基礎プログラミングI

第8回　正規表現

メディア情報コース
平居　悠（ひらい　ゆたか）

到達目標

計算機上での情報の取り扱い方の基礎の理解

1. コンピュータに指示を送る基本を理解する。
2. 文字列や数値を基本とした値の概念とそれを格納する変数の仕組みを理解する。
3. 変数一つで集合を表す概念を理解する。
4. ２進数を基本としたコンピュータの内部表現について基本を理解する。
5. 現実社会の簡単な問題を扱うプログラムを作成できるようになる。

前回

第１回	４月９日	プログラミングの基礎
第２回	４月１６日	変数・制御構造
第３回	４月２３日	メソッド、値の型変換
第４回	４月３０日	確認テスト
第５回	５月１４日	集合処理１（配列）
第６回	５月２１日	集合処理２（CSVとデータ処理）
第７回	**５月２８日**	**集合処理３（ハッシュ）**
第８回	６月１１日	正規表現
第９回	６月１８日	計算機の内部表現
第１０回	６月２５日	スタイルとデバッグ、実用的なプログラム
第１１回	７月　２日	作品の公開
第１２回	７月　９日	チーム課題作成
第１３回	７月１６日	チーム課題発表
第１４回	日程未定	期末試験

前回の目標

CSVをハッシュの集合として処理し集計結果などが作れる

まとめ

- 特定のキーワードに対応する値を持つ関係性の集合を**ハッシュ**という。
- 各関係の左辺にある元となるキーワードのことを**キー (key)**、右辺にある対応値のことを**バリュー (value)**という。
- ヘッダ行を持つCSVファイルを読み込むときはCSV.read(ファイル名, **headers: true**)とする。

今回

第１回	４月９日	プログラミングの基礎
第２回	４月１６日	変数・制御構造
第３回	４月２３日	メソッド、値の型変換
第４回	４月３０日	確認テスト
第５回	５月１４日	集合処理１（配列）
第６回	５月２１日	集合処理２（CSVとデータ処理）
第７回	５月２８日	集合処理３（ハッシュ）
第８回	**６月１１日**	**正規表現**
第９回	６月１８日	計算機の内部表現
第１０回	６月２５日	スタイルとデバッグ、実用的なプログラム
第１１回	７月　２日	作品の公開
第１２回	７月　９日	チーム課題作成
第１３回	７月１６日	チーム課題発表
第１４回	日程未定	期末試験

今回の目標

正規表現を利用したデータのフィルタリングができる

導入課題

チーム内で2人もしくは3人組を作り、以下から1つ問題を選んで協力して解答せよ。解答はs4基礎プロI (G, H)の「#08 導入課題」に指示通り書き込む。ただし、チーム内の組ごとに異なる問題を解答すること。書き込みは全員行うこと。

1. **正規表現 /sai?toh/はどのような意味か？**
2. **正規表現/ai[sz]awa/は文字列“ayukawa”とマッチするか？**
3. **正規表現/Sai?to/は文字列“sato”とマッチするか？**

タイピング練習スケジュール

第１回　　ホームポジション
第２回　　ローマ字
第３回　　英語初級
第４回　　日本国憲法（trr試験）
第５回　　ホームポジション
第６回　　ローマ字
第７回　　英語初級
第８回　　日本国憲法（trr試験）
第９回　　ホームポジション
第１０回　ローマ字
第１１回　英語初級
第１２回　日本国憲法（trr試験、合格スコア150）

trr起動方法

1. ブラウザを起動し、https://www.koeki-prj.org/trr/ に繋ぐ。
2. 学籍番号（Cは大文字、省略なし８桁）を入力する。
3. Koeki MAILに届いたパスコードをPasscode: 欄に入力する。

ホームポジション


左手でタイプするキー
右手でタイプするキー
小指
薬指
中指
人差指
人差指
中指
薬指
小指
漢字
Back space
Tab
Enter
Caps Lock
Shift
Shift
Ctrl
Fn
Alt
Space
Alt
Fn
Ctrl
左手の人差指から小指までのホームポジション
両手の親指のホームポジション
右手の人差指から小指までのホームポジション

今回の内容

- egrepコマンドによる正規表現の実験
- Rubyで使う正規表現
- 正規表現検索の練習

今回の内容

- egrepコマンドによる正規表現の実験
- Rubyで使う正規表現
- 正規表現検索の練習

名簿データダウンロード

名簿データ：

https://www.koeki-prj.org/~yuuji/2026/pf1/regex/meibo.csv

リンクを右クリックし“Save link as”で、~/Rubyに保存する。

egrepコマンド

正規表現を使ってファイルから
特定の行を検索するコマンド

egrep “正規表現パターン” [ファイル群]

SAITOHまたはSATOHの検索

SAITOHまたはSATOHのいずれかを含む行を全部選んでみよう。この場合、大文字のIがあってもなくても良いので、正規表現パターンは“SAI?TOH”となる。

egrep “SAI?TOH” meibo.csv

大文字小文字の同一視

検索の際、大文字小文字を区別しないためには、egrepコマンドに-iオプションを指定する。

```
egrep –i “sai?toh” meibo.csv
```

日本語（カタカナ）での検索

```
egrep –i “サイ?トウ” meibo.csv
```

今回の内容

- egrepコマンドによる正規表現の実験
- **Rubyで使う正規表現**
- 正規表現検索の練習

指定したファイルからsai?tohというパターンを検索するプログラム

右のプログラム、saito.rbを書いて実行してみよう。

```
#!/usr/koeki/bin/ruby
# -*- coding: utf-8 -*-
require 'csv'

meibo = CSV.read(ARGV[0], headers:true)
meibo.each do |row|
  if /sai?toh/i =~ row["Name"]
    print row.to_s
  end
end
```

指定したファイルからsai?tohというパターンを検索するプログラム

コマンドラインで指定したmeibo.csvがARGV[0]に代入され、それがヘッダ付きCSVとして読み込まれる。

```
#!/usr/koeki/bin/ruby
# -*- coding: utf-8 -*-
require 'csv'

meibo = CSV.read(ARGV[0], headers:true)
meibo.each do |row|
  if /sai?toh/i =~ row["Name"]
    print row.to_s
  end
end
```

指定したファイルからsai?tohというパターンを検索するプログラム

meibo変数にあるレコードを１つ１つ取り出して繰り返す。

```
#!/usr/koeki/bin/ruby
# -*- coding: utf-8 -*-
require 'csv'

meibo = CSV.read(ARGV[0], headers:true)
meibo.each do |row|
  if /sai?toh/i =~ row["Name"]
    print row.to_s
  end
end
```

指定したファイルからsai?tohというパターンを検索するプログラム

各レコード中のNameフィールドを取り出し、それが正規表現にマッチしているかの判定を行う。

```
#!/usr/koeki/bin/ruby
# -*- coding: utf-8 -*-
require 'csv'

meibo = CSV.read(ARGV[0], headers:true)
meibo.each do |row|
  if /sai?toh/i =~ row["Name"]
    print row.to_s
  end
end
```

指定したファイルからsai?tohというパターンを検索するプログラム

「正規表現にマッチしているか？」を意味する条件式は

正規表現 =~ 文字列

と書く。

```
#!/usr/koeki/bin/ruby
# -*- coding: utf-8 -*-
require 'csv'

meibo = CSV.read(ARGV[0], headers:true)
meibo.each do |row|
  if /sai?toh/i =~ row["Name"]
    print row.to_s
  end
end
```

指定したファイルからsai?tohというパターンを検索するプログラム

正規表現にマッチしたフィールドを文字列に変換して出力。

```
#!/usr/koeki/bin/ruby
# -*- coding: utf-8 -*-
require 'csv'

meibo = CSV.read(ARGV[0], headers:true)
meibo.each do |row|
  if /sai?toh/i =~ row["Name"]
    print row.to_s
  end
end
```

指定したファイルからsai?tohというパターンを検索するプログラム

ifに対応するend

```
#!/usr/koeki/bin/ruby
# -*- coding: utf-8 -*-
require 'csv'

meibo = CSV.read(ARGV[0], headers:true)
meibo.each do |row|
  if /sai?toh/i =~ row["Name"]
    print row.to_s
  end
end
```

指定したファイルからsai?tohというパターンを検索するプログラム

eachに対応する
end

```
#!/usr/koeki/bin/ruby
# -*- coding: utf-8 -*-
require 'csv'

meibo = CSV.read(ARGV[0], headers:true)
meibo.each do |row|
  if /sai?toh/i =~ row["Name"]
    print row.to_s
  end
end
```

問題

saito.rbを以下のように改良したものを作成せよ。

1. 検索パターンを「サイ？トウ」に変えて、検索フィールドをナマエ（第1項目）として検索した結果を示すsaito-katakana.rb
2. 検索対象を1レコード全体(row.to_s)として検索するsaito-line.rb

今回の内容

- egrepコマンドによる正規表現の実験
- Rubyで使う正規表現
- 正規表現検索の練習

任意のパターーンを検索するプログラム

右のプログラム
meibosearch.rbを
書いてみよう。

```
#!/usr/koeki/bin/ruby
# -*- coding: utf-8 -*-
require 'csv'
meibo = CSV.read(ARGV[0], headers:true)

print "検索パターン: "
pattern = STDIN.gets.chomp!
regexp = Regexp.new(pattern, nil)

meibo.each do |row|
 line = row.to_s
 if regexp =~ line
  print line
 end
end
```

任意のパターーンを検索するプログラム

任意のCSVファイルを読み込む。

例：

./meibosearch.rb meibo.csv

```
#!/usr/koeki/bin/ruby
# -*- coding: utf-8 -*-
require 'csv'
meibo = CSV.read(ARGV[0], headers:true)

print "検索パターーン: "
pattern = STDIN.gets.chomp!
regexp = Regexp.new(pattern, nil)

meibo.each do |row|
  line = row.to_s
  if regexp =~ line
    print line
  end
end
```

任意のパターーンを検索するプログラム

標準入力で与えた文字列を正規表現に変換

```
#!/usr/koeki/bin/ruby
# -*- coding: utf-8 -*-
require 'csv'
meibo = CSV.read(ARGV[0], headers:true)

print "検索パターーン: "
pattern = STDIN.gets.chomp!
regexp = Regexp.new(pattern, nil)

meibo.each do |row|
 line = row.to_s
 if regexp =~ line
  print line
 end
end
```

ひらがなを含む行の検索

ひらがなは正規表現[ぁ-ん]で表現できる。

```
./meibosearch.rb meibo.csv
検索パターン: [ぁ-ん]
サトウ ムネユキ,SATOH Muneyuki,さとう胸幸,青葉台
サイトウ テツヤ,SAITOH Tetsuya,斎藤徹夜,白夜の国
```

「サ」で始まる人全てを検索

行の先頭は^で指定する。^サとすると行の先頭が「サ」で始まるもののみにマッチする。

```
./meibosearch.rb meibo.csv
検索パターン: ^サ
サトウ ムネユキ,SATOH Muneyuki,さとう胸幸,青葉台
サイトウ テツヤ,SAITOH Tetsuya,斎藤徹夜,白夜の国
```

同じ文字（空白以外）が繰り返す行の検索

グルーピングを使うと繰り返しを記述できる。空白以外の文字は\Sである。これを括弧でグルーピングして、\数字で参照する。正規表現は(\S)\1となる。

```
./meibosearch.rb meibo.csv
検索パターン: (\S)\1
イイモリ ハナコ,IIMORI Hanako,飯森花子,飯森山
ゲゲノ キタロウ,GEGENO Kitarou,下々野喜太郎,魔界村
```

「イ」と「ゲ」が2字連続で出てくるのでこれにマッチする。

下の名前に「ヒミ」が付く人を検索

苗字と下の名前の区切りにスペースがあることを利用し、

行頭＋苗字＋空白＋下の名前

というパターンを記述する。苗字は単語だったら何でも構わないので「1文字以上の非空白文字」を意味する\S+でマッチさせる。下の名前は「ヒミ」がどこに出てきても良いように\S*ヒミ\S*\bとする。

./meibosearch.rb meibo.csv
検索パターン: ^\S+\s+\S*ヒミ\S*\b
ヤマタイ ヒミコ,YAMATAI Himiko,邪馬台氷見子,大和

名前が「たろう」の人

「タロ」の次が「ウ」でも「ー」でもマッチするようタロ[ウー]とする。ただし、名前の途中にそれらが含まれるものは除外したいので単語境界を意味する\bで絞り込みを行う。

./meibosearch.rb meibo.csv
検索パターン: \bタロ[ウー]\b
ナカマチ タロウ,NAKAMACHI Taro,中町太郎,酒田
ナニワノ タロー,NANIWANO Taro,浪花之太朗,探偵騎士王国

提出課題（必須:s4提出）

問題１

課題7aで作成したmydata.csvに含まれる特定のフィールドから特定のパターンにマッチするレコードのみを出力するプログラムmydata-search.rbを作成せよ。

レポート提出方法

s4基礎プロIG, Hの「#08 提出課題」に以下の通り書き込む。

- クラス、学籍番号、氏名
- 作成したプログラムの仕組みの説明
- 実行結果（約20行以内）
- 所用時間と共同作成者
- 作成したmydata-search.rbの添付

締め切り：6月15日 (月)

ボーナス課題

問題 2

自作したmydata.csvに数値項目がない場合は追加せよ。その上で、数値による絞り込み検索ができるプログラムmydata-N.rbを以下の仕様により作成せよ。

(イ)./mydata-N.rb mydata.csv 下限値　上限値

と2つの数値を第2第3引数（つまりARGV[1], ARGV[2]）に指定するとその数値項目の値が指定した範囲内にある場合のみそのレコードを出力する。

(ロ) ./mydata-N.rb mydata.csv　下限値

のように上限値指定を省略した場合は、その数値項目が下限値以上のものを全て出力する。

ボーナス課題

問題３

自作したmydata.csvの任意の項目で検索できるようなプログラムfield-search.rbを作成せよ。

ボーナス課題は以下の通りkoeki MAILで提出する。

- 提出先：yutaka.hirai@koeki-u.ac.jp
- 件名：#08 正規表現 (解いた問題番号)
- 本文：以下の順番で記載せよ。

1. **クラス・学籍番号・氏名**
2. **作成したプログラムの中身の説明**
3. **プログラムを動かして正規表現検索した結果の説明**
4. **参考文献**
5. **実行結果**
6. **共同作成者・所要時間**
7. **感想**
8. **作成したファイルの添付**

締め切り：6月17日 (水)

今回の目標

正規表現を利用したデータのフィルタリングができる

今回の内容

- egrepコマンドによる正規表現の実験
- Rubyで使う正規表現
- 正規表現検索の練習

まとめ

- データから何かを検索するときに、**正規表現**を用いる。
- Rubyでは、正規表現を/パターン/と表記する。
- **メタキャラクタ**を用いて検索方法を指定することができる。

次回

第１回	４月９日	プログラミングの基礎
第２回	４月１６日	変数・制御構造
第３回	４月２３日	メソッド、値の型変換
第４回	４月３０日	確認テスト
第５回	５月１４日	集合処理１（配列）
第６回	５月２１日	集合処理２（CSVとデータ処理）
第７回	５月２８日	集合処理３（ハッシュ）
第８回	６月１１日	正規表現
第９回	**６月１８日**	**計算機の内部表現**
第１０回	６月２５日	スタイルとデバッグ、実用的なプログラム
第１１回	７月　２日	作品の公開
第１２回	７月　９日	チーム課題作成
第１３回	７月１６日	チーム課題発表
第１４回	日程未定	期末試験

事前課題

1. 第9回解説動画(https://youtu.be/Gm_Fs9qVG1s)を視聴する。
2. 第9回授業資料「計算機の内部表現」(https://www.koeki-prj.org/~yuuji/2026/pf1/internal/index.html)を読む。

事前課題

3. (1) クイズ

quiz ruby-digitalを3回以上行ったベストスコアを記載せよ。

事前課題

3. (2) 説明資料

以下テーマについて使い方と具体的なプログラム例を独自に作成して説明する資料を提出せよ。

1. gets, to_i, to_f, to_sとprintfの%d, %s, %x
2. if/while/breakと真偽値
3. 配列と添字, <<, length, eachとforによる繰り返し
4. CSVファイルの構造と作成方法・作成時の注意
5. ヘッダ付きCSVファイルのRubyプログラムからの処理の仕方
6. 正規表現とはどういうものでどういうときに使うものか
7. 正規表現の特殊文字の使い方 . [] ? + * () $1, $2, $3, ...

選んだテーマにある項目を全て含むプログラムを作成し、その部分部分の働きを図などを用いて説明する資料(PDF)とともに提出せよ。図は手書きのものを写真で取り込んでも良い。

レポート提出方法

s4基礎プロIG, Hの「#09 事前課題★説明資料★」に以下の通り書き込む。

- クラス、学籍番号、氏名、選んだ項目
- quiz ruby-digitalのベストスコア
- どんな例題プログラムを作成したか
- ファイル添付（プログラムと説明資料）

締め切り：6月17日 (水)