

HTML での通信の実際 (telnet で Web サーバと通信)

1 準備

※ telnet のインストール

telnet がインストールされていない場合は、「スタート」右クリック - 「アプリと機能」 - 「プログラムと機能」 - 「Windows の機能の有効化と無効化」をクリックして、画面に出てくるダイアログボックスから「telnet クライアント」にチェックを入れてください。しばらくすると、telnet のインストールが始まります。そして、すぐに終わります。

まず、sample.html と sample.php をつくります。内容は、以下です。

sample.html

```
<!DOCTYPE html>
<html lang="ja">
  <head>
    <meta charset="utf-8">
    <title>sample</title>
  </head>
  <body>
    <h1>sample</h1>
    <form action="sample.php" method="post">
      <input type="text" name="name">
      <input type="submit">
    </form>
  </body>
</html>
```

sample.php

```
<?php $name = $_POST['name']; ?>
<!DOCTYPE html>
<html lang="ja">
  <head>
    <meta charset="utf-8">
    <title>sample</title>
  </head>
  <body>
    <h1>sample</h1>
    <p><?php print $name; ?></p>
    <p><a href="sample.html">もどる</a></p>
  </body>
</html>
```

2 GET での通信

まず、コマンドプロンプトで、

```
> telnet localhost 80 <Enter>
```

とします。(<Enter> は、エンターキーを押下すること)

そのあと、何やら一瞬文字が表示されたのち画面が消えて、新しい画面になります。

そこで、

```
Ctrl キー + ]
```

とします。画面が切り替わるので、

```
telnet > set localecho <Enter>
```

とします。

```
ローカルエコー オン
```

```
telnet >
```

と表示されるので、そのまま <Enter> キーを押します。

すると、また、何もない画面になるので、以下のコマンドを入力します。

※Windows 版 telnet は、BS キーが使えません。入力間違いを訂正することができないのです。したがって、まずメモ帳で入力すべきコマンドを記述してから、それをコピーし、コマンドプロンプトの左上アイコンをクリックして現れるメニューの「編集」-「貼り付け」で、ペーストします。

```
GET /php/sample.html HTTP/1.1
```

```
Host: localhost
```

(Host: の下の行は、<Enter>キーを入力するだけです。)

すると、以下のような表示が現れるはずです。

```
HTTP/1.1 200 OK
Date: Sun, 11 Feb 2018 03:55:30 GMT
Server: Apache/2.4.18 (Ubuntu)
Vary: Accept-Encoding
Content-Length: 516
Content-Type: text/html; charset=UTF-8

<!DOCTYPE html>
<html lang="ja">
  <head>
    <meta charset="utf-8">
    <title>sample</title>
  </head>
  <body>
    <form action="sample.php" method="post">
      <input type="text" name="name">
      <input type="submit">
    </form>
  </body>
</html>
```

さきほどの

```
GET /php/sample.html HTTP/1.1
```

```
Host: localhost
```

を「リクエスト範ヘッダ」といいます。

そのあとの「空行」は、「ヘッダ部」の終了を相手に伝えてます。そのあとに「ボディ部」がくることもあるのですが、GET の場合は通常、ボディ部はありません。

「空行」によって、ホストは「ヘッダ部」の終了を検知したので、返答を開始します。それが以下の部分です。

```
HTTP/1.1 200 OK
Date: Sun, 11 Feb 2018 03:55:30 GMT
Server: Apache/2.4.18 (Ubuntu)
Vary: Accept-Encoding
Content-Length: 516
Content-Type: text/html; charset=UTF-8

<!DOCTYPE html>
<html lang="ja">
( 以下、略 )
```

上の6行が「レスポンス範ヘッダ」で、「空行」で区切られたあと、「ボディ部」が現れます。

3 POST での通信

POST の場合は、以下のようにリクエストを送ります。

telnet コマンドでの入力方法は同じです。以下は、GET コマンドを入力するところで POST コマンドを入力しています。

```
POST /php/sample.php HTTP/1.1
```

```
Host: localhost
```

```
Content-Type: application/x-www-form-urlencoded
```

```
Content-Length: 10
```

```
( 空行 )
```

```
name=maido
```

POST の場合は、Content-Type の指定と Content-Length の指定が必要です。今回は、POST データで「maido」という文字列を送っています。「name=maido」で10バイトになります。

POST でデータを送る場合は、ボディ部に埋め込んで送られるというのがよくわかります。また、<input>で複数のデータを送る場合は、以下のように「&」で連結して送ります。

```
name=maido&age=23&gender=man
```

さて、10 バイト送られたので、ホストは、以下のレスポンスを送ってきます。

```
HTTP/1.1 200 OK
Date: Sun, 11 Feb 2018 04:07:09 GMT
Server: Apache/2.4.18 (Ubuntu)
Vary: Accept-Encoding
Content-Length: 521
Content-Type: text/html; charset=UTF-8

<!DOCTYPE html>
<html lang="ja">
  <head>
    <meta charset="utf-8">
    <title>sample</title>
  </head>
  <body>
    <h1>sample</h1>
    <p>maido</p>
    <p><a href="sample.html">もどる</a></p>
  </body>
</html>
```

今回も、レスポンスは、「ヘッダ部」と「ボディ部」に分かれていて、「空行」で区切られているのがわかります。

このような、クライアントとサーバーの HTML データのやり取りは、Chrome や Firefox のデベロッパーツール「**network**」で見ることができます。
(この操作の方法は、今回は省略します)

追記

curl コマンドを使うと、telnet でおこなったことを簡単に見れます。

```
> curl -v localhost  範範範 リクエストとレスポンスの両方を見る
> curl -I localhost  範範範 レスポンスのヘッダーだけ見る。(-I はアイの大文字)
> curl -i localhost  範範範 レスポンスのヘッダーとボディを見る
> curl -v localhost -X POST -d 'name=Taro'
localhost に POST で name=Taro というデータを送る。
```

curl コマンドは、Windows 10 Ver.1803 (RS4) のプレビュー版 (Build 17063) にはいったということなので (<http://ascii.jp/elem/000/001/630/1630041/>)、次期大幅アップデートのときに入るのでしょう。今のところは、Windows Subsystem for Linux で使えます。Mac のターミナルでも使えるはずです。