情報理工学実験 II
データベースと WEB サービス構築
フレームワークなしの Web サーバプログラミング
レポート

Klak Bartosz
クラク バルトッシュ
学生番号 02166102

2018 年 10 月 25 日

1　基本課題 2-1

Listing 1　FizzBuzz.rb

```
puts("Insert number:")
maxn = gets().chomp

puts
puts("Answer:")
1.upto(maxn.to_i) do |i|
  if i % 5 == 0 and i % 3 == 0
    puts "Fizz Buzz"
  elsif i % 5 == 0
    puts "Buzz"
  elsif i % 3 == 0
    puts "Fizz"
  else
    puts i
  end
end
```

Listing 2　FizzBuzz.txt

```
[02166102@PC169 2]$ ruby FizzBuzz.rb
Insert number:
17

Answer:
1
2
Fizz
4
Buzz
Fizz
7
8
Fizz
Buzz
11
Fizz
13
14
Fizz Buzz
16
17
```

2　基本課題 2-2

Listing 3　tryLibrary.rb

```
require 'bigdecimal'
var1 = 4
var2 = BigDecimal("1000.0000000000000000000000000000000001")
print var1 + var2
puts ''

require 'matrix'
mat1 = Matrix[[1,2,3],[4,5,6],[7,8,9]]
mat2 = Matrix[[9,8,7],[6,5,4],[3,2,1]]
print mat1 + mat2
puts ''
mat3 = mat1*mat2
print mat3
puts ''
print mat3.eigensystem.eigenvalues
puts ''
```

Listing 4　tryLibrary.txt

```
[02166102@PC169 2]$ ruby tryLibrary.rb
0.10040000000000000000000000000000000000001E4
Matrix[[10, 10, 10], [10, 10, 10], [10, 10, 10]]
Matrix[[30, 24, 18], [84, 69, 54], [138, 114, 90]]
[187.26987657639725, 1.730123423602656, -1.4656551669782677e-14]
```

そのプログラムは、bigdecimal と matrix という library の働き方を示す。

3　基本課題 2-3

Listing 5　min-Chat-Server.rb

```
require 'socket'

server = TCPServer.new 2000
a = 1

client = server.accept

while a == 1
        puts client.gets
end

client.close
```

Listing 6　min-Chat-Client.rb

```
require 'socket'

socket = TCPSocket.new('127.0.0.1', 2000)
a = 1

while a==1
chat = gets().chomp
if chat == "bye"
        socket.puts chat
        break
        socket.close
else
        socket.puts chat
end
end
```

Listing 7　min-Chat.txt

```
[02166102@PC169 2]$ ruby min-Chat_Server.rb
hello
hi
whats up

(...)

^C
min-Chat_Server.rb:9:in `puts': Interrupt
        from min-Chat_Server.rb:9:in `puts'
        from min-Chat_Server.rb:9:in `<main>'

----------------------------------------------------------

[02166102@PC169 2]$ ruby min-Chat_Client.rb
hello
hi
whats up
bye
```

4　基本課題 2-4

Listing 8　reply-WebServer.rb

```ruby
require 'webrick'
CGI_PATH = '/usr/bin/ruby'
srv = WEBrick::HTTPServer.new({ :DocumentRoot => './', :BindAddress => '127.0.0.1', :Port => 2000, :CGIInterpreter => CGI_PATH})
srv.mount('/time', WEBrick::HTTPServlet::FileHandler, 'time.html')
srv.mount('/fizzbuzz', WEBrick::HTTPServlet::CGIHandler, 'FizzBuzz_cgi.rb')
trap("INT"){srv.shutdown}
srv.start
```

Listing 9　reply-WebServer.txt

```
http://127.0.0.1:2000/time

Server time: Mon Oct 22 2018 16:17:52 GMT+0900 (JST)

------------------------------------------------------------

http://127.0.0.1:2000/fizzbuzz?num=16

1       1
2       2
3       Fizz
4       4
5       Buzz
6       Fizz
7       7
8       8
9       Fizz
10      Buzz
11      11
12      Fizz
13      13
14      14
15      FizzBuzz
16      16

------------------------------------------------------------

[02166102@PC169 2]$ ruby reply_WebServer.rb
[2018-10-22 16:17:36] INFO  WEBrick 1.3.1
[2018-10-22 16:17:36] INFO  ruby 2.0.0 (2014-11-13) [x86_64-linux]
[2018-10-22 16:17:36] INFO  WEBrick::HTTPServer#start: pid=23524 port=2000
localhost.localdomain - - [22/Oct/2018:16:17:52 JST] "GET /time HTTP/1.1" 304 0
- -> /time
[2018-10-22 16:17:52] ERROR ‘/serverDate.js’ not found.
localhost.localdomain - - [22/Oct/2018:16:17:52 JST] "GET /serverDate.js HTTP/1.1" 404 283
http://127.0.0.1:2000/time -> /serverDate.js
/usr/share/ruby/webrick/httpservlet/cgihandler.rb:47: warning: Insecure world writable dir /home in PATH, mode 040777
localhost.localdomain - - [22/Oct/2018:16:18:56 JST] "GET /fizzbuzz HTTP/1.1" 200 42
- -> /fizzbuzz
localhost.localdomain - - [22/Oct/2018:16:19:09 JST] "GET /fizzbuzz?num=16 HTTP/1.1" 200 542
- -> /fizzbuzz?num=16
^C[2018-10-22 16:21:52] INFO  going to shutdown ...
[2018-10-22 16:21:52] INFO  WEBrick::HTTPServer#start done.
```

Listing 10　time.html

```html
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    <title>Server time</title>
    <script language="javascript" src="serverDate.js"></script>
  </head>
  <script language="javascript">
  var serverTime = new Date();
  document.write("Server time: " + serverTime + "<br>");
  </script>
  <body>
  </body>
</html>
```

Listing 11　FizzBuzz-cgi.rb

```
require "cgi"
cgi = CGI.new

cgi.out("type" => "text/html", "charset" => "UTF-8") do
  number = cgi["num"].to_i
  html = "<html><body><table>\n"
  number.times do |n|
if (n+1)%3==0 && (n+1)%5==0
    html = html + "<tr><th>#{n+1}</th><th>FizzBuzz</th></tr>"
elsif (n+1)%3==0
    html = html + "<tr><th>#{n+1}</th><th>Fizz</th></tr>"
elsif (n+1)%5==0
    html = html + "<tr><th>#{n+1}</th><th>Buzz</th></tr>"
else
    html = html + "<tr><th>#{n+1}</th><th>#{n+1}</th></tr>"
  end
end
  html = html + "</table></body></html>"
  html
end
```

5　基本課題 2-5

XSS - cross site scripting: XSS では攻撃する側は、Client 側のスクリプトを Website に入力する。それによって、User の Cookie が盗まれたら、その User のアカウントをハイジャックできる。その問題の対策の一つとして、SameSite cookie parameter という技術がある。それは、Cookie が SameSite=Strict というパラメータをもち、Cross のリクエストを防ぐ。

CSRF - cross site request forgeries: CSRF は、Web が知っているユーザーから無用なコマンドを送信する Website である。その対策として STP という技術などが存在する。STP(Synchronizer Token Pattern) は Token、つまり秘密でユニークに作成された値が Server 側にチェックされている。

Cross site cooking は攻撃の側は Browser の Cookie を別の Server の Cookie Domain に入力するという攻撃である。

ソース：

https://en.wikipedia.org/wiki/Cross-site_request_forgery

https://en.wikipedia.org/wiki/Cross-site_scripting

https://en.wikipedia.org/wiki/Cross-site_cooking

6　応用課題 2-1

Listing 12　bulletinBoard-Server.rb

```ruby
require 'webrick'
CGI_PATH = '/usr/bin/ruby'
srv = WEBrick::HTTPServer.new({ :DocumentRoot => './', :BindAddress => '127.0.0.1', :Port => 2000, :CGIInterpreter => CGI_PATH})
srv.mount('/write', WEBrick::HTTPServlet::CGIHandler, 'write_cgi.rb')
srv.mount('/index', WEBrick::HTTPServlet::CGIHandler, 'index_cgi.rb')
srv.mount('/read', WEBrick::HTTPServlet::CGIHandler, 'read_cgi.rb')
trap("INT"){srv.shutdown}
srv.start
```

Listing 13　bulletinBoard-Server.txt

```
http://127.0.0.1:2000/write?msg=king

your message: king

-------------------------------------------------------------------------------------------------------
http://127.0.0.1:2000/write?msg=door

This is the forbidden word!

-------------------------------------------------------------------------------------------------------
http://127.0.0.1:2000/index

id: 1
id: 2
id: 3

-------------------------------------------------------------------------------------------------------
http://127.0.0.1:2000/read?id=2

your message was: hello

-------------------------------------------------------------------------------------------------------
[02166102@PC169 2]$ ruby bulletinBoard_Server.rb
[2018-10-22 16:23:47] INFO  WEBrick 1.3.1
[2018-10-22 16:23:47] INFO  ruby 2.0.0 (2014-11-13) [x86_64-linux]
[2018-10-22 16:23:47] INFO  WEBrick::HTTPServer#start: pid=24919 port=2000
/usr/share/ruby/webrick/httpservlet/cgihandler.rb:47: warning: Insecure world writable dir /home in PATH, mode 040777
localhost.localdomain - - [22/Oct/2018:16:24:00 JST] "GET /write?msg=hi HTTP/1.1" 200 50
- -> /write?msg=hi
localhost.localdomain - - [22/Oct/2018:16:24:06 JST] "GET /write?msg=door HTTP/1.1" 200 58
- -> /write?msg=door
localhost.localdomain - - [22/Oct/2018:16:24:13 JST] "GET /write?msg=hello HTTP/1.1" 200 53
- -> /write?msg=hello
localhost.localdomain - - [22/Oct/2018:16:24:19 JST] "GET /write?msg=king HTTP/1.1" 200 52
- -> /write?msg=king
localhost.localdomain - - [22/Oct/2018:16:24:35 JST] "GET /index HTTP/1.1" 200 63
- -> /index
localhost.localdomain - - [22/Oct/2018:16:25:08 JST] "GET /read?id=2 HTTP/1.1" 200 59
- -> /read?id=2
^C[2018-10-22 16:25:15] INFO  going to shutdown ...
[2018-10-22 16:25:15] INFO  WEBrick::HTTPServer#start done.
```

Listing 14　write-cgi.rb

```ruby
require "cgi"
cgi = CGI.new
cgi.out("type" => "text/html", "charset" => "UTF-8") do
  mssg  = cgi["msg"]
  id=1
  html = "<html><body>\n"
  if mssg == "door"
        html = html + "This is the forbidden word!</p>"
  else
        f="index.txt"
        if File.exist?(f)
```

```
                f = File.open("index.txt", "r")
                f.each_line do |line|
                        id+=1
                end
                f = File.open("index.txt", "a")
                f.puts("#{id}: #{mssg}\n")
                f.close
                html = html + "<p>your message: #{mssg}</p>"
        else
                f = File.open("index.txt", "w")
                f.puts("#{id}: #{mssg}\n")
                f.close
                html = html + "<p>your message: #{mssg}</p>"
        end
  end
  html = html + "</body></html>"
  html
end
```

Listing 15 index-cgi.rb

```
require "cgi"
cgi = CGI.new

a=1

cgi.out("type" => "text/html", "charset" => "UTF-8") do
  id=0
  html = "<html><body>\n"
  f="index.txt"
  if File.exist?(f)
        f = File.open("index.txt", "r")
        f.each_line do |line|
                id+=1
        html = html + "<p>id: #{id}</p>"
        end
        f.close
  else
        html = html + "<p>no id available</p>"
  end
  html = html + "</body></html>"
  html
end
```

Listing 16 index.txt

```
1: hi
2: hello
3: king
```

Listing 17 read-cgi.rb

```
require "cgi"
cgi = CGI.new

cgi.out("type" => "text/html", "charset" => "UTF-8") do
  index = cgi["id"].to_i
  id=1
  html = "<html><body>\n"
  f="index.txt"
  if File.exist?(f)
        f = File.open("index.txt", "r")
        f.each_line do |line|
                if index == id
                        a = line.split(':')[1]
                        html = html + "<p>your message was: #{a}</p>"
                end
                id+=1
        end
  end

  html = html + "</body></html>"
  html
end
```