文字コードへの道

The Road to Character Encodings

pixiv Inc.
USAMI Kenta

pixiv

2024-01-28 #retro games any
レトロゲームから得た学びの発表 #01

お前誰よ

- うさみけんた（@tadsan）/ **Zonu.EXE** / **にゃんだーすわん**
- ピクシブ株式会社 pixiv事業本部 Webエンジニアリングチーム PHPer
- Emacs PHP Modeを開発しています（2017年-）
- プログラミング言語にちょっとこだわりのある素人（spcamp2010）
- ゲーム開発経験は学生時代にDXLibとか一瞬使ったくらい
- 好きなゲームはスーパーロボット大戦シリーズ

今回の発表の背景 (phpcon2024)

PHP Conference Japan 2024

採択 2024/12/22 11:25〜 トラック3 - 4F コンベンションホール 梅 レギュラートーク(25分)

入門 文字列

 うさみけんた tadsan

☆ 11

文字列(string)はPHPのみならず多くのプログラミング言語で提供されている基本的な機能のひとつであり、誰でもあたりまえに使っているものです。しかしその実態は多様で、一筋ではいかない概念であります。

本トークでは文字列という概念の概観を掴み、一筋ではいかないということを納得して向き合えるようになることを目指します。

- PHPの文字列の性質
- 文字コードとは何か
- UnicodeとUTF-8
- プログラミング言語は文字をどう扱うか
- レガシー文字コードに立ち向かう

ただし、以下の内容については詳しく取り扱いません。

- mbstringモジュール内部構造の最新動向

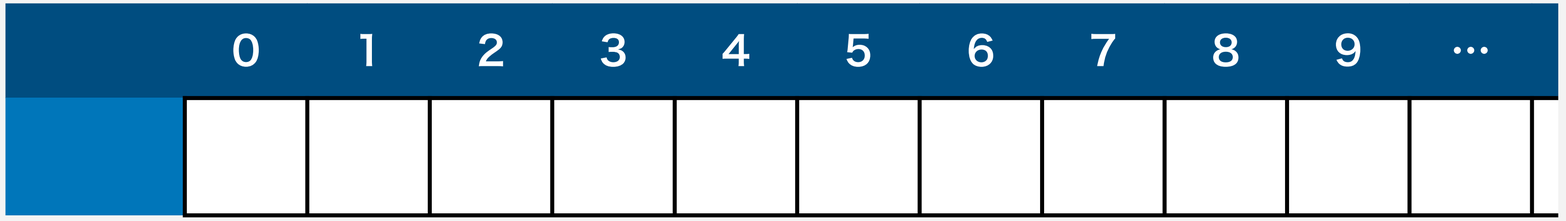
pixiv

さて

コンピュータ内のデータのイメージ

非常に大雑把なイメージ

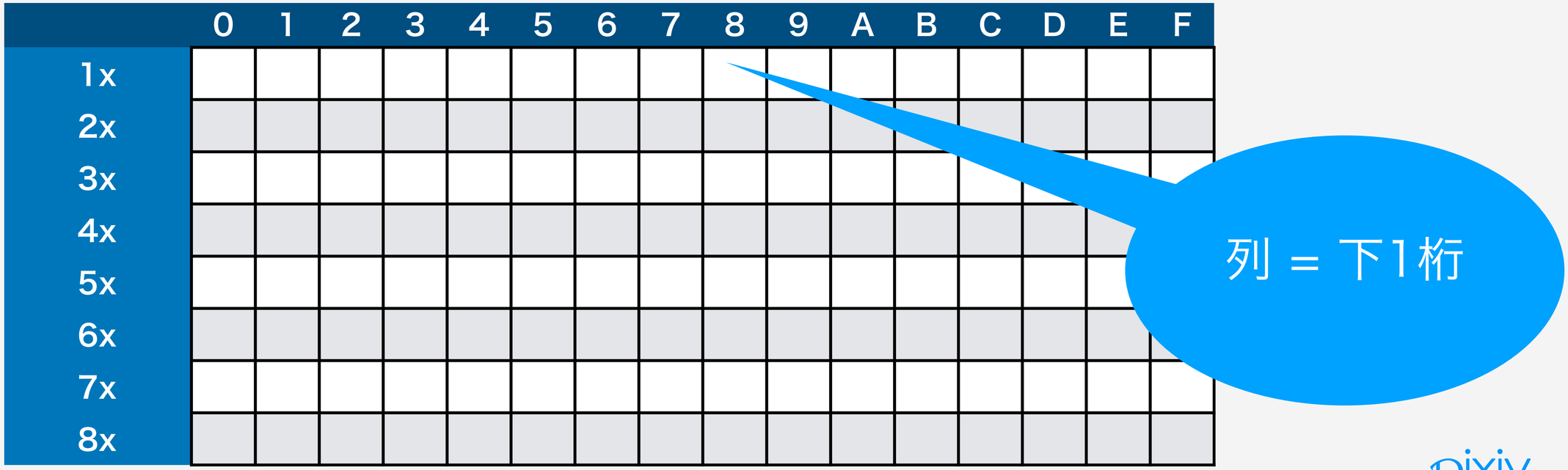
コンピュータのメモリ
＝ データを格納する箱が並んでいる



pixiv

データを整列して見る

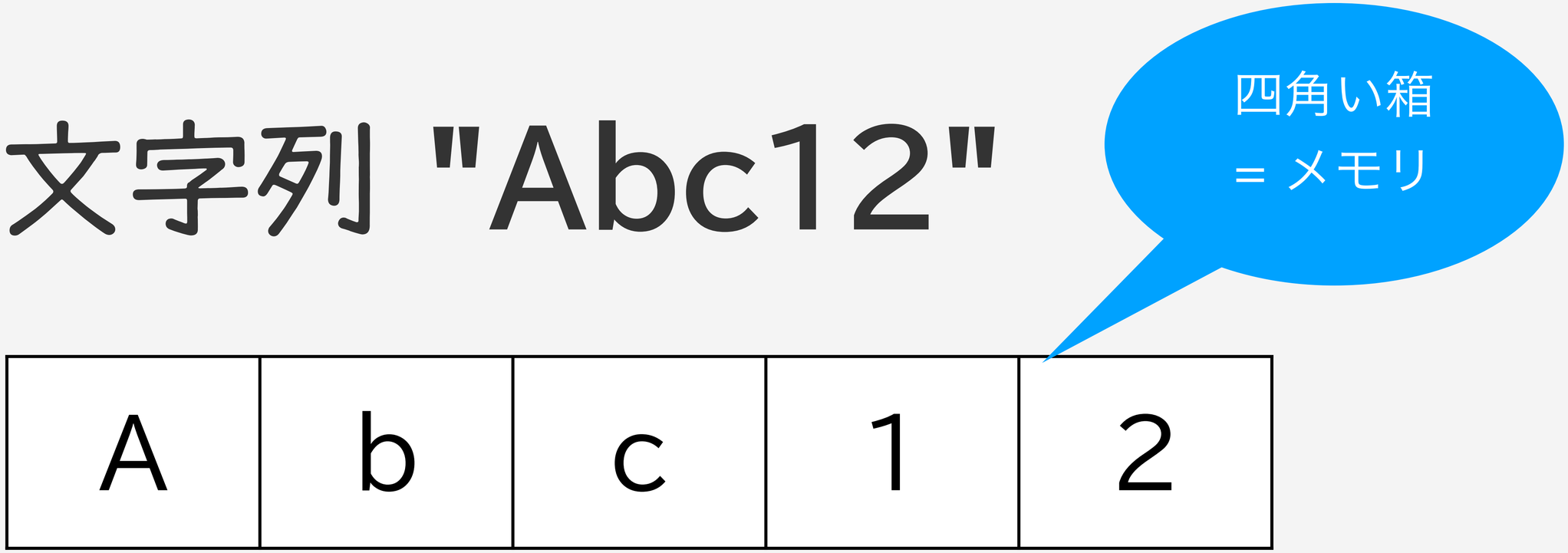
1列で表示するとわかりにくいので
16個単位で折り返すことがよくある



pixiv

非常に大雑把なイメージ

文字列 "Abc12"



pixiv

ところでコンピュータって
0と1の世界
なのでは…？

0と1の世界

0と1の世界
＝ binary (2進法)

0と1の世界
＝ binary (2進法)
＝ bit (2値データ)

粒度が細かすぎる

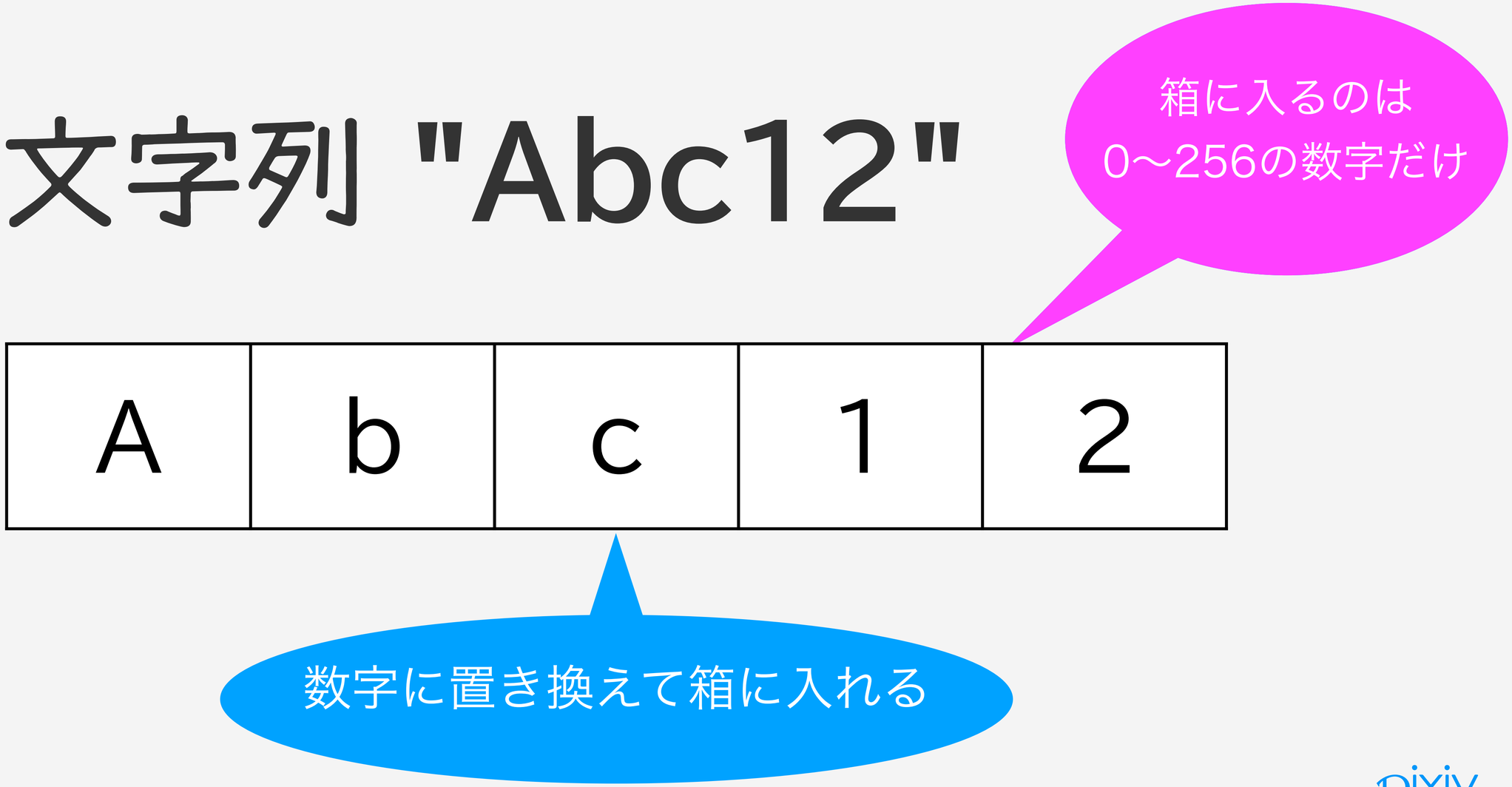
粒度が細かすぎる
→ まとめて扱おう

粒度が細かすぎる
→ まとめて扱おう
＝ bytes（バイト）

pixiv

非常に大雑把なイメージ (再掲)

文字列 "Abc12"



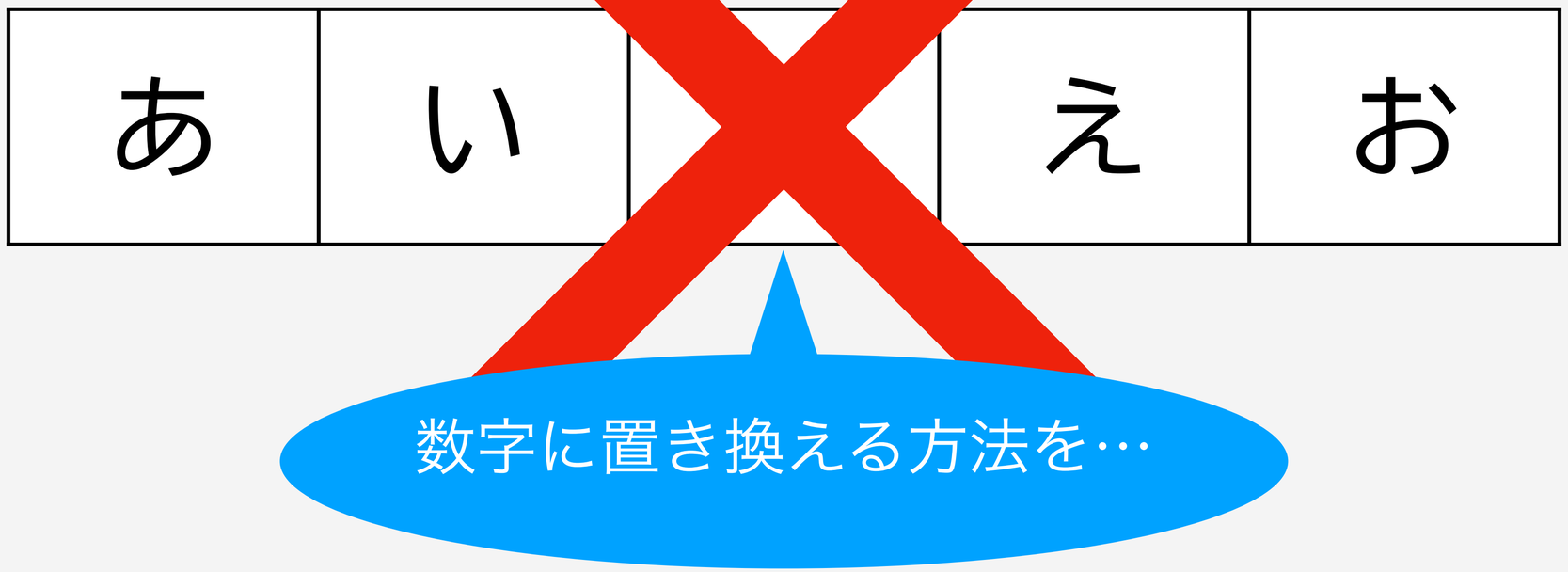
pixiv

データをまとめて扱う

- コンピュータ上のデータを0と1だけで表現するのはわかりにくい
- 現代では「**1バイト＝8個のビット**」の塊として扱うのが標準的
  - 歴史的には、1バイト＝7個のビットで扱う流れもあった
  - この「7」という数は現在までひっそりと影響を残している…
- 8バイトの場合、0～2^8(=256)までの数を格納できる

日本語文字列をメモリに入れよう

文字列"あいうえお"



pixiv

文字をデータに
してみよう

英語の文字をデータにしてみよう

- A, B, C, ... Z までのアルファベットだけを扱う
- アルファベット26種類なので、
  5bit ＝ 2^5 ＝32(種類)で表現できる

2進数	10進数	文字
00000	0	A
00001	1	B
00010	2	C
00011	3	D
00100	4	E
00101	5	F
00110	6	G
00111	7	H
…		…
11001	25	Z

ひらがなをデータにしてみよう

- あいうえお… わ を ん までの五十音を扱えるように
  - 6bit ＝ 2^6 ＝64(種類)に収まる

2進数	10進数	文字
000000	0	あ
000001	1	い
000010	2	う
000011	3	え
000100	4	お
…	…	…
101100	44	わ
101101	45	を
101110	46	ん

ひらがなをデータにしてみよう

- あいうえお… わ を ん までの五十音を扱えるように
  - 6bit ＝ 2^6 ＝64(種類)に収まる
- カタカナや濁点を扱うには… どうすれば？

2進数	10進数	文字
000000	0	あ
000001	1	い
000010	2	う
000011	3	え
000100	4	お
…	…	…
101100	44	わ
101101	45	を
101110	46	ん

ひらがなをデータにしてみよう

- あいうえお… わ を ん までの五十音を扱えるように
  - 6bit ＝ 2^6 ＝64(種類)に収まる
- カタカナや濁点を扱うには… どうすれば？
  - 💡 ＼ﾋﾟｺｰﾝ／ そうだ、先頭ビットを濁点とカタカナに

2進数	10進数	文字
000000	0	あ
000001	1	い
000010	2	う
000011	3	え
000100	4	お
…	…	…
101100	44	わ
101101	45	を
101110	46	ん

ひらがなをデータにしてみよう

- あいうえお… わ を ん までの五十音を扱えるように
  - 6bit ＝ 2^6 ＝64(種類)に収まる
- カタカナや濁点を扱うには… どうすれば？
  - 💡 ＼ﾋﾟｺｰﾝ／ そうだ、先頭ビットを濁点とカタカナに
  - ↑ 小文字とか半濁点のこと考えてないよね…

2進数	10進数	文字
000000	0	あ
000001	1	い
000010	2	う
000011	3	え
000100	4	お
…	…	…
101100	44	わ
101101	45	を
101110	46	ん

16進数表

- 文字エンコーディングとかバイナリファイルを向き合っているとちょくちょくこの形式の表を見ることになるかと思います
  - メモリやディスク上のデータの表示（バイナリエディタ）
  - エンコーディングのバイトデータと文字の対応表

そういうことを考えると
文字コードを設計できる

現在のコンピュータは
どうやって文字を扱うか

ASCII (アスキー)

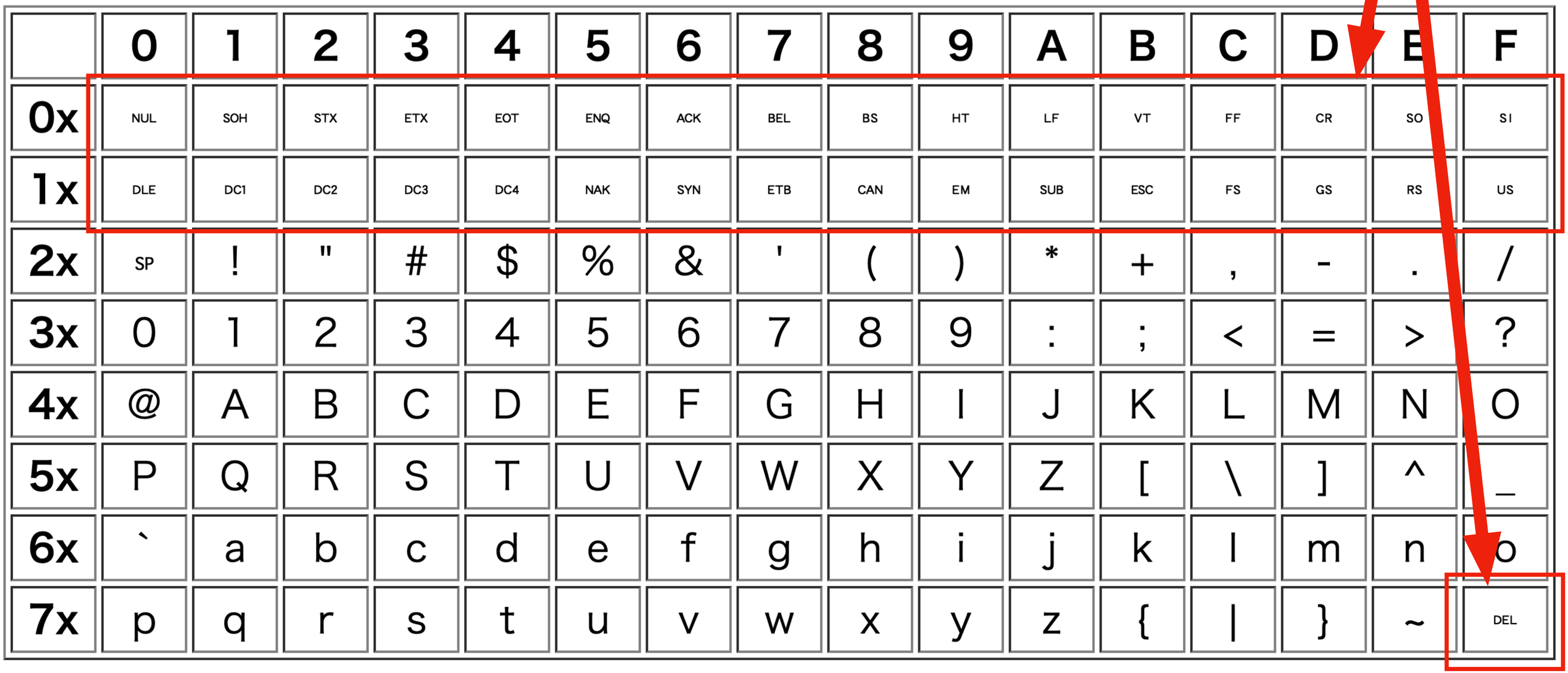
- American Standard Code for Information Interchange
  - ↑ 正式名称は覚えなくていい
- いわゆる「半角文字」「半角英数」とか呼ばれる文字コード
  - アメリカ英語で使用されるアルファベット(ラテン文字)をカバー
- 0～127までの7bitだけを使うので、1バイト=8bitとして扱うと必ず先頭が0になる

Unicode

- 大統一文字コード
  - 世界中の文字を統合して一個の表にしようぜ！ プロジェクト
  - 表に割り当てられた文字を「コードポイント」と呼ぶ
    - U+1234 のように書く
- 最初は16bit = 2^{16} = 65536種類の空間あれば十分じゃね？と思われていたらしい

UTF-8

- 8bit固定で1～4バイトの可変長エンコーディング
  - ASCIIの文字(いわゆる半角文字)は1バイトで扱える
  - そのほかの文字は可変長
    - 典型的な日本語の文字は3バイト
    - 半角カナは4バイト


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

21世紀の環境で Unicode以外を 採用する動機は激減

pixiv

かつてはそうではなかった（と伝え聞く）

歴史的な事例

有名な逸話

名言こぼれ話

本当にギリギリ！『DQⅠ』のデータ容量

『DQI』の発売当初はカートリッジに保存できるデータ容量がいまより格段に小さく、完成にこぎつけるには極限までデータ容量を削る必要があった。その削減は相当で、下表のカタカナのうち、赤字のものしかゲームに搭載していないという凄まじさだ。

そして、ローラ姫を抱きかかえた主人公のグラフィックも、容量不足から考えられた苦肉の策だった。しかし、それが「ゆうべは おたのしみでしたね」という名ゼリフを生んだと考えると……。なんだか感慨深い言葉にも思えてくるのでは？

ワ	ラ	ヤ	マ	ハ	ナ	タ	サ	カ	ア
	リ		ミ	ヒ	ニ	チ	シ	キ	イ
ヲ	ル	ユ	ム	フ	ヌ	ツ	ス	ク	ウ
	レ		メ	ヘ	ネ	テ	セ	ケ	エ
ン	ロ	ヨ	モ	ホ	ノ	ト	ソ	コ	オ

『DQⅠ』に搭載されたカタカナ文字（赤字）

『しんでしまうとは なにごとだ!』堀井雄二（編集・執筆）
株式会社スクウェア・エニックス
デジタル版 Ver.1.00 2018年9月1日（Kindle）
p11より引用

ドラゴンクエスト(ファミリーコンピュータ)

- 1986年5月27日発売
- **512Kbit**ロムカセット（64KBytes）
  - プログラムROMとキャラクタROMがそれぞれ32キロバイト
    - モンスターやキャラクターなどの画像とフォントなどを
      全部この32キロバイトに押し込める必要がある
    - プログラムとテキストデータが残り32キロバイト

ドラゴンクエスト (FC) 解析資料

Published on: 2022-03-24
Updated on: 2022-03-28

文字コード

文字コードの範囲は `0..=0x56` | `0x58..=0x6C` で、割り当ては下表の通り。
'※' は特殊文字。

	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xA	xB	xC	xD	xE	xF
0x	0	1	2	3	4	5	6	7	8	9	あ	い	う	え	お	か
1x	き	く	け	こ	さ	し	す	せ	そ	た	ち	つ	て	と	な	に
2x	ぬ	ね	の	は	ひ	ふ	へ	ほ	ま	み	む	め	も	や	ゆ	よ
3x	ら	り	る	れ	ろ	わ	を	ん	っ	ゃ	ゅ	ょ	※	※	※	メ
4x	ラ	ロ	※	ル	レ	コ	ン	タ	シ	ホ	イ	マ	カ	ム	ー	。
5x	、	※	※	゛	゜	＊	※		：	※	※	※	※	※	※	※
6x	？	！	「	H	M	P	G	E	ミ	ス	キ	ト	…			

カタカナの 'へ', 'リ' は存在せず、ひらがなの 'へ', 'り' で代用されている。

特殊文字は以下の通り:

ドラゴンクエスト（FC）解析資料 - 文字コード より引用（2025年1月28日閲覧）

データをじっと見る

実は結構大事なことだったりします
まずはCHRエディタで開くことにでもしましょうか？
R@Mはデータとプログラムによって構成されているのですが今ここで解析するのはデータ部分です
まずはデータの中のCHRは

まぁ当たり前って感じですね
CHRは圧縮されてなければこのように画像として見えます
まぁ画像として見えなきゃエディタとしての意味は無いのですがね(^^;;;;

文字コード

文字コードの範囲は `0..=0x56` | `0x58..=0x6C` で、割り当ては下表の通り。
'※' は特殊文字。

	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xA	xB	xC	xD	xE	xF
0x	0	1	2	3	4	5	6	7	8	9	あ	い	う	え	お	か
1x	き	く	け	こ	さ	し	す	せ	そ	た	ち	つ	て	と	な	に
2x	ぬ	ね	の	は	ひ	ふ	へ	ほ	ま	み	む	め	も	や	ゆ	よ
3x	ら	り	る	れ	ろ	わ	を	ん	っ	ゃ	ゅ	ょ	※	※	※	メ
4x	ラ	ロ	※	ル	レ	コ	ン	タ	シ	ホ	イ	マ	カ	ム	ー	。
5x	、	※	※	゛	゜	＊	※		：	※	※	※	※	※	※	※
6x	？	！	「	H	M	P	G	E	ミ	ス	キ	ト	…			

ドラゴンクエスト（FC）解析資料 - 文字コード より引用（2025年1月28日閲覧）

第三章 自分だけのオリジナルマップを作ろう（kuwatan.jp）より引用
（2025年1月28日閲覧）

ドラゴンクエストV (スーパーファミコン)

- 1992年9月27日発売
- **12MBit**ロムカセット (1.5MBytes)
  - DQ1の時代から考えれば無限に広く見えるが…
  - プログラム・テキスト・グラフィック・音楽などあらゆる要素がリッチに
  - テキストデータはハフマン符号による可変長ビット(!)
    ドラクエ命 第 II 部 解析 第 3 章 SFC 版ドラクエ 5 (1992) 3.6. テキスト解析

ポケットモンスター(ゲームボーイ)

- 1996年2月27日発売（DQのほぼ10年後）
- **8Mbit**ロムカセット（1MBytes）：DQの16倍
  - DQ1の時代から考えれば無限に広く見えるが…
    151匹のポケモンを押し込むには明らかに大変
  - 文字列は独自の2バイトコードを定義している
    - アルファベットは13文字しか含まれていない(!)

pokemonbug @ ウィキ

文字コード対応表

Pokémon Bug Litches(リンク切れ)より

第一・第二世代共に文字コードはおおむね統一されているが、第二世代では一部違う使われ方をする文字コードが存在する。

メッセージウインドウ(第一世代)

メッセージ版

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	[無効]	イ゛	ウ゛	エ゛	オ゛	ガ	ギ	グ	ゲ	ゴ	ザ	ジ	ズ	ゼ	ゾ	ダ
1	ヂ	ヅ	デ	ド	ナ゛	ニ゛	ヌ゛	ネ゛	ノ゛	バ	ビ	ブ	ボ	マ゛	ミ゛	ム゛
2	ィ゛	あ゛	い゛	う゛	え゛	お゛	が	ぎ	ぐ	げ	ご	ざ	じ	ず	ぜ	ぞ
3	だ	ぢ	づ	で	ど	な゛	に゛	ぬ゛	ね゛	の゛	ば	び	ぶ	べ	ぼ	ま゛
4	パ	ピ	プ	ポ	ぱ	ぴ	ぷ	ぺ	ぽ	ま゛	み゛	[文字送り]	[自動送り]	も゛	[1行改行]	[改行]
5	[終]	[文字送り]	[主人公]	[ライバル]	ポケモン	[文字送り]	……	[無効]	[改ペ]	[ポケ1]	[ポケ2]	パソコン	わざマシン	トレーナー	ロケットだん	､､゛
6	A	B	C	D	E	F	G	H	I	V	S	L	M	:	ぃ	ぅ
7	「	」	『	』	・	…	ぁ	ぇ	ぉ	┌(二重)	＝	┐(二重)	‖	└(二重)	┘(二重)	[空白]
8	ア	イ	ウ	エ	オ	カ	キ	ク	ケ	コ	サ	シ	ス	セ	ソ	タ
9	チ	ツ	テ	ト	ナ	ニ	ヌ	ネ	ノ	ハ	ヒ	フ	ホ	マ	ミ	ム
A	メ	モ	ヤ	ユ	ヨ	ラ	ル	レ	ロ	ワ	ヲ	ン	ッ	ャ	ュ	ョ
B	ィ	あ	い	う	え	お	か	き	く	け	こ	さ	し	す	せ	そ
C	た	ち	つ	て	と	な	に	ぬ	ね	の	は	ひ	ふ	へ	ほ	ま
D	み	む	め	も	や	ゆ	よ	ら	り	る	れ	ろ	わ	を	ん	っ
E	ゃ	ゅ	ょ	ー	゛	゜	?	!	。	ァ	ゥ	ェ	[白カーソル]	[黒カーソル]	▼	♂
F	円	×	.	／	ォ	♀	0	1	2	3	4	5	6	7	8	9

文字コード対応表 - pokemonbug @ ウィキ - atwiki より引用（2024年12月22日閲覧）

岩田　そこで、わたしは任天堂の人でもなかったのに、なぜか
任天堂と石原さんの間を取り持つようなことをしていました（笑）。

石原　そうでした。

岩田　当時、わたしは任天堂の人ではなくて
ＨＡＬ研究所の社長だったのですが、
同時にクリーチャーズの役員でもあったご縁があって
『赤・緑』の海外版のローカライズがどうやったらできるのか、
その分析の仕事に関わることになったんですね。
そこで『赤・緑』のプログラムソースをあずかって、
それを読み込むようなことをして、
「こうすればローカライズできますよ」と
任天堂の担当部門につなぐようなことをやりました。

石原　それから、ほとんど同じ時期に
『ポケモンスタジアム』（※16）にも。

社長が訊く『ポケットモンスター ハートゴールド・ソウルシルバー』より引用（2025年1月28日閲覧）

今や文字コードを独自設計したり、フォントをビット単位で押し込めるような時代ではなくなった

サイズ削減のためバンドルするフォント
ファイルの文字種を削ったり、
独自の文字を表示するためにUnicodeの
私用領域を使うことはあると思う