

# コンピュータ概論 A/B (竹澤先生,栗野)

-- mathematica --

数学科 栗野 俊一

2010/10/19 コンピュータ概

# 伝言

---

## □ 補習

- 今日、1221 室で、5 限で、補講を行います

- ▶ Mathematica は本日中でインストールする事!!

## □ 調べておきましょう

- 次のキーワードを **google** で調べておきましょう

- ▶ List 処理

- 便利と思ったページの **URL** は、**skype** で交換しましょう

# 注意

---

## □ 講義前の注意

### ○ 講義開始前にすませておくこと

▶ PC の電源を入れる/ネットワークに接続しておく/今日の資料に目を通す

### ○ 過去ログ(記録)を参照しましょう

▶ 新しいメールが届いていないか確認しましょう

▶ CST Portal のフォーラムの新着の記事に目を通しましょう

▶ skype を起動し、新しいメッセージがないか確認しましょう

▶ ついでに何か発言する習慣を身に付けましょう(挨拶も可)

## □ やる気のある方へ

### ○ 今日の資料は、すでに上っています

▶ どんどん、先に進んでかまいません

# 先週の復習

---

## □ 先週の内容

### ○ 講義

- ▶ Mathematica の変数

### ○ 実習

- ▶ [演習 1] Mathematica の変数
- ▶ [演習 2] 課題の作成

## □ 講義内容

### ○ Mathematica の変数

- ▶ 変数：値(式)を代入する事ができ、後から参照ができる
- ▶ `= / :=` で変数に値を代入する
- ▶ 式中に値が設定されている変数があると、その値に置き換えられる

### ○ ポイント

- ▶ 計算の途中結果を変数に入れ、何度も利用できる
- ▶ 変数に値をいれて行くことにより、環境が変わってゆく
- ▶ 関数の定義もほぼ、変数への代入と同じ形

# 本日の予定

---

## □ 講義

- Mathematica によるプログラミング

## □ 実習

- [演習 1] Mathematica での関数の作成
- [演習 2] List 処理
- [演習 3] 基本変形で rank の計算
  - ▷ 課題はこの結果

# 本日の課題 (2010/10/19)

---

## □ 前回 (2010/10/12) の課題

○ 次のファイルを提出しなさい

▶ 表題 : Mathematica の課題

▶ ファイル名 : 20101005-QQQQ.nb (QQQQ は学生番号)

▶ 詳しくは、配布した sample-20101005.nb の内容を参照

## □ 今週 (2010/10/19) の課題

○ 次のファイルを提出しなさい

▶ 表題 : 基本変形による rank の計算

▶ ファイル名 : 20101019-QQQQ.nb (QQQQ は学生番号)

▶ 内容 : 与えられた行列の rank の計算を途中経過をも含めて計算する

▶ 詳しくは、配布した sample-20101019.nb の内容を参照

# [実習 1] Mathematica での関数の作成

---

## □ Mathematica では関数も定義可能

○ 基本：「関数名[引数のパターン] := 式」

○ 例

▷ `fac[0] := 1`

▷ `fac[n_] := n * fac[n-1]`

○ ポイント

▷ 引数に「\_」を付けるとパターンとなる

▷ 「\_」の前の名前に、一時的にパターンにマッチする値が入る

▷ 一つの式が複数のパターンに一致した場合は、より具体的なパターンの値が使われる

○ 計算の様子

▷ `fac[3]` : 引数が 0 と一致しないので、一般的な `n_` に一致 / `n == 3` となり「`3 * fac[2]`」となる

▷ `3 * fac[2]` : 上と同様「`3 * 2 * fac[1]`」となる

▷ `3 * 2 * fac[1]` : 上と同様「`3 * 2 * 1 * fac[0]`」となる

▷ `3 * 2 * 1 * fac[0]` : `fac[0]` は、両方にマッチするがより具体的な上が選ばれ 1 となる

▷ `3 * 2 * 1 * fac[0]` : `fac[0]` は、両方にマッチするがより具体的な上が選ばれ「`3 * 2 * 1 * 1`」となる

▷ `3 * 2 * 1 * 1` : 計算されて「6」になる

▷ 答 : `fac[3] == 6`

# List 処理

---

## □ List : 何か並び

### ○ List の再帰的定義(帰納的定義)

▷ {} は List

▷ L がリストならば、Prepend[L,X] もリスト

## □ List の関数

### ○ L が {} でないリスト ( {a,b,c} の時 )

▷ First[L] : L の最初の要素 ( First[L] == a )

▷ Rest[L] : L の先頭の要素を取り除いた残り、リスト ( Rest[L] == {b,c} )

▷ L == Prepend[First[L],Rest[L]]

▷ Part[L,n] : L の n 番目を取る ( First[L] == Part[L,1] )

### ○ 詳しくは Mathematica の Help を参照の事

## [実習 2] List 処理関数の定義

---

### □ List を処理する関数定義のパターン ( list.txt 参照 )

- L が {} の処理を考える
- L の先頭の処理と残りの処理を考える

### □ 例 1 : リストの長さを求める

- `lengthOfList[{}]` := 0
- `lengthOfList[list]` := `lengthOfList[Rest[list]]` + 1

### □ 例 2 : 要素を探す

- `findOfMember[list_,member_]` := `findOfMember[list_,member_,1]` (\* 先頭から探す \*)
- `findOfMember[{},member_,pos_]` := -1 (\* みつからなかった \*)
- `findOfMember[list_,member_,pos_]`
  - ▷ := `If[ member == First[list], pos, findOfMember[Rest[list],member,pos + 1]`

# [実習 3] 基本変形

---

## □ 行列の基本変形

- 代数幾何で利用する基本変形を `mathematica` で実現

- ▷ `primitive.txt` を参照

## □ `Mathmatica` を利用して `rank` 計算

- 行の操作関数を作って、`rank` 計算を試みる

- ▷ `sample-20101019.nb` を参照