

情報科学演習: 第 3 回

コンピュータの構成, 2 進法, データの表現

講義の補足

この講義の Web page を作成しました.

<http://www.math.u-ryukyu.ac.jp/~suga/joho/2021/index.html>

です. 講義資料を, HTML 化して置いてあります. 講義の後に, 間違い部分を修正して載せていきます. PDF も置いてありますので, 復習する際には好みの方をご利用下さい .

課題のレポートメール

受け取ったレポートメールには, 私から返事を書いています. Web Mail で確認して下さい. さらに,

<http://www.math.u-ryukyu.ac.jp/~suga/joho/2021/reports/index.html>

に, 受け取った人の学籍番号を載せていきます. 更新は, 平日の 17:00 頃にします. 課題メールを送ったのに, このリストにないと言う人は, まず Web Mail の送信簿を確認してください. 送信済みであることが確認できたら, 改めて, suga@math.u-ryukyu.ac.jp まで連絡してください. 「下書き」に残ってたら, それは送信されていませんので, 改めて送信の操作をしてください.

メールの補足

琉球大学のメールアカウントのような「公式なメール (あるいは PC のメール)」は, 携帯電話で利用するメールとは, 扱われ方が違います (配送や読む技術が全く同じなのは, 前回の資料で述べた通り). 一番大きな違いは, 次です.

メール到着通知は基本来ない 携帯電話と違って, メールが届いても, PC に通知されません. そのような仕組みを用意することもできますが, そうしていない事も普通です.

基本的に, 電話と書面の中間の媒体であると考えて下さい.

上のことから, 電話のような即時性はありませんので, 急ぎの用事には不向きです. しかし, 記録が残ったり, 即時性が要求されない落ち着いた対応ができるものです.

このような特性を理解して, メールを利用して下さい.

1 コンピュータの仕組み

今回は、コンピュータがどのようなものかの概観を述べます。高校の時の情報科の授業で、学習したことがあると思いますが、その復習でもあります。

Computer という英語のそもそもの意味は、計算機あるいは計算する人という意味です。

計算機、すなわち、計算するための道具であれば、そろばん、計算尺、機械式計算機などがあります (ありました)。現実には、これらを用いて計算をすることは、ほとんどなくなって来ています。また、電卓に対応する英語は、Calculator です。現在では、コンピュータと言えば、いわゆる PC (Personal Computer) とその親戚の事を指します。この講義でも、そのようなコンピュータについての解説を行います。講義の目標は、使い方を知るといよりも、付き合い方を知るという方向です。

PC は personal computer の略ですが、パーソナルでない (個人利用を想定していない) コンピュータも多量にあります。典型的な例は、スーパーコンピュータ (super computer) です。また、携帯電話やタブレット端末も、この講義で扱うコンピュータと同じものです。これらを用いて、Web を見たり、電話したり、音楽を聴いたり、ゲームをしたりしますが、これらは、現実には「計算」を実行した結果なのです。ここでの計算は、足し算や掛け算だけではない、少し広い意味での計算を意味します。

この講義で扱うコンピュータは、次の節の特徴を持つものです。次世代の計算法の候補として、「量子計算」がありますが、量子計算の事も述べません。

今のコンピュータの仕組みを、歴史的な発展とともに、比較的詳しく解説してある書籍として、次があります。

Raspberry Pi で学ぶコンピュータアーキテクチャ、

E. Upton, J. Duntemann, R. Reberts, T. Mamtara, B. Everard 著、

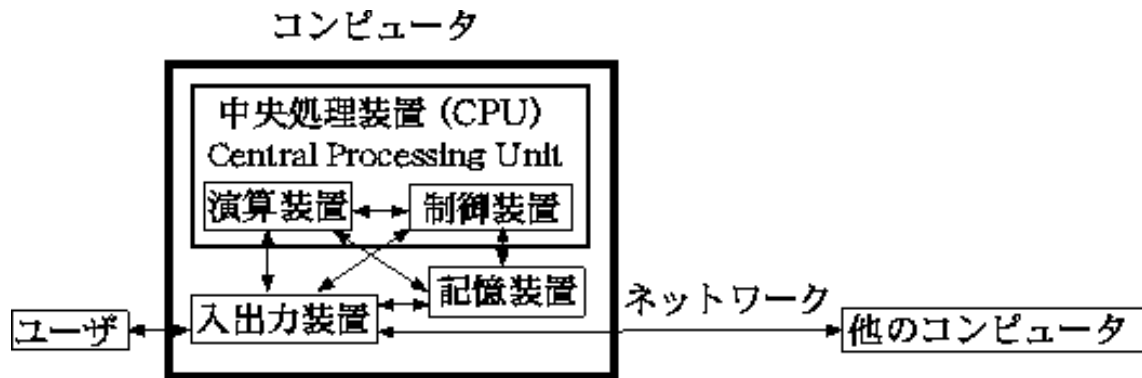
宮下健輔、坂下秀監訳、株式会社クイープ訳、オライリー・ジャパン、2019 年

500 ページを超える大著で、値段も 4200 円しますが、歴史を含めて、今のコンピュータの成り立ちを丁寧に解説してあります。Raspberry Pi とは、タブレット端末からタッチパネルを取り去ったような教育用コンピュータで、それに、電源やキーボード、マウス、モニタ (地デジテレビをモニタにも使える) を足すと使える、5000 円くらいの手のひらサイズの機械です (とは言え、この講義の内容は、全て簡単に実行できます)。上の本は、Raspberry Pi はあくまでも例として利用しており、今のコンピュータがどのようなもので作られているかが、詳しく書かれています。歴史の部分を読むと、例えば、初代ウルトラマンの科学特捜隊本部のコンピュータが、あのように描かれている理由も、少しわかります。

興味のある方は、何かの機会に、上の本を読んでみてください。Raspberry Pi (ラズベリーパイ) は、日本では、ラズパイと略されてます。こっちも、ネットで調べてみてください。

2 コンピュータはどのような構成になっているか

皆さんが使うコンピュータは、大ざっぱに言って図のような構成になっています。



入出力装置 データを入力し出力するための装置です。これを通じて人がコンピュータとやり取りします。この講義では、入力装置としてキーボードとマウスを使い、出力装置はディスプレイを使います。

中央処理装置 (CPU) データの処理を実際に行うコンピュータの心臓部です (制御装置と同じパッケージになっていることが多い)。CPU の以下の 4 つのステップを電源の入っている限り繰り返します。

1. 命令を主記憶から読み出す
2. 命令を解釈する
3. その命令を実行する
4. 割り込み (緊急を要する処理) の有無を調べ、あれば受け付ける

記憶装置 データを処理のために蓄えておく場所です。蓄えられている時間によって 1 次記憶装置 (主記憶装置、電源を切れば記憶した内容が消える) と 2 次記憶装置 (補助記憶装置、電源を切っても記憶した内容は消えない) に分類されます。1 次記憶装置 (主記憶装置) には、半導体メモリが主に使われ、2 次記憶装置 (補助記憶装置) には磁気ディスク (ハードディスク) が使われることが多いのですが、最近、様子が変わってきました。少し前までは、2 次記憶装置は、多くの場合「ディスク (Disk)」, 「ドライブ (Drive)」, 「ディスクドライブ (Disk Drive)」と呼んでいましたが、今では、「ストレージ (Storage)」という言葉が使われつつあるようです。電源を切ってもデータを保持し続けることができる、大容量半導体装置が開発され、市場に流通するようになったからです。

今のコンピュータの特徴付けは、だいたい次のようになります。

- 電子式: 半導体素子を使った電氣的なスイッチを利用してる。
- デジタル: 2 進法を用いた有限桁数の数で計算している。
- プログラム内蔵方式: プログラムを取り替える事が簡単。
- 逐次処理: 基本的なところでは 1 つ 1 つを順に処理しており、多数の同時進行はできない。

2.1 コンピュータと同じ働きをしているもの

身近なもので、上のコンピュータと(ある意味で数学的に)同じ働きをしているものは、「自動販売機」です。

自動販売機は、お金の投入口という入力と、商品を出す出力部分があります。動作としては、入力された金額を記憶していき、販売する商品の金額を超えた時点で、商品と、お釣りを出力します。つまり、入力の応じた適切な処理を実行して出力をするわけです。定期的に人間が、売り上げの回収や商品の補充というメンテナンス(機能維持のための作業)をするのもコンピュータと同じです。上のような動作を、商品販売ではなく、事務作業やゲームなどの方向に進化させたのが、コンピュータです。上のような動作は、オートマトン (automaton) という概念で抽象化され、数学基礎論と呼ばれる数学の分野と関係します。

自動販売機とコンピュータの大きな違いは、一つのコンピュータが事務作業機にも、計算機にも、ゲーム機にも、電話機にもなるところです。自動販売機は、例えば、飲料の自動販売機をチケットの自動販売機に変えようとなると、大変なことになります。コンピュータでは、使うソフトウェアを交換するだけで、その操作時間もあまりかかりません。この部分が、今のコンピュータの飛躍的な発展に繋がった部分です。これを実現するのが、前ページの最後の方に書いてある「プログラム内蔵方式」です。von Neumann(フォン ノイマン) という人が、このアイデアを最初に発表したため、フォン ノイマン型と呼ばれています(実際は、米国で最初に実用的に使われた ENIAC というコンピュータを開発した、エッカート、モークリーあたりがこのアイデアの出所らしい)。

2.2 ハードウェアとソフトウェア

コンピュータの実体を構成する物理的な物体を、ハードウェアといいます。具体的には、CPU、記憶装置等を含むコンピュータ本体、入力装置のキーボード、マウス等、出力装置のディスプレイ、プリンタ等です。

ソフトウェアとはデータ処理の手順の総称をいいますが、マウスやキーボードからの入力を受け付けて、その内容を判断し、別のアルゴリズムに渡すなど、ハードウェアを直接制御するソフトウェアはとくにオペレーティングシステム(OS, 基本ソフトウェア)と呼ばれます。この講義が使う予定であった実習室では、Windows 10 と Linux のディストリビューションの 1 つである CentOS が使えます。情報処理センターでは、これ以外にも macOS X が導入されており、共通教育棟の 2-200 では、macOS X も使えます。

更に、アプリケーションソフトウェア(応用ソフトウェア)があります。この分類も、正確な定義があるわけではありませんが、ある目的のために作られた特別なプログラムという意味で使われます。通常 OS を通してハードウェアを使い処理を進めていきます。(昔の)Play Station で言うと、本体の中身に入っているソフトウェアが OS で、ディスクの中に入っているのがアプリケーションです。

2.3 コンピュータ内のデータその 1: 2 進法と浮動小数点

今のコンピュータを実現しているのは、多量のスイッチ (任天堂のゲーム機ではなく、電流を流したり切ったりするやつ) です。現在は、そのスイッチは半導体素子を用いて実現されていますが、過去には、電磁石でスイッチの開閉をする「リレー式計算機」も作られました。今の 1 台のコンピュータの中にあるスイッチの個数は、億単位です。

電気をを用いた計算では、電圧が有る無しで数の 1, 0 の状態を作り出します。これと、スイッチを組み合わせて、様々な計算をすることができる電気回路を作ることができます。即ち、コンピュータの内部は、2 進法の数だけが存在する世界です*¹。

まず、一般的な数の表記 (記数法) について述べます*²。

通常、数は 789.012 のように書かれます。この意味は、 $7 \times 10^2 + 8 \times 10^1 + 9 \times 10^0 + 0 \times 10^{-1} + 1 \times 10^{-2} + 2 \times 10^{-3}$ の 10 のべき乗の係数を順に並べて書いたもので、10 進法表記と言います。10 のことをこの表記の基数と言います。10 進法表記では、数を表すのに 0 ~ 9 の 10 個の文字を用います。10 を基数とするのは、ヒトの両手の指の総数が 10 本であるからだと考えられていますが、数学的に合理的である事が証明されている訳ではないと思います。ヨーロッパ系言語の数詞を見ますと、必ずしも言葉は 10 進法ではありません。従って、10 進法は歴史的には割に新しい数の表記法です。基数を 10 にするのは人間的な都合ですが、10 以外の基数であっても、上のような数の表記は、筆算が可能であると言う意味では合理的です。例えば、ローマ数字を用いたかけ算を考えると、実行がとても難しい事は、ちょっとやればわかります (ローマ数字については、各自調べて下さい)。

上の基数を 2 に変えたものが、2 進法です。2 進法では、数を表すのに 0, 1 の 2 つの文字しか用いません。2 進法で 101.011 は、 $1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2} + 1 \times 2^{-3}$ の意味で、これを 10 進法で書くと $\frac{43}{8} = 5.375$ になります。

コンピュータの世界では 2 進法は自然なのですが*³、人間に取っては小さい数を表すのにも多量の桁数を必要とするため、双方で便利な記法として $2^4 = 16$ を基数とする 16 進法もよく用いられます。この場合、数を表記するのに 16 個の文字が必要となりますが、数字としては存在しないので、A, B, C, D, E, F をそれぞれ 10 進法の 10, 11, 12, 13, 14, 15 を表す文字として用います*⁴。

もう 1 つ、(数学ではあまり現れませんが) 浮動小数点表記というものがあります。化学で習ったアボガドロ数 6.02×10^{23} や物理で習った電気素量 (電子 1 個の電荷) 1.6×10^{-19} クーロンです。これらは、人間の普段の暮らしからすると大きすぎたり、(絶対値が) 小さすぎたりする値で、表記の桁数を少なくするために、上の様に数桁の有効数字に 10 のべき乗を掛けた形で書かれます。このような表記の仕方を、浮動小数点数 (浮動小数点表記) と言います。左側にある有効数字の部分を仮数部、右側にある 10 のべきの部分指数部と言います。コンピュータでも浮動小数点数の計算は可能であり、上の例では 10 進法で表記していますが、コンピュータでの実際の計算では、仮数部、指数部共に 2 を基数とする数に変換されて計算されます。

通常の数、浮動小数点数ともに、コンピュータの中にあるのは有限桁数の数であり、数学のような理想的な整数や実数が計算機で実現できている訳ではありません。ただし、普段の人間の実用に必要な範囲での数の世界は、実現されていると言って良いでしょう。

*¹ +, 0, - を利用して 3 進法コンピュータというのも考えられますが、おそらく技術的な理由で成功していません。

*² 高校の 数学 A や情報の授業で勉強して来て欲しい内容です

*³ 従って情報系の分野では、2 を底とする対数 $\log_2$ が数の桁数に対応するため、最も用いられる。

*⁴ 10 に近いと言う意味で、8 進法を利用していた時期もあるが、最近は廃れている。

2.4 コンピュータ内のデータその 2, 文字コード: 例として, ASCII コード

上で述べたように, 全てのデータは 0, 1 の列, 即ち 2 進法の数です. 2 進法 1 桁を 1bit(ビット, binary digit の略) といいます. 例えば, Nintendo 64 の 64 は 64 ビットに由来します. さらに 8bit を 1Byte(バイト) といいます. 2 進法の世界では $1024 = 2^{10}$ 毎に単位が K(キロ), M(メガ), G(ギガ), T(テラ) の単位をつけます.

しかし 2 進法では, 簡単な情報を表すのにも膨大な桁数が必要です. そこで, 2 進法を 4bit 毎に区切って, 16 進法を使う事が普通です. 16 進法では 1Byte が 2 桁の数になります. 16 進法では 10 進法の 10, 11, 12, 13, 14, 15 に文字 a, b, c, d, e, f (もしくは, A, B, C, D, E, F) を使います.

コンピュータの中では全てが 2 進法の数ですから, 文字も数に変換されています. この文字と数の対応を文字コードといいます^{*5}. 文字コードにはいくつかの種類がありますが, ここでは最も良く使われ, かつ簡単な ASCII(American Standard Code for Information Interchange) コードを紹介します. 次の表で最初の 2 列は制御文字と呼ばれる物です.

ASCII コード表

下位 \ 上位	0(0000)	1(0001)	2(0010)	3(0011)	4(0100)	5(0101)	6(0110)	7(0111)
0(0000)	NULL	DLE	Space	0	@	P	'	p
1(0001)	SOH	DC1	!	1	A	Q	a	q
2(0010)	STX	DC2	"	2	B	R	b	r
3(0011)	ETX	DC3	#	3	C	S	c	r
4(0100)	EOT	DC4	\$	4	D	T	d	t
5(0101)	ENQ	NAC	%	5	E	U	e	u
6(0110)	ACK	SYN	&	6	F	V	f	v
7(0111)	BEL	ETB	'	7	G	W	g	w
8(1000)	BS	CAN	(	8	H	X	h	x
9(1001)	HT	EM	)	9	I	Y	i	y
A(1010)	LF	SUB	*	:	J	Z	j	z
B(1011)	VT	ESC	+	;	K	[	k	{
C(1100)	FF	FS	,	<	L	\	l	
D(1101)	CR	GS	-	=	M	]	m	}
E(1110)	SO	RS	.	>	N	^	n	~
F(1111)	SI	US	/	?	O	_	o	DEL

表より, Ascii コードでは, 大文字の O は 16 進で数 4F(10 進で 79, 2 進で 01001111) に対応します. 通常は 16 進の数は前に, 0x をつけて, 0x4F のように表記します.

^{*5} より細かくは, 利用可能な文字の集合である「文字セット」と, それらをどのように 2 進法の数に対応させるかという「エンコーディング」の 2 つの部分からなりますが, そのような話は複雑なので省略します.

文字コードに対するいくつかの注意

ASCII コードは 7bit 上の表からわかるように, ASCII コードは 7bit です (最上位の bit は常に 0). しかし今のコンピュータはデータを Byte 単位で処理を実行しますので, 最上位のビットを 0 として, 8 bit のコードになっています. ASCII コードが 7bit になった理由のひとつが, 1 ページに挙げた参考書に書かれています.

文字コードは何種類もある 上でも言いましたが, 数と文字の対応表は他にもあります. ASCII 以外で代表的なのが大型汎用計算機 (Main Frame, 銀行のオンラインシステムなどで使われている) で使われている EBCDIC と呼ばれる文字コードです.

日本語の扱い 日本語, 中国語, ハングルなどの文字数の多い言語では, 文字コードはより複雑になります. 日本語の文字コードの代表的な規格は, JIS コード 2 種類と, EBCDIC の 3 つです.

インターネットでは 通信に於けるデータ交換では, お互いの合意があればどのようなデータ交換も可能です. しかし, 通信をする度に文字コードをいちいち決めていたのでは大変です. そこで, インターネット通信で標準的に用いられる文字コードが決められています. 他の言語と一緒に送っても問題の出ない文字コードは, 日本語については 昔の JIS コードから派生した iso-2022-jp という規格と Unicode から派生した iso-10646 という規格です.

過去のしがらみ 以前, JIS X 0201 1976 という 8bit の文字コードが定義されました. これは, 最初の 7 ビットの部分を Ascii コードの 2 カ所に変更を加え, 残った部分にカタカナを定義しました. この規格は日本のパソコンで多く用いられてきました. これは 2000 年に新しい JIS 規格に吸収されましたが, その規格にソフトウェアが対応していないことが数多くあります. この規格で定義されているカナ文字が, 半角カナ文字と呼ばれる物です. 上にも述べたように, きちんとした規格通りに動作しないソフトウェアが沢山あるので, これで定義されている半角カナ文字を使った文書を他人に送ったり渡したりしてはいけません. 迷惑行為です.

現在の Windows 10 (Windows 7, Windows 8 とかも) の日本語環境では, \ (バックスラッシュ) のコードに対して ¥ 記号を表示しています. これはこの過去のしがらみ (JIS X 0201 1976) のためで, 実際には間違った表記となっています. 日本語以外の Windows では, \ が表示されます. また, JIS X 0201 1976 の前半 7bit 部分を Ascii コードと解説している Web ページも沢山存在しますが, それらも間違ったコード表記です.

半角/全角 元々は印刷業界用語で, 文字幅の意味です. 文字コードには文字幅は定義されていず, 数と文字の対応だけが定義されています. 多くの環境で, 1byte 文字が 2byte 文字の半分の幅で表示されたため, 半角文字という俗称が生まれました. 2000 年にはこの俗称を追認する形で JIS X 0201 1976 カナ文字に half-width という名称がつけましたが, 今では意味がありません. この授業では (昔の風習を用いて) 1byte 文字と言う事にします.

他の言語では ASCII コードが合衆国でしか通用しないのは明らかです. ヨーロッパ圏やトルコ語, タイ語用の規格として多くで用いられていたのは, ISO-8859-1~ISO-8859-16 と決められている物です. これらは最近, 下の UTF-8 に置き換わっていったようです.

この講義では この講義では, 主に UTF-8 と呼ばれるコードを用います. これは上に述べた ISO-10646 の文字集合を符号化したもので, インターネットではデファクトスタンダード (de Facto Standard, 事実上の標準という意味のラテン語 + 英語) となっているものです.

昔、日本で売られたパソコンでは MS-Kanji と呼ばれる文字コード、およびこれから派生した文字コードが使われていました。これはマイクロソフトウェア・アソシエーツと三菱電機 (アスキー社も加わっていた?) が定めた日本語の文字コードを少し変形したものです。上で述べた、JIS X 0201 1976 の規格を 2byte にし、増えた部分に JIS で定められた文字セットを少し番号をずらして割り振ってできたものです。今でこそ、過去との互換性のために国際的に登録された文字コードとなっていますが、元となった JIS 規格の規格違反をしている、多数の方言がある、ASCII 以外のコードとの共存ができない等問題点の多い文字コードです。

なお、コンピュータは、上で述べた main frame のように、パーソナルでないものもあります。過去には、何種類かの日本語文字コードが、コンピュータシステム毎に存在していました。それらの中で、代表的なものひとつが、EUC (Extended Unix Code) です。

文字コードはこれからも変化する 日本語に限っても文字コードはまだまだ不完全な物です。これから先も変化し続けると考えられます。その変化を追いかける専門家になる事は不要ですが、その変化に対して合理的な行動がとれるように心がけて下さい。

現在でも、Unicode の空き領域に絵文字が多量に追加され続けています。

より詳しく知りたいなら Wikipedia(<http://ja.wikipedia.org/>) の文字コードの項は、割と良く書けている。本気で勉強するなら、一冊の専門書を読む必要があります。

私の講義を受ける上で注意してほしいこと

漢字まで込めて、日本語がコンピュータでそこそこ使えるようになったのは、1978 年に制定された、JIS X 0208 と呼ばれる文字コードです (JIS = Japan Industrial Standard: 日本工業規格)。これは 7bit × 2 バイト (もしくは 8bit × 2) の文字コードでした。

問題になるのは、この文字コードに数字やアルファベットが定義されていたことです。つまり、ASCII コードと同じ文字が異なる数に対応するように JIS でも定義されました。

コンピュータ自体、あるいはそれを動かす基本ソフトは、ほとんどの場合アメリカで開発されましたから、当然、ASCII コードを基本とする処理系が作られます。日本では、それに上の JIS X 0208 を加えて動かすというシステムになりました。そのための工夫が、MS-KANJI だったり、EUC だったわけです。

その世界では、アルファベットの A に対応する数が 2 種類あることになります。そこで、ASCII の文字幅を JIS X 0208 の文字幅の半分で表示するフォント (Font, 文字の形を表示するためのデータ) が作られました。これが、全角/半角という言葉の起源です。

しかし、プログラムなどを作る側から考えるとそのようなシステムは、不便です。「1」という文字を探すのにも、ASCII と JIS X 0208 の両方を探さなければいけないからです。また、プログラミングなどでは、プログラムで意味を持つ単語は、ASCII 文字を使うのが通常で、JIS X 0208 のような文字は、文字列定数としてしか使わないのが基本です。

現在、上のような「ASCII + 現地語」のようなシステムは、ほぼ無くなっていると思います。しかし、残念ながら、Unicode を策定するときにも、「複数の数が同じ文字に対応する」という愚行が継承されました。

この授業では、HTML を講義する予定ですが、それ以外の場面でも私の授業では、

ASCII で定義されている文字は、半角を利用する。

を守ってください。前回の資料でも述べましたが、特に邪悪なのが「全角の空白」です。これが原因でうまくないことが往々にしてあり、見えないので、原因究明に時間がかかるのです。他にも、括弧記号とかにも注意してください。

前回の資料で、「全角の空白で位置揃えをするな」と書きました。ほとんどの場合、位置揃えには全角空白以外の方法が用意されており、位置揃えをすることを意識的に行えるようになっています。つまり、「全角の空白」というのは、全く存在価値のない文字なのです。

次の問は、高校の数学 A の復習です。レポート問題ではありませんが、数理の学生としては、このくらいはできるようになっておいてください。

問 10 進法で 0.1 と表記される数を 2 進法で表すとどうなるか。

今後の予定

次週、4 月 30 日は、木曜の授業日なので、この講義のデータの更新はありません。

次回 5 月 7 日には、もう一度、コンピュータに対する一般的な話をした後、テキストエディタのインストールの解説を書きます。

その次は、琉球大学の図書館利用についての解説を、図書館の方が作った資料をもとに、皆さんで自習してもらいます。資料の場所は、この Web class でお知らせします。私の方からは、数理科学科の図書室に対する資料を提示します。

上のことが終わる頃には、皆さんの PC の環境整備ができると思うので、Web page の作成に入ります。

Web page の作成では、なんでも良いのでオリジナルなページを作成していただきます。連休中に素材を集めるみたいなことも、考えて下さい。ただし、感染症予防は徹底して下さい。