

第 1 章

Introduction

1.1 準備

1.1.1 この講義の目的

大学生活において、さらには社会人生活において、何かを的確に表現することは非常に重要と
 なってくる。この講義では以下の 3 項目を主目的として進めていく。

1. 情報化社会で必要となる知識を身につけ、活用させる能力を修得すること。
 レポート作成や卒業研究に必要となるツール (LaTeX) の利用方法を学習する。
2. 中学・高校の教員になって役立つ情報を得ること。
 GeoGebra を利用した視覚的な中学・高校数学の学習し、実習を行う。
3. 2 年次以降の講義で必要となる予備知識を身につけること。
 プログラミング言語の導入として、簡単なプログラミングの仕組みについて Scratch の実
 習を行う。

1.1.2 ディレクトリ構造

コンピューターにおいて、ファイルの管理は、ディレクトリ (≡ フォルダ) の階層構造で行わ
 れている。Windows パソコンならエクスプローラーから確認ができる。

Windows 10 なら検索 () からエクスプローラーと入力 (図 1) すると、表示さ
 れる (図 2) エクスプローラーを起動する。

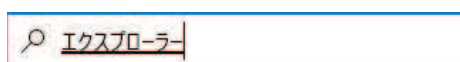


図 1



図 2

なお、応用数学科 計算機室のパソコンではツールバーにあるエクスプローラー    を起動する。

エクスプローラーでディレクトリとファイル見ると、右の図のような階層構造をしている (一部のみ掲載)。

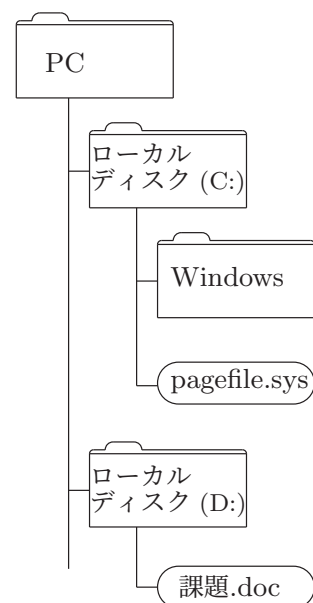
PC、デスクトップや、ローカルディスク (C:) というディレクトリがあり、それぞれのディレクトリには (サブ) ディレクトリやファイルが収納されている。

♠ **重要** 応用数学科 計算機室のパソコンでは、PC の下層ディレクトリとして、

PC の下層ディレクトリ

ローカルディスク (C:)

qemu(¥¥172.16.2.2) (D:) (*)



があることに注意する。今後、

『ローカルディスク D』, 『D ドライブ』, 『ボリューム D』

と言ったりすることがあるが、いずれもここにある (*) のことをさしている。

また、ローカルディスク (C:) にファイルを保存した場合、次回起動時には消えているので、必ず (*) に保存すること。

課題 1.1.1. 応用数学科 計算機室のパソコンを使用している場合は、D ドライブ (自分のパソコンの場合は C ドライブ) の直下にサブフォルダ files を作成し、files のサブフォルダとして

scratch、tex、geogebra

の 3 つのフォルダを作成せよ。

サブフォルダの作成は、エクスプローラーの左側のウィンドからサブフォルダを作成したいフォルダをマウスで右クリックし、新規作成からフォルダを選ぶ。

1.1.3 ファイル名と拡張子

この図にあるファイル `pagefile.sys` や `課題.doc` の ドット (.) 以前をこのファイルのファイル名、以降をファイルの 拡張子 という。

したがって、課題.doc のファイル名は 課題 で、拡張子は doc である。

拡張子には “ファイルの種類を識別する” という重要な役割がある。拡張子の例として

.sys .exe .com .bat .txt .html .doc .jpg .c .JAVA .mp4 .wma .tex

などなど沢山ある。

Windows パソコンを扱っていると拡張子を見る機会が減っているが、無くなったわけではなく、隠れているだけである。拡張子を確認したい場合は、エクスプローラーの設定を変更したり、コマンドプロンプトを利用して確認出来る。

♡ 注意! Windows パソコンでは 基本、ドット (.) の左側がファイル名、右側が拡張子なので、課題 3.2.1.tex や HandM.tar.gz のようなファイルを作成しないこと。

♡ 注意! また、ファイル名には空白を使わないことを推奨する。アプリケーションに依存するが、空白があるファイルを正しく扱えないことが多い。

1.1.4 ショートカット

パソコンでの作業でマウスを使う場面は多いが、文章の入力中にマウスに手を持って行くのは手間である。そこで、いくつか便利なショートカットキーを紹介しておく。

ただし、これらは Microsoft Windows の場合であり、Mac OS では異なる。

[Shift] + ↑ ↓ ← →	文字の“センタク”
[Ctrl] + ← →	“カタマリ”の前後まで移動 ([Shift] 併用可)
[Ctrl] + a	表示されているモノ全てを“センタク”
[Ctrl] + c	“センタク”したものを“キオク”する
[Ctrl] + s	編集中のファイルを上書き保存する
[Ctrl] + v	“キオク”したものを貼り付ける
[Ctrl] + x	“センタク”したものを削除して“キオク”する
[Ctrl] + z	アンドゥ (取り消し) をする
[Ctrl] + Tab	タブの切り替え
[Alt] + Tab	ウィンドウの切り替え
[Ctrl] + r	置換のウィンドウを開く
[Ctrl] + f	検索のウィンドウを開く
[Ctrl] + n	新規ウインドウ, 文章を開く
[Ctrl] + w	ウインドウを閉じる ([Alt] + F4)
[Home]	カーソルキーのある行の先頭に移動する
[End]	カーソルキーのある行の終端に移動する

♠ 補足 (挿入モード、上書きモード) ショートカットキーではないが、入力方法では“挿入”と“上書き”の2つのモードがある。これらの切替は [Insert] キーまたは [Ins] キーで行う。

ab|cd の状態 (|はカーソルキーの位置) で、“挿入”モードのとき e を入力すると abe|cd となり、“上書き”モードのとき e を入力すると abe|d となる。

例 1.1.1. 以下の (A) の状態からショートカットキーを用いて (B) の状態に編集する。ただし、カーソルキーの初期位置は 789 の 7 の前とする。

(A) 789 456 123

(B) 123 456 789

このときの入力は以下の通りとなる。

```

[End]
[Shift] + [Ctrl] + ←
[Ctrl] + x
[Ctrl] + ← ←
[Ctrl] + v
[Shift] + [Ctrl] + →
(Ctrl を離して) [Shift] + ←
[Ctrl] + x
[Ctrl] + → →
[Ctrl] + v

```

この例では文字数が少ないので再度入力した方が楽かもしれないが、コピー&ペーストやカット&ペーストが多い場合は非常に作業が楽になる。

課題 1.1.2. 以下の文章を TeXworks で入力し、課題 1.1.1 で作成した tex フォルダの中にファイル名 A-2.txt で保存せよ (拡張子に注意)。

```

abcdefg hijklm nopqrst uvwxyz ABCDEFG HIJKLM NOPQRST UVWXYZ

ABCDEFGH hijklm nopqrst uvwxyz abcdefg HIJKLM NOPQRST UVWXYZ
ABCDEFGH HIJKLM nopqrst uvwxyz abcdefg hijklm NOPQRST UVWXYZ
ABCDEFGH HIJKLM NOPQRST uvwxyz abcdefg hijklm nopqrst UVWXYZ
ABCDEFGH HIJKLM NOPQRST UVWXYZ abcdefg hijklm nopqrst uvwxyz

ABCDEFGH uvwxyz ABCDEFG uvwxyz ABCDEFG uvwxyz ABCDEFG uvwxyz
HIJKLM nopqrst HIJKLM nopqrst HIJKLM nopqrst HIJKLM nopqrst
NOPQRST hijklm NOPQRST hijklm NOPQRST hijklm NOPQRST hijklm
UVWXYZ abcdefg UVWXYZ abcdefg UVWXYZ abcdefg UVWXYZ abcdefg

```

課題 1.1.3. 課題 1.1.2 で作成した A-2.txt を、メールに添付し yoshiyuki-mori@ous.ac.jp まで送ること。なお、送信は sXXmYYYab@ous.jp のアドレスからとする。

・ 件名は、学生番号 (必ず半角) 半角スペース 1 つ 課題 半角スペース 1 つ 課題番号 の順番で書き (今後も厳守!)、

例) S25M999 課題 A.

・ 本文は空欄でよい。

1.2 ギリシャ文字、記号、文字コード

1.2.1 ギリシャ文字

ギリシャ文字は複数の書体があるが、ここでは、一般的な (数学で用いられる) 文字について紹介しておく。

大文字	小文字	読み方	大文字	小文字	読み方
A	α	アルファ	N	ν	ニュー
B	β	ベータ	Ξ	ξ	クシー, クサイ, グザイ
Γ	γ	ガンマ	O	o	オミクロン
Δ	δ	デルタ	Π	π, ϖ	パイ
E	ϵ, ε	イプシロン	P	ρ, ϱ	ロー
Z	ζ	ゼータ, ツェータ	Σ	σ, ς	シグマ
H	η	イータ	T	τ	タウ
Θ	θ, ϑ	シータ	Υ	υ	ウプシロン
I	ι	イオタ	Φ	ϕ, φ	ファイ
K	κ	カッパ	X	χ	カイ
Λ	λ	ラムダ	Ψ	ψ	プサイ
M	μ	ミュー	Ω	ω	オメガ

1.2.2 記号

この講義で使われている記号について、読み方を紹介しておく。

記号	読み方	記号	読み方
	パイプライン、縦棒	*	アスタリスク
^	ハット	{ }	波カッコ, 中カッコ
/	スラッシュ	\	バックスラッシュ
"	ダブルクォーテーション	.	ドット、ピリオド
'	シングルクォーテーション	,	カンマ, コンマ
:	コロン	;	セミコロン
-	アンダーバー, アンダースコア	@	アットマーク
•	ナカグロ, 中点	!	エクスクラメーションマーク
~	チルダ	#	シャープ, いげた
¥	エンマーク	\$	ドルマーク

1.2.3 文字コード

コンピュータで文字を扱う場合、文字そのものを扱うのではなく、文字を表す“文字コード”を用いる。例えば、1Byte で表すことが出来るものとして、ASCII コードと呼ばれる文字コード (0 から 127 まで) があり、A が 65, B が 66, ..., Z が 90 で、a が 97, b が 98, ..., z が 122 となっている。

その他、! が 33, " が 34, # が 35 など定義されている。

	0	1	2	3	4	5	6	7	8	9
0	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT
10	LF*	VT	FF*	CR	SO	SI	DLE	DC1	DC2	DC3
20	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS
30	RS	US	SP	!	"	#	\$	%	&	'
40	(	)	*	+	,	-	.	/	0	1
50	2	3	4	5	6	7	8	9	:	;
60	<	=	>	?	@	A	B	C	D	E
70	F	G	H	I	J	K	L	M	N	O
80	P	Q	R	S	T	U	V	W	X	Y
90	Z	[	\	]	^	_	`	a	b	c
100	d	e	f	g	h	i	j	k	l	m
110	n	o	p	q	r	s	t	u	v	w
120	x	y	z	{		}	~	DEL		

1.2.4 バックスラッシュと円マークについて

文字コードの 92 にある \ (バックスラッシュ) は日本では ¥ が用いられるが、文字コードでは \ が表示される。逆に、メーラやエディタでは、¥ が \ と表示される場合がある。

Windows PC の日本語キーボードでは右 [Shift] キー周辺の [ろ] のキーに \ があるが、押すと ¥ となる。

♡ 注意! MacOS を使用している場合は注意が必要で、普通に入力した ¥ は正しく動作しない ([¥] は ¥ として機能しない) ことがある。

MacOS では [option] キーを押しながら [¥] キーを押して表示される \ を使う。

なお、全ての MacOS と、エディタが上記の仕様となっているかは不明なので、最初に使う時に確認してみてもらいたい。

1.3 正規表現

正規表現を用いると検索、置換が容易に行える。詳しく説明することは避けるが、こういった方法があることを知っておけば、必要となったとき調べることが出来るであろう。

正規表現を用いて検索、置換を行う場合に特殊な文字として **メタ文字** というものがある。検索用と置換用では使用が異なるものが多いので、検索用の一部を紹介しておく。

検索用メタ文字 (の一部)

.	任意の 1 文字
*	直前のパターンの 0 回以上繰り返し (最長一致)
+	直前のパターンの 1 回以上繰り返し (最長一致)
?	直前のパターンの 0~1 回繰り返し (最長一致)
*?	直前のパターンの 0 回以上繰り返し (最短一致)
+?	直前のパターンの 1 回以上繰り返し (最短一致)
??	直前のパターンの 0~1 回繰り返し (最短一致)
<i>A 列 B 列</i>	の左右の文字列のいずれか (A 列または B 列)
[文字列]	文字列のいずれか 1 文字 (例: [abc] や [0-9])
(文字列)	置換後へ継承。継承する場合は ¥1 などとする。(例 2-2)
¥	直後のメタ文字をエスケープする (直後の文字を正規表現のメタ文字として扱わないことを指定。 例えば、() や [] なら ¥(¥) や ¥[¥] のようにする)
¥n	改行文字
¥d	すべての半角数字
¥D	半角数字以外すべて
¥w	すべての半角英数字とアンダースコア
¥W	半角英数字とアンダースコア以外すべて
¥l	半角英小文字すべて
¥L	半角英小文字以外すべて (英大文字、数字、全角文字など)
¥u	半角英大文字すべて
¥U	半角英大文字以外すべて (英小文字、数字、全角文字など)
¥n	改行文字
¥s	空白 (スペース)

♡ 注意! エディタによって正規表現を使える、使えないがあり、使える場合も形式が異なっていることがある。

例 1.3.1. (正規表現を用いた置換の例 1)

A regular expression, regex or regexp is, in theoretical computer science and formal language theory, a sequence of characters that define a search pattern. Usually this pattern is then used by string searching algorithms for “find” or “find and replace” operations on strings.

という文章に対して、正規表現の置換で `¥s..¥s` を `_(__)_` に変換させると

A regular expression, regex (__) regexp is, (__) theoretical computer science and formal language theory, a sequence (__) characters that define a search pattern. Usually this pattern (__) then used (__) string searching algorithms for “find” (__) “find and replace” operations (__) strings.

となる。これは空白と空白の間に 2 文字あるものを見つけ変換している。従って、*is*, が変換されていないことが解る。(`_` は半角スペースを意味する。)

例 1.3.2. (正規表現を用いた置換の例 2)

one, two, three, four, five, six, seven, eight, nine, ten.

に対して、`¥s...([,.] )` を `____¥1` に変換すると

____, ____, three, four, five, ____, seven, eight, nine, ____.

となる。これは空白 '¥s' に続く 3 文字があり、その後に , または . があるものを検索している。また、() で囲むことによって置換した時に継承している。すなわち ¥1 は検索でヒットした文字が , なら , を継承し、. なら . を継承して表示している。

例 1.3.3. (正規表現を用いた置換の例 3)

$f(x,y)=6x+y-3$, $g(x,y)=x+y$ に対して、 $g(0,0)=-3$, $g(1,0)=3$, $g(0,1)=-2$

と入力した時点で、後半は g でなく f であることに気付いた。ここで、`g¥((¥d),(¥d)¥)` を `f(¥1,¥2)` に置換すると

$f(x,y)=6x+y-3$, $g(x,y)=x+y$ に対して、 $f(0,0)=-3$, $f(1,0)=3$, $f(0,1)=-2$

となる。¥d は数字なので、 x や y は (すなわち $g(x,y)$ は) ヒットしない。また、() はメタ文字なので、検索するためには ¥が必要となる。継承するものが 2 つあるので ¥1 と ¥2 を使う。

数字が 2 桁以上のものを含む場合は ¥d+ とする。

したがって、上記の場合 $g(10,20)$ などはヒットしない。