第 6 回：Unicode での文字の取り扱い

本日のポイント：

- Unicode のもう少し詳しい紹介
  - Unicode の歴史と現状
  - Unicode で扱うことのできるさまざまな文字
  - Unicode の基本原則
- Unicode 文字の入力：文字の探し方
  - 入力補助ツールを利用して入力：Word2002 の「記号と特殊文字」，IME2002 の「IME パッド」，Windows の「文字コード表」
  - 文字のコード値を調べて入力：IME2002 と Windows の Unicode コード値入力機能，Word 2002 と EmEditor のコード変換・表示機能
- Unicode を扱う際に気をつけるべきこと（1）
  - 字体の異なる漢字の統合（Unification）
  - UTF-16LE と UTF-16BE，BOM
  - よく似た記号

1. Unicode のもう少し詳しい紹介

1.1. Unicode の歴史

- 1984 年，ISO (International Organization for Standardization) と IEC (International Electrotechnical Commission) の 2 つの国際標準化団体が合同で委員会を設立し，ISO/IEC 10646 という統一されたエンコード規格の検討をはじめる。(初期の規格は ISO で否決され，廃案になってしまった。)
- 1987 年ごろから企業間で Unicode の策定機運が高まり，1991 年に正式に非営利団体 Unicode Consortium（法人名 Unicode, Inc）が設立される。同団体の Unicode Technical Committee (UTC) が実際の規格（Unicode Standard「Unicode 標準」）の策定と技術レポート（Unicode Technical Report, UTR）の発行を行っている。
- 1991 年, 業界標準としての Unicode と国際標準規格の ISO/IEC 10646 が同一の文字集合 character set とコード割り当て codepoint (文字とコードの対応づけ) を用いることで合意。
- 1993 年，統一された内容に基づく初めての規格書が両団体から発表される（ISO/IEC 10646-1:1993； Unicode Standard, Version 1.1)。
- 以来，２つの標準は共同歩調をとり，収録文字の拡張と文字コードの理論的な枠組みの検討を ISO/IEC が, Unicode はソフトウエアへのコードの実装方法の開発を，という役割分担のもと，作業をおこなっている。
  - 2 つの規格 ISO/IEC 10646 と Unicode Standard は, 収録する文字とそのコード割り当てについては全く同じである。用語やコードの表記方法など，いくつか異なる点があるが, この授業では Unicode Standard に従って解説する)。

1.2. Unicode の現状

- Unicode の最新バージョンは以下のとおり。
  - ISO/IEC 10646-1:2000 Universal multi-octet Character Set (UCS) Part 1: Architecture and Basic Multilingual Plane (1993 年版の改訂版, 2000 年)
  - Unicode Standard, Version 4.0.1 (2004 年 3 月発行)
- Unicode コンソーシアム Web ページ（URL: http://www.unicode.org）から，Unicode

担当：千葉庄寿（schiba@reitaku-u.ac.jp）西田文信（fnishida@reitaku-u.ac.jp）

に関する多くの情報が入手できる。

- Unicode Standard は発展中の規格で，新しいバージョンがリリースされるたびに収録文字の拡張や仕様の改訂がおこなわれている（例えばユーロ記号はバージョン 2.1 (1998 年）で追加された)。現在でも多くの文字が規格化待ちのリストに入っている。
- Unicode の規格は順次更新されているが, 実際に利用できる Unicode のバージョンは, ソフトによってさまざまである。現在 Unicode の実装が最も進んでいる OS は Windows XP であり，Unicode 3.0 相当を利用できる（Windows2000 では Unicode 2.1)。

Unicode のバージョンごとの収録文字数

バージョン（公開)	文字数	漢字（ハングルを除く）の数	16bit 領域の空き
Unicode 2.0 (1996-07)	38,885	21,204	18,136
Unicode 3.0 (1999-09)	49,194	27,786 (拡張 A 6,582 文字ほか)	7,827
Unicode 4.0 (2003-04)	96,382 [1]	71,098 (拡張 B 42,711 文字ほか)	6,323

参考： 各国語の標準的なエンコード方式とその収録文字数（第 7 回で詳しく扱う）
日本語 Shift JIS: 6802;　　簡体字中国語 GB2312: 7445;
繁体字中国語 BIG5: 10353;　韓国語 EUC-KR: 8224

1.3. Unicode で扱うことのできるさまざまな文字

Unicode では，主要な現代語の文字はもちろん，音声記号などの特殊な記号や文字も収録されている。その結果，複数の言語や記号体系を自由に混在させて使うことができる。以下のサンプル Web ページを WWW ブラウザで閲覧し，その有効性を体感してみよう。

Unicode Transcriptions:
http://www.macchiato.com/unicode/Unicode_transcriptions.html
UTF-8 Sampler:
http://www.columbia.edu/kermit/utf8.html

Unicode では，文字は種類別に収録されている。以下の表は 16 ビット（0 か 1 かの 2 進数 16 桁 = 2^{16} = 65,536 個の文字を表現可能，16 進数であらわすと 0000～FFFF）で表現される文字の種類を示したものである（出典：清水『図解でわかる文字コードのすべて』p.64)。

[PDF 文書では省略]

[1] Unicode 3.1 (2001 年 3 月策定）より，16 ビット以上で表現する文字も含まれるようになり，収録可能文字が 2^{16} = 65,536 を超えた。

BMP (Basic Multilingual Plane の略) は ISO/IEC 10646 の用語で，16 ビットで表示できるコードの領域 (0000～FFFF) に収録する文字が規定されている。主要な文字や記号が収録されており，BMP で各文字にあてがわれているコードは，Unicode の主要なエンコード方式である UTF-16 でそのまま使われる。

1.4. Unicode の基本原則

Unicode の開発にあたって，10 の基本原則が掲げられている。

(1) 16 ビットの文字コード	16 ビットの固定長のエンコード方式
(2) 効率	コードが並ぶだけの単純なテキスト構造
(3) 字体でなく，文字	字体 glyph が異なるだけでは異なる文字として収録しない
(4) 文字のプロパティ	文字の特徴を記述
(5) プレーンテキスト	フォントの違いは表現しない
(6) 論理的な順番	テキストの方向や文字の組み合わせの順番を規定
(7) ユニフィケーション	異なる言語の字体が同じ文字は統一する
(8) 動的合成	アクセント記号つき文字は合成できる
(9) 等価な文字列の規定	一つのコードとして収録された合成文字は，必ず対応する組み合わせ文字列をもつ
(10) 変換可能性	Unicode と他のエンコード方式との対応を規定

解説 (要点のみ)：

(1) **16 ビット固定長の文字コード：** 西ヨーロッパ言語 ISO 8859-1 では 8 ビットが，日本語 Shift JIS では 8 ビット (1 バイト [2]) 長 と 16 ビット (2 バイト) 長の 2 種類の文字が混在して使われている。これに対し，Unicode では，文字は種類の別なく全て 16 ビット，つまり 8 ビット 2 つ分 (2 バイト) のコードで表す。このように文字あたりのビット数を統一することで，コンピュータのデータ処理が単純になる。
　また，16 ビット長のエンコード方式をとることで，収録できる文字数が飛躍的に増大し，単純計算でも $2^{16} = 65{,}536$ 個の文字が表現可能 [3] になる。Unicode では，テキストとして流通する全ての文字をコード化し，収録することが目的になる。(**なお，バイト，ビットについては次回授業で詳しく解説・実習する。**)

(3) **「文字」の収録：** 「言葉を書き表すときに。意味をもつ最小の構成要素」(*Unicode Standard 3.0* での文字の定義)。同じ意味をもつ文字に，書き方が 2 通りあった場合，両方を別々のコードとして登録してあると，どちらを使ったらよいかが分からなくなる。そこで，意味が同じである限り，文字には単一のコードで表し，字体 (「グリフ」Glyph という) が異なるだけでは別のコードを与えない。a でも **a** でも，*a* でも，同じ「a」(LATIN SMALL LETTER A) である。逆に，同じ文字でも，異なる意味で使われる場合には，異なるコードを与える。

(5) **プレーンテキスト** plain text**：** フォント情報を表すコードは含まれない。ただし，テキストの方向性については指示が可能になっている (授業では扱わない)。

---

[2] バイトという用語は 7 ビットでも 8 ビットでも使われるので，Unicode では特に 8 ビットを単位とするコードにはオクテット Octet (Octopus の Oct と同語源) という用語が用いられる。

[3] サロゲート surrogate という仕組みを利用すると，16 ビットの制限を越え，さらに 100 万を越える文字の使用が可能になる (ただし，Unicode 3.1 以降でこの方式を使い追加された文字を実際に利用できるソフトウエアやフォントはまだ多くなく，本授業ではこれ以上触れない)。

担当：千葉庄寿 (schiba@reitaku-u.ac.jp) 西田文信 (fnishida@reitaku-u.ac.jp)

(7) **文字統合（Unification）：** 日本や中国，台湾，韓国などで使われている漢字（Unicode では，China, Japan, Korea の頭文字をとって CJK Ideographs（CJK 漢字）と呼ばれる）のように，同じ文字群に属し，言語間で重複している文字は統合し，一つのコードを割り当てる（Unification については 3.1.で詳しく扱う)。

(10) **コード変換の機軸としての役割：** (1) により，Unicode には既存の文字エンコード方式に含まれる文字全てが収録されていることが期待される。よってあらゆる国，地域の文字エンコード方式を Unicode に変換して処理できる。また，ローカルなエンコード間の変換も Unicode を介することでより効率よくできる。

2. Unicode 文字の入力―文字の探し方―

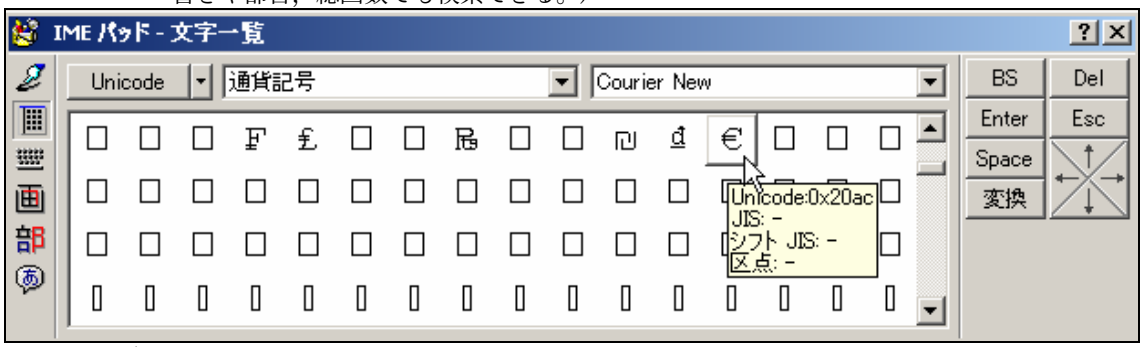
外国語の入力方法として最も有効なのは，Windows に用意されている入力ロケールを使う方法である。しかし，Unicode には通常あまり使わない文字や記号も大量に収録されており，それら文字を探して入力することも多く，入力ロケールを覚えるのでは効率が悪い。ここでは，Unicode に収録されている文字を探して入力するために利用できる補助入力ツールを紹介し，さらに文字を Unicode のコード値を使って入力する方法を紹介する。

2.1. 入力補助ツールを利用して入力

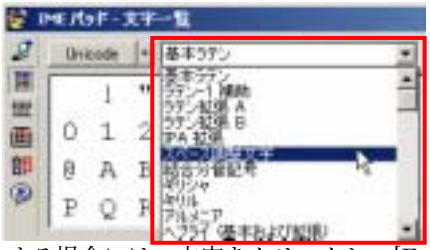
(1) Word2002 の [挿入]→[記号と特殊文字] (第 3 回資料 §3. で解説)

(2) IME2002 の「IME パッド」を利用

- ✓ 「文字一覧」アップレット を開く。(漢字は手書きや部首，総画数でも検索できる。)



- ✓ 「Shift JIS 順」と「Unicode 順」の 2 種類の文字コード順で文字の一覧を表示できる。日本語 Shift JIS に入っていない文字は Unicode 順で表示する必要がある。
- ✓ 文字の上にマウスをのせると Unicode や ShiftJIS でのコード値が表示される。上記のユーロ記号の場合，Unicode の文字コードとして 0x20ac と表示されており，コード値は 20AC である。(0x はコードが 16 進数で表記されていることを表す。)
- ✓ 使用するフォントを指定できる (指定したフォントに収録されていない文字は中黒(・)や四角(□←通称「トーフ」)で表示される)。
- ✓ 文字種を指定し，入力したい文字の場所を大雑把に指定できる。



- ✓ 実際に文字を入力する場合には，文字をクリックし，[Enter]キー（ないしアップレットの Enter ボタン）で確定する。あらかじめ入力用のソフトを選択しておく必要がある。

(3) 「文字コード表」

- ✓ Windows XP では，標準で付属する「文字コード表」を使って Unicode 文字を検索したり，文字を選択してアプリケーションに貼り付けたりすることができる。
  - ※ Windows 2000 でも，同等の機能を利用できる。Windows95, 98, ME にも「文字コード表」は付属しているが，Unicode の文字を扱うことはできない。
- ✓ 起動方法：[スタート]→[プログラム]→[アクセサリ]→[システムツール]→[文字コード表]
  - ※ 使用するフォントを指定できる（指定されたフォントに収録されていない文字は表示されない）
  - ※ 文字を選択し，コピーして貼り付けることができる。文字を選んで「選択」ボタンを押す(もしくはダブルクリックする)と，「コピーする文字」欄に文字が入力される。入力したい文字列ができたら「コピー」を押すと，文字列がクリップボードに記憶され，アプリケーションに貼り付けることができる。

- ✓ 「詳細表示」をチェックするとより詳細に条件を指定して文字を検索できる。
  - ① 文字セット： Unicode や各地域の Windows コードページ（第 9 回補足資料を参照）を指定する。指定した文字コードによる文字の一覧が表示される。各言語の標準的な文字エンコード方式で利用できる文字を特定するのに有用だが，Windows がコードページに独自に追加した文字も含まれる。

  - ② グループ： 以下のようなカテゴリでグループごとのリストを表示させ，文字を効率よく探すことができる。
    - Unicode カテゴリ（左図）
    - Shift JIS カテゴリ
    - 簡体字中国語ピンイン（声調なし）
    - 繁体字中国語（台湾の Bopomofo という音声記号による）
    - 韓国 漢字とハングル読み，部首，部首の画数
  - ③ 検索する文字の名前： Unicode の文字名での検索をおこなう。Unicode では，全ての文字にアルファベットで名前がつけられている。

    例： a = `Latin Small Letter A`
    A = `Latin Capital Letter A`
    ä = `Latin Small Letter A With Diaeresis`
    あ = `Hiragana Letter A`

    ※ 検索語はスペースで区切って複数入力する。
    ※ 「検索」ボタンを押すと，検索語にマッチした文字だけが表示される。(検索後や検索をやり直す際には結果を「リセット」すること。)
    ※ 漢字はすべて「CJK Unified Ideograph」という名前で分類されているだけなので，個別に検索することはできない。「グループ」や IME パッド等を使う。
  - ④ Unicode で指定： Unicode のコード値を直接指定して検索ができる(「文字セット」に Unicode を指定しておくこと)。**コード値は 16 進数で指定する。**

担当：千葉庄寿（schiba@reitaku-u.ac.jp）西田文信（fnishida@reitaku-u.ac.jp）

2.2. 文字のコード値を調べて入力

Windows XP では，文字の Unicode でのコード値が分かっている場合に，コード値を利用して入力をおこなう方法が 2 つある。また，Word 2002 および EmEditor にも独自のコード⇔文字変換方法があり，文字の入力やコードの調査に便利に使うことができる（**方法** 3, 4)。

方法 1： 日本語 IME2002 の機能を利用する

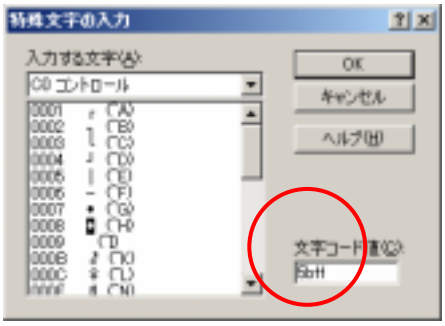
1. 日本語 IME を ON にする
2. 文字の Unicode での値を入力する。Shift JIS であればそのままコード値を，Unicode であれば初めに u を入力し，その後にコード値を入力する（「う５ｂｆｆ」のように妙な表示になるが気にしなくてよい)。**コード値は 16 進数で入力する。**
3. [F5]キーを押す。
4. IME パッドが起動し，該当する箇所の文字一覧表が表示される。必要ならばフォントを変更して文字を確認し，マウスで文字をクリックして編集中のアプリケーションに挿入し，Enter で確定する。
5. 必要ならばフォントの種類を調整する。

方法 2： Unicode コード値入力機能を利用する

1. [NumLock] キーを ON にする（テンキー上の NumLock ランプが点灯)。
2. [Alt] キーを押しながら，テンキー(数字キー)を使って Unicode でのコード値を入力する。**コード値は 10 進数で入力する。**
3. コード値をすべて入力したら，[Alt]キーを離す。文字がアプリケーション上に挿入される。

方法３： Word 2002 以降では，コード値を 16 **進数で**入力後，[Alt] + [x] で文字に変換してくれる（コード値をマウスで選択してもよい)。また，同様の方法で，文字を 16 進数コード値に変換することもできる（Unicode のコード値を調べる場合に便利である)。これらの機能は Word 2002 から搭載された新しい機能である。

方法４： EmEditor では，[編集]→[高度な操作]→[特殊文字の入力]を使って文字のコード値を 16 進数で指定し，文字を入力することができる。また，[表示]→[文字コード値]で文字の 16 進数コード値を表示することもできる（ショートカットは Ctrl + w)。



3. Unicode を扱う際に気をつけるべきこと (1)

Unicode は，これまでコンピュータで使われていたエンコード方式とは異なる多くの特徴をもつ。そのため，Unicode を使いこなすためには，Unicode のもつ問題点や，Unicode 独特の注意点をしっかり把握しておく必要がある。今回はそのうち 3 点をまとめる。

3.1. 異なる言語間で字体の異なる漢字の統合 (Unification)

Unicode では，漢字のように，同じ文字群に属し，言語間で字形が重複している漢字は一つの文字と考える，という原則（「統合」= Unification）がある（§ 1.4.を参照)。確かに，同じ漢字が違うコードで複数個登録されており，言語によって使い分けなければならないのはわずらわしく，また収録すべき文字数が大幅に増大してしまう。Unification の原則により，日本 (J)，中国・台湾 (C)，韓国 (K) それぞれの文字エンコード方式に含まれる約 54,000 の字形を，Unicode では「CJK 統合漢字」20,902 字にコンパクトに統合することに成功している。

しかし, Unification の原則により, 困った問題もおきる。まず, 字体が違うにも関わらず, 同じコードとして登録されている漢字がある。例：「平」(Unicode 5E73)「骨」(Unicode 9AA8)：

日本語 フォント	平	骨	MS 明朝
中国語(台湾) フォント	平	骨	MingLiU
中国語(大陸) フォント	平	骨	SimSun
韓国語 フォント	平	骨	BatangChe
Unicode 汎用フォント	平	骨	Arial Unicode MS

Word2002 (および Word 2000) では，入力ロケールの切り替えと同時にフォントも切り替えられるので，その地域の漢字の字体がうまく保持され，それほど混乱は生じない。しかし，Unicode をテキストファイルの文字エンコード方式として使う場合，フォントの情報は保持されないので，環境によって漢字が見慣れない形で表示されてしまう可能性がある。

また一方で，字形が似ているにもかかわらず，各言語・地域の標準的な文字エンコード方式にある漢字が別々のコード値で登録されている場合もある：

字体	コード値	言語
説	8AAC	日本語
說	8AAA	中国語の繁体字
说	8BF4	中国語の簡体字

このように，Unicode においてどの漢字が「統合」されているかは直感的には分からない。同じように見える漢字が，実は全く別のコード値をもっている場合がある。各言語の漢字がどのように Unicode に収録されているかは，最終的には Unicode の規格書や，Unicode のコード値が記された漢字辞典を調べる必要がある。

なお，Unification はあくまで漢字の字形の整理を目的としておこなわれるのであって，その文字が言語間で同じ意味をもつかどうかとは全く関係ないので注意。例えば「机」という漢字は, 言語・地域によって全く意味が異なるが, 同じ字形を持つ「文字」であり, Unicode では一つのコード値（Unicode 673A）しかもたない（清水『図解で分かる文字コードの全て』p.73)：

日本語	「つくえ」
中国語(大陸)	「機」の簡体字
中国語(台湾)	「つくえ」
韓国語	~~（存在せず）~~

担当：千葉庄寿（schiba@reitaku-u.ac.jp）西田文信（fnishida@reitaku-u.ac.jp）

漢字とそのコード値の関係は，各国や地域でどのような漢字が使われているかという問題，各エンコード方式でどの漢字が登録されているかという問題，さらには Unicode がその漢字をどう収録しているかという問題とが絡み合っており，大変複雑である。そのため漢字のコードを細かくチェックする必要がしばしば出てくる。漢字を専門的に扱う必要がある場合は，文字コードを掲載した辞典（字典）がいくつか出版されているので参照することを強く勧める。ここでは Unicode と関連して一冊だけ紹介する（資料末に一部を掲載する）。

- ユニコード漢字情報辞典編集委員会（編）『ユニコード漢字情報辞典』2000 年. 三省堂. (5,000 円)

  Unicode2.0 に基づき，日本語，中国語（簡体字，繁体字），韓国語の漢字情報を網羅的に収録した本邦初の辞典（ただし，繁体字中国語の文字エンコード方式には，Big5 ではない政府標準の CN コードが記載されている）。漢字ごとに各言語のエンコード方式でのコード値や異体字の情報が収録されている。

3.2. UTF-16LE と UTF-16BE，BOM

Unicode でエンコードされたテキストは，UTF-16 や UTF-8 に代表される，さまざまなエンコード方式を使って保存することができる（第 5 回資料§2.4.参照）。このうち，UTF-16 にはコンピュータの CPU（中央演算装置）の数値処理の性質により 2 つの種類が存在する。

- Windows 系の PC の CPU（Intel, AMD など）：**リトルエンディアン** little endian (LE)
- Macintosh などの CPU：**ビッグエンディアン** big endian (BE)

前者の CPU で最も早く処理できるようバイト列を並べたものを，UTF-16 LE（little endian），一方，後者の CPU に特化した UTF-16 を UTF-16 BE（big endian）と呼ぶ。

Unicode **のコード値と各変換方式の実際のバイト列 * の比較**

	t	**e**	**s**	**t**	**て**	**す**	**と**
ASCII	74	65	73	74	-	-	-
Shift JIS	74	65	73	74	82C4	82B7	82C6
Unicode	0074	0065	0073	0074	3066	3059	3068
little endian (UTF-16LE)	7400	6500	7300	7400	6630	5930	6830
big endian (UTF-16BE)	0074	0065	0073	0074	3066	3059	3068
UTF-8	74	65	73	74	E381A6	E38199	E381A8

*** 通常，各変換方式のバイト列はアプリケーション内で自動的に処理されるので，我々がバイト列を個別に編集操作する必要はない**

Unicode 対応アプリケーションの多くは，UTF-16 LE と UTF-16 BE と UTF-8 の全てが利用できるようになっているが，UTF-16 の 2 つの形式はバイトの並べ方が正反対なので，確実に区別できねばならない。このため，UTF-16 では Unicode テキストの先頭に特殊な印をつけてバイトの並びをあらかじめ記述することになっている。これを BOM（Byte Order Mark）といい，アプリケーションは BOM から UTF-16 のテキスト内のビットの並びを判断する。

アプリケーションでエンコード方式を選択する際，単に「UTF-16」ないし「Unicode」という場合は，「それぞれのパソコンの CPU に合ったエンディアン(LE か BE)で，BOM が付いたもの」を指すことが多い（つまり，Windows では「UTF-16 LE で BOM つき」）。

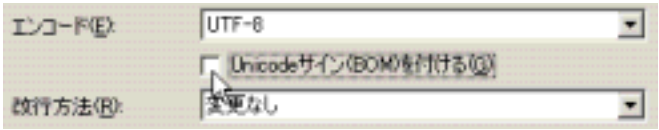
BOM は UTF-16 だけでなく, UTF-8 など Unicode の他の変換方式にもつけることができる。ただし，UTF-16 と違い，UTF-8 の場合は CPU エンディアンによってバイトの並びが変わることはないので，BOM はつけなくともよい。

※ テキストエディタ EmEditor は非常に細かく Unicode での保存形式を指定できる。

- 「エンコード」: Unicode の変換形式の指定（第 5 回資料§2.4.参照）

  - Unicode = UTF-16 LE (little endian) ←Windows 系 CPU に合ったエンディアン
  - Unicode big endian = UTF-16 BE
  - UTF-8 = Unicode の 8bit 変換方式
  - UTF-7 = Unicode の 7bit 変換方式

- 「Unicode サイン(BOM)をつける」: チェックをはずすことで BOM をつけないファイルを作成することができる。(BOM はアプリケーションの内部処理に必要なもので，通常私たちが気にする必要はない。Word 2002 や「メモ帳」などでは全ての変換方式で自動的に BOM が付く。)



  - UTF-8 (および UTF-7) には BOM は必ずしも必要ない。
  - EmEditor では UTF-16 の BOM を外すことができるが, エンディアンの区別をするために BOM は非常に重要である。Unicode の指針でも UTF-16 には BOM を必ずつけることになっている（上記の解説を参照)。

3.3. よく似た記号

コンピュータでの文字の検索は，コードに基づいて「厳密に」行われるため，よく似た文字でもコードが違えば見つけることができない。Unicode は収録文字数が多いため，形のよく似た（意味が異なる）記号が多数登録されており，検索が難しい。たとえば，§2.1.で紹介した「文字コード表」を使い，Unicode の文字名に Hyphen と Dash を含むものを検索すると，以下のように 9 種もの記号が含まれることがわかる（このうち，ASCII のハイフンに対応しているのは Hyphen-Minus (16 進数コード値 002D) である)：

文字 (Arial Unicode MS で表示)	Unicode コード値 (16 進数)	文字名
-	002D	Hyphen-Minus
-	00AD	Soft Hyphen
-	2010	Hyphen
-	2011	Non-Breaking Hyphen
–	2012	Figure Dash
–	2013	En Dash
—	2014	Em Dash
–	2015	Horizontal Bar
·	2027	Hyphenation Point

同様に，「～」と「～」はどちらも Shift JIS の「全角ダッシュ」（または「波ダッシュ」）記号に相当するする Unicode 文字であるが，それぞれ 301C, FF5E というコード値をもつ別の記号である。

本日の課題：

1. 第 5 回授業の宿題となっていた，(1) 外国語の基本的な挨拶表現とその日本語訳（10 通り）と (2) 外国語と日本語の簡単な自己紹介，の 2 つのファイルの冒頭に自分の学籍番号と氏名を入力し，印刷して提出しなさい。印刷の際は，FinePrint の「まとめ印刷」機能と「レイアウト」機能を使って複数ページ A4 の用紙 1 枚にまとめること（文字は小さくてもかまわない)。

2. file_server の Kadai にある [schiba] → [2005fl] フォルダにある [No6] フォルダを file_server の Home（の「2005fl」フォルダ）にコピーしなさい。No6_kadai.doc を Word で開き，指定された作業をおこなって印刷し，提出しなさい。

参考：『ユニコード漢字情報辞典』の一部

[PDF 文書では省略]