文字化け

とみたまさひろ

bitの会

2015-06-06

自己紹介

- とみた まさひろ
- 長野県北部在住プログラマー
- 言語: C(1989〜) Ruby(1998〜)
- 日本MySQLユーザ会代表
- 長野ソフトウェア技術者グループ(NSEG)

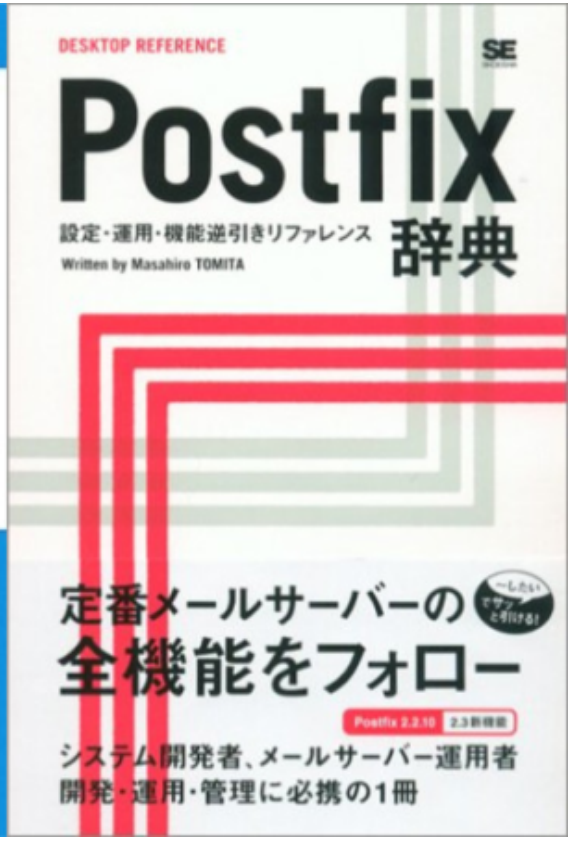
自己紹介

- http://**tmtms**.hatenablog.com
- http://twitter.com/**tmtms**
- https://github.com/**tmtm**
- MySQL 3.21 の日本語対応 (1998)
- MySQLのRubyバインディング作成 (1998)

執筆書籍

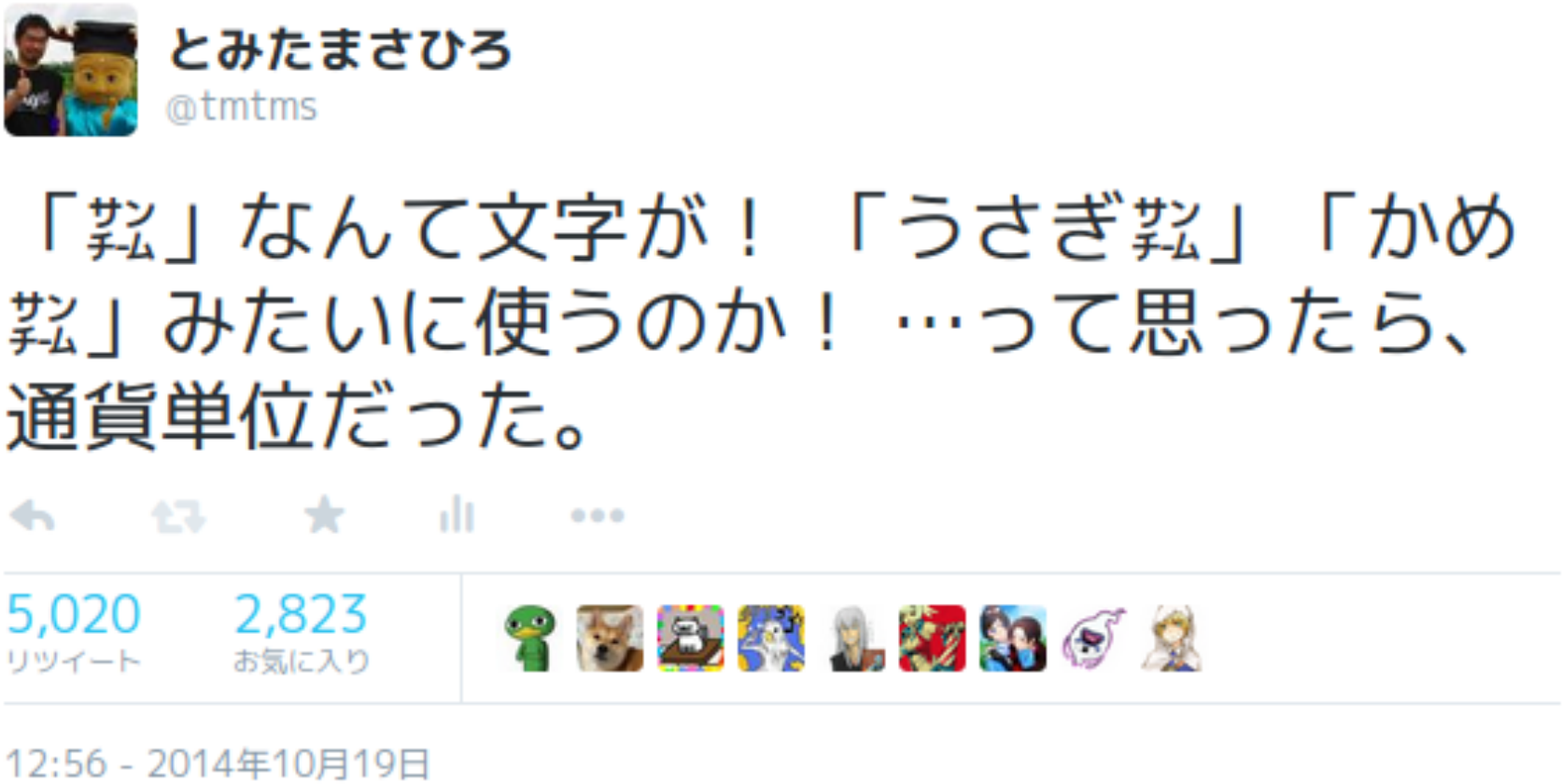

MySQL
第3版
徹底入門
～5.5新機能対応～
日本MySQLユーザ会
オープンソースRDBMSをマスターするための定番書
進化したMySQL 5.5
お待ちかねの新版！改訂！
Linux/Mac/Windows環境へのMySQL導入に対応
C言語/PHP/Ruby/Perl/Java/Access/Excel/Wordからのデータベースアクセスを解説
レプリケーション機能やUnicode対応
充実するGUI管理ツールなど、
知りたいところを丁寧に解説
SE


DESKTOP REFERENCE
SE
MySQL
辞典
コマンド・関数・パラメータ、
目的別逆引きリファレンス
Written by Kei SAKAI / Masahiro TOMITA / Noriyuki TSURUTA
Shinichi TSURUNAGA / Yoshinori MATSUNOBU
定番 オープンソースDBの
すべてのエッセンスを
凝縮
データベースの設計・開発から運用・管理まで
エンジニア必携の一冊！
MySQL
5.0 対応


DESKTOP REFERENCE
SE
Postfix
設定・運用・機能逆引きリファレンス
辞典
Written by Masahiro TOMITA
定番メールサーバーの
全機能をフォロー
Postfix 2.2.10
2.3新機能
システム開発者、メールサーバー運用者
開発・運用・管理に必携の1冊

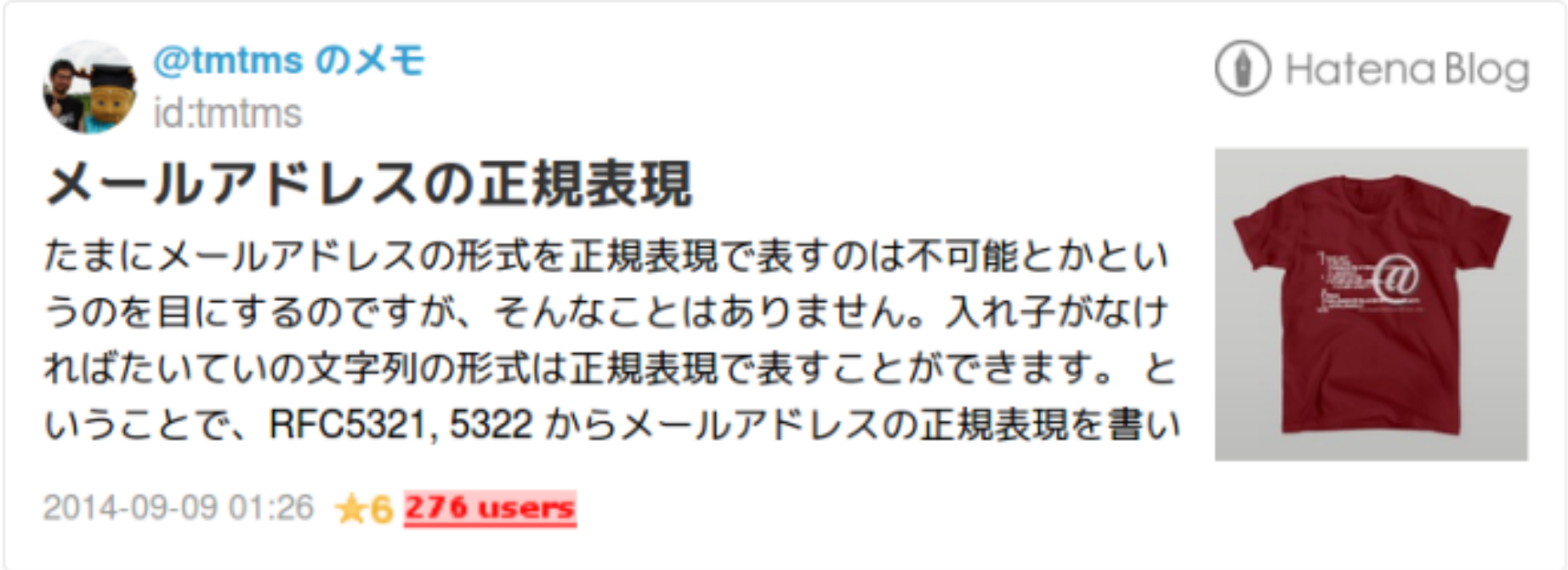
自己紹介

- もっともRTされたツイート



自己紹介

- もっともブクマされたブログ

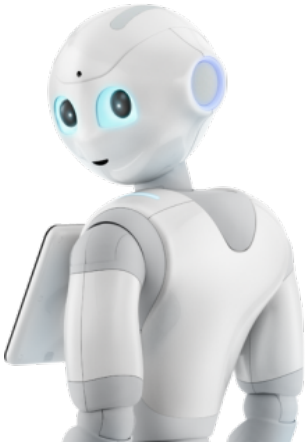


宣伝その1 NSEG

- http://nseg.jp
- 長野のIT技術者のゆるい集まり
- 2010年2月発足
  (発起人:おびなたさん、しむやさん、おかもとさん)
- 毎月勉強会開催(今までに63回)主に長野市
- 「理論から学ぶデータベース実践入門」読書会（隔週水曜日） at ギークラボ長野

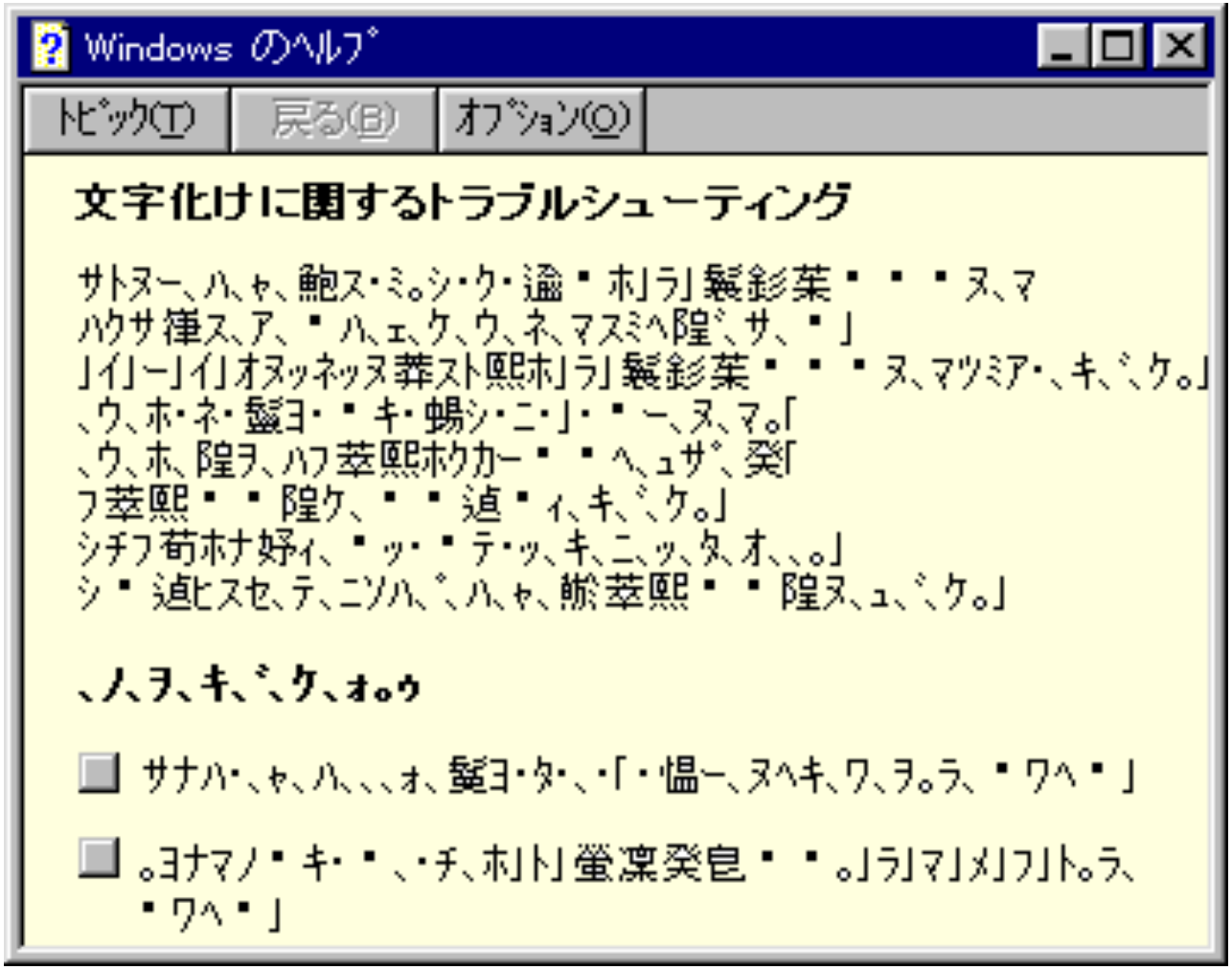
宣伝その2 ギークラボ長野

- http://geeklab-nagano.com
- 県庁近く
- 無料で利用できる作業スペース
- 電源 & WiFi & Pepperくん あり
- 6/13 ハンズオン「はじめての Unity ～2D ゲームを作ってみよう～」



文字化け

文字化けに関するトラブルシューティング



http://www.geocities.co.jp/Playtown/7711/gallaly/trableshooting.html

文字化けを直してみた

残念ながら現バージョンのＷｉｎｄｏｗｓでは
文字化けをなおすことは出来ません。
２０２５年発売予定のＷｉｎｄｏｗｓでは対応します。
このトラブルシューティングでは、
このような問題の原因を突き止め、
問題を解決する手順を示します。
質問の答えをクリックしてください。
手順に従って進みながら問題を解決できます。

どうしますか？

□ 仕方がないから「ダイアログで遊ぼう」を閲覧。
□ 「渡部シンイチのＤｅｊａｖｕ　ＷＯＲＬＤ」を閲覧。

文字化けの原因

1. 文字コードが正しくない

文字コード

- Charset, 文字集合
- エンコーディング方式
- シフトJIS, 日本語EUC, UTF-8 等
- 文字のバイト表現

日本語の文字コード

- UTF-8 (Unicode 今は普通はこれ)
- SHIFT_JIS (Windows)
- EUC-JP (昔の UNIX)
- ISO-2022-JP (メール)

同じ文字でも異なるコード

「あ」

- UTF-8 (E3 81 82)
- EUC-JP (A4 A2)
- SHIFT_JIS (82 A0)

同じバイト列でも別の文字

C2 A9

- UTF-8「©」
- EUC-JP「息」
- SHIFT_JIS「ツゥ」

バイト列だけでは
文字コードはわからない

ところで

日本語文字のことを「**2バイト文字**」と言ってませんか？

UTF-8は日本語はだいたい**3バイト**

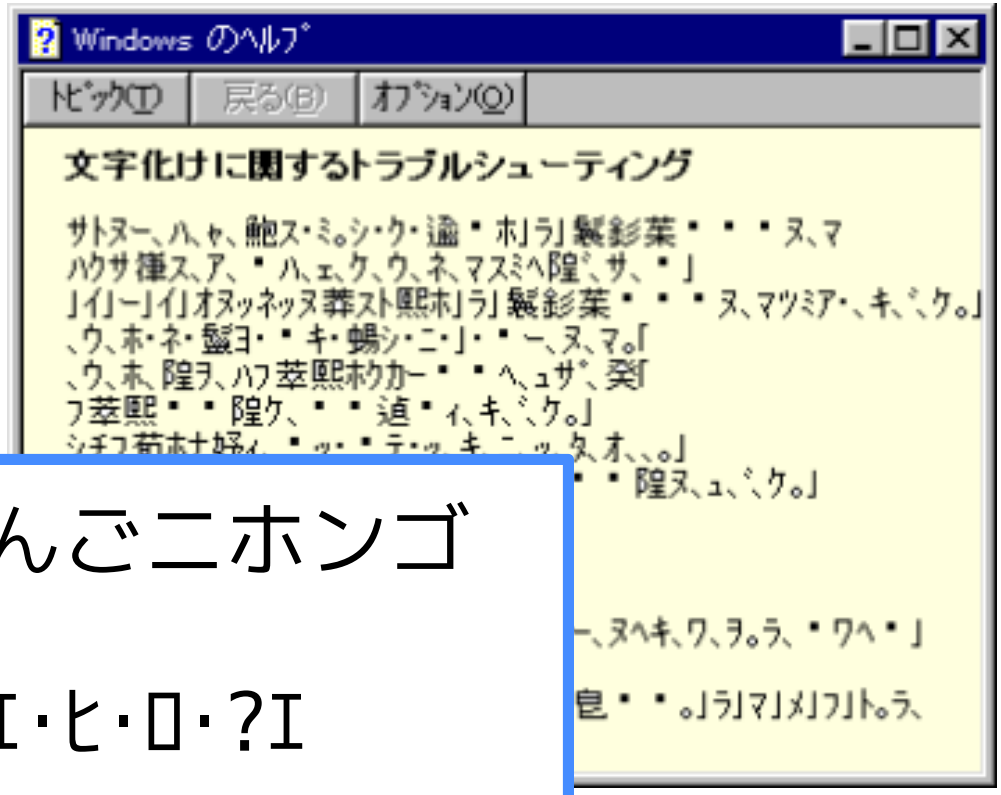
- 「あ」(E3 81 82)
- 「ｱ」(EF BD B1)

文字化けした文字列の元の文字コードの見分け方

慣れるとパッと見でわかる

EUC-JP を SHIFT_JIS として表示

半角カナが多い



日本語にほんごニホンゴ

ﾆ?ﾜｸ?ﾋ､ﾛ､?ｴ･ﾋ･ﾛ･?ｴ

UTF-8 を SHIFT_JIS として表示

画数の多い漢字の中に半角カナが少々混じってる

日本語にほんご二ホンゴ

謌･譖ｬ隱槭↓縺ｻ繧薙＃繝九?繝ｳ繧ｴ

ISO-2022-JP の ESC 落ち

ASCII文字だけで構成されて「$B」「(B」がある

日本語にほんごニホンゴ

?$BF|K\8l$K$[$s$4%K%[%s%4?(B

文字化けの原因

2. 文字コードが不明

自動判定もできるけど
完全じゃない

同じバイト列でも別の文字(再掲)

C2 A9

- UTF-8「©」
- EUC-JP「息」
- SHIFT_JIS「ﾂｩ」

自動判定

- バイト列が文字コードの正当な範囲にあるかどうかで判定
- 文字列が短いと難しい
- 文字列が長ければ精度は上がる
- へんなバイトが混在してたらアウト
- 日本語じゃないのに日本語と誤判定したり

EUC - Extended Unix Code

- EUC-JP：日本語EUC
- EUC-KR：韓国語EUC
- EUC-CN：簡体字中国語EUC

文字コードの範囲は同じだけど文字が違う

- BB FA：字(JP) / 샙(KR) / 机(CN)

文字境界

- EUC
  - 1バイト文字 (00-7F) ← ASCII
  - 2バイト文字 (A1-F4 | A1-FE)
  - 文字列の先頭から見ないとわからない
- SHIFT_JIS
  - 1バイト文字 (00-7F A1-DF) ← JISX0201
  - 2バイト文字 (81-9F E0-FC | 40-7E 80-FC)
  - **2バイト目に ASCII 範囲含んでてやばい**

文字境界

- UTF-8
  - 1バイト文字 (00-7F) ← ASCII
  - 2バイト文字 (C2-DF | 80-BF)
  - 3バイト文字 (E0-EF | 80-BF | 80-BF)
  - 4バイト文字 (F0-F7 | 80-BF | 80-BF | 80-BF)

文字境界判別しやすい

文字化けの原因

3. 機種依存文字

ISO-2022-JP

- いわゆる「JISコード」
- 日本語メールで使われる文字コードは未だに ISO-2022-JP が主流
- \ と ￥ を使い分けできる
- ISO-2022-JP に無い文字
  - ① ② ③ Ⅰ Ⅱ Ⅲ ㈱ ㎏ ㎝ 髙 﨑 ｶﾀｶﾅ

ISO-2022-JP に無い文字を ISO-2022-JP と偽ってメールする奴がいる

文字化け

昔の Mac だと ① が (月) に見えたり

だいたいマイクロソフトが悪い

MS の ISO-2022-JP

- CP50221
- ISO-2022-JP に無い文字を含む
  - ① ② ③ Ⅰ Ⅱ Ⅲ ㈱ ㎏ ㎝ 髙 﨑 ｶﾀｶﾅ
- \ と ￥ の区別がつかない

まともなメールアプリが「ISO-2022-JP」を信じると文字化け

まともなメールアプリが責められる「Outlookだと見えるのに」

ISO-2022-JP は CP50221 とみなして表示

ISO-2022-JP を ISO-2022-JP として扱うアプリが絶滅

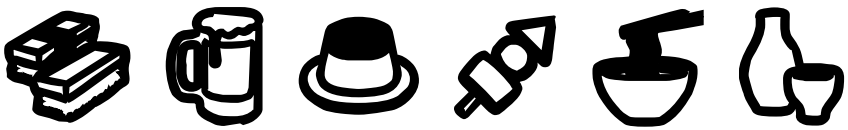
文字コードの置き換え

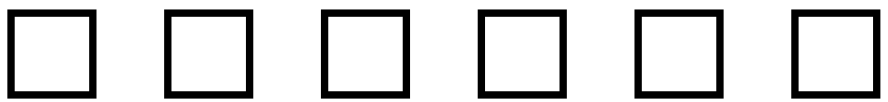
- ISO-2022-JP : CP50221
- EUC-JP : CP51932
- SHIFT_JIS : CP932

だいたいマイクロソフトが悪い

文字化けの原因

4. フォントが無い





対応フォント入れれば解決

発表の途中ですがお知らせです

FAQ: 下のウサギとカメは何？

Rabbit

- Ruby製プレゼンツール
- http://rabbit-shocker.org/
- ウサギとカメはタイムキーパー
- カメに追い抜かされなければ時間内
- プレゼンがつまらなくても和む
- ソースはテキスト(RD / Markdown)
- PDF ビューアとしても利用可

Ruby

Ruby

- プログラミング言語
- オブジェクト指向スクリプト言語
- オブジェクト毎に異なる文字コードが可能
- ファイル毎に異なる文字コードが可能
- 嫌な予感…

ファイル

同じファイルからの読み込みでも異なる文字コードになることも

```
f = File.open("何か.txt", "r:utf-8")
f.gets       #=> 1行文字列 UTF-8
f.read(10)   #=> 10バイトデータ ASCII-8BIT
```

異なる文字コード同士

```
utf8 = "ほげ"
sjis = "ほげ".encode("cp932")
utf8 == sjis          #=> 偽
utf8 + sjis           #=> エラー
utf8 =~ /#{sjis}/     #=> エラー

utf8 = "ASCII"
sjis = "ASCII".encode("cp932")
utf8 == sjis          #=> 真
```

混ぜるな危険！

全部 UTF-8 に統一すれば安全

MySQL

オープンソースの RDBMS

文字コード

- クライアントライブラリ
- クライアント-サーバー接続
- データベース
- テーブル
- カラム

…毎に設定可能

嫌な予感…

utf8 と utf8mb4

- UTF-8 文字コード
- utf8 は 3バイトまで
- utf8mb4 は 4バイトまで
- 絵文字( や )は4バイト

utf8カラムに4バイト文字を入れようとすると

接続が utf8 の場合:

```
mysql> insert into t (c) values ('今日は📠と🍺');
Query OK, 1 row affected, 1 warning (0.09 sec)
mysql> select * from t;
+------------+
| c          |
+------------+
| 今日は     |
+------------+
```

utf8カラムに4バイト文字を入れようとすると

接続が utf8mb4 の場合:

```
mysql> insert into t (c) values ('今日は📠と🍺');
Query OK, 1 row affected, 1 warning (0.09 sec)
mysql> select * from t;
+----------------+
| c              |
+----------------+
| 今日は?と?     |
+----------------+
```

utf8mb4カラムに4バイト文字を入れようとすると

接続が utf8mb4 の場合:

```
mysql> insert into t (c) values ('今日は📚と🍺');
Query OK, 1 row affected (0.06 sec)
mysql> select * from t;
+----------------------+
| c                    |
+----------------------+
| 今日は📚と🍺          |
+----------------------+
```

utf8mb4カラムに4バイト文字を入れようとすると

接続が utf8 の場合:

```
mysql> insert into t (c) values ('今日は📚と🍺');
Query OK, 1 row affected, 1 warning (0.06 sec)
mysql> select * from t;
+----------------------+
| c                    |
+----------------------+
| 今日は????と??       |
+----------------------+
```

4バイト文字を utf8 接続から取り出すと

```
mysql> select * from t;
+------------------+
| c                |
+------------------+
| 今日は?と?       |
+------------------+
```

「?」の調査

```
mysql> select hex(c) from t;
+----------------------------------------+
| hex(c)                                 |
+----------------------------------------+
| E4BB8AE697A5E381AFF09F8DA3E381A8F09F8DBA | ←ちゃんと入ってる
| E4BB8AE697A5E381AF3F3F3FE381A83F3F     | ←データが「?」
+----------------------------------------+
```

混ぜるな危険！

全部 utf8 / utf8mb4 に統一すれば安全

まとめ

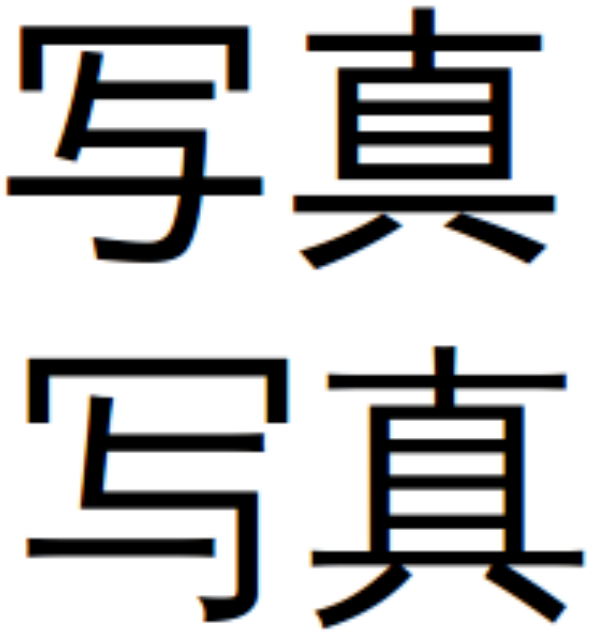
歴史的経緯により日本語には複数の文字コードが使用されている

これからは UTF-8 を使えばみんなハッピー

本当に？

Unicode の闇

中国語と日本語で文字が統一



合成文字

- が (E3 81 8C)
- が (E3 81 8B E3 82 99)
- 「か」＋「゛」の二文字
- 比較には正規化が必要

色付き絵文字の肌色問題

- 絵文字は日本発祥
- 肌色が一種類
- 「人種差別だ！」
- 合成文字で解決

これからの絵文字の実装指針、UTR #51“Unicode Emoji”とはなにか -INTERNET Watch
http://internet.watch.impress.co.jp/docs/special/20150131_686161.html

俺達の戦いはこれからだ！

おわり