

情報工学実験 Perl[2]

≡ tags	情報工学実験
🕒 Created time	@July 10, 2023 2:28 PM

1.目的

代表的なスクリプト言語である Perl の特徴を学び、スクリプト言語の活用例の仕組みと基礎的な開発について理解を深めることを目的としています。

2.実装

- 課題1 (01.pl)

```
use strict;
use warnings;

print "start> ";
chomp(my $start = <STDIN>);
print "end> ";
chomp(my $end = <STDIN>);
print "result: ";

my $sum = 0;
while($start <= $end){
    $sum += $start;
    $start++;
}

print $sum;
```

- 課題2 (02.pl)

```
use strict;
use warnings;

my @week = ("Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday");
my @input_list;

print "index> \n";

while (1){
    my $num = <STDIN>;
    chomp $num;
    #空白が入力されたら終了
    last if $num eq " ";
    push @input_list, $num;
}

#昇順にソート
@input_list = sort { $a <=> $b } @input_list;

#各整数値に対応する曜日を出力
foreach my $num (@input_list) {
    print "week[$num]: $week[$num]\n";
}
```

- 課題3 (03.pl)

```
use strict;
use warnings;

sub above_average{
```

```

my $ave = ($_[0] + $_[1] + $_[2] + $_[3] + $_[4] + $_[5]) / 6;
my @return_list;

for(my $i=0; $i<6 ; $i++){
    if($ave < $_[1]){
        push(@return_list, $_[1])
    }
}
print "ave: $ave\n";
return @return_list;
}

my @result = &above_average(10, 80, 32, 58, 13, 3);

print "@result\n";

```

- 課題4 (04.pl)

```

use strict;
use warnings;

my @keys = sort keys %ENV;

foreach my $key (@keys) {
    my $value = $ENV{$key};
    print "$key: $value\n";
}

```

- 課題5 (05.pl)

```

use strict;
use warnings;

while(1){
    print "input> ";
    my $in = <STDIN>;
    chomp($in);

    last if $in eq " ";

    if ($in =~ /^[01]+5$/) {
        print "accepted\n";
    } else {
        print "not accepted\n";
    }
}

```

- 課題6 (counter.cgi)

```

#!/C:\xampp\perl\bin\perl.exe"
use CGI;

# カウンタのファイルパス
my $counter_file = './data/counter.txt'; # カウンタのファイルパスを適宜変更してください

# カウンタの初期値を設定
my $counter = read_counter_from_file($counter_file) || 0;

# カウンタのインクリメント
$counter++;

# カウンタをファイルに保存
write_counter_to_file($counter_file, $counter);

# CGIオブジェクトの作成
my $cgi = CGI->new;

# レスポンスのヘッダを出力
print $cgi->header('text/html');

# レスポンスのボディを出力
print "<html>\n";

```

```

print "<head><title>Counter</title></head>\n";
print "<body>\n";
print "<h1>Counter</h1>\n";
print "<p>This page has been loaded $counter times.</p>\n";
print "</body>\n";
print "</html>\n";

# カウンタの値をファイルから読み込むサブルーチン
sub read_counter_from_file {
    my ($file) = @_;
    if (-e $file) {
        open(my $fh, '<', $file) or die "Failed to open counter file: $!";
        my $counter = <$fh>;
        close($fh);
        chomp($counter);
        return $counter;
    }
    return undef;
}

# カウンタの値をファイルに保存するサブルーチン
sub write_counter_to_file {
    my ($file, $counter) = @_;
    open(my $fh, '>', $file) or die "Failed to open counter file: $!";
    print $fh $counter;
    close($fh);
}

```

- 課題7,8 (forum.cgi)

```

#!"C:\xampp\perl\bin\perl.exe"

use CGI;
$cgi = CGI->new();

# データの保存先のファイル
my $filepath = "../data/forum.txt";
my $data;

# カウンタ
my $counter_file = '../data/counter.txt'; # カウンタのファイルパスを適宜変更してください
my $counter = read_counter_from_file($counter_file) || 0;
$counter++;
write_counter_to_file($counter_file, $counter);

# カウンタの値をファイルに保存するサブルーチン
sub write_data{
    my ($name, $message, $num) = @_;
    open(my $fh, '>>:encoding(utf8)', $filepath);
    print $fh "$num :$name<br>$message<br>\n";
    close($fh);
}

# データをファイルから読み込むサブルーチン
sub read_print_data{
    my ($file) = @_;
    if (-e $file) {
        open(my $fh, '<', $filepath) or die "Failed to open forum file: $!";
        my @lines = <$fh>;

        # 読み込んだ行を表示する
        foreach my $line (@lines) {
            $cgi->print("$line<br>");
        }

        close($fh);
        return @data;
    }
    return undef;
}

# カウンタの値をファイルから読み込むサブルーチン
sub read_counter_from_file {
    my ($file) = @_;
    if (-e $file) {
        open(my $fh, '<', $file) or die "Failed to open counter file: $!";
        my $counter = <$fh>;
        close($fh);
    }
}

```

```

        chomp($counter);
        return $counter;
    }
    return undef;
}

# カウンタの値をファイルに保存するサブルーチン
sub write_counter_to_file {
    my ($file, $counter) = @_;
    open(my $fh, '>', $file) or die "Failed to open counter file: $!";
    print $fh $counter;
    close($fh);
}

# -----HTML-----
print $cgi->header('text/html');

# レスポンスのボディを出力
print "<html>\n";
print "<head><title>Forum</title></head>\n";
print "<body>\n";

# フォームデータの取得と表示
if ($cgi->param('submit')) {
    # フォームからのデータの取得
    my $username = $cgi->param('username');
    my $message = $cgi->param('message');

    # データの出力はここから
    print "<h1>View Data</h1>";

    # ファイルにデータを書き込み
    if($username eq '' || $message eq ''){
        print "<h5>Username or message is empty!</h5>\n";
    }
    else {
        write_data("$username", "$message", "$counter");
    }

    # ファイルからデータを読み込み
    my @data = read_print_data($filepath) || "no data to display.";
    print "<p>$data</p>\n";
}

print $cgi->start_form; # フォームの開始

print "<h1>Forum site</h1>\n";
print "<label for='username'>name:</label>\n";
print $cgi->textfield('username');
print "<br>\n";
print $cgi->textarea(-name => 'message', -rows => 6, -cols => 40);
print "<br>\n";

print $cgi->submit('submit', 'Submit'); # 送信ボタン

print $cgi->end_form; # フォームの終了

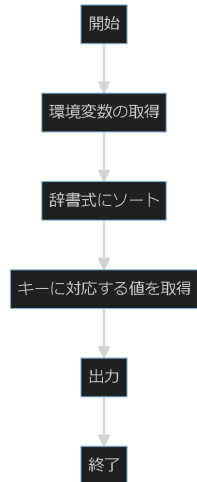
print "</body>\n";
print "</html>\n";

```

3.解説

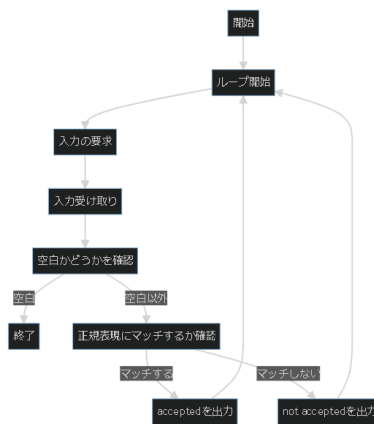
- 課題4

perlは実行時に環境変数を%ENVに格納するため、keyを指定して対応する値を簡単に呼び出すことができる。



- 課題5

perlの言語的強みは正規表現で、ライブラリを使用せずとも簡単に利用できる。



cgiファイルの実行環境

ファイル構成は以下のようになっており、apacheを起動し ブラウザから`http://localhost/cgi-bin/`実行したいcgiファイル/ にアクセスすると、実行することができる。実行したいcgiファイルを書き換え、再読み込みを行うと変更が反映される。

```

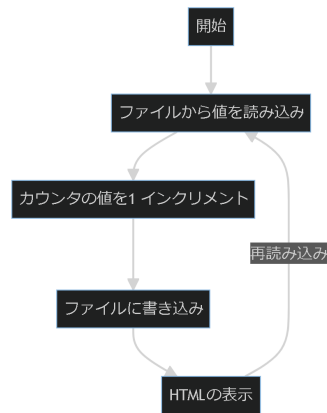
/
├── c
│   └── xampp
│       └── cgi-bin
  
```

```

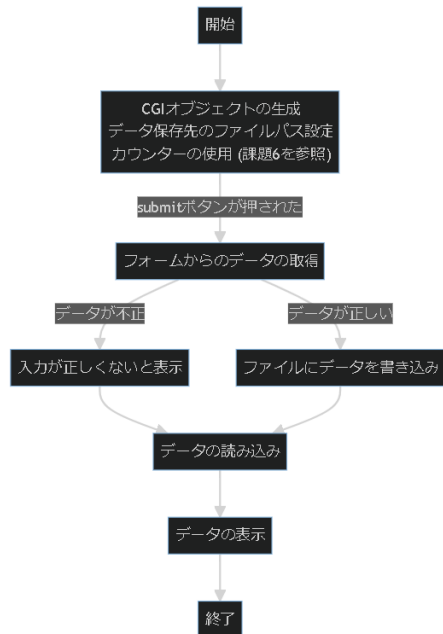
├─ forum.cgi (掲示板)
├─ counter.cgi (カウンター)
└─ data (データベース)
    ├─ counter.txt (カウンタの情報を保存)
    └─ forum.txt (掲示板の書き込みを保存)

```

• 課題6



• 課題7,8



4.結果

- 課題1

```
start> 5
end> 10
result: 45
```

- 課題2

```
index>
5
2
1

week[1]: Monday
week[2]: Tuesday
week[5]: Friday
```

- 課題3

```
ave: 32.6666666666667
80 58
```

- 課題4

```
ACLOCAL_PATH: /mingw64/share/aclocal:/usr/share/aclocal
ALLUSERSPROFILE: C:\ProgramData
APPDATA: C:\Users\inoue\AppData\Roaming
CHROME_CRASHPAD_PIPE_NAME: \\.\pipe\LOCAL\crashpad_7992_PUCQP0GPRFECKLBH
COLORTERM: truecolor
COMMONPROGRAMFILES: C:\Program Files\Common Files
COMPUTERNAME: INYT113
COMSPEC: C:\WINDOWS\system32\cmd.exe
CONFIG_SITE: /etc/config.site
CommonProgramFiles(x86): C:\Program Files (x86)\Common Files
CommonProgramW6432: C:\Program Files\Common Files
DISPLAY: needs-to-be-defined
DriverData: C:\Windows\System32\Drivers\DriverData
EFC_8792: 1
ESET_OPTIONS:

EXEPATH: C:\Program Files\Git\bin
FPS_BROWSER_APP_PROFILE_STRING: Internet Explorer
FPS_BROWSER_USER_PROFILE_STRING: Default
GIT_ASKPASS: c:\Users\inoue\AppData\Local\Programs\Microsoft VS Code\resources\app\extensions\git\dist\askpass.sh
GIT_INSTALL_ROOT: C:\Users\inoue\scoop\apps\git\current
HOME: /c/Users/inoue
HOMEDRIVE: C:
HOMEPATH: \Users\inoue
HOSTNAME: inyt113
INFOPATH: /usr/local/info:/usr/share/info:/usr/info:/share/info
LANG: en_US.UTF-8
LOCALAPPDATA: C:\Users\inoue\AppData\Local
LOGONSERVER: \\INYT113
MANPATH: /mingw64/local/man:/mingw64/share/man:/usr/local/man:/usr/share/man:/usr/man:/share/man
MINGW_CHOST: x86_64-w64-mingw32
MINGW_PACKAGE_PREFIX: mingw-w64-x86_64
MINGW_PREFIX: /mingw64
MSYSTEM: MINGW64
MSYSTEM_CARCH: x86_64
MSYSTEM_CHOST: x86_64-w64-mingw32
MSYSTEM_PREFIX: /mingw64
NODE_PATH: C:\Users\inoue\scoop\apps\yarn\current\global\node_modules
NUMBER_OF_PROCESSORS: 8
ORIGINAL_PATH: /mingw64/bin:/usr/bin:/c/Users/inoue/bin:/c/Program Files (x86)/Common Files/Oracle/Java/javapath:/c/WINDOWS/system32:/c/WIN
ORIGINAL_TEMP: /tmp
ORIGINAL_TMP: /tmp
```

```

ORIGINAL_XDG_CURRENT_DESKTOP: undefined
OS: Windows_NT
OneDrive: C:\Users\inoue\OneDrive
PATH: /c/Users/inoue/bin:/mingw64/bin:/usr/local/bin:/usr/bin:/bin:/mingw64/bin:/usr/bin:/c/Users/inoue/bin:/c/Program Files (x86)/Common F
PATHEXT: .COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC;.RB;.RBW
PKG_CONFIG_PATH: /mingw64/lib/pkgconfig:/mingw64/share/pkgconfig
PLINK_PROTOCOL: ssh
PROCESSOR_ARCHITECTURE: AMD64
PROCESSOR_IDENTIFIER: Intel64 Family 6 Model 126 Stepping 5, GenuineIntel
PROCESSOR_LEVEL: 6
PROCESSOR_REVISION: 7e05
PROGRAMFILES: C:\Program Files
PS1: \[\033]0;$TITLEPREFIX:$PWD\007\]\n\[\033[32m]\u@h \[\033[35m]$MSYSTEM \[\033[33m]\w\[\033[36m]`__git_ps1`\[\033[0m]\n$
PModulePath: C:\Program Files\WindowsPowerShell\Modules;C:\WINDOWS\system32\WindowsPowerShell\v1.0\Modules
PUBLIC: C:\Users\Public
PWD: /c/Users/inoue/Desktop/perl
ProgramData: C:\ProgramData
ProgramFiles(x86): C:\Program Files (x86)
ProgramW6432: C:\Program Files
SESSIONNAME: Console
SHELL: /usr/bin/bash
SHLVL: 1
SSH_ASKPASS: /mingw64/bin/git-askpass.exe
SYSTEMDRIVE: C:
SYSTEMROOT: C:\WINDOWS
TEMP: /tmp
TERM: xterm-256color
TERM_PROGRAM: vscode
TERM_PROGRAM_VERSION: 1.80.0
TMP: /tmp
TMPDIR: /tmp
USERDOMAIN: INYT113
USERDOMAIN_ROAMINGPROFILE: INYT113
USERNAME: inoue
USERPROFILE: C:\Users\inoue
VSCODE_GIT_ASKPASS_EXTRA_ARGS: --ms-enable-electron-run-as-node
VSCODE_GIT_ASKPASS_MAIN: c:\Users\inoue\AppData\Local\Programs\Microsoft VS Code\resources\app\extensions\git\dist\askpass-main.js
VSCODE_GIT_ASKPASS_NODE: C:\Users\inoue\AppData\Local\Programs\Microsoft VS Code\Code.exe
VSCODE_GIT_IPC_HANDLE: \\.\pipe\vscode-git-1b5038f3ab-sock
WINDIR: C:\WINDOWS
_: /usr/bin/perl
perl: C:\xampp\perl\bin

```

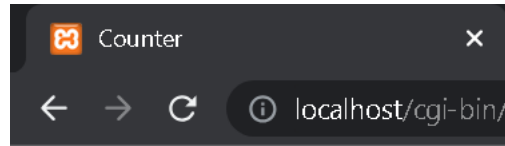
- 課題5

```

input> 10
not accepted
input> 15
accepted
input> 1015
accepted
input> 5
not accepted
input> 25
not accepted
input>

```

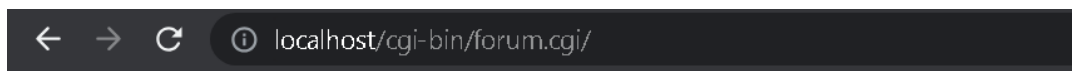
- 課題6



Counter

This page has been loaded 18 times.

- 課題7



View Data

1 :inoue
hello my name is inoue
i made this forum site

2 :takahashi
Hi, good job!

Forum site

name:

Submit

5.その他のスクリプト言語についての調査

1. Python

- 言語の特徴: シンプルで読みやすい文法、動的型付け、豊富な標準ライブラリ、オブジェクト指向
- 主な用途: Web開発、データサイエンス、機械学習、自動化スクリプト、科学計算、バックエンド開発

2. JavaScript

- 言語の特徴: ウェブブラウザ上で実行される、動的型付け、イベント駆動、クライアントサイドとサーバサイドの両方で使用可能(Node.jsを使用)。
- 主な用途: ウェブ開発、フロントエンド開発、バックエンド開発 (Node.jsを使用)、モバイルアプリ開発 (React Nativeを使用)

3. Ruby

- 言語の特徴: シンプルで読みやすい文法、動的型付け、オブジェクト指向
- 主な用途: Web開発、スクリプト作成、プロトタイピング、ウェブフレームワーク (Ruby on Rails)

4. PHP

- 言語の特徴: ウェブ開発のために設計されている、動的型付け、ウェブサーバーとの統合が容易、開発者コミュニティが大規模であること
- 主な用途: ウェブ開発、サーバーサイドスクリプティング、フォーム処理、データベース連携、ウェブフレームワーク

6.結論

UIをあまり気にしないような動的なwebサイトを構築するにはHTMLに埋め込んでかけるような perl, PHPなどのスクリプト言語が適している。サーバーサイドで実行されるので、クライアントのブラウザが異なっても同じ挙動をさせられるのもメリットの一つだと思った。