

基礎プログラミングII

第7回 CGIプログラム (1)

CGI (Common Gateway Interface) とは

Webサーバーが外部プログラムを呼び出して動きのあるページを作るための仕組み

CGIの仕組み

CGIのデータ入力窓の集合を**入力フォーム**という。

名前↓

さんの誕生日は?

1997 年 月 日

OK

reset

CGIの仕組み

前ページの入力窓に入れた値は全て**入力名**という特別な変数**CGI**スクリプトに送られる。

CGIの仕組み

自分で作るプログラムではその変数を受け取りそれに応じた処理をした上で、結果を表すHTML文書を出力する。

CGIの仕組み

[taro]さんの誕生日は？
[1992]年[3]月[1]日
[OK][reset]

「Rubyスクリ
プト」値を受
け取りHTML
を出力

```
<html>
<head>
<title>結果発表!</title>
</head>
<body>
<p>.....</p>
</body>
</html>
```

CGIの作り方

CGIでは、利用者にデータを入力してもらうための**HTML文書**とそこから送られて来るデータを受け取って処理をする**プログラムファイル**を作る。

CGIの作り方

Webサーバに対して「ここでCGIプログラムを利用する」ことの宣言を行う必要がある。

CGIの作り方

1. CGI利用宣言→.htaccessファイルの作成
2. 入力フォームのHTML文書作成
3. 処理プログラムの作成

CGIプログラムの利用宣言

「この場所でCGIを利用する」ためのディレクトリを作成し、そこに.htaccessという名前のファイルを作成する。

CGIプログラムの利用宣言

AddHandler cgi-script .rb

Options +ExecCGI

AddType "text/html; charset=utf-8" .html

1, 2行目：このディレクトリに作成する.rbという名前で終わるファイルはCGIスクリプトと認識される。

CGIプログラムの利用宣言

```
AddHandler cgi-script      .rb  
Options +ExecCGI  
AddType "text/html; charset=utf-8".html
```

3行目：HTMLファイルの文字コードがUTF-8であることを宣言する。

HTML文書の構成

入力フォームを構成するHTML文書を作る。入力フォームはform要素を使って記述する。

```
<form method="メソッド" action="/スクリプト">  
~~~  
</form>
```

メソッドの部分はGETまたはPOSTのどちらかを指定する。文章など、長いデータを入力させたいときはPOSTを指定する。

CGIスクリプトの構成

CGIスクリプトとなるRubyプログラムでは、最低限以下の内容を記述して作成する。

```
#!/usr/bin/env ruby
```

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

```
require 'cgi'
```

```
c = CGI.new(:accept_charset => "UTF-8")
```

```
print "Content-type: text/html; charset=UTF-8\n\n"
```

変数cはどんな名前でも良い。このあとにHTML文書となり得る文字列を順次出力していく。

CGIスクリプトの構成

requireは、プログラムに必要な外部モジュールを現在のプログラムに取り込む記法で、

require 'cgi'

はCGIプログラムを組むのに便利なクラスが定義された**cgi.rb**を取り込む宣言である。これでCGIクラスが使えるようになる。

CGIスクリプトの構成

CGIプログラムに渡された入力値を取り出すには、`CGI.new`を受け取った変数のハッシュ値として取り出す。

ヒアドキュメント(<<)とは

<<の次の行から、<<の直後に書かれた単語が見つかる行までをまとめて1つの文字列として括る記号のこと。

ヒアドキュメント

改行文字を含む長文文字列を扱いたいときに便利な記法。文字列中に“”や“”を自由に入れることもできるため、“”を多様することの多いHTML文の出力に有用

#{...}

その部分をRubyの式としてその値に置き換える働きを持つ。

#{...}

例えば、

```
print “1+2は#{1+2}です\n”
```

とすると、

```
1+2は3です
```

と出力される。HTML出力文に変数結果を入
れたい場合に有用。

CGIプログラムのデバッグ手順

1. ターミナルで直接動かしてエラーをなくす
2. ローカルPC（目の前で使っているPC）で動かして詳細デバッグ
3. 本番サーバで動かして最終調整

フォームで使える入力用要素

input要素

なんらかの値を入力させる場合に、**input要素**に入力方法と入力名を指定する。

入力方法は**type属性**で、入力名は**name属性**で指定する。

input要素 : text

1行だけの文字列入力

<p><label>メッセージをどうぞ :

<input name="v_text" type="text" value="Hello" size="40">

</label></p>

メッセージをどうぞ: Hello

[送信]

入力名v_textに初期値“Hello”を入れて40字分の枠を作る。初期値を省略すると空欄になる。

input要素 : passwd

textと同様だが、入力文字をアスタリスクで隠す。

<p><label>パスワード :

```
<input name="v_passwd" type="passwd" size="20"
maxlength="12">
```

</label></p>

パスワード:

[送信]

入力名v_passwdに最大文字数12字のパスワードをいれる。ただし、パスワードの値がCGIプログラムに渡される時は**暗号化されない**。

input要素：checkbox

ONのときの値とOFFのときの値をvalue属性で指定する。

```
<p><label><input name="v_checkbox_1" type="checkbox" value="morning">朝御飯OK</label><br></p>
```

```
<p> <input name="v_checkbox_2" type="checkbox" value="evening">夕飯OK </p>
```

「朝御飯OK」がチェックされていたら入力名v_checkbox_1を“morning”に、「夕飯OK」がチェックされていたら入力名v_checkbox_2に“evening”を代入させる。

☐ 朝御飯OK

☐ 夕飯OK

[送信]

input要素 : checkbox

```
<p><label><input name="v_checkbox_1" type="checkbox" value="morning">朝御飯OK</label><br></p>
```

```
<p> <input name="v_checkbox_2" type="checkbox" value="evening">夕飯OK </p>
```

- 「朝御飯OK」はlabel要素で結びついているので、「朝御飯OK」をクリックするとチェックが入る。
- 「夕飯OK」はボタンをクリックしないとチェックが入らない。

input要素 : radio

ラジオボタンを作成する。name属性で同じ入力名を持つもの同士がグループ化され、どれか一つをチェックすると他はクリアされる。

```
<p>  
<label><input name="v_radio" type="radio" value="1">地球人</label>  
<label><input name="v_radio" type="radio" value="2">宇宙人</label>  
<label><input name="v_radio" type="radio" value="3">仙人</label>  
</p>
```

☐ 地球人 ☐ 宇宙人 ☐ 仙人

例えば「仙人」を選ぶと入力名v_radioに“3”が代入される。どれも選択されていない場合は空文字列が代入される。

input要素 : submit, reset

入力値の送信またはリセットボタンを作成する。

```
<p>  
<label><input type="submit" value="[送信]">  
<label><input type="reset" value="[リセット]">  
</p>
```

[送信]

[リセット]

input要素 : hidden

隠れ変数を設定する。CGIプログラムに必ず渡したい値を見えないところで指定する。

```
<input name="v_hidden" type="hidden" value="第3ステージ">
```

上記を書いてもWebページにな何も表示されないがCGIプログラムには入力名v_hiddenに「第3ステージ」という値が代入される。

hidden変数の活用

1つのCGIプログラムに複数のWebページ出力を管理させるには、現在の状態を示す変数を**hidden変数**に埋め込んで、次に起動されるCGIプログラムでその値を参照して出力するHTMLを切り替えればよい。

select要素

選択肢を提示してそのうち1つまたは複数を選ばせる。

好きな季節

```
<select name="v_select">  
  <option>春</option>  
  <option value="Summer" selected>夏</option>  
  <option value="Autumn">秋</option>  
  <option value="Winter">冬</option>  
</select>
```

好きな季節

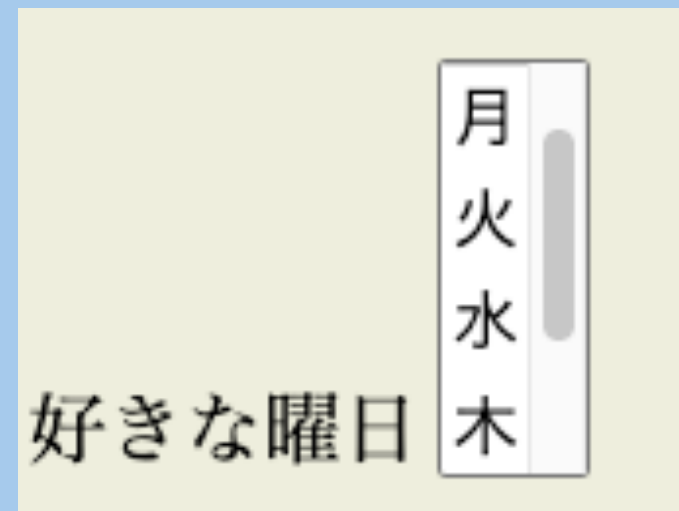
- `value`指定のない「春」を選択すると「春」そのものが値となる。
- `selected`を付けておいたオプションが選択初期値となる。
- `selected`を付けなければ先頭項目が選択された状態で始まる。

select要素

multiple属性を付けると複数選択が可能になる。Firefoxで複数選択するにはCtrlクリックで行う。

好きな曜日

```
<select multiple name="v_select_multi">  
  <option value="Sunday" selected>日</option>  
  <option value="Monday">月</option>  
  <option value="Tuesday">火</option>  
  <option value="Wednesday">水</option>  
  <option value="Thursday">木</option>  
  <option value="Friday">金</option>  
  <option value="Saturday" selected>土</option>  
</select>
```



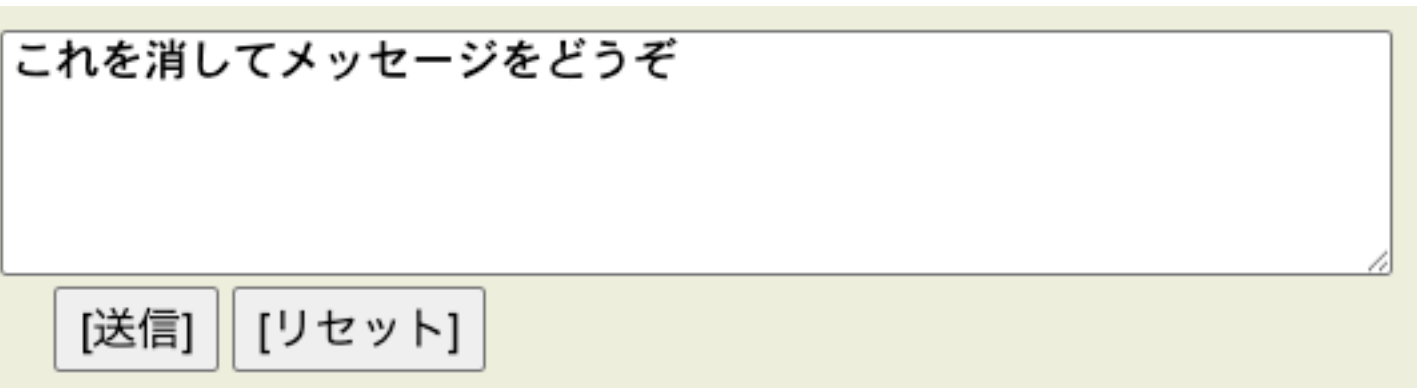
テキストエリア

長文を入力させるためにはtextareaを用いる。入力エリアの行と桁数はそれぞれrows、cols属性で指定する。

```
<textarea name="v_textarea" cols="60" rows="5">
```

これを消してメッセージをどうぞ

```
</textarea>
```



A screenshot of a web form. It features a large text area with a light gray border and a light gray background. Inside the text area, the text "これを消してメッセージをどうぞ" is displayed in a dark gray font. Below the text area, there are two buttons: "[送信]" (Send) and "[リセット]" (Reset), both with light gray borders and light gray backgrounds.

その他のフォーム

フォーム制御の詳細はHTML Standard 4.10 フォーム (<https://momdo.github.io/html/forms.html#forms>)が参考になる。

コンソールプログラム

コマンドラインで起動し、入出力を
すべてコマンドラインで完了させる
プログラム

コンソールプログラム

コンソールプログラムで値を利用者から読み取るときは主に`gets`を利用しその結果を変数に代入する。

```
STDERR.print(“年齢を入力してください : ”)  
age = gets.to_i
```

コンソールプログラムのCGI化

前ページのようなものをCGI化するには、
入力に関わる部分をCGIパラメータから
受け取るように変え、出力する部分を
HTML形式で書き出すように変更する。

コンソールプログラムのCGI化

履修科目データベース検索をCGI化する。「開講時期」をjiki、「担当教員」をkyoiで受け渡すHTMLフォームは以下の通り。

```
<form method="POST" action="db-cgi.rb">
<p>検索条件を入力して下さい。</p>
<table border="1">
  <tr><td>開講時期</td><td><select name="jiki">
    <option value="">----- (指定なし)</option>
    <option>春学期</option>
    <option>秋学期</option>
  </select></td></tr>
  <tr><td>担当教員</td><td><input name="kyoi" size="8"></td></tr>
</table>
</form>
```

検索条件を入力して下さい。

開講時期	----- (指定なし) ▼
担当教員	

コンソールプログラムのCGI化

入力される値をjiki, kyoïから受け取るRubyプログラムは以下の通り。

```
require 'cgi'
c = CGI.new(:accept_charset => "UTF-8")
term = c["jiki"]
whom = c["kyoi"]
```

この結果出力をHTML形式で行えばCGI化は完了する。

入力フォームの自動出力

検索オプションが多数になるとoption要素一覧を用意するのが困難になる。

データの値に関連する選択肢を入力フォームにいれる場合、入力フォーム自体もデータの値を管理するCGIプログラムで生成すると良い。

データベースからHTML文を出力する手順

1. データベースを読み込む
2. 自分自身を呼ぶform文を出力する
3. formの値が入力されていたら検索結果を出力

まとめ

- CGIとは、Webサーバーが外部プログラムを呼び出して動きのあるページを作るための仕組み。
- Rubyでは、「require 'cgi'」でCGIプログラムを組むのに便利なクラスを取り込むことができる。
- hidden要素を活用すると、1つのCGIプログラムに複数のwebページ出力を管理させ、紙芝居のようにページを進めるCGIプログラムを作成できる。