文字コード

データベースと情報検索　第6回

横浜市立大学　坂巻顕太郎
kentaro.sakamaki@gmail.com

ディレクトリ（フォルダ）の確認

- エクスプローラのアドレスバーに
  ¥¥wsl$
  と入力

- 以下にファイルがあるはず
  ¥¥wsl$¥Ubuntu-20.04¥home¥ユーザー名

ワイルドカード

- ワイルドカード
  - * は任意の文字列
  - ? は任意の文字という意味

- ls コマンド
  - 引数に与えられたファイルを表示するコマンド
  - ls *.jpg とすると、拡張子がjpg のファイルをすべて表示

- 練習

```
ls /usr/bin/
ls /usr/bin/g*
ls /usr/bin/g??
```

の3つを試してみよ

lsを使ってみる

- ls *.jpg > pictures.txt
  とすればカレントディレクトリにあるjpeg ファイルのリストができる
  - 上の例ではpictures.txt が存在すれば上書きされる
  - ls *.jpg >> pictures.txt とすればpictures.txt に追記される

- nkf やiconv と組み合わせれば、文字コードを変換したファイルを作れる

文字コード

- コンピュータの中では文字は数として記憶されている
- コンピュータやOS によって様々な文字コードが使われている
- 半角の英数字や記号に関しては「ASCII コード」とよばれる文字コードが一般的に使われている
  - たとえば、’A’ は65 (16 進数で41)、’B’ は66 (42) ’a’ は97(61)

ASCII コード

- アルファベット、数字、記号などを表す文字コードの一つ
  - 最も基本的な文字コード
  - 主に英語で必要な文字を収録したコード規格
  - 多くの文字コードがASCIIコードの拡張になるよう実装されている

- 文字を7ビット（0～127）で表しており、128文字が表現できる
  - 0から127までの値（正確には2進数で0000000から1111111まで）に対して、どの文字を意味するかという対応がある

ASCII コード（つづき）

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

テキストファイル

- テキストファイル
  - 文字コードが保存されている
- ファイルの中身を見る

```
od -tx1z -Ax ファイル名
```

- 練習
  - 以下のようなファイルをエディタで作り、中身を見てみよう

```
ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz
1234567890
!"'+-*/=()[]|
```

odコマンド

- ファイルを8進数や16進数などでダンプする際に使用するコマンド
  - 「ダンプ（dump）」は記録などの中身をまとめて表示したり、記録したりするという意味

-A 指定	出力位置の表示形式を指定 「o（8進数）」「x（16進数）」「d（10進数）」
-j バイト数	先頭からスキップするバイト数を指定
-N バイト数	表示するバイト数を指定
-S バイト数	可読文字が指定したバイト数分連続していたら、その箇所を表示
-t タイプ	出力フォーマットを指定
-v	「*」マークによる出力行の省略をやめる
-wバイト数	1行当たりの出力バイト数を指定

様々な日本語のコード

- ISO-2022-jp (JISコード)
  - ASCIIコードと同様、7bitで日本語文字を表現する方式
  - 互換性のため、電子メールなどの古い規格ではJISコードが使われる

- EUC-JP (Extended Unix Code)
  - UNIXで用いられている
  - 8bitコードの半角カナを扱う場合に不具合が起きることがある

- Shift-JIS (SJIS)
  - Microsoftとアスキーなどが開発した文字コード
  - 8bitコードの半角カナを残して、余ったコードで全角文字を表現
  - 多くのコンピュータや様々な機器で使用されている

- Unicode
  - 世界中の文字を統一的に扱うことが目的の文字コード
  - 主要なコンピュータやスマホ向けのOSで標準となってきている
  - UTF-8やUTF-16などのバリエーションがある

8bitコード

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	DLE	SP	0	@	P	`	p				ー	タ	ミ		
1	SOH	DC1	!	1	A	Q	a	q			。	ア	チ	ム		
2	STX	DC2	"	2	B	R	b	r			「	イ	ツ	メ		
3	ETX	DC3	#	3	C	S	c	s			」	ウ	テ	モ		
4	EOT	DC4	$	4	D	T	d	t			、	エ	ト	ヤ		
5	ENQ	NAK	%	5	E	U	e	u			・	オ	ナ	ユ	未	
6	ACK	SYN	&	6	F	V	f	v			ヲ	カ	ニ	ヨ		
7	BEL	ETB	'	7	G	W	g	w			ァ	キ	ヌ	ラ	定	
8	BS	CAN	(	8	H	X	h	x			ィ	ク	ネ	リ		
9	HT	EM	)	9	I	Y	i	y			ゥ	ケ	ノ	ル	義	
A	LF	SUB	*	:	J	Z	j	z			ェ	コ	ハ	レ		
B	VT	ESC	+	;	K	[	k	{			ォ	サ	ヒ	ロ		
C	NP	FS	,	<	L	¥	l	|			ャ	シ	フ	ワ		
D	CR	GS	-	=	M	]	m	}			ュ	ス	ヘ	ン		
E	SO	RS	.	>	N	^	n	‾			ョ	セ	ホ	゛		
F	SI	US	/	?	O	_	o	DEL			ッ	ソ	マ	゜		

文字コードの互換性

- 日本語の文字コードとして4種類ある
  - ある文字を表す数値が文字コードによって異なる

- ひらがなの「あ」
  - JISコード（16進数）：2422
  - EUC-JP：A4A2
  - Shift-JIS：82A0
  - UTF-8：E38182
  - UTF-16：3042

文字コードを知る方法

- Fileコマンドからファイルの文字コードがわかる

```
file ファイル名
```

  - File –e encoding ファイル名

- File コマンドは文字コード以外にもファイルの種類を教えてくれる

nkfコマンド：漢字コードの変換

- 昔は漢字コードの変換にはnkf というフリーのプログラムを使っていた
  - nkf: Network Kanji Filter

- WSL のUbuntu にはnkf が入っていない（はず）
  - sudo apt install nkf を実行すればインストールできる

文字コードの変換: iconv

- ```
  iconv -t UTF-8 a.txt > b.txt
  ```
  - a.txt というテキストファイルの文字コードをutf-8 に変換して、b.txtに保存
  - 入力ファイルの文字コードを指定する場合は、-f オプション（iconv -f –t）を用いる

- iconvで用いることのできる文字コードの確認

```
iconv -l | less
```

- iconvの使い方のヘルプ

```
iconv --help | less
```
```

iconvコマンド

- ファイルなどのエンコーディング（文字コードを変換）するためのコマンド

		意味
-f コード	--from-code=コード	入力のエンコーディング（文字コード）
-t コード	--to-code=コード	出力のエンコーディング（文字コード）
-c		変換できなかった文字を出力しない
-s	--silent	対応していないなどで変換できなかった場合にエラーメッセージを表示しない
	--verbose	処理中のメッセージを表示する
-o	--output=ファイル	保存先のファイル
-l	--list	対応しているコードを表示する

練習

- 先程作ったファイルを色々な文字コードに変換し、テキストエディタなどで中身を見てみよう

- fileコマンドで確認してみよう

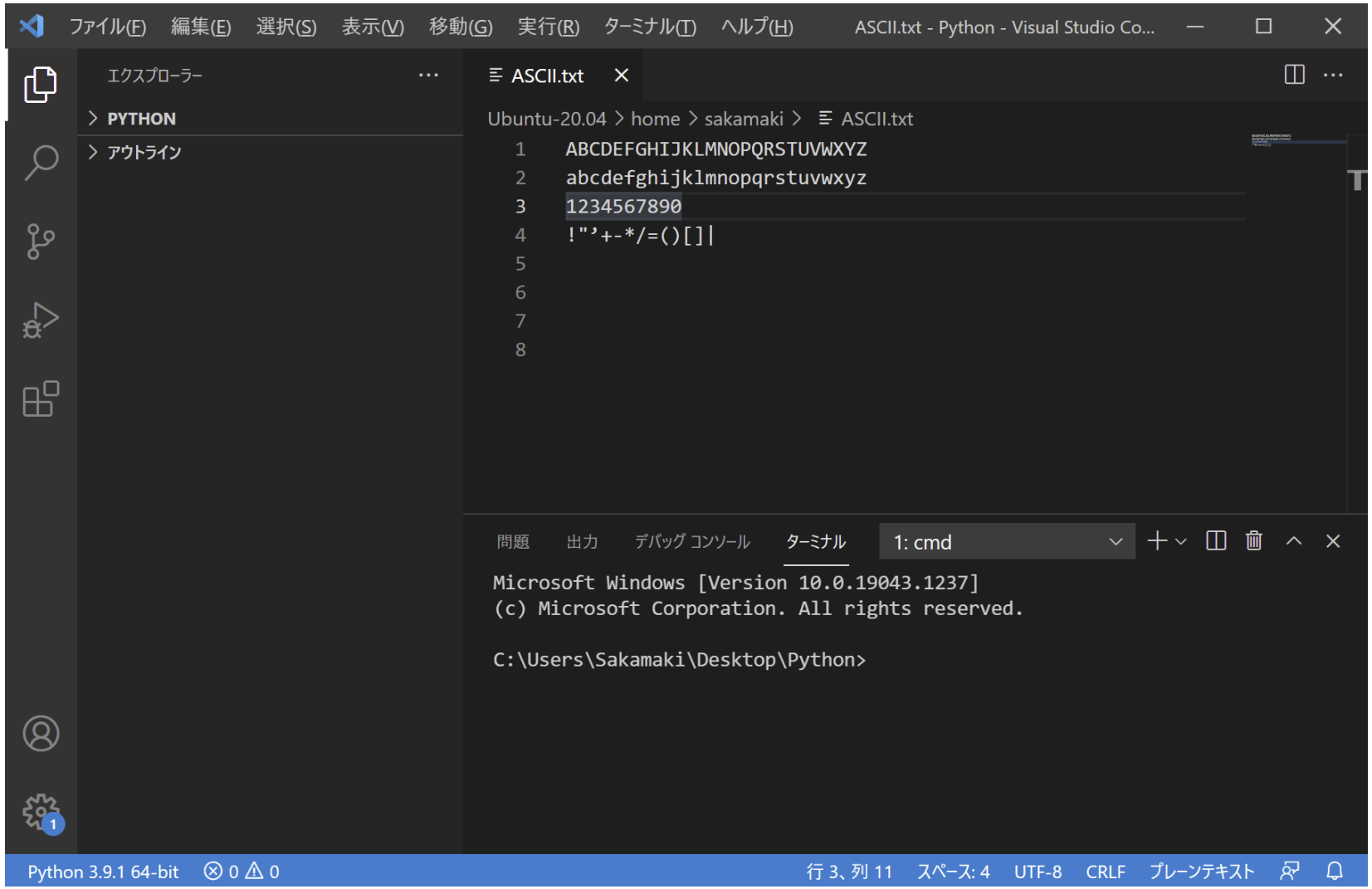
```
file –e encoding
file –i ファイル名
```

互換性のある文字コードのみの場合

- ASCII/UTF-8/Shift-JISは同じに見えることがある
  - 閲覧環境のエンコーディングのデフォルト設定によって、ASCII, UTF-8, Shift-JISのいずれかが表示される可能性がある
  - ASCIIだけだとUTF-8に変換しようとしてもASCIIにみえる

- ASCIIとの互換性
  - UTF-8
    - ASCIIと互換性をもたせるため、ASCIIで定義されている記号や英数字部分は同じ1バイトで表現し、その他は2～6バイトで表現する
  - Shift-JIS
    - JIS X 0201は半角英数とカタカナを1バイト、JIS X 0208はその他の日本語を2バイトで表現する
    - JIS X 0201の"¥"（エン）と"‾"は、ASCIIでは"¥"（バックスラッシュ）と"~"

VScodeで開いてみると



改行コード

- Windows, Mac, Unix で異なる
  - Windowsでは、0x0D + 0x0A の2 バイト(¥r¥n) (CRLF)
  - 昔のMacでは、0x0D (¥r) (CR)
  - MacとUnixでは、0x0A (¥n) (LF)

- 日本語文字コード以外ではまりやすい罠の一つ

- `od -c ファイル名`
  - Windows のメモ帳で作ったファイルと echo をリダイレクトして作ったファイルの改行コードを比べてみよう

fileコマンドで改行コードの確認

- file ファイル名
  - 例: UTF-8 Unicode text, with CRLF line terminators.
- "with CRLF line terminators"と表示されればCRLF, 表示がなければLF

改行コードの変換

- nkf でできる
  - nkf -Lu hoge.txt > unix.txt
  - nkf -Lw hoge.txt > windows.txt
  - nkf -Lm hoge.txt > mac.txt

- nkf がない場合、tr コマンドを使う
  - tr ¥¥r ¥¥n < mac.txt > unix.txt
  - tr -d ¥¥r < windows.txt > unix.txt
  - perl -p -e ’s/¥n/¥r¥n/’ < unix.txt > windows.txt

trコマンド

- 文字を置き換えるためのコマンド
  - 指定した文字を削除したり、文字が連続している場合、1つにまとめたりすることもできる
- 標準入力と標準出力が使われる
  - ファイルを処理する場合、catコマンドなどを使用してパイプするか、リダイレクトを用いる

BOMにも注意

- BOM: byte order mar
  - Unicodeの符号化形式で符号化したテキストの先頭につける数バイトのデータ
  - Unicodeで符号化されていることおよび符号化の種類の判別に使用する

- BOMの追加

```
nkf --overwrite --oc=UTF-8-BOM ファイル名
```

- BOMの削除

```
nkf --overwrite --oc=UTF-8 ファイル名
```

head, tail

- head ファイル名
  - ファイルの先頭10 行が見れる
  - "head -n 行数ファイル名"とすれば、任意の行数を表示できる
- tail ファイル名
  - ファイルの終り10 行が見れる
  - "tail -n 行数ファイル名"とすれば、任意の行数を表示できる
- ファイル名を指定しないと、head, tailも標準入力から入力する

パイプ

- ファイルを表示するようなUnixコマンドは、ファイル名を指定されないと標準入力を出力する
- 一方、コマンドの出力は標準出力される
- ２つのコマンドを“| ”で繋げることで、コマンドの標準出力を別のコマンドの標準入力とすることが可能
  - ls /usr/bin | more などとすれば、ファイルが沢山あるディレクトリのファイル名もチェックできる

応用

- `ls -c | head -n 5`
  - 最近更新したファイルが5 つ表示される
- `ls | head -n 100 | tail -n 50`
  - アルファベット順で51 番目から100 番目のファイルが表示される
- 練習
  - 適当なテキストファイルを用意し、その3行目だけを表示してみよ

課題

- テキストファイルの作成
  - 1行目に「自分の名前」、2行目に「いま使っているPCのOS」、3行目に「学んできたプログラミング言語」、4行目以降はPC スキルなどを自由記述

- 文字コードと改行コードの変換
  - UTF-8 でUNIX の改行コードのu.txt
  - SHIFT-JIS でWindows の改行コードのw.txt