エディタ・ターミナル

クラスは？

1. S1-01～03（1年生）
2. S1-04（1年生）
3. S1-05（1年生）
4. S1-15（1年生）
5. 他クラス（1年生文系）
6. 他クラス（1年生理系）
7. 他クラス（2年生）
8. 上記以外

なぜアルゴリズム入門を受講したか？

1. もともと情報科学に興味があった
2. １学期の「情報」が面白かった
3. 先輩から勧められた
4. 教員から勧められた
5. 担当教員が仏だと聞いて期待した
6. 担当教員が鬼だと聞いて燃えた
7. 時間帯が良い
8. 上記以外

なぜアルゴリズム入門を受講しないか？

1. もともと情報科学に興味がない
2. １学期の「情報」で疲れた
3. 先輩から避けるように言われた
4. 教員が仏だと聞いてなえた
5. 教員が鬼だと聞いてなえた
6. お昼過ぎは眠たい
7. 上記以外

ディレクトリは？

1. 授業用のディレクトリで作業している
2. ホームディレクトリで作業している
3. デスクトップ(自体)で作業している
4. ホームディレクトリ／デスクトップって何？
5. 上記以外

「パス」は？

1. 絶対パス・相対パスともに理解している
2. 絶対パスがわからない
3. 相対パスがわからない
4. 絶対パスも相対パスも理解できていない

エディタは

1. Emacs を使ったことがある/ている
2. miを使ったことがある/ている
3. テキストエディットを使ったことがある/ている
4. その他のエディタを使ったことがある/ている
5. 使ったことがない
6. エディタって何？

「GUI」「CUI」は？

1. 理解している
2. GUI, CUI ってなに？

ターミナルとファイルシステム

1学期の復習

「ls -l」の結果 mydir1 の先頭は？

1. drwxr-xr-x
2. drwx------

ターミナルのコマンドのまとめ

ディレクトリ ＝ フォルダ

(ワーキング)ディレクトリを移動して作業する

cd (change directory)

ディレクトリの移動

pwd (print working directory)

ワーキングディレクトリの表示

ls (list)

(ワーキング)ディレクトリ内のファイルの一覧

mkdir/rmdir (make / remove directory)

ディレクトリの作成 / 消去

パスは？

1. 理解した
2. 絶対パスがわからない
3. 相対パスがわからない
4. まったく理解できていない

cd, pwd, ls, mkdir コマンド

1. 理解して使えそう
2. まったく理解できない
-. 特に難しいコマンド名

授業用のディレクトリを作って，カレントディレクトリにする

ホームディレクトリかデスクトップの直下に
　Algorithmというディレクトリを作る
（すでに作ってある場合には，作成しなくて良い）
カレントディレクトリをAlgorithmに移動する

移動が終わったら投票

1. 終わった

ターミナルのコマンド

ディレクトリ ＝ フォルダ

(ワーキング)ディレクトリを移動して作業する

cd (change directory)

ディレクトリの移動

pwd (print working directory)

ワーキングディレクトリの表示

ls (list)

(ワーキング)ディレクトリ内のファイルの一覧

mkdir/rmdir (make / remove directory)

ディレクトリの作成 / 消去

エディタ Emacs

Emacs

- Emacs (イーマックス) は歴史の長い代表的なテキスト・エディタである。
- コントロール・キーを使うのが由緒正しいが、とりあえず、メニューや矢印キーを使ってテキストを編集しよう。
- irb でロードする前にセーブを忘れずに。
- 自分の慣れたエディタがあれば、それを使ってもよいです。ただし、プレーン・テキストのファイルが作れないと駄目。

Emacs

- Emacs (イーマックス) の基本的なコマンド。

  File → Visit New File... (C-x C-f)

  　ファイルの読込み，新しいファイルの作成

  File → Save (C-x C-s)

  　ファイルの書出し

  File → Save As... (C-x C-w)

  　(新しいファイル名での)ファイルの書出し

  Edit → Undo (C-_)

  　直前の編集作業(群)の取消

Emacs

1. 理解して使えそう
2. まったく理解できない

-. Emacs 以外のエディタを使う場合

bmi.rb というファイルを作る

```
# BMI of a person with height (cm) and weight (kg)
def bmi(height , weight )
    weight / ( height /100.0) ** 2
end
```

ファイルを作ったら投票

1. 終わった

Emacs

- Emacs (イーマックス) の基本的なコマンド。

  File → Visit New File... (C-x C-f)

  　ファイルの読込み，新しいファイルの作成

  File → Save (C-x C-s)

  　ファイルの書出し

  File → Save As... (C-x C-w)

  　(新しいファイル名での)ファイルの書出し

  Edit → Undo (C-_)

  　直前の編集作業(群)の取消

数の計算と関数(続き)

教科書第1章（15ページから）

ファイルに保存した関数定義 ---ファイルからの読み込み

#以下
行末まで
コメント

```
# BMI of a person with height (cm) and weight (kg)
def bmi(height , weight )
   weight / ( height /100.0) ** 2
end
```

bmi.rb

```
irb(main):001:0> load("./bmi.rb")
=> true
irb(main):002:0> bmi(188.0, 104.0)
=> 29.425079221367138
```

ファイルに保存した関数定義

```
def feet_to_cm(f, i)
   (f + i/12.0)*30.48
end
def pound_to_kg(p, o)
   (p + o/16.0)*0.4536
end
```

yardpound.rb

```
irb(main):003:0> load("./yardpound.rb")
=> true
irb(main):004:0> feet_to_cm(5, 11)
=> 180.34
```

ファイルを読み込むファイル

```
load ("./bmi.rb")
load ("./yardpound.rb")

def bmi_yp(f,i,p,o)
    bmi(feet_to_cm(f,i), pound_to_kg(p,o))
end
```

bmi_yp.rb

```
irb(main):005:0> load("./bmi_yp.rb")
=> true
irb(main):006:0> bmi_yp(5,11,170,0)
=> 23.710342996960538
```

検査プログラム check.rb

- 授業のページから検査プログラム check.rb をダウンロードする（右クリック→「リンク先のファイルを別名で保存…」）
- 授業のページから検査ファイル ex01.rb をダウンロードする（右クリック→「リンク先のファイルを別名で保存…」）
- １つのディレクトリにまとめ、以下を実行する

  ci123456m:~ yama $ ruby check.rb ex01.rb

定数関数

```
# BMI of a person with height (cm) and weight (kg)
def bmi(height , weight )
   weight / ( height/100.0) ** 2
end
def k_height()       #K選手の身長
   188.0
end
def k_weight()       #K選手の体重
   104.0
end
```

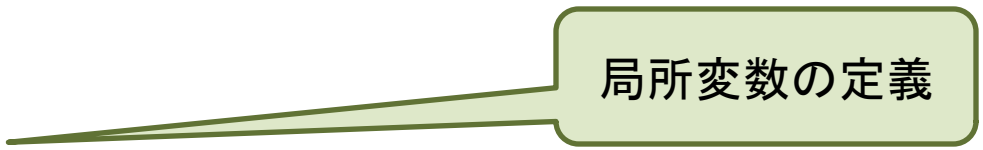
bmi.rb

定数関数

```
irb(main):005:0> load("./bmi.rb")
=> true
irb(main):006:0> k_weight()
=> 104.0
irb(main):007:0> bmi(k_height(), k_weight())
=> 29.425079221367138
```

局所変数

```
include(Math)
def heron(a,b,c)
    s = 0.5*(a+b+c)
    sqrt(s * (s-a) * (s-b) * (s-c))
end
```



heron.rb

練習

- 定数関数 k_height と k_weight も定義されたファイル bmi.rb を作り、ロードして動作を確認せよ。（教科書 17ページ、ファイル 1.6）
- ファイル heron.rb を作り、ロードして動作を確認せよ。1行目に「include(Math)」を入れておくこと。（教科書 11ページ）

進捗状況の確認

1. bmi ができた時点で投票してください。

bmi.rb – 17ページ

heron.rb – 11ページ

できてしまった人は、教科書の練習1.3 さらに練習1.9をやってください。

進捗状況の確認

1. すべてできた
2. bmi だけできた
3. heron だけできた
4. 関数が実行できない（エラーなど）

できてしまった人は、教科書の練習1.3 さらに練習1.9をやってください。