

# コンピュータ概論 A/B

-- TeX Macro (2) / サクラ Macro --

数学科 栗野 俊一 (TA: 浜津 翔 [院生 2 年])

2014/11/25 コンピュータ概

# 伝言

---

## 私語は慎むように !!

### □ 席は自由です (出席パスワード : 20141125)

- できるだけ前に詰めよう

- 教室にきたら直ぐにやる事

  - ▶ PC の電源 On / ネットワーク接続 / Web を参照する / skype を起動する

### □ 色々なお知らせについて

- 栗野の Web Page に注意する事

<http://edu-gw2.math.cst.nihon-u.ac.jp/~kurino>

- 2014/12/01(来週の月曜日) の 2 限目に「補習」を行います

  - ▶ 場所 : 1214 (この部屋の前) / 時間 : 2 限 / 内容 : 質問等.. / 対象 : 希望者

### □ Local Server

  - ▶ 10.9.209.128 (VNC) : 画面の操作を見ることができます (PW : vnc-2014)

  - ▶ 10.9.209.88 (Web) : 普段のサーバより速いはずです

# 前回(2014/11/18)の内容

---

## □ 前回(2014/11/18)の内容

### ○ ファイルシステムと木構造

▶ Windows のファイルシステムは木構造(但し、ドライブ内)になっている

### ○ パス名 ( ファイルやフォルダの「位置」を表す表現 : アドレス )

▶ 絶対パス名 : ドライブのルート(根)からの経路(パス)で指定 (一通り)

▶ 相対パス名 : 現在の場所(カレントディレクトリ)からの経路で指定 (複数あり)

### ○ TeX マクロ

▶ `\newcommand` を利用して、新しい独自マクロを作る事ができる

# 本日(2014/11/25)の予定

---

## □ 講義

### ○ TeX のマクロ(2)

▶ 引数付きマクロ

### ○ Sakura Editor のマクロ

▶ キーボードマクロの定義と利用

## □ 実習

○ [演習 1] texDviPdf.js のインストール

○ [演習 2] TeX の引数付マクロの作成

○ [演習 3] Sakura Editor のキーボードマクロの作成

# 本日の課題 (2014/11/25)

---

## □ 前回 (2014/11/18) の課題

### ○ 次のファイルを提出しなさい

- ▶ 表題：自分の名前のロゴマクロ
- ▶ ファイル名：20141118-QQQQ.sty (QQQQ は学生番号)
- ▶ 詳しくは、配布した sample-20141118.sty の内容を参照

## □ 今回 (2014/11/25) の課題

### ○ 次のファイルを提出しなさい

- ▶ 表題：自家製サクラマクロ
- ▶ ファイル名：20141125-QQQQ.mac (QQQQ は学生番号)
- ▶ 詳しくは、配布した sample-20141125.mac の内容を参照

# 演習 1: texDviPdf.js のインストール

---

## □ [実習 1.1] texDviPdf.js の入手とインストール

○ 講議のページから texDviPdf.js をダウンロード

▶ 保存先は C:\Users\[自分のID]\AppData\Roaming\sakura

▶ [注意 1] 「自分のID」の所は自分の Windows の LoginID が入るので、人によって異なる(僕の場合は pc が入る)

▶ [注意 2] 『C:\Users\[自分のID]』は「個人フォルダ」なので、そこから「\AppData\Roaming\sakura」としても良い

## □ [実習 1.2] サクラエディタで texDviPdf.js を利用

○ 講議のページから tex-001.tex をダウンロード

▶ 保存先は C:\usr\tex\20141125

○ サクラエディタで tex-001.tex を開く

○ サクラエディタで

▶ [ツール(T)]→[名前を指定してマクロ実行(E)]→[texDivPdf.js]→[開く(O)]

▶ 自動的に typeset, dvipdfmx, acrRd32 が実行される

▶ # 今迄のコマンドプロンプトでの苦行は何だったのか.. ?

# マクロ(Macro)

---

## □ Macro(マクロ) とは

○ Micro(マイクロ) の反対、では Micro は.. ? Macro の反対.. ;-D

▶ というの循環定義(論法)という.. (駄目な定義の典型例..)

○ 再度 Macro の意味 : 「大きな」という接頭語

▶ Macro : 大きい <-> Micro : 小さい

○ でも、何を基準に「大小」を考える .. ?

▶ 最小のモノ(基本命令/基本操作/基本関数)は Micro(マイクロ)

▶ それを「いくつかまとめたもの」→ Micro でない→ Macro

○ 計算機における「Macro」の意味

▶ 基本的なモノ(何が「基本」かはシステムによって異なる)を、いくつかまとめたモノ

▶ Macro 定義 : Macro に名前を付ける(事によって名前で Macro が参照できるようにする)事

# 色々なマクロ(Macro)とその意義

---

## □ Macro の事例

- Mathematica : 幾つか関数の合成 (に名前を付けるのが「関数定義」)
- TeX : 幾つかの文字の並び(に名前を付けるのが「マクロ定義」)
- Sakura Editor : 幾つかの編集操作の並び(今週の話)

## □ Macro の意義 : メタ操作になっている

- システムが決まるという事は、対象物と基本操作が決まるという事
  - ▶ 基本操作を対象とする機能(メタ機能)があれば、高機能化が可能
- メタ操作により「基本機能」が増える (操作効率が向上する)
  - ▶ システムの「拡張」が可能になる → 自分の好みのシステムに変更できる
  - ▶ 他人の「拡張」が「再利用」できる → 更に高々機能化される

## □ Macro の運用 : 「数学」の運用と同じ

- 1 段階 : 他人のマクロが利用できる (「数学」を応用する人)
- 2 段階 : 自分でマクロが作成できる (「数学」を創出する人 : You !!)

# TeX のマクロ機能(再)

---

## □ TeX のマクロ機能 (tex-001.tex)

### ○ マクロ機能

▶ TeX では、「『文字列』に『名前』を付けて、参照する」事ができる

### ○ マクロ名：文字列に付けられた名前

▶ 「名前」は習慣により「\`\`(バックスラッシュ)」で始まる物にする

### ○ マクロ定義：『文字列』に『名前』を付する事

▶ マクロ定義の表現：`\newcommand{新しい名前}{文字列}`

### ○ 例：`\newcommand{\MyJapaneseName}{栗野俊一}`

▶ 文字列「栗野俊一」に、マクロ名「`\MyJapaneseName`」を付けた

▶ マクロ「`\MyJapaneseName`」を、定義した(マクロ定義した)

▶ 文字列「栗野俊一」は、マクロ「`\MyJapaneseName`」の定義内容

### ○ マクロ展開(参照)

▶ 文書内で定義済のマクロ名を記述すると「その定義内容」に置き換わる

▶ 例：「私の名前は`\MyJapaneseName` です」→「私の名前は栗野俊一です」

▶ ※ マクロ名の後ろには「空白(改行でも良い)」を入れること

# 引数付きマクロの定義

---

## □ 引数付きマクロ

- マクロの内容の一部を、「後から指定したい」場合がある

- ▶ ほとんど同じだが、一部だけ異なる文字列を何度も利用したい場合

- ▶ 引数付きマクロによって対応可能

## □ 引数付きマクロの定義(三つのポイント)

- 引数付きマクロの場合は、引数の個数を「[n] (n は個数)」で指定する

- ▶ 例 : `\newcommand{\ThreeArgMacro}[3]{ .. }` : 引数は三つある

- 内容の中に「#1 ~ #n」を含める

- ▶ 例 : 上記で { 最初は #1, 次は #2, 最後が #3 } とする

- マクロを利用する時にマクロ名の後ろに n 個の引数を指定する

- ▶ 例1 : `\ThreeArgMacro{芋虫}{蛹}{蝶}`

- ▶ 例2 : `\ThreeArgMacro{おたまじゃくし}{足が生えてきて}{蛙}`

- ▶ 例3 : `\ThreeArgMacro{ちよろちよろ}{ぱっぱ}{子供が泣いても蓋取るな}`

# 引数付きマクロの作り方

---

## □ 何時、引数付きマクロを作るか (When) ?

- 似ているマクロが出て来た時 ( cf. 連分数を沢山定義してみる .. )

  - ▶ # 「『今』でしょう」とは.. 言わない... :-)

- 初めっから引数付きマクロを作る必要はない

  - ▶ 初めっから複数の似ているマクロを作る場合はある..(その時には最初から..)

## □ どうやって、引数付きマクロを作るか (How to) ?

- 基本は、似ているマクロの統合(抽象化)

  - ▶ 似ているマクロの形を揃えて、「同じ所」と「違う所」が解るようにする

  - ▶ 「同じ所」はそのままで、「違う所」を引数参照 (#n) に変更

  - ▶ 新しい(引数付き)マクロを定義し、それに引数の個数(#n の個数)を追加

  - ▶ 「違う所」はマクロ参照の引数(マクロの後ろに{}で記述..)で指定する

# 実習 2: TeX の引数付きマクロ

---

□[実習 2] 引数付きマクロを作成してみる

○sample-20141125.tex の中に色々な連分数を定義して表示してみる

▶ 参考 : <http://d.hatena.ne.jp/rikunora/20090830>

# サクラエディタのマクロ

---

## □ サクラエディタのシステム

○ 対象物：文字列 / 操作：文字の編集命令(カーソルの移動命令..)

▶ サクラエディタのメタシステム：「操作」を「編集」する

▶ サクラエディタのマクロ：「操作」列

○ [ポイント]

▶ サクラエディタでは、「操作」が編集でき、「実行」できる

▶ サクラエディタでは、マクロを記述する言語が沢山ある(js/mac/wsh/etc..)

## □ キーボードマクロ

○ 命令の並び(編集操作)を、そのまま記録する仕組み

▶ キーボードやマウス操作が「命令列(を表す文字列)」に変換され、保存できる

▶ 「命令列」は(文字列なので..)編集できる (メタ!!)

○ (応用)マクロを直接作らず、キーボードマクロで雛形を作り、それを編集する

# 実習 3: サクラエディタでのマクロ利用

---

## □ [実習 3.1] sample-20141125.mac の利用

- sample-20141125.sty をダウンロードしてサクラエディタで開く
  - ▶ sample-20141125.mac を実行するとどうなるか？

## □ [実習 3.2] キーボードマクロの記録

- 何かファイルを開く
- [ツール]→「キーボードマクロの記録開始(R)」とする
  - ▶ 何か色々操作をする(この内容が記録される)
- [ツール]→「キーボードマクロの記録終了&保存(M)」とする
  - ▶ 保存先は C:\Users\[自分のID]\AppData\Roaming\sakura のままで良い
  - ▶ 名前を 20141125-QQQQ と指定して保存する (自動的に .mac が付く)
- [ファイル]→[開き直す(W)]→[開き直す(W)]
  - ▶ 「変更を破棄するか？」と尋ねられるが構わず [OK] を押す → 元に戻る

## □ [実習 3.3] キーボードマクロの実行

- [ツール]→「名前を指定してマクロ実行(E)」とする
  - ▶ 先刻の変更内容が再度行われた事を確認

## □ [実習 3.4] 課題の提出

- 上記の 20141125-QQQQ.mac を CST Portal に提出
  - ▶ [注意] C:\Users\[自分のID]\AppData\Roaming\sakura にある

# 実習 4: サクラエディタでのマクロ利用(2)

---

## □ [実習 4.1] texDviPdf.js をキーだけで実行

○ 手順: 以下の手順の組み合わせで行う

▶ 外部マクロの登録する

▶ ショートカットキーを割り当てる

## □ [実習 4.2] 外部マクロの登録する (texDviPdf.js を登録)

○ [設定]→[共通設定]→[マクロ]

▶ [名前] 好きな名前 (例えば tex2pdf)

▶ [File] texDviPdf.js を指定する

○ [設定]→[OK] で、登録

## □ [実習 4.3] ショートカットキーの割り当てる(Ctrl+Shift+Z)

○ メニュー[設定]→[共通設定]→[キー割り当て]

○ [種別]を[外部マクロ]にする

○ キー(Ctrl+Shift+Z)と割り当てたいマクロ名(tex2pdf)を選ぶ

▶ キーは「未割り当て」なものであれば、何でも OK

○ [割付]→[OK]を押して完了

## □ [実習 4.4] 試してみる

○ tex ファイルを開いて [Ctrl]+[Shift]+[Z] を押してみる