

# ネットワーク システム管理 #02

たかさきこうや  
1限 (09:00-10:45)

## 先週のお題

- Wireshark、TeraTerm、WinSCPの3つのソフトをとりあえずインストールして起動できるようにしておいてください
- 皆さんの身の回りで「数字で管理されているもの」のうちもうじき「桁あふれ」してしまいそうな「もの」とそれが予測される「時期」を考えて例示してください
- その他

1

2

## まずは基本的なこと…

- CUCのメールアドレスを書いてください…
  - 学外のメールアドレスを書かれても、僕には「名寄せ」が出来ません…
    - 名寄せ：課題提出者が、本学の学生としては「誰」に相当するのか
    - 学籍番号の一意性を使わないと、同姓同名がいる場合個人特定ができません
- 「あなたのメールアドレスを記入してください。(学籍番号でも、ログイン名でもありません)」
  - なので、学籍番号を書いた人はアウト
  - @st.cuc.ac.jp を書いていない人はアウト

3

## 学籍番号/アカウント名/メールアドレス

- 2240XXX …学籍番号
- c240XXX …アカウント名
- c240XXX@st.cuc.ac.jp …メールアドレス
- それぞれ意味が違うので、区別すること

4

## 桁あふれの解答例

- カレンダー
- 携帯電話番号
- うん、それ、講義中に俺、話した
  - 補足：携帯番号については、「番号は**使いまわされる**」「日本の人口はこれ以上増えない」「**060**が今後用意される」ので、人が使う分については足らなくなることはおそくない
  - IoT機器が使う番号(通話を伴わない、発信のみ、等)は足らなくなったため、2021年に**14桁**に増やされた

5

## 桁あふれの解答例で面白かったもの

- 運転免許証(の12桁目)
  - 再発行のたびに、ここの数字が増えていく
  - 9回目の再発行の次は**0**に戻らず、**1**になる
  - ただし、5回目以降あたりから警察に「お前なんで無くしとんねん」とかなりきつく詰問される(犯罪に使われる可能性があるので)
  - よほどでない**と9**まで使い切って**1**になる人はいない
  - というわけでループする決まりなので、「桁あふれ」は起きない
- 銘柄番号
  - 株式市場で使われる4桁の企業識別番号
  - 2024年から2桁目と4桁目にアルファベットが使われることに
    - CUC方式だから、「桁があふれたわけではないが…」というパターン

6

## 桁あふれの解答例で面白かったもの

- 銀行の支店コード
  - 銀行のコードは国際機関によって決まっている
  - それぞれの国で金融機関ごとに4桁の番号
    - 使いまわさない
  - 金融機関ごとに3桁の支店コード
    - 使いまわさない
    - 1つの金融機関が「1000店舗」持つと桁あふれが起きる可能性はある
      - ただし、3桁は国際的な仕様なので、どちらかというとこれは「一銀行が1000も支店を持つな」という縛りと捉えられる
  - 支店ごとに7桁の口座番号
    - 使いまわさない
      - 1店舗に1千万人も顧客は抱えられない

7

## 桁あふれの解答例で勘違いしてるもの

- 世界人口
- 迷惑メールの総件数
- Gmailで他と被らないメールアドレス
- switchソフトの販売本数

8

## マイナンバー

- 12桁、うち1桁はチェックデジットのため実質11桁
- 使いまわさない
- $10^{11}$ となるため、1000億に対応
- 生涯同じ番号を用いる  
(漏洩し悪用される恐れがある場合のみ変更可能)
- 日本の人口はおよそ1.2億
- 1年あたり、1/80が入れ替わると仮定する
- …そんなに簡単に足りなくなる？

9

## 車のナンバープレート

- 車のナンバープレートは再利用制度が（現在は）ない
- 最近桁あふれを起こし（そうになっ）たため、「地名+〇〇」の〇〇の部分にアルファベットを導入した(2017年)
  - ※〇〇は「分類番号」
  - 3桁のうち下2桁に、アルファベットが10文字追加された
  - $10 \times 10 \times 10 \rightarrow 10 \times 20 \times 20$ なので、4倍に拡張されたことと同義

10

## クレジットカード

- については、ちょっとおもしろいので掘り下げる
- まず、クレジットカードの番号はそもそも「ISO/IEC 7812」で規定されている「磁気ストライプ型IDカード番号」と同じもの
  - ISO/IEC 7812では、全19桁を有することができる
  - うち、先頭の数3,4,5,6がクレジットカードに付与されている
  - 医療には先頭8が付与されていたり
- 磁気カードそのものは、カードの磁気ストライプ部分にデータを書き込むだけなので、複製は容易

11

## クレジットカード

- 現行(最大)16桁、うち最初の6桁は発行者識別番号
  - 最初の6桁で、クレジットカード会社がわかる
  - 最後の1桁がチェックデジット
  - 残りが個人番号
- つまり「楽天VISAカード」などのカードごとに現行  $10^9$ (=10億人)の顧客を抱えることができる
  - 漏洩したら再発行
  - 番号の使いまわしはなし
  - とはいえ、世界人口の1割強の顧客を抱えるカード会社は存在しないので、かなり潤沢

12

## その他(1)

- お勧めのゲームはありますか？
  - ーごめん、ゲームはあんまりやらないんだ
- IT系の企業に就職するならとった方が良い資格はありますか？
  - ーITパスポート、基本情報、応用情報 あたりはそれなりに知名度がある。右に行けば行くほどスキルがある扱いされる
  - [https://www.cuc.ac.jp/campus/expenses/shoreikin/mstsp000000b23o-att/kyotsu\\_shikaku2023.pdf](https://www.cuc.ac.jp/campus/expenses/shoreikin/mstsp000000b23o-att/kyotsu_shikaku2023.pdf)
  - 奨励金も出るので在学中にやっくと良いと思う
  - あと俺、「NTT.comの『インターネット検定』の委員」もやります

13

## 先週の話

- 10進数、2進数、16進数
- CUCでは学生さん一人一人を一意な学籍番号で識別している
- 桁と桁あふれ
- bit(ビット)とbyte(バイト)
- コンピュータでは、「文字」「音声」「画像」「映像」「データ」など、さまざまなものを電気を使ってやり取りする

15

## おさらい：単位

- 日本(や中国とか)は、 $10^4(10 \times 10 \times 10 \times 10 = 10^4)$ で繰り返すけど、欧米では $10^3(10 \times 10 \times 10 = 10^3)$ で繰り返す
- だから、アラビア数字は0が3つ出てくるとにカンマで区切る(彼らはそれで数えやすくなる)
- 当然、アジア言語圏の人にとっては数えにくい

17

## その他(2)

- ついていきたいですが、数学的な思考が必要だとちょっとヤバいです
  - ー必要はあります
  - ーというか、もともと「商大」って数学的思考は要るよ？
- 遠隔授業が減って困ってます
  - ー俺の講義に限って言うと「みんなの手作業の進捗度」は、教室回ってみんなの画面見ながらじゃないと絶対に確認できないので、演習系講義は遠隔だとシンドいんですね…
  - ー気持ちも分かる (あくまで「気持ち」だけけど)

14

## おさらい：通信速度

- コンピュータが、ネットワークを使って、情報のやりとりをする
- 電気のon/offのみを使うので、1秒間ずっと電気が流れればなしだとそこにonが何個あるか分からない
- だから「1秒間に何bit分」送るかをお互いに決めて通信をする
- 1秒間に300個分のon/offを流す場合「300bps(bit per second)」と言う
- 読みは「さんびゃくビーピーエス」
- 今どきの有線の通信速度は「1Gbps」

16

日本の読み方	アラビア数字	英語での読み方	単位
いち(一)	1	One	
じゅう(十)	10	Ten	
ひゃく(百)	100	One hundred	
せん(千)	1,000	One thousand	k(キロ)
いちまん(一万)	10,000	Ten thousand	
じゅうまん(十万)	100,000	One hundred thousand	
ひゃくまん(百万)	1,000,000	One million	M(メガ)
せんまん(千万)	10,000,000	Ten million	
いちおおく(一億)	100,000,000	One hundred million	
じゅうおおく(十億)	1,000,000,000	One billion	G(ギガ)
ひゃくおおく(百億)	10,000,000,000	Ten billion	
せんおおく(千億)	100,000,000,000	One hundred billion	
いちっしょう(一兆)	1,000,000,000,000	One trillion	T(テラ)

18

## 単位

- 1の1000倍 = 1キロ (1k)
  - 1キロの1000倍=1メガ (1M) = 1,000,000
  - 1メガの1000倍=1ギガ (1G) = 1,000,000,000
  - 1ギガの1000倍=1テラ (1T) = 1,000,000,000,000
  - 1テラの1000倍=1ペタ (1P) = 1,000,000,000,000,000
- IT関連だとこのぐらいの単位までは出てくるので慣れてください

19

## 文字の話

- 文字をコンピュータ同士でやりとりしたい
- そのためには、まず何を決めなくてはいけないか

20

## 決めるべきこと

- やりとりするのは何処の誰とか
- やり取りしたい文字は何種類あるか
- それは何桁あれば足りるか
- それを、どのように一意に番号を割り当てるか

21

## 先の例でいうと

- 性別という情報をやり取りしたい
  - 種類は？
    - 昔：男と女しかないんだから2種類だ
    - 今：男、女、肉体男で精神女、肉体女で精神男…etc …etc
- 桁数は？
- どう割り当てる？
  - 男=0、女=1？
  - それとも女=0、男=1？

22

## 昔話：コンピュータは

- ——英語圏で生まれた
  - 彼らにとって、文字は
    - 「アルファベットの大文字と小文字」
    - 「数字」
    - 「あといくつかの記号」
  - しかなかった
- それは100種類もない
- それを2進法で表現したい、と思ったら、何桁必要か

23

## 組み合わせの総数

- 一体0と1の組み合わせが何桁あれば足りる？
  - 1桁なら2通りの文字しか表現できない
  - 2桁なら組み合わせて4通り
  - 3桁なら8通り
  - 4桁なら16通り
  - 5桁なら32通り
  - 6桁なら64通り
  - 7桁なら128通り
  - 8桁なら256通り

24

さて、数えてみよう

- 数字は10文字
- アルファベットは大文字が26文字
- 小文字も26文字
- 記号は30個くらい？
- →これなら7桁(7ビット=128パターン)あれば余裕で足りる
- 当時はまだコンピュータ資源が貴重な時代
  - 使うかどうか分からないもののために、貴重な枠を大量に用意するわけにはいかなかった

25

この7桁を使い

- それぞれの番号に、文字を割り当てていく
  - 例えば以下の通り
    - 0100001 = !
    - 0101111 = /
    - 0110000 = 0
    - 0111001 = 9
    - 0111010 = :
    - 1000000 = @
  - 1000001 = A
  - 1011010 = Z
  - 1011011 = [
  - 1100000 = `
  - 1100001 = a
  - 1111010 = z
  - 1111011 = {
  - 1111110 = ~
- ASCII表は次ページ

27

さて、実はASCII表には

- ハート(♥)がない
  - 他にも色々ないし、漢字も日本語もない
    - さっきも書いたけど「この当時の偉い人たちは、英語圏の文字以外を定義するつもりがなかった」
- 限られた枠の中で、どの文字を採用し「定義」するか？
- これを一般的に「文字集合(文字セット)」と言う

29

で、英語圏の

- ——偉い人たちは考えた
- 数字は、全部で0と1を8つ組み合わせよう
- ただし、本当に使うのは7つで、8つ目は特殊な用途に使う
  - パリティビット
  - 誤り検出
- こうして決まったのが「ASCII規格」と呼ばれるもの
- ※ American Standard Code for Information Interchange

26

32	0100000	20	SP	56	0110000	38	8	90	1010000	50	P	104	1101000	68	h
33	0100001	21	!	57	0110001	39	9	91	1010001	51	Q	105	1101001	69	i
34	0100010	22	"	58	0110010	3A	:	92	1010010	52	R	106	1101010	70	j
35	0100011	23	#	59	0110011	3B	;	93	1010011	53	S	107	1101011	71	k
36	0100100	24	\$	60	0111000	3C	<	94	1010100	54	T	108	1101100	72	l
37	0100101	25	%	61	0111001	3D	=	95	1010101	55	U	109	1101101	73	m
38	0100110	26	&	62	0111010	3E	>	96	1010110	56	V	110	1101110	74	n
39	0100111	27	'	63	0111011	3F	?	97	1010111	57	W	111	1101111	75	o
40	0101000	28	(	64	1000000	40	@	98	1011000	58	X	112	1110000	76	p
41	0101001	29	)	65	1000001	41	A	99	1011001	59	Y	113	1110001	77	q
42	0101010	2A	*	66	1000010	42	B	90	1011010	5A	Z	114	1110010	78	r
43	0101011	2B	+	67	1000011	43	C	91	1011011	5B	[	115	1110011	79	s
44	0101100	2C	,	68	1000100	44	D	92	1011100	5C	\	116	1110100	80	t
45	0101101	2D	-	69	1000101	45	E	93	1011101	5D	]	117	1110101	81	u
46	0101110	2E	.	70	1000110	46	F	94	1011110	5E	^	118	1110110	82	v
47	0101111	2F	/	71	1000111	47	G	95	1011111	5F	_	119	1110111	83	w
48	0110000	30	0	72	1001000	48	H	96	1100000	60		120	1111000	84	x
49	0110001	31	1	73	1001001	49	I	97	1100001	61	a	121	1111001	85	y
50	0110010	32	2	74	1001010	4A	J	98	1100010	62	b	122	1111010	86	z
51	0110011	33	3	75	1001011	4B	K	99	1100011	63	c	123	1111011	87	{
52	0110100	34	4	76	1001100	4C	L	100	1100100	64	d	124	1111100	88	
53	0110101	35	5	77	1001101	4D	M	101	1100101	65	e	125	1111101	89	]
54	0110110	36	6	78	1001110	4E	N	102	1100110	66	f	126	1111110	90	^
55	0110111	37	7	79	1001111	4F	O	103	1100111	67	g	127	1111111	91	_

28

でも

- でも、世の中は英語だけではない
- 僕は日本語を使いたい
- それも、複数のコンピュータで情報が交換できるようにしたい

30

## 日本語を扱うためには

- 何桁要る？
  - 平仮名50文字ぐらい？
  - 片仮名50文字ぐらい？
  - 濁点、半濁点、拗音、促音、撥音、長音
  - 普通使う漢字が3000字弱
  - あまり使わないけどある漢字が3000字強
  - 記号とか
    - 全部で1万字ぐらい？
- じゃあ、ASCIIに倣って日本語を定義しよう！

31

## まずは

- どの文字を扱うかを定める
  - Ex: 常用漢字
    - 文科省が決めている
    - 「日本人が日常生活で使う文字(とその読み)の一覧」
    - 漢字は2136文字ある
    - ほぼ全てを小中学校で習う
- 文字集合
  - 日本における文字集合はJIS X 0221:2014が(多分)最新

32

## JISってなに？

- JIS = Japanese Industrial Standards(日本産業規格)
- かつては「日本工業規格」と呼ばれた
- 国内産業で仕様の統一が出来るよう、色々な統一規格を決めたりしている
- 帳票なども含まれる
- JIS規格の履歴書、なんてものもある

33

## 文字集合上の制約

- 例えば、僕の名前(たかさき)の漢字である「崎」は、2004年制定のJIS X 0213:2004で初めて定義された
- 「高」の方は、JIS X 0221(≒ISO/IEC 10646)で定義された
  - ISO/IEC 10646の翻訳規格にすぎないため、JIS X 0221で定義されている漢字は所謂「JIS漢字」ではない
  - IBMの定義した外字に登録されており、Windowsでも使えた

34

## 文字集合

- 扱う文字を決めたら、今度は、どの文字にどの番号を割り当てるか決める必要がある
- 文字符号化方式
- 国内で主に使われる符号化方式は「3種類」「あった」

35

## 符号化方式

- 先ほども書いた通り、日本語はおよそ1万字分の割り当てが出来れば良い
- 2の16乗=65,536
- 2の14乗=16,384
- 2の13乗=8,192
- なので、16ビット=2バイトあれば足りる、という計算
- 実際、日本国内用文字符号化方式はいずれも16ビット

36

## 文字符号化方式

- ISO-2022-JP
  - 電子メールで現在も利用されることがある
- Shift\_JIS
  - Windows系で現在も利用されることがある
- EUC-JP
  - UNIX系の一部で現在も利用される

37

## エスケープシーケンス

- 「ここから日本語が始まります」
- 「ここで日本語が終わります」
- を意味する特殊な記号を用いて、普通のASCII文字と日本語を区別する
- 例えば、「航」を表すJISコードは00111001 01010010
  - でもこれだと、
- 9(00111001) R(01010010)
- と区別がつかない！

39

## ISO-2022-JP(JIS)の制約事項

- 半角カナは使えない
  - 使えないこともないが、伝わる保証がない
- もはや、レガシーな電子メールでしか使われない
  - が、逆にこれを使っていれば、日本語のメールを送ったときに伝わる確率が上がる
- 日本語と英語が変わる度に制御記号が入るので、文字数とファイルサイズが一定ではなく、ファイルサイズから文字数が類推できない

41

## ISO-2022-JP(JIS)

- ISOという規格で決められている、日本語を表現するコード体系
  - 単にJISとも呼ばれている
- 通常、電子メールの送受信に使われる
- 2バイトコード（2バイトで1文字を表現）
- 8ビット目を使わない
- 「エスケープシーケンス」という技術を使って、英語と日本語の部分を区別する

38

## だから

- エスケープシーケンスを使い、日本語部分を「挟む」ことで、日本語と英語を区別している
- でも、頻繁に日本語と英語を切り替えると、無駄にデータ量が増える
- また、JISでは一般的に「半角カナ」は定義されていない
- 使える場合、そのソフトウェアは「ISO-2022-JP」の規格を無視している

40

## Shift\_JIS

- WindowsやMacで使われていた文字コード
  - 今でも使える
- エスケープシーケンスなし
- 7ビットではなく、8ビットを存分に使っている
  - つまり、エラー検出用ビットを使わない

42

## でもShift\_JISでは

- 1byteコード(ASCIIコード)にもあった「とある文字」に割り当てているコードを2バイト目に使った
  - 厳密に言うと、1バイト目には0x81~0x9fもしくは0xe0~0xfcが出て来る
  - 2バイト目には0x40~0x7eもしくは0x80~0xfcが出て来る
- この中でも、2バイト目の「0x5c」がヤバイ

43

## エスケープ文字

- プログラム上、「特殊な意味」を持つ記号を、普通の文字として使いたくなることがある
- \$とか&とか…まあ色々
- 特殊な文字の直前に「¥」を書くことで、それらを普通の文字として認識させることができる
- これをエスケープ文字と言う

45

## EUC-JP

- エスケープシーケンスなし
- 8ビットを使う
- 2バイトではなく、3バイトの文字も混在する
- Extended UNIX Codeの略
  - 古いタイプのUNIXで使うときは幸せ
  - 最近のUNIXは別
  - UNIXってなに? っていう話は、2週間後ぐらいにやるはずなので割愛

47

## 0x5c

- 0x5cは、半角の「\」
- 日本語環境だと「¥」が出る
- でも、Windowsでは、この¥マークは「フォルダの区切り記号」に使われていたりする
- プログラミング方面では、「エスケープ記号」として使われている
- エスケープシーケンスとは違う

44

## Shift\_JISの制約事項

- 不用意にShift\_JISを使うと、「¥」が問題のある振る舞いをして、プログラムが異常動作を起こすことが多い
- 電子メールでもそのままでは使えない
  - 静的な(プログラム処理をしない)Webページだとならん問題はない
  - 「インターネットでは半角カナは使っちゃダメ」みたいな言説はここから来ている

46

## EUC-JPの制約事項

- 電子メールでは使えない
- Shift\_JISと、似たようなコード体系を使っている
- しかも、エスケープシーケンスがないため、判別が出来ない
  - 現状では、「Shift\_JISっぽいか」「EUC-JPっぽいか」を機械が頑張って判断しているだけ
- 短い文章(数文字)だと、全く判別不能

48



さて…

- 当然のことながら、「ISO-2022-JP」「Shift\_JIS」「EUC-JP」のいずれも、日本語以外の文字集合は扱えない
- 日本以外の国でも、自国の文字のための文字集合を決めてもそれを日本では扱うことができない
- インターネットが普及したことに伴い、すべての国の文字を一つの文字集合で表す必要が出てきた

49

## Unicodeの

- 文字集合的な特徴は以下の通り
- 現在111万の空間が用意され、うち14万弱が文字に割り当てられている
- 21ビットを使用
- 絵文字にも対応
- 中国語、日本語、韓国語の漢字を統一

51

## 演習

- IMEパッドを開いてみる
- その中から、自分の名前の漢字を探してみよう
- Macの人はことえりの文字パレットから出せるらしいよ？



53

## 全世界統一の

- 文字集合があれば、一つのファイル内に複数の言語が書ける
- Unicodeという、全世界の文字を統一的に扱う規格が生まれた
- 現在、Windows、Mac、iPhone、Android、Unixの大部分はUnicodeを標準の内部コードにしているので、おのずとUnicodeが扱えるようになっていく

50

## 符号化方式の特徴

- 主に使われるのはUTF-8、UTF-16
- UTF-7、UTF-32などもある
- 例えばUTF-8は、1~4バイトの変長であり、ASCIIコードなどは1バイト、漢字などは3バイトになるので、バイト数だけみれば欧米が有利、漢字を使うものが不利になる
- UTF-32は全文字が4バイトの固定長
- データとして見た時に判別しやすくする方法として「BOMを付けるか付けないか」や「ビッグエンディアンにするかリトルエンディアンにするか」があるがここでは割愛

52

## 調べてみよう

- 僕の名前の「航」の字
- Unicodeでは U+822a が割り当てられている
  - UTF-16では0x822aと出てきている
  - 「a」が数字中に出てきているのでこれは16進数(4bit)だ
  - 先頭に0xとついているので16進という意味でもある
  - それが4桁なので、4bit x 4=16bit=2byte と思われる

54

例えば…

- この、「822a;」の前に&#xをつけ、「&#x822a;」とだけ書いて、test.htmlファイルとして、デスクトップに保存
  - このファイルをダブルクリックすると、何が出て来る？

55

試してみよう

- 自分の名前の文字コードを全て調べる
- その16進数に&#xをつけて記述してtest.htmlファイルとして保存
- こういう風に、文字そのものではなくそのコードを記述する手法を「実体参照形式」なんて言う
  - Web界隈では頻繁に用いられる方法で、ISO10646(≒Unicode)の文字番号で入力できるので、ファイルの文字コードがなんであっても表現できるし、ファイルの文字コードがUnicodeである必要もない

56

余談：test.htmlの拡張子

- Windowsでは、拡張子を見て「起動するアプリ」を決めている
  - .htmlだったら、「Edgeを開き、該当ファイルを読み込め」とか
  - .txtだったら「メモ帳を開いて該当のファイルを読み込め」とか
- ただ、拡張子が.exe(とか.batとか)だったら、「そのファイルをプログラムとして実行しろ」という意味になる
  - なので、ファイルだと思ってダブルクリックしたらプログラムでした→それが悪意あるプログラムだったので感染しましたみたいなことになる

57

メモ帳と.html

- メモ帳は「テキストファイル」を作るプログラム
- 普通に保存すると、拡張子は.txtになる
- しかし、.txt以外も作れないわけではない(.htmlとか)
- ただし、.htmlを開こうと思ってダブルクリックすると、「関連付け」された「Edge」が開く
- この場合、「まずメモ帳を開いて」から、「ファイルを開く(.htmlファイルを)」と、.txtの拡張子以外のファイルも開ける…という寸法

58

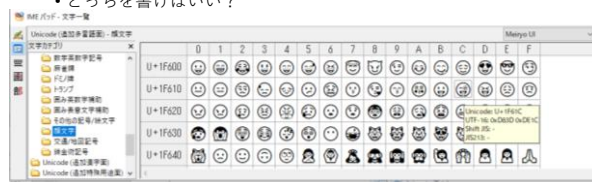
本来であれば

- .htmlファイルを開くと、その時点でインターネット上のファイルを読み込みかねない
- (本音を言えば)ブラウザと関連付けされているべきではない
- 自分で適当なテキストエディタをインストールし、.htmlファイルと関連付けてしまうのが安全ではある
  - …という考え方ができると、少し成長ではある

59

じゃあ

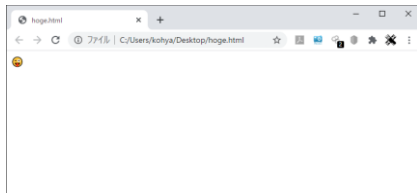
- この辺はどうだろう？
- UnicodeとUTF-16が違う
- どっちを書けばいい？



60

## 答え合わせ

- &#x1F61C; とだけ書くと、右のような表示になる
- 君、なんかさっきと違うくない？



61

## コンピュータを使ってネットワーク上で

- 正確に文字をやり取りするには
  - お互いが同じ「文字集合」を使っている
  - お互いが使う「文字符号化方式」を知っていて相互に変換できる
- これに加え
  - 発信者が想定した文字に、受信者の環境が対応している(該当のフォントがある)
- ことが必要

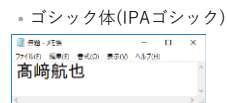
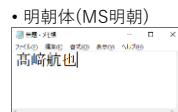
62

## フォントってなに？

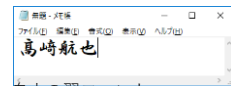
- 字の形、のこと
- 細かいことは割愛
- ざっくり言うと
  - 「文字のそれぞれの線の太さが一定か、そうでないか」
  - 「文字の装飾があるか、そうでないか」
  - 「個々の文字が同じ幅か、そうでないか」
  - で大別される

63

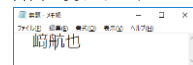
## こういうやつ



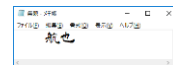
- 明朝体(HGP行書体)



- 自由の翼フォント



- 青柳隸書



64

## 見ての通り

- 自由の翼フォントでは「高」の字がない
- 青柳隸書フォントには「高」も「崎」もない
- 文字集合にあり符号化された文字であっても、フォントがない場合、表示させたい文字が正しく表示できる保証はない
  - 商用のフォントは、この辺の「普段使わないだろこんな字」みたいな字のフォントもちゃんと作ってる(ことが多い)
    - 故に高価(なことが多い)。ただし高価な理由はほかにもある
      - 気の利いたシステムなら、別のフォントを使って表示してくれることもある

65

## よく聞く「全角」「半角」

- 「0」と「O」は別の文字である
- 半角のゼロと、全角のゼロ
  - 前者は「U+0030」、後者は「U+FF10」だ
  - アルファベットにも全角「A」と半角「A」がある
- 両者は同じ文字に見えるが、コンピュータの中では「違う文字」として扱われる
- 「コンピュータに歩み寄ってあげる」なら、これは明確に違うものとして扱ってあげるべし
  - お勧めは「半角で表現できるものは半角を使うこと」

66

## 文字以外のデータの話

- 音声、映像、画像なども何らかの方法でデータに置き換える
- 当然箱(枠)が重要になる

67

## だから

- 自分で光を出さないものの場合
  - 色 = 「どんな波長の光を吸収したか」
  - Ex: 絵具やインクなど
    - 「全ての色を混ぜる」と「全ての波長を吸収する」
    - ⇒ 黒になる
- 自分で光を出すものの場合
  - 「どんな波長の光を出したか」
  - Ex: ブラウン管や液晶など
    - 「全ての色を混ぜる」と「全ての波長が出ている」
    - ⇒ 白になる

69

## 赤、緑、青が

- 全部均等に、全開まで光を出したら「白」に見える
- 全部が光を出してなかったら「黒」に見えるはずなので、撮った写真をよく眺めてみよう

71

## 画像の色の多さ

- 色が、3原色で表現出来るのは知ってる？
  - 光の3原色 = 赤、緑、青
    - 全部混ぜると白になる
  - 色の3原色 = シアン、マゼンタ、イエロー
    - 全部混ぜると黒になる(ということになっている)
  - そもそも、「色」とは「光を当てたとき、どんな波長の波が返ってくるか」を表現したものだ
  - 何故3種類なのか？
    - 人間の錐体細胞が3種類だから

68

## 詳しいことは

- 色彩学のページかなにかを見て調べること
- ブラウン管や液晶みたいな「光る物」は
  - 赤い光
  - 緑の光
  - 青い光
  - の「強さ」を変えることで、様々な色を表現している

70

## この、赤や緑や青の

- 光が、完全に消えている状態から全開に光を出している状態までを、  
**一体何段階で区切るか**  
が、微細な色を表現できるかどうかの分かれ目になる
- 例えば
  - 赤 消灯-点灯(2段階)
  - 緑 消灯-点灯(2段階)
  - 青 消灯-点灯(2段階)
- だと、全部で8色の色を表現できる

72

## でもディスプレイはもっとカラフル

- 例えば、赤、緑、青に対して8bitの箱を用意する
- 8bit=2 5 6種類の情報を用意して、それぞれを光の強さに割り当てると、全部で  
「2の8乗」×「2の8乗」×「2の8乗」で、  
「2の24乗」= **16777216色**  
の色が表現できる
  - 「24ビットフルカラー」という単語はこれのこと

73

## Powerpointと電卓をたちあげて

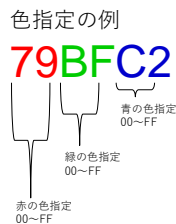
- なんかPowerPointで適当な図形を作り、ここに色を設定してみよう
- バージョンによってはHexという記述がないが…
- 赤、緑、青について、0~255までの色が指定できる
- 適当な色を指定したら、それを電卓を使って10進数⇒16進数に直してみよう
- そしてそれを赤、緑、青の順番に16進6桁で記述してみよう



74

## すると…

- 赤 FF0000
- 緑 00FF00
- 青 0000FF
- 黄 FFFF00
- 灰 888888
- など、など、など…



75

## さっき作ったファイル

- ```
<body
  bgcolor="#ffffff"
  text="#000000">
  &#32384; &#22883; &#42439;
</body>
```
- みたいに書き足して保存してみる(赤字が追記箇所)
  - 消しちゃった人は大した手間じゃないから作り直そう

76

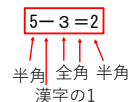
## …で、

- ```
<body
  bgcolor="#ffcc33"
  text="#002244">
  &#32384; &#22883; &#42439;
</body>
```
- 赤字の箇所(色を24bitで指定している)を自分の好きな色に変えてみよう
  - HTMLカラーコードなどで検索してみよう

77

## 本日のまとめ

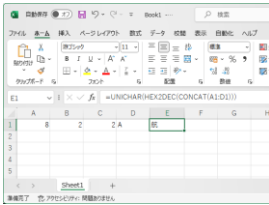
- 自分が書いている文字や文章が、見た目的に正しいかどうかより「コンピュータの中で使っているコード的に正しいかどうか」で文章を書こう
  - 間違っても、右のようなことはしないように…
- 文字コードの存在を理解しよう
  - 文字集合と符号化形式、フォントの概念を押さえよう



78

オマケ(みんなExcel大好きだよ！)

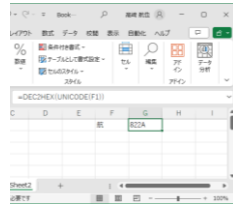
- Excelだとこんなことができる



79

では、本日のお題

- Webサイト上のURLにアクセスし、設問に答えてください
- 本日の目標
  - 10進数の数を、自在に2進数、16進数に置き換えられるようになってください
  - 文字とコードを変換できるようになってください
  - **数字やアルファベット**は、**半角英数**で入力しましょう



80