

コンピュータ概論 A/B

-- マクロ --

数学科 栗野 俊一 (TA: 佐藤 淳 [院生 1 年])

2015/12/22 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 : 20151222)

○ 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>

□ 本日 (2015/12/22[火]) 「補習」を行います

○ 場所 : 1211 (この部屋) / 時間 : 5 限 / 内容 : 質問等.. / 対象 : 希望者

□ 次回 (2016/01/12)

○ 教室変更があります : 1211 → 1221 (二階)

今後の予定

□ 今後の予定(後ろから)

○ 2015/01/26 (講義最終日)

▶ 試験を行う

○ 2015/01/19 (試験前)

▶ 模擬試験を行う

○ 2016/01/12(講義最終日)

▶ Power Point とプレゼンテーション

▶ 教室変更があります : 1211 → 1221 (二階)

○ 2015/01/05, 2014/12/29

▶ 冬期休暇期間中

○ 2014/12/22

▶ マクロ

▶ 5 限目に補習を予定

前回(2015/12/15)の内容

□ 前回(2015/12/15)の内容

○ 講義

- ▶ Excel の応用と TeX との連携
- ▶ 他のソフトで作成した情報(表・グラフ)をファイルに保存
- ▶ そのファイルを参照して、TeX が資料(ドキュメント)を整形する

○ <<ポイント>>

- ▶ ファイルに保存できれば、どのソフトで作成したかとは無関係
- ▶ (形式さえ整っていれば..) インターネット経由でダウンロードしたファイルも利用可能
- ▶ (もちろん..) MS-Word でも同等な事はできる

○ 実習

- ▶ Excel によるグラフの作成
- ▶ 「ペイント」による、画像ファイル(jpeg形式)の作成
- ▶ Excel マクロ「Excel2LaTeX.xla」による、表ファイル(TeX形式)の作成
- ▶ TeX による、ファイルの取り込み

本日(2015/12/22)の予定

□ 本日(2015/12/22)の予定

○ 講義

▶ マクロ

○ 実習

▶ 実習 1: サクラエディタでのマクロ利用

▶ 実習 2: TeX でのマクロ利用

本日の課題 (2015/12/22)

□ 前回 (2015/12/15) の課題

○ 次のファイルを提出しなさい

▶ 表題 : TeX と MS-Excel の連携

▶ ファイル名 : 20151215-QQQQ.pdf (QQQQ は学生番号)

▶ 詳しくは、配布した sample-20151215.tex, sample-20151215.xlsx の内容を参照

□ 今回 (2015/12/22) の課題 (二つあるので注意)

○ 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 : 20151222-QQQQ.mac (QQQQ は学生番号)

▶ 詳しくは、配布した sample-20151222.mac の内容を参照

○ 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 : 20151222-QQQQ.sty (QQQQ は学生番号)

▶ 詳しくは、配布した sample-20151222.sty の内容を参照

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(を記録するのが「マクロ定義」)

□ Macro の意義 : メタ操作になっている

- 「システムが決る」という事は、「対象物と基本操作が決る」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 !!)

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 1: サクラエディタでのマクロ利用

□ [実習 1.1] sample-20151222.mac の利用

- sample-20151222.sty をダウンロードしてサクラエディタで開く
- 更に、サクラエディタで、以下をすると、どうなるか？
 - ▶ [ツール(T)]→[名前を指定してマクロ実行(E)]→[sample-20151222.mac]→[開く(O)]

□ [実習 1.2] 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 20151222-YYYY 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 1.3] 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 1.4] 課題の提出

- 上記の 20151222-YYYY.mac を CST Portal に提出

TeX の復習

□ pLaTeX システムとは(What)

- tex 形式の文章ファイル(*.tex)を typeset(整形)して dvi 形式ファイルを作るソフト
 - ▶ dvi 形式のファイルは更に dvipdfmx で pdf 形式に変換できる
 - ▶ 一度、pdf 形式にすれば、表示も印刷も綺麗にできる

□ TeX を利用する理由(Why)

- 印刷物(「内容」と「形式」の二つの情報から成る)の作成を「内容」だけに専念したい
 - ▶ 「形式」は、TeX が対処してくれる(形式は style ファイルの形で別に記述されているので、その御仕着せに頼る)
- tex 形式は基本 Text 形式である
 - ▶ 「編集」を「Text Editor(サクラ)」で行いたい / 高速にかつ汎用で、しかも、手慣れている
 - ▶ pLaTeX がなくても、ある程度、中身が理解できる (e-mail での利用)
- 色々なシステムで動く
 - ▶ MS-Windows は勿論、Linux (栗野愛用)や、MacOS でも利用可能
- 数式を使いたい
 - ▶ 数式を扱いたければ、(残念な事に、未だに..) 一番便利

実習 2: pLaTeX の利用 (復習)

□[実習 2] pLaTeX の利用

○ tex ファイルの作成 (tex-000.tex) : Text エディタ(サクラエディタ)を使う

- ▶ 拡張子は「tex」にする
- ▶ sample file をダウンロードしてもよい

○ platex による typeset

- ▶ ubuntu を利用して「platex tex-000.tex」とする
- ▶ 問題がなければ、「tex-000.dvi」(dvi 形式ファイル)が作られる

○ dvipdfmx による pdf 形式への変換

- ▶ ubuntu を利用して「dvipdfmx tex-000.dvi」とする
- ▶ 問題がなければ、「tex-000.pdf」(pdf 形式ファイル)が作られる

○ pdf ファイルの表示

- ▶ ubuntu を利用して「evince tex-000.pdf」とする
- ▶ windows で tex-000.pdf を「開いて」もよい

TeX のマクロ機能

□ 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`」をつけた

▶ 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)

▶ マクロ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」

Style ファイルとマクロの分離

- マクロは何度も使い回す事が多い
 - 毎回同じ事を書くのは面倒 / 一度作成した内容を使い回す
 - ▶ マクロ定義だけを記述した **Style** ファイルを作成して読み込む (`\input`)
- マクロ集としての **style** ファイル
 - TeX の特徴の一つ : 内容と形式が分離できる (内容だけに専念できる)
 - ▶ 内容は `.tex` ファイルに記述。では、形式は .. ? 実は **style** ファイルに
 - 色々な **style** ファイル
 - ▶ 中に、色々なマクロが定義されている
 - ▶ 内容で、同じ名前のマクロを利用しても、マクロの内容が異れば、異なる結果になる
 - ▶ 「形式」の情報を「マクロ」にしておくことにより、形式と内容を分離できる
 - `\usepackage` マクロ : 実は、**style** ファイルを読み込んでいる

pLaTeX vs TeX

□ pLaTeX vs TeX

○ 今迄、pLaTeX は TeX の一種と言っていたが.. ?

▶ 実は、pLaTeX は TeX そのもの / 単に、pLaTeX 専用のマクロが事前に定義されて(読み込まれている..)だけ..

○ じゃあ、なぜ別の名前 ?

▶ マクロの量が多く、また、利用形式が独特なので、まるで違ったように見える

▶ 違う物は違う名前の方が混乱がすくない(cf. 役者はドーランを塗ったら別人)

色々な TeX マクロ

□ 表示位置を変更するマクロ

○ 文字の位置を上にする

▷ `\raise<サイズ>\hbox{<文字列>}`

▷ 例: `\raise.4ex\hbox{野}`: 「野」の表示位置を x の高さ(ex)の 0.4 倍だけ上に

○ 文字の位置を下にする

▷ `\lower<サイズ>\hbox{<文字列>}`

▷ 例: `\lower.2ex\hbox{俊}`: 「俊」の表示位置を x の高さ(ex)の 0.2 倍だけ下に

○ 文字と文字の間隔を調整する

▷ `\kern<サイズ>`

▷ 例: `俊\kern.1em一`: 「俊」と「一」の間を m の幅(em)の 0.1 倍だけ広げる

▷ サイズに負の数を指定すると幅を狭める事もできる

実習 3: TeX マクロ

□[実習 3.1] マクロを利用する

- `tex-001.tex` の `\MyJapaneseName` の定義を変更して試してみる

□[実習 3.2] style ファイルの作成

- `tex-001.tex` の `\MyJapaneseName` の定義を切り取り、別のファイル(`myinfo.sty`) に保存

- `tex-001.tex` のプリアンブルに `\input{myinfo.sty}` を追加

▶ `tex-002.tex` を DL してもよい

□[実習 3.3] 名前ロゴの作成

- `tex-003.tex` を参考にロゴを作るマクロを作成する

□[実習 3.4] 引数付きマクロの作成

- `tex-004.tex` を参考に引数付きマクロを作ってみる

□[実習 3.5] 課題の作成

- 自分ロゴのマクロを作る

▶ `sample-20151222.tex`, `sample-20151222.sty` を参考に `sample-20151222.sty` を作る

▶ `sample-20151222.tex` を `typeset` して、きちんと、pdf が自分の情報になっている事を確認

引数付きマクロの定義

□ 引数付きマクロ

- マクロの内容を一部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}{子供が泣いても蓋とるな}`

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 4: TeX の引数付きマクロ

□[実習 4] 引数付きマクロを作成してみる

○sample-20151222.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>

メタ, 関数定義, マクロ定義

□ 計算機操作におけるメタ

○ ファイル(対象), ソフト(操作)

▶ メタは?: インストール / プログラム作成

□ プログラミング

○ 計算機操作におけるメタの典型

▶ Mathematica の関数 / TeX のマクロ

□ コンピュータ概論の*裏*目標

○ 「計算機に使われる」のではなく「計算機が使える」人材の育成

▶ ソフトを使うだけの人 (実は、ソフトに使われている人: ×)

○ プログラミングができる人

▶ ソフトを作る人: つまり、「計算機が使える」人

○ 講義では「概念」だけを述べている

▶ 自分で色々試してみよう (結局、何時ものセリフ)