています。

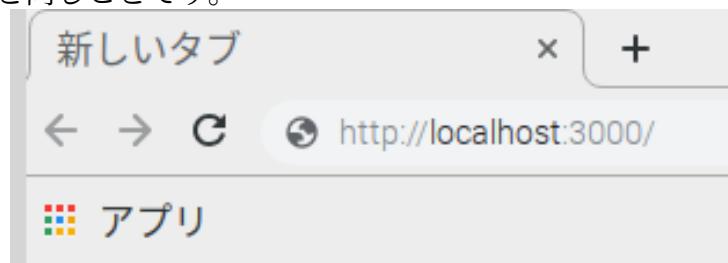
<http://localhost:3000/>

をブラウザのアドレスバーに打ってみましょう。

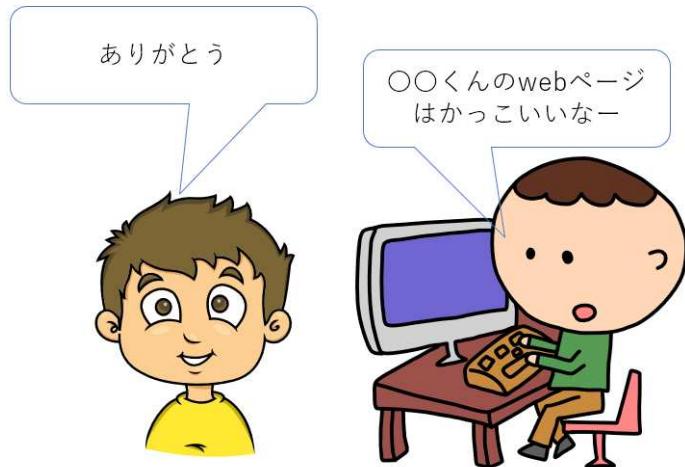
みんなが作ったウェブページのHTMLが表示されましたね。

これは

<http://localhost:3000/index.html>  
と入力しているのと同じことです。



## 2 - 3 例題 7-6 友だちのホームページを見てみよう



1. 左隣の友達のIPアドレスを確認しよう。

友達のターミナルで次のコマンドをうってもらいましょう。

hostname -I

そのIPアドレスをメモしましょう。

[ ] さんのIP: [ ]

2. 友達のウェブページを見て見ましょう。

自分のサーバーを見るときは

<http://localhost:3000/>

でした。localhostというのが自分のサーバーという意味でした。

今回は友達のサーバーを見たいのでlocalhostの代わりに友達のIPアドレスを打ちましょう。

<http://xxx.xxx.xxx.xxx:3000/>

<http://xxx.xxx.xxx.xxx:3000/index.html>

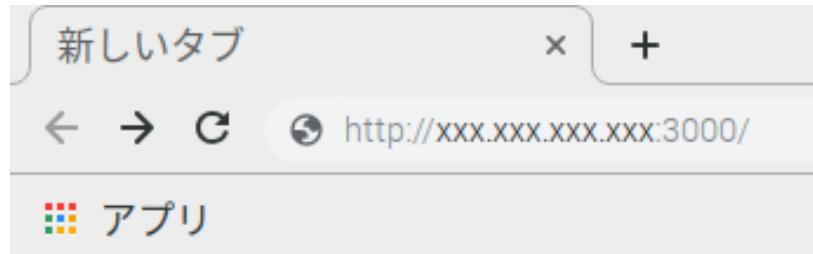
をブラウザのアドレスバーに打ってみましょう。

もし左隣の友達のIPアドレスが192.162.33.5だったら

<http://192.162.33.5:3000/index.html>

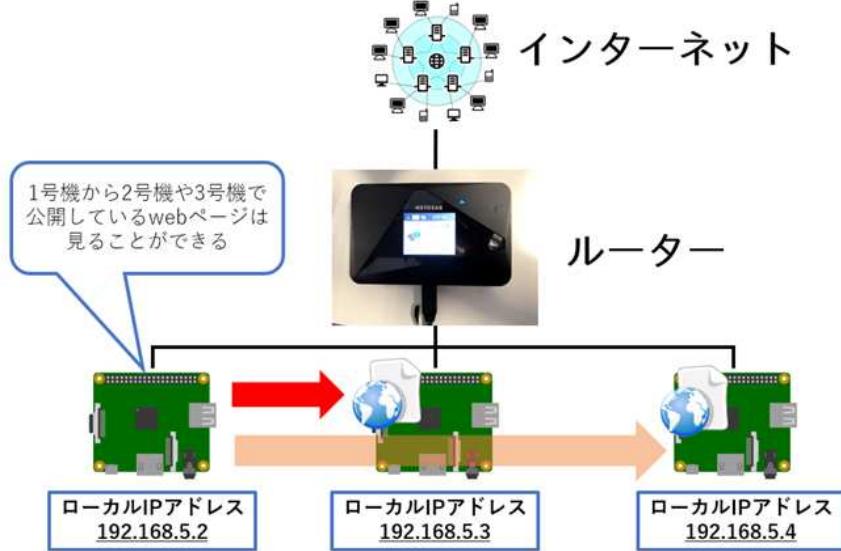
となります。

友達のウェブページが表示されたら成功です！

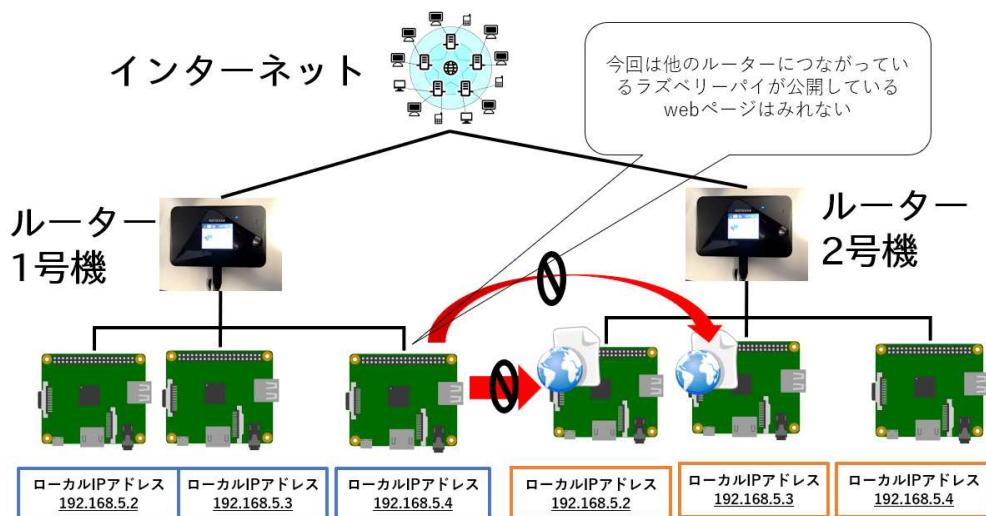


## 注意

例題 7-6 の友達のホームページを見てみようでは、以下の図のように同じルータにつながっているラズベリーパイが公開しているホームページは、見ることができます。基本同じグループの子たちのホームページは見れるので確認してみてください。



あと、他のルータに接続しているラズベリーパイが公開しているホームページは今回みることができません。実際に試してみても面白いと思います。



問題 7-8 他のグループの友達のホームページを見てみよう、自分とは違う Wi-Fi ルータに接続している友達のホームページは見れたかどうか答えに書こう

答え：

---

### 3 - 1 CGIについて勉強しよう

ここではCGIについて勉強しよう。第1回で作成したwebページはいわば静的なページです。静的なページとは、常に同じ画面を表示するページのことです。ですが、CGI（Common Gateway Interface）を用いることによって動きのあるページ、動的なページを作成することができます。動的なページとは表示させるたびに違う画面を表示することができるページのことです。アクセスカウンターはたまにwebページについていることがあります。アクセスカウンタというのとは、そのwebページに今までどれぐらいの人がアクセス（接続）したのかを表示させるものです。webページにアクセスするたびにアクセスカウンタの数は増えていき、毎回同じ数字にはなりません。これはアクセスするというアクションによってページに変化を起こしているのです。さらにCGIを用いることによって以下の機能を作成することができます。

- ・アクセスカウンター



SOS団のサイトによこそ！



- ・アンケートフォーム

A screenshot of a survey form titled 'アンケートにご協力ください。' (Please cooperate with the survey). It asks for the user's name and then lists two sets of questions with multiple choice options. The first question is 'あなたはどの食べ物が好きですか？' (Which food do you like?) with options: 和食 (Japanese), 中華 (Chinese), フランス料理 (French), イタリアン料理 (Italian), and その他 (Others). The second question is 'あなたのご予算はいくらぐらいですか？' (How much is your budget?) with options: 1000円未満 (Under 1000 yen), 1000円以上2000円未満 (1000 yen to 2000 yen), 2000円以上3000円未満 (2000 yen to 3000 yen), 3000円以上4000円未満 (3000 yen to 4000 yen), and 4000円以上 (Over 4000 yen). At the bottom are '送信' (Send) and 'リセット' (Reset) buttons.

- ・掲示板

掲示板

名前（必須項目）：

メールアドレス：

URL： http://

削除パスワード：

内容（必須項目）：

最初の 10 件を表示

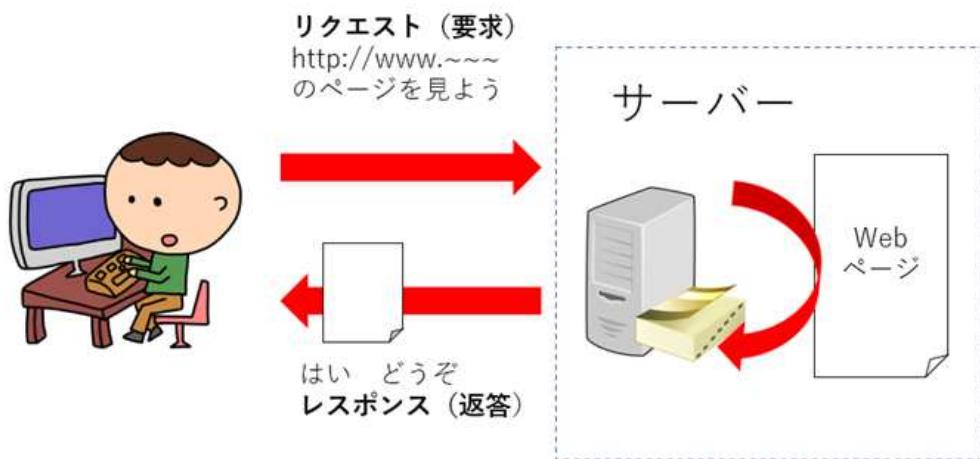
OK! [URL: http://www.♦URL.ne♦.jp/ok/] 書き込み 9/8(火)03:10:31 [レス] [削除]

OK!

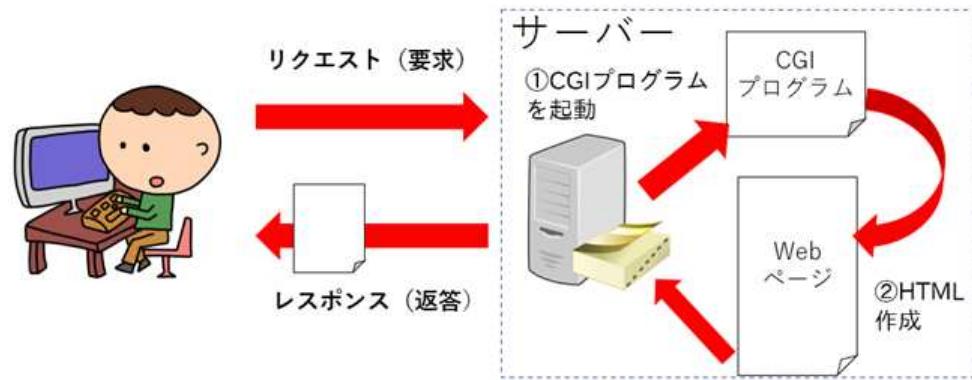
Test [URL: http://www.♦URL.ne♦.jp/test/] 書き込み 9/8(火)03:06:40 [レス] [削除]

Test

みんなが作った web ページやほかの静的なページはどのようなやり取りで表示させているか見ていきましょう。次の図を見てください。



web ブラウザで検索したりして web ページを表示させることをリクエスト（要求）といい、そのリクエストを受け取ったその web ページのあるサーバーが要求したところにレスポンス（返答）を返します。要求されたら返すのみで、web ページに変化はなく同じものを見せていました。ですが次の画像みてください。



CGIを使った動的ページでは、サーバーが皆さんからwebページを要求されるたびに、CGIプログラムで新しいHTMLを生成しています。それにより、毎回違うwebページを見せることが可能なのです。この教科書では皆さんのラズベリーパイと今までの講義で学んだことを生かしたCGIを体験していきます。

**問題 7-9** 動的なページとはどのようなものでしょうか。また、動的なページを作成する場合に用いるものは何でしょう。

答え：  
\_\_\_\_\_

答え：  
\_\_\_\_\_

## 例題 7-7 CGIを使ってウェブページからLEDをつける

考え方

まずはraspberry piにセンサーボードを取り付けましょう。今回はFaBoは使用しません。

ウェブサーバはウェブブラウザから要求を受けます。”ファイル名.hsp”ファイルへのアクセスをブラウザから行うとサーバはCGIプログラムをサーバ側で起動します。結果はHTMLとしてブラウザ（クライアント）へ返されます。

試しに、CGIプログラムを起動させてみましょう。例題のプログラムはLEDがすでに点灯していれば消灯、消灯してれば点灯するプログラムです。ウェブサーバがウェブブラウザから要求を受けて、CGIプログラムを起動するのでウェブサーバが起動している必要があります。まずはウェブサーバを起動させましょう。

ターミナルを開いて

```
cd ~/07/www/
```

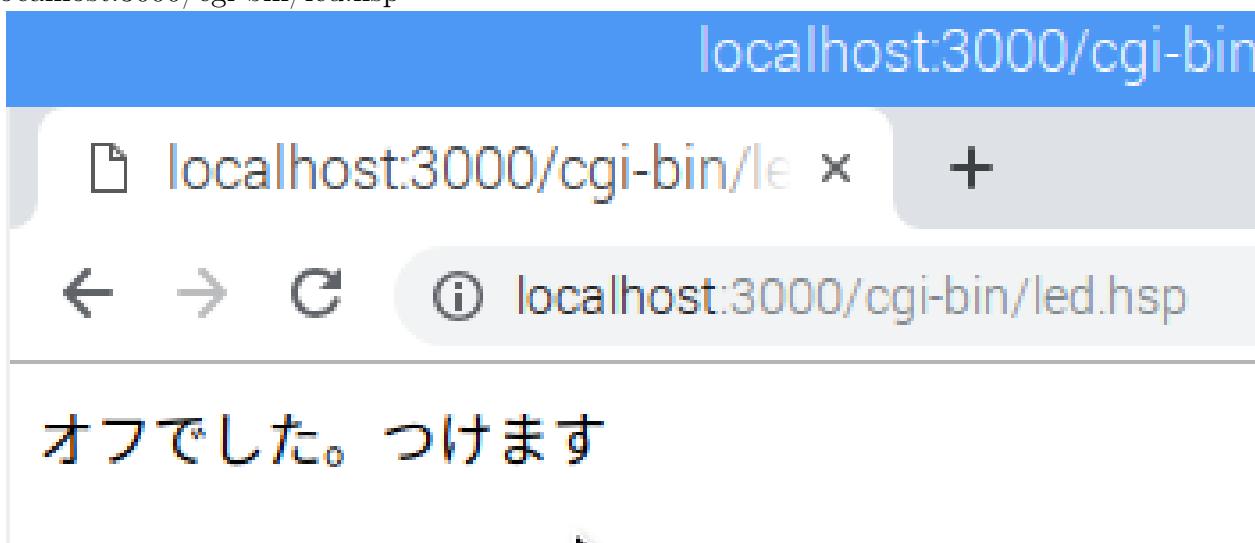
```
./webserver.py
```

 で起動できます。

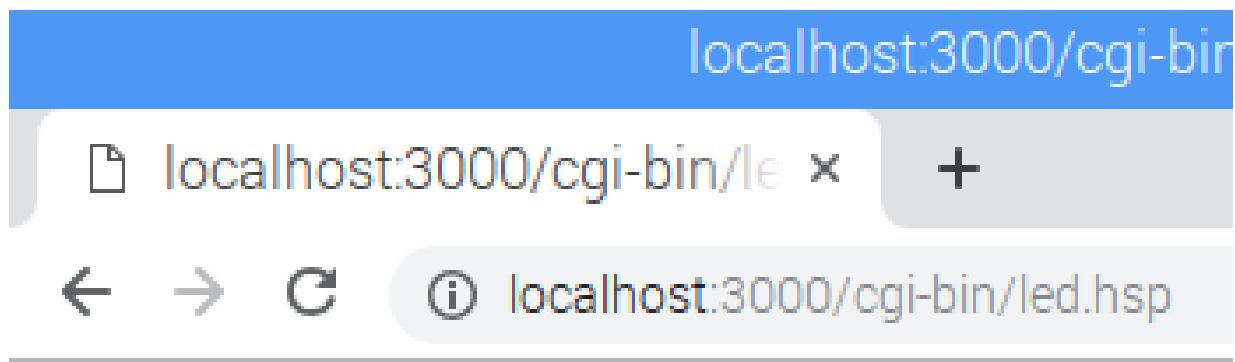
次にウェブブラウザからウェブサーバへCGIの要求を出してみましょう。

ブラウザを開いて、

```
localhost:3000/cgi-bin/led.hsp
```



にアクセスをしてみましょう。アクセスをすると、プログラムが起動してLEDが点灯します。ブラウザのリロードボタンを押すと消灯します。



cgi-bin ディレクトリは CGI プログラムを入れるディレクトリです。~/07/www の下にあります。このディレクトリ内の  
”ファイル名”.hsp  
は CGI プログラムとして扱われます。led.hsp は CGI プログラムです。

## プログラム解説

```
1. #include "hsp3cl.as"
2. #include "rpz-gpio-cl.as"
3.
4. mes "Content-type: text/html\n"
5. mes "<html><head><meta charset=\"utf-8\"></head><body>"
6.
7. ; 現在の GPIO 17 の 値 を取得 1 == ON, 0 == OFF
8. ; CGI からは cgpioin を使う (使い方は gpioin と同じ)
9. prev_led = cgpioin(17)
10.
11. if prev_led = 0{
12. ; ついていなかった
13. mes "<p>オフでした。つけます</p>"
14. next_led = 1
15. }else{
16. ;
17. mes "<p>オンでした。けします</p>"
18. next_led = 0
19. }
20. mes "</body></html>"
21. ; CGI からは cgpio を使う (使い方は gpio と同じ、0 以外を書いた場合プログラムが
22. ; 終了しても LED は点灯し続ける)
23. cgpio 17, next_led
24. end
```

第3回に使用した命令ではない命令が入っています。

詳細は P45 に載っています。確認してください。

プログラム解説の4行目

mes "Content-type: text/html\n"

は CGI プログラムがウェブサーバに結果を返すときに、結果が HTML であることを伝えるために必要です。 \n は改行を意味します。本来は改行は 2 つ必要ですが、mes 命令は自動で改行を入れてくれるので、 \n は 1 つで十分です。

4行目以降の mes 命令で出力するものは、HTML の一部として扱われます。

6行目では、文字コードを設定しています。

9行目から LED 点灯消灯のプログラムが始まります。

```
prev_led = cgpioin(17)
```

では、GPIO17 番の状態を調べています。LED が点灯していれば 1 になり、消灯していれば 0 になります。

10～17 行目では、**条件判断**の if 文を使って読み取った LED の状態を判断します。消灯状態であれば、9 行目で

```
mes "<p>オフでした。つけます</p>"
p タグを使ってメッセージを表示します。
```

12 行目で

```
next_led = 1
```

として LED を点灯させるための値を用意します。

消灯以外の状態（つまり点灯状態）の場合は、13 行目で

```
mes "<p>オンでした。けします</p>"
p タグを使ってメッセージを表示します。
```

18 行目で

```
next_led = 0
```

として LED を消灯させるための値を用意します

20 行目で実際に値を GPIO17 へ書き込みます。

```
cgpio 17, next_led
```

21 行目の

```
end
```

でプログラムの終了をします。

第7回の CGI を使うプログラムでは、第3回の授業とはちがう命令を使います。

第3回の命令で CGI を使おうとすると、上手く動かなくなってしまうからです。

下の表にそれぞれの命令をまとめておきます。

第3回に使った 命令名	CGI を使うときの命令名	効果
#include “ rpz-gpio.as ”	#include “ rpz-gpio-cl.as ”	プログラムのはじめに書くことで LED の点灯や消灯の命令が使えるようになります。
onoff = gpioin(17)	onoff = cgpioin(17)	17 番の GPIO の LED が点灯していれば変数 onoff に 1 が、消灯していれば変数 onoff に 0 が入ります。17 をほかの番号に変えることで、ほかの LED の状態を確認できます。
gpio 18, 1	cgpioin 18, 1	18 番の GPIO の LED を点灯します。1 を 0 にすると消灯になります。18 をほかの番号に変えることで、ほかの LED を点灯、消灯できます。

### 問題 7-10

友達の LED を CGI で点灯させてみよう。

点灯、消灯時のウェブページに表示されるメッセージを へんこう 変更してみよう。

他の色の LED も点灯、消灯できるようにしよう。

Fabo のセンサ (LED, しんどうし 振動子などのデジタルセンサ) を接続して CGI から動かせるようにしてみよう。

## 例題 7-8 URLを使って CGIに情報を渡す

### 考え方

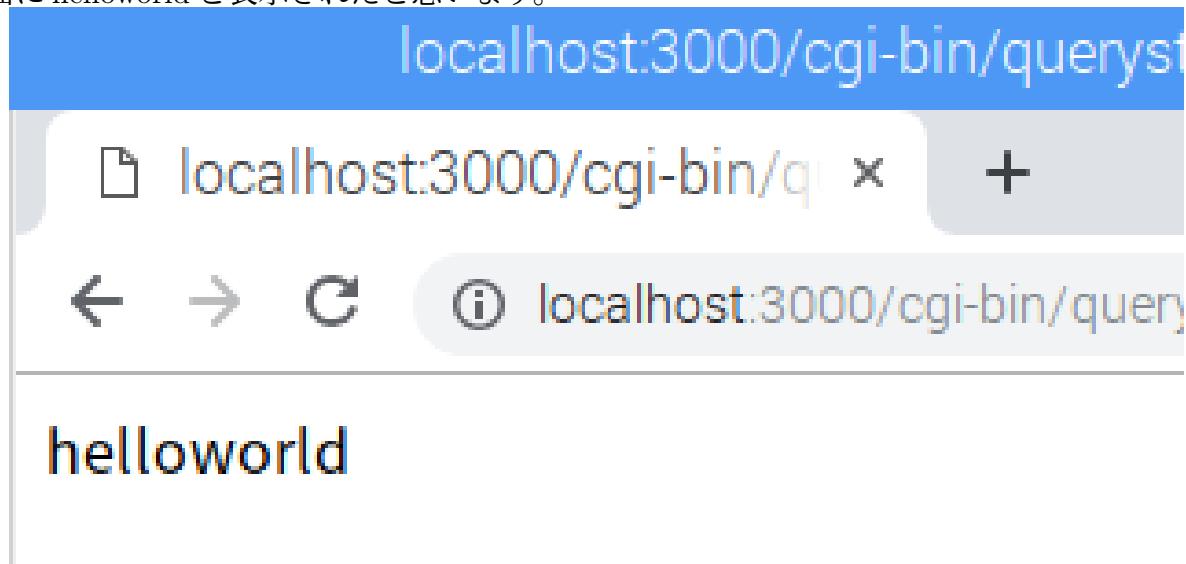
CGI のプログラムに情報を渡すことができればより便利なプログラムが作れます。ここでは単純な URL を用いた方法を使います。URL に埋め込まれた情報はクエリストリングと呼ばれます。クエリストリングはまずウェブサーバへ送られ、ウェブサーバがCGI プログラムへクエリストリングを渡します。これにより CGI プログラムは URL に埋め込まれた情報を扱うことができます。

まずは、ブラウザを開いて、

localhost:3000/cgi-bin/querystring.hsp?msg=helloworld

を入力してみましょう。

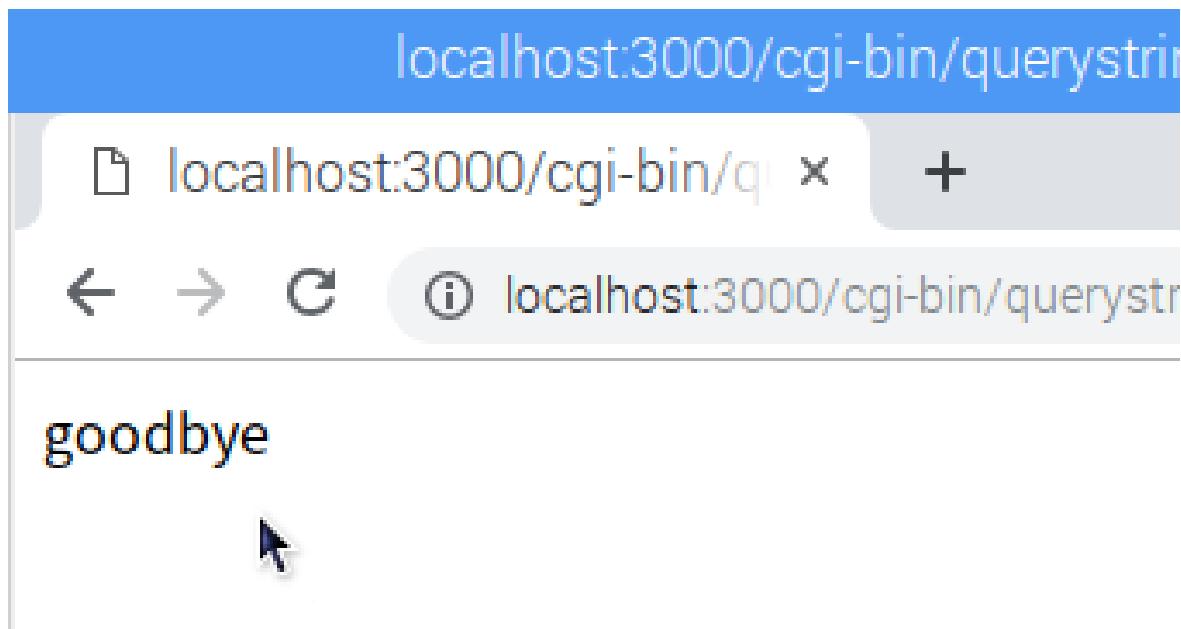
画面に helloworld と表示されたと思います。



localhost:3000/cgi-bin/querystring.hsp?msg=goodbye

に変更してみてください。

次は goodbye と表示されたと思います。



このように URL に情報を埋め込んでおくことで CGI のプログラムはこれを受け取り処理することができます。URL に埋め込まれた情報はクエリストリングと呼ばれます。この情報はまずウェブサーバが受け取ります。ウェブサーバは少し処理をして CGI プログラムへ渡します。クエリストリングは URL の最後に?をつけて始めます。

クエリストリングの形は

名前=値

のようになっています。この例では

msg=helloworld

msg=goodbye

などとなっています。これは msg という名前の値は helloworld, goodbye であることを意味します。HSP の変数と同じような感じです。

次に CGI プログラムでどのようにクエリストリングを扱っているのか見てみましょう。

### プログラム解説

```
1. #include "hsp3cl.as"
2. #include "cgi.as"
3. mes "Content-type: text/html\n"
4. mes "<html><head><meta
5. charset=\"utf-8\"></head><body>"
6. getqueryval "msg", q
7. mes q
8. mes "</body></html>"
9. end
```

6行目の

getqueryval “ msg ”, q

はクエリストリング内の名前が ” msg ” に対応する値を変数 q に入っています。

localhost:3000/cgi-bin/querystring.hsp?msg=helloworld

という URL があった場合、

変数 q には helloworld が文字列として入ります。

7行目で表示をしています。

このように簡単にプログラムでクエリストリングを受け取ることができます。クエリストリングが使えると、<form></form>を使って CGI に処理を依頼できたり、CGI の処理をより高度にすることができます。次の例では、クエリストリングを使って LED を点灯消灯します。ほとんどのウェブサイトは検索をするさいに URL に検索ワードを埋め込んでいます。

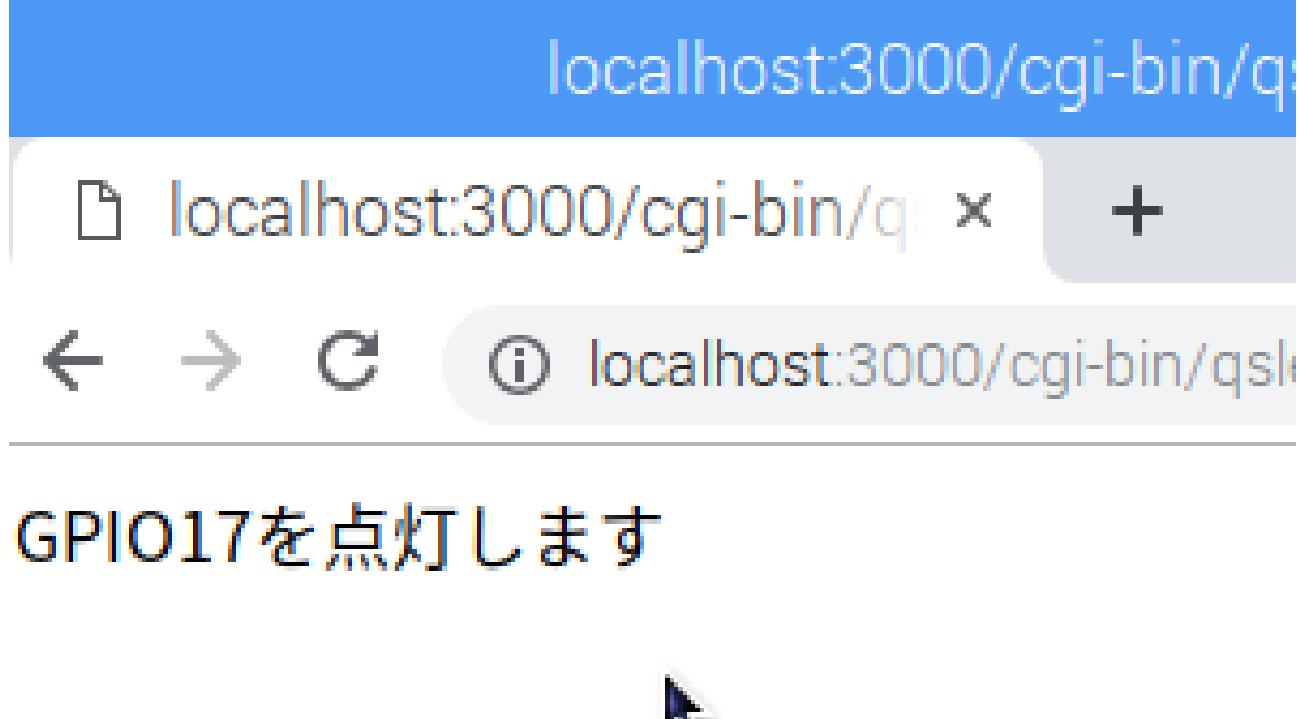
**問題 7-11** 画面に”ありがとう”と表示させよう。

## 例題 7-9 クエリストリングを使って LED を操作する

この例では、クエリストリングを使って LED を点灯消灯させてみましょう。

まずは、ブラウザを開いて、

localhost:3000/cgi-bin/qsled.hsp?led=17&val=1  
を入力してみましょう。



実行すると LED17 が光ります。val=1 を val=0 にすると消えます。URL に埋め込まれた情報をもとに CGI プログラムは LED を操作していることがわかります。URL に埋め込まれた情報はクエリストリングと呼ばれます。

クエリストリングを詳しく見てみましょう。

クエリストリングとは URL の最後に?をつけて始めます。?のあとには

名前(変数名みたいなもの) = 値

の形を続けます。例えば、

localhost:3000/cgi-bin/qsled.hsp?led=17

のようにかけます。この場合は、

名前が led、値が 17 となります。

複数の情報を渡したい場合は&を使います。

localhost:3000/cgi-bin/qsled.hsp?led=17&val=1

のようになります。

名前1が led、値1が 17 となり

名前2が val、値2が 1 となります。

このクエリストリングをプログラムから扱う方法を次に学びましょう。

## プログラム解説

```
1. #include "hsp3cl.as"
2. #include "rpz-gpio-cl.as"
3. #include "cgi.as"
4. mes "Content-type: text/html\n"
5. mes "<html><head><meta
6. charset=\"utf-8\"></head><body>"
7. ; クエリストリングから led を探して、値を led_port へ入れる
8. getqueryval "led", led_port
9. ; クエリストリングから val を探して、値を led_val へ入れる
10. getqueryval "val", led_val
11. ; 文字列なので数値へ変換する
12. led_port = int(led_port)
13. led_val = int(led_val)
14.
15. ; 点灯消灯を判断して、メッセージを p タグで表示する
16. if(led_val = 1) {
17. mes "<p>GPIO" + led_port +
18. "を点灯します</p>"
19. } else {
20. mes "<p>GPIO" + led_port +
21. "を消灯します</p>"
22. }
23.
24. ; クエリストリングの led で指定された GPIO に val で指定された値を書き込む
25. ; localhost:3000/cgi-bin/qsled.hsp?led=17&val=1
26. ; の場合、LED17 を点灯させる
27. cgpio led_port, led_val
28. mes "</body></html>"
29. end
```

8行目、10行目で使用している

getqueryval 命令でクエリストリングを受け取ります。この命令は

getqueryval 名前, 値を入れる変数

のように使います。クエリストリングから名前を探して、変数へ値を文字列としていれます。

例えば、

getqueryval led, led\_port

の場合に localhost:3000/cgi-bin/qsled.hsp?led=17

とクエリストリングを与えたとすると、

led\_port には 17 が入ります。

12,13 行目で

文字列から数値へ変換しています。

16～20 行目では、点灯するのか消灯するのかを条件判断を使って判断しています。led\_val が 1 のとき (val=1 がクエリストリングで与えられたとき) に点灯する、その他のときは消灯するとなります。

25 行目で実際に点灯、消灯を行います。

cgpio 命令は数値で GPIO 番号、オンオフ (1, 0) を受け付けるので、文字列から変換をしました。

### 問題 7-12

GPIO17 以外の LED を光らせてみよう。

## 例題 7-10 CGIを使ってOpenJtalkに読み上げをさせよう

考え方

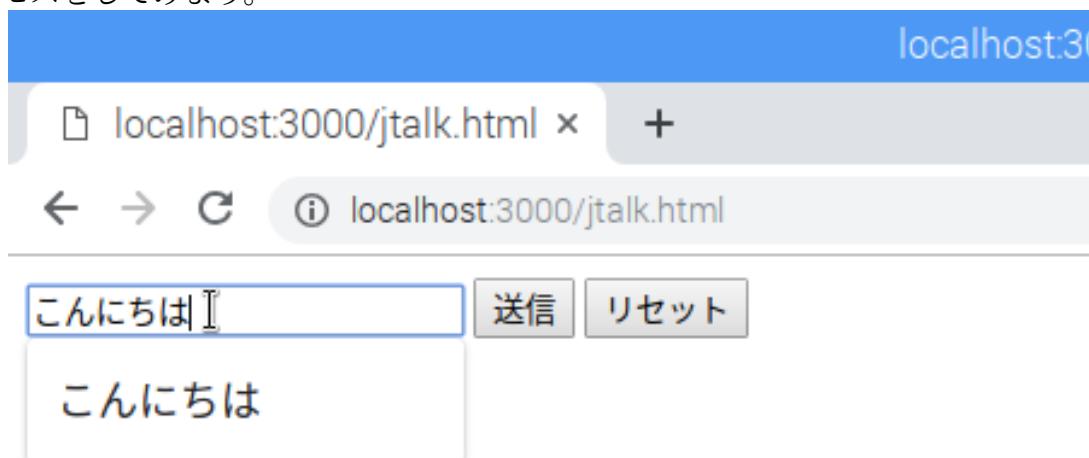
ウェブページから読み上げさせたい文章を受け取って、OpenJtalkにCGIから読み上げをさせよう。

まずは、試してみよう。

ブラウザを開いて

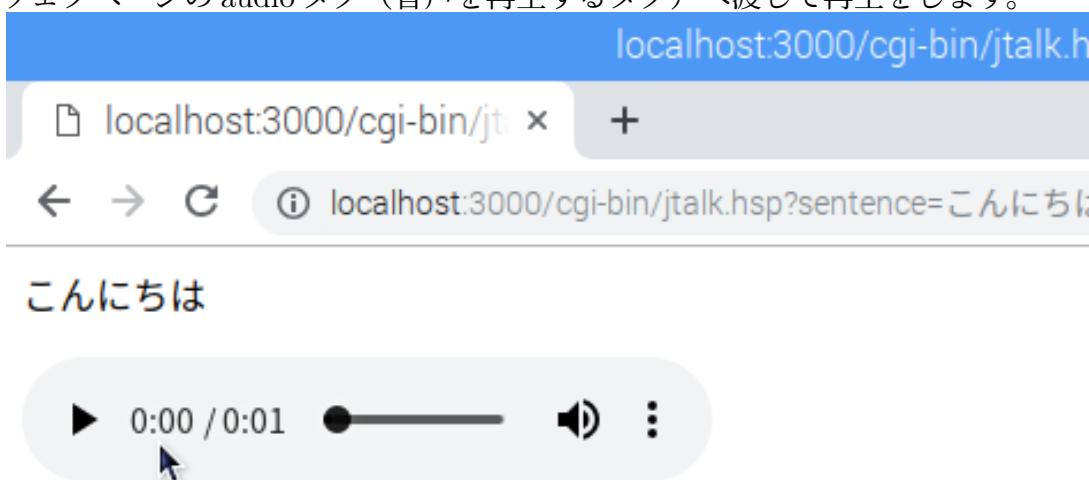
localhost:3000/jtalk.html

にアクセスをしてみよう。



テキストボックスに読み上げさせたい文章を入れて送信をしてみよう。送信を押すとCGIプログラムへ文章が送られ、OpenJtalkで音声合成をします。

結果をウェブページのaudioタグ（音声を再生するタグ）へ渡して再生をします。



## プログラム解説

~/07/www/jtalk.html

```
1. <!DOCTYPE html>
2. <html>
3. <head>
4. <meta charset="utf-8">
5. </head>
6. <body>
7. <form action="cgi-bin/jtalk.hsp" method="GET">
8. <input type="text" name="sentence">
9. <input type="submit" value="送信">
10. <input type="reset" value="リセット">
11. </form>
12. </body>
13. </html>
```

7～11行目では、CGI ようにフォーム (form) を用意しています。

文字を入れるテキストボックスが一つと送信用のボタンが一つ、テキストボックスをリセットするボタンが一つあります。

7行目の

<form action="cgi-bin/jtalk.hsp" method="GET">  
action= " cgi-bin/jtalk.hsp "

はフォームに入力された情報を受ける(処理する)CGI プログラムを指定します。

method= " GET "

はクエリストリングとしてフォームの情報を CGI に渡すことを指示しています。

8行目の

<input type="text" name="sentence">

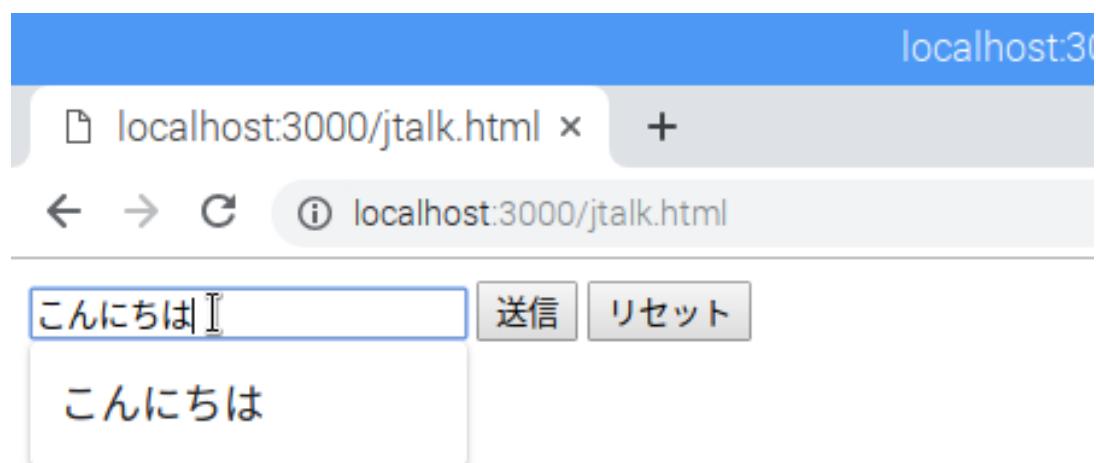
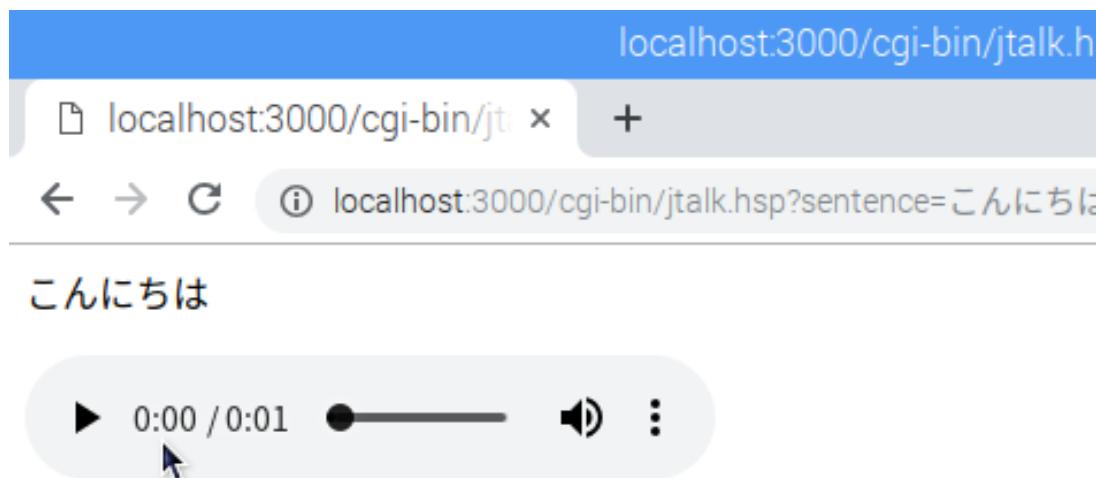
では、テキストボックスを作っています。

name= " sentence "

はクエリストリングの名前を指定します。入力された文字列が値となります。ちなみに sentence とは日本語にすると”文章”という意味になります。

例えば、テキストボックスに

“こんにちは”とテキストボックスに入力され、送信された場合



ウェブサーバにこのような要求を出します。

IP アドレス:3000/cgi-bin/jtalk.hsp?sentence=%E3%81%82%E3%81%97%E3%81%99

実際には”こんにちは”という文字列は URL 内ではある規則に沿って記号、数値に変換されています。

このことをパーセントエンコーディングといいます。CGI プログラムにクエリストリングが渡される前にウェブサーバは記号、数値に変換された文字列をもとに戻します。よって、プログラム内では入力された文字列と同じものを扱うことができます。

~/07/www/cgi-bin/jtalk.hsp

```

1. #include "hsp3cl.as"
2. #include "jtalk.as"
3. #include "cgi.as"
4.
5. mes "Content-type: text/html\n"
6. mes "<html><head><meta charset=\"utf-8\"></head><body>"
7.
8. getqueryval "sentence", sentence
9. jtsave sentence, wav_file
10. mes "<p>" + sentence + "</p>"
11.
12. mes "<audio src=\"" + wav_file + "\" type=\"audio/wav\" controls>"
13. mes "</body></html>"
14. end

```

8行目で jtalk.html のフォームの

<input type="text" name="sentence">

に入っている値を取り出しています。取り出した値は sentence 変数へ入れています。

9行目の

jtsave sentence, wav\_file

は sentence 変数の中の文字列を音声合成して音声ファイルのファイル名を wav\_file 変数へ入れています。

12行目の

mes "<audio src=\"" + wav\_file + "\" type=\"audio/wav\" controls>"

<audio>タグは音声ファイルを再生することができます。

src= " ファイル名 "

では、再生するファイルを指定します。指定するファイルはドキュメントルート以下にないといけません。つまり、ドキュメントルートである~/07/wwwの中にある必要があります。

jtsave 命令で作った音声ファイルは~/07/www/tmpの中にあります。ファイル名は wav\_file 変数に入っています。

““ 中で ” を使うには\” のように書きます。

type= " audio/wav "

では、音声ファイルの種類を決めます。jtsave 命令が作る音声ファイルの場合は、

type= " audio/wav " にします。

controls をつけると、再生ボタン、ミュートボタンなどが表示されます。

再生ボタンを押すと音声合成した音声ファイルが再生されます。

### 問題 7-13

”おはようございます””こんばんは””ありがとうございます”を読ませてみよう。

できる人は自分の自己紹介ページに (`~/07/www/index.html`) に作ったフォームを変更して、こちらから CGI プログラムを起動するようにしましょう。

HINT: jtalk.html を参考にしてください。

## 例題 7-11 CGI でセンサーの情報を表示させる

考え方

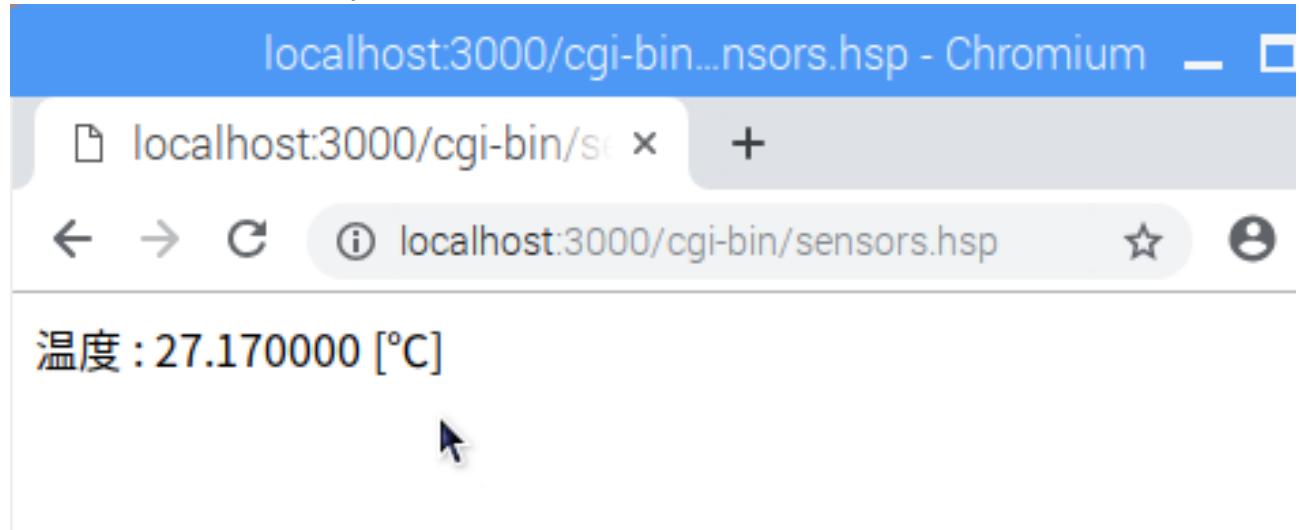
温度、湿度、<sup>きあつ</sup>気圧、照度センサーの情報を CGI を使ってウェブページから見れるようにしましょう。

まずは、プログラム実行してみましょう。

ブラウザを開いて、

localhost:3000/cgi-bin/sensors.hsp

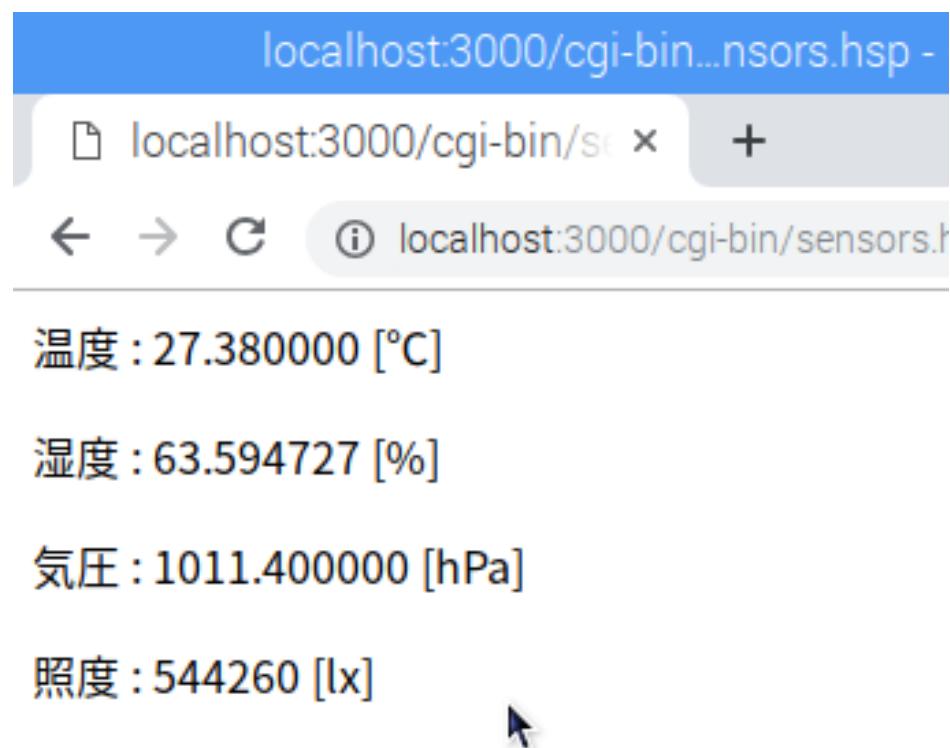
にアクセスしてください。



温度センサーの値が表示されていると思います。

リロードするたびに値が更新されます。

今は温度センサーしか表示をしていないので、他のセンサーの値も表示させるように変更してみましょう。



次にプログラムを見てみましょう。

## プログラム解説

```
1. #include "rpz-gpio-cl.as"
2. #include "hsp3cl.as"
3.
4. mes "Content-type: text/html\n"
5. mes "<html><head><meta charset=\"utf-8\"></head><body>"
6.
7. i2c_ch_bme = 0
8. i2c_ch tsl = 1
9.
10. fail = init_bme(i2c_ch_bme) ; 温湿度気圧センサ bm280 を初期化する
11. if fail { ; 初期化成功チェック
12. mes "failed to init bme: " + fail
13. end
14. }
15.
16. init_lux i2c_ch_tsl ; 照度センサ
17. tsl2572 を初期化する
18.
19. *main
20. temp = get_temp(i2c_ch_bme) ; 温度取得
21. hum = get_hum(i2c_ch_bme) ; 湿度取得
22. press= get_press(i2c_ch_bme) ; 気圧取得
23. lux = get_lux(i2c_ch_bme) ; 照度取得
24.
25.
26. ; 取得したデータの表示
27. mes "<p>温度：" + temp + " [°C]</p>"
28.
29. mes "</body></html>"
30. end
```

この例題は~/03/sensors.hsp をもとにしています。詳しい解説はそちらを参照してください。

20~23行目で温度、湿度、気圧、照度センサーから情報を取得しています。

結果はそれぞれ temp, hum, press, lux 変数へ入っています。

27行目で、温度センサーの値を p タグで表示しています。

#### 問題 7-14

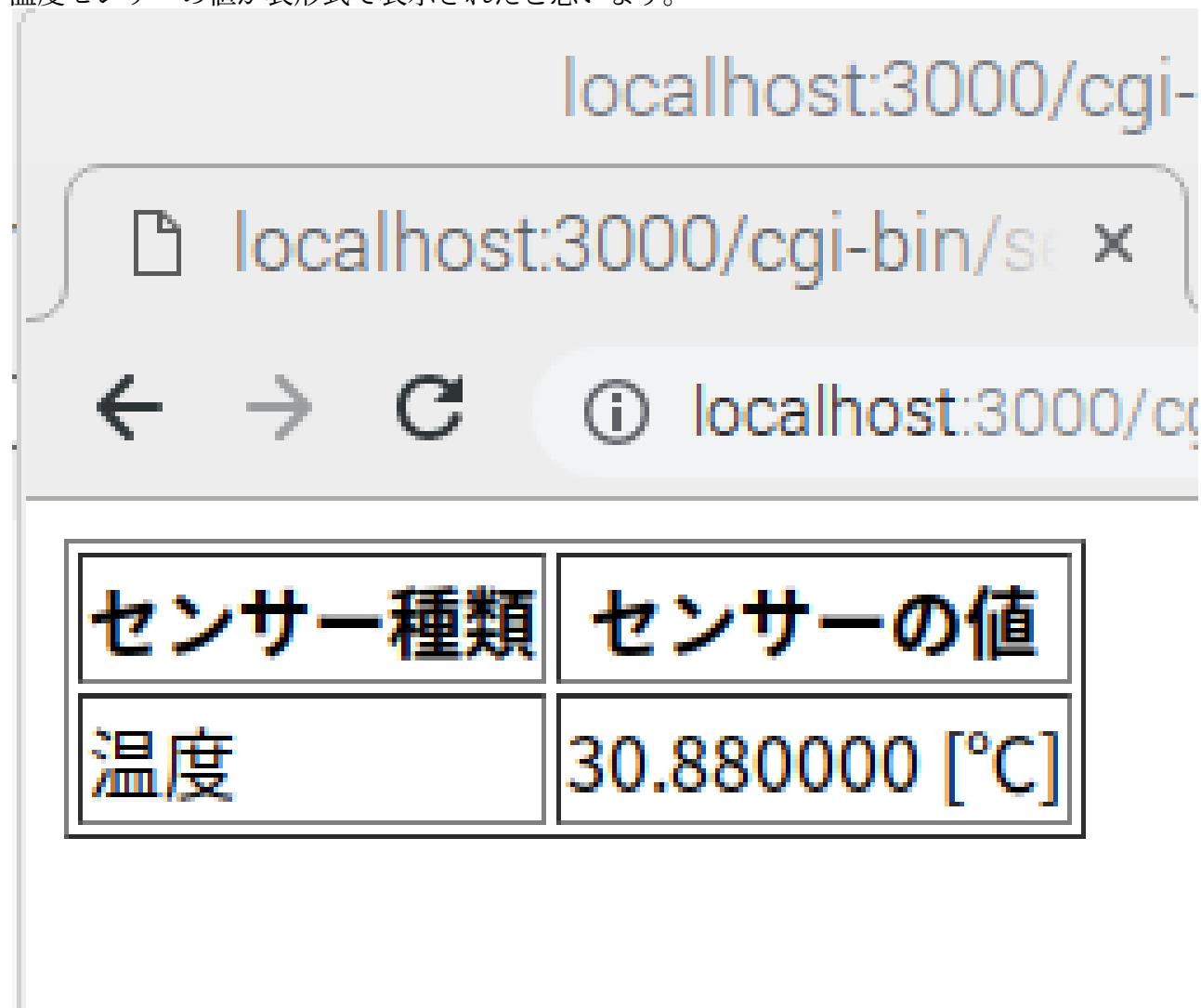
友達の CGI ヘアクセスして、センサーの値を確認してみよう。

温度センサーと同様に、湿度、気圧、照度センサーの値も表示させてみましょう。

localhost:3000/cgi-bin/sensors\_table.hsp

にブラウザでアクセスをしてみましょう。

温度センサーの値が表形式で表示されたと思います。



次は、湿度、気圧、照度センサーの値を表形式で表示するように変更してみましょう。

## 例題 7-12 赤外線をウェブページから送信する

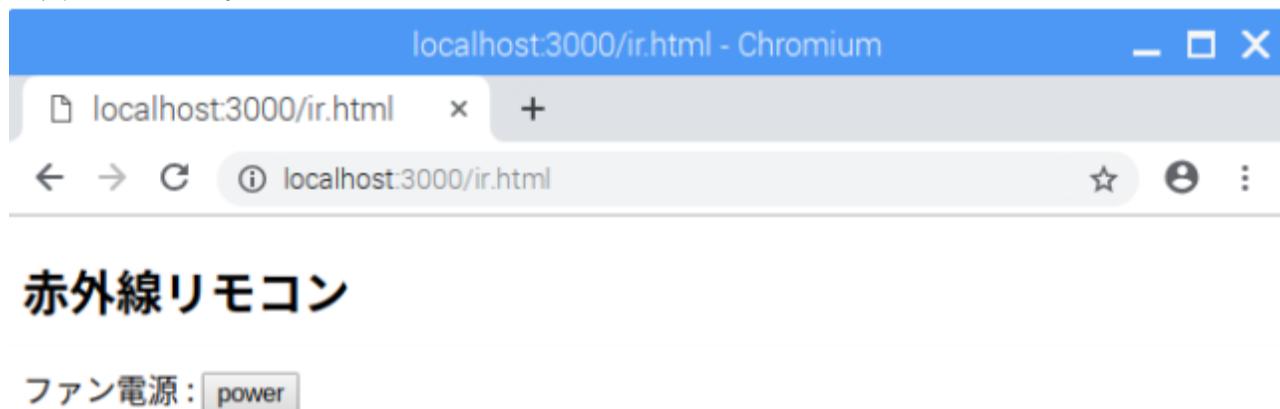
CGIを使ってウェブページから赤外線を送信できるようにしましょう。

まずは、サンプルを動かしてみましょう。

ウェブブラウザを開いて

localhost:3000/ir.html

を開きましょう。



The screenshot shows a Chromium browser window with the URL `localhost:3000/ir.html`. The page title is "赤外線リモコン". Below the title, there is a text input field with the placeholder "ファン電源:" followed by a button labeled "power".

ボタンを押すと CGI で赤外線の送信をするコマンドが実行されます。

コマンドは、

`irsend SEND_ONCE fan onoff`

が実行されます。リモコンの名前は `fan`、信号の名前は `onoff` になっています。

リモコンの名前、信号の名前が違う場合は、

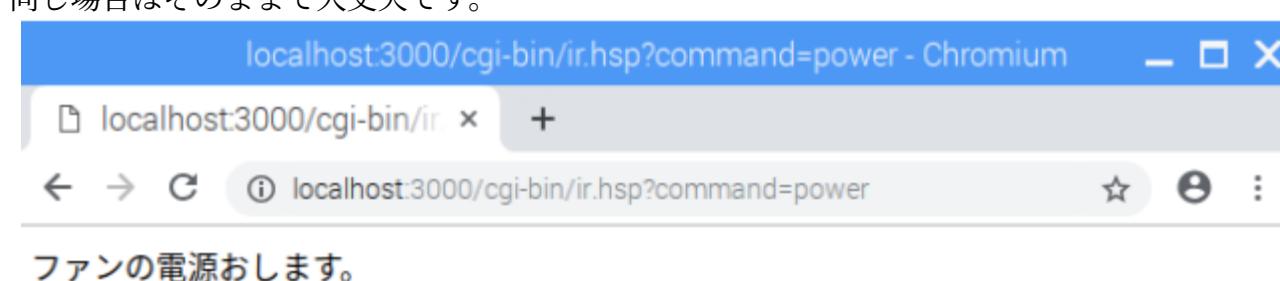
`~/07/www/cgi-bin/ir.hsp`

を開いて

`exec "irsend SEND_ONCE fan onoff"`

を変更して保存しましょう。

同じ場合はそのまま大丈夫です。



The screenshot shows a Chromium browser window with the URL `localhost:3000/cgi-bin/ir.hsp?command=power`. The page displays the message "ファンの電源おします。" (Fan power off).

ブラウザのファン電源ボタンを押すと、CGI が起動して  
赤外線を送信します。

## プログラム解説

~/07/www/ir.html

8行目でフォームの内容を受け取る CGI を設定しています。

```
1. <!DOCTYPE html>
2. <html>
3. <head>
4. <meta charset="utf-8">
5. </head>
6. <body>
7. <h2>赤外線リモコン</h2>
8. <form action="cgi-bin/ir.hsp" method="GET">
9. <p>
10. ファン電源 : <input type="submit" value="power" name="command">
11. </p>
12. </form>
13. </body>
14. </html>
```

~/07/www/cgi-bin/ir.hsp

が CGI として起動します。

10行目で送信ボタンを作っています。

<input type="submit" value="power" name="command">

クエリストリングの名前は command で値は power となります。このボタンが押されると CGI が起動します。

```
1. #include "hsp3cl.as"
2. #include "cmdexec.as"
3. #include "cgi.as"
4.
5. mes "Content-type: text/html\n"
6. mes "<html><head><meta charset=\"utf-8\"></head><body>"
7.
8. getqueryval "command", cmd
9. if cmd = "power" {
10. mes "<p>ファンの電源をおします。</p>"
11. exec "irsend SEND_ONCE fan onoff"
12. } else {
13. mes "<p>コマンドが正しくありません</p>"
14. mes "<p>" + cmd + "</p>"
15. }
16.
17. mes "</body></html>"
18. end
```

8行目でクエリストリングから名前が command に対応する値を取り出して cmd 変数へ入れています。

9～15行目で受け取った cmd 変数をもとに条件判断をしています。cmd が power のときにファンの電源をつける赤外線を送っています。

それ以外の場合は正しいコマンドでないとして、メッセージを表示しています。

### 問題 7-15

ファンの電源をつける以外の赤外線送信機能を追加してみよう。

HINT：まずは、ir.html にボタンを追加しよう。

その後、ir.hsp の条件判断を追加しよう。

ir.html のフォームを自己紹介ページの一番下に付け加えよう。

## CGI を使うときに便利な HSP の命令一覧

命令名/使い方	例題	効果
#include "cgi.as" getqueryval "name", var	例題 7-8 例題 7-9 例題 7-10 例題 7-12	クエリストリングから名前 "name" を探し て文字列として値を変数 var へ入れる。 <b>localhost:3000/cgi-bin/querystring.hsp?name=val</b> の 場合、var には val が入る。
#include "rpz-gpio-cl.as" cgpio 17, 1	例題 7-7 例題 7-9 例題 7-11	gpio 命令と同じ。GPIO17 番に 1 を書き込む。gpio 命令との違いはプログラ ム終了時にも書き込んだ値が持続する。例 えば、cgpio 17, 1 を実行すると、プログラ ム終了時でも 17 番の GPIO は 1 のままに なる。
#include "rpz-gpio-cl.as" onoff = cgpioin(17)	例題 7-7	gpio 命令と同じ。cgpio でプログラム終了 時でも効果を持続させたい場合はこちらを 使用する必要がある。
#include "rpz-gpio-cl.as"	例題 7-7 例題 7-9 例題 7-11	#include "rpz-gpio.as" のコマンドライン 版コマンドラインや CGI で動くプログラ ムを書く場合はこちらを使う。cgpio, cgpioin 命令以外は同じ命令が使える。
#include "jtalk.as" jtsave "こんにちは", hello	例題 7-10	"こんにちは" を OpenJtalk で音声合成を して、作った音声ファイルのファイル名を hello 変数へ入れる。音声ファイルは/tmp/ ディレクトリの下にランダムなファイル名 で入る。 例えば、hello 変数の中身は "/tmp/tmp.abzgda" のような感じになる。
end	すべての例題	プログラムの終了を意味する。この命令は CGI 用のプログラムを終了するときに必要。 この命令がないと CGI のプログラムは終了 しない。(ブラウザの読み込みが終わらなく なる。)

## 付録 ウェブサーバの停止方法

電源を切る前、ウェブサーバが起動しているターミナルを閉じる前、ウェブサーバを停止したいときは、ウェブサーバを起動したターミナル (`./webserver.py` を実行している画面) を選択して

Ctrl と C を同時に押します。

それでウェブサーバは終了します。