

コンピュータ入門

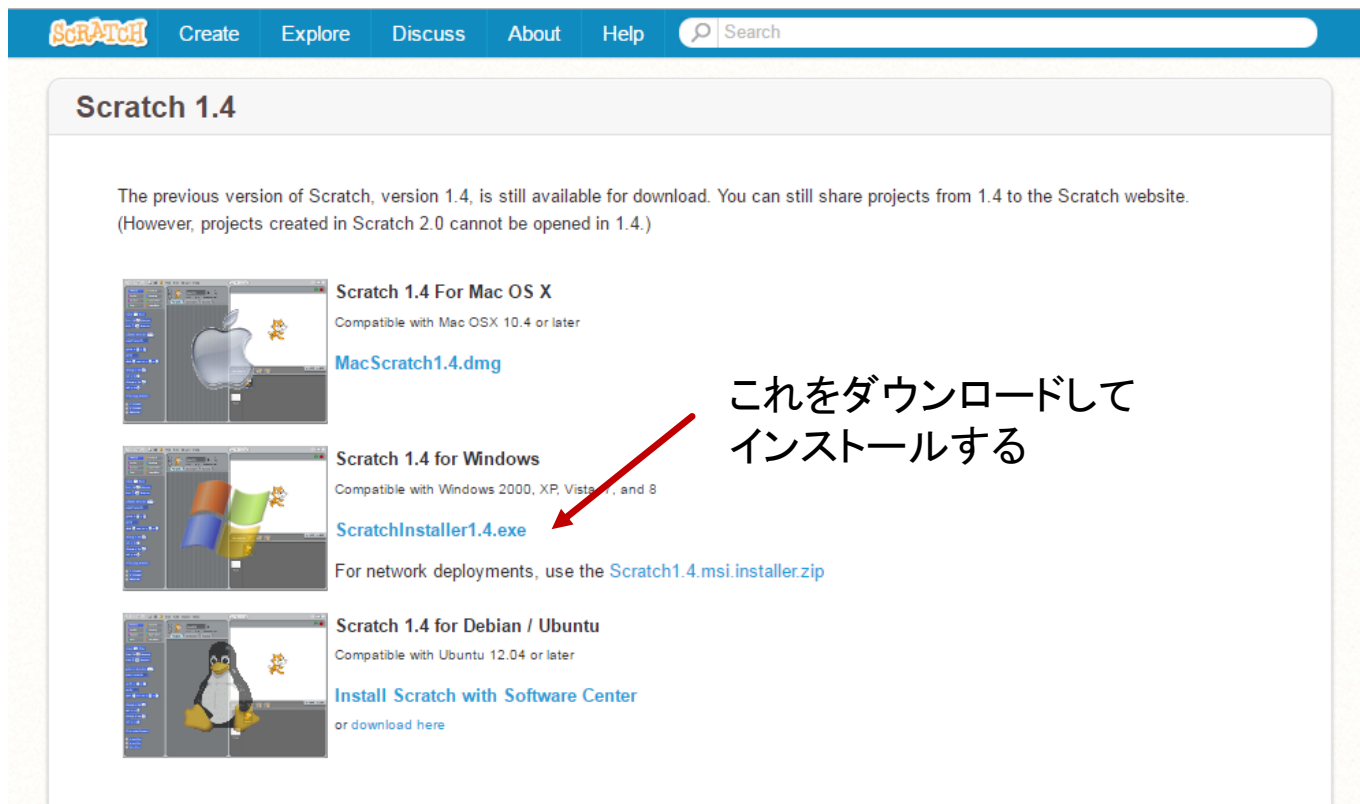
第9回 UNIX(3) 文字コードとシェル

授業計画と課題

#1	オリエンテーションとコンピュータ・リテラシー	(4月16日)
#2	Windows入門	(4月23日)
#3	Excel入門	(5月7日)
#4	Excelによるデータ分析	(5月14日)
#5	WordとExcelを用いたレポートの書き方	(5月21日)
#6	PowerPointを使ったスライド資料作成法の理解とプレゼンテーション演習	(5月28日)
#7	UNIX(1) 基本操作	(6月4日)
#8	UNIX(2) ディレクトリとファイルシステム	(6月11日)
#9	UNIX(3) 文字コードとシェル	(6月18日)
#10	プログラムとは	(6月25日)
#11	プログラムの基本処理	(7月2日)
#12	分岐処理とは	(7月9日)
#13	反復処理とは	(7月16日)
#14	分岐・反復処理の応用	(7月23日)
#15	プレゼンテーションと相互評価	(7月30日)

来週までの課題

- Scratch 1.4を各自のPCにインストールしておくこと
(https://scratch.mit.edu/scratch_1.4/)



本日の予定

- 文字コード
- シェル
- UNIXコマンド操作の小テスト

文字コード

- コンピュータ内部で、文字・数字・記号を表わすために各文字に割り当てられる数値
 - 英数字は1バイト(256文字)で表現される
 - 日本語等は2バイト(65536文字)で表現される
 - 1バイト(256文字)では字数が足りないため

文字コード: ASCII

- 英語などラテン文字を中心とした文字コード
 - ASCII: American Standard Cord of Information Interchange
 - 7bitコード(128文字分 ($2^7 = 128$))
 - 印字可能文字(96文字): 半角英数字および記号 (日本語は含まれない)
 - 制御記号(32文字): Enterやタブなど

文字コード: ASCII

- 文字等は数値で表現される
 - 例: 'J'のコードは0x4A
 - man ascii

下位(16進)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
2		!	"	#	\$	%	&	'	()	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[¥]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{		}	~	□

上位

ASCIIコード表(一部)

文字コード: 日本語文字コード

- 日本語(全角ひらがな, 漢字, 全半角カタカナ), 英数字, 記号が使える文字コード
 - 主に2byte(16bit)で表わす
- 数種類の日本語文字コードがある
 - JISコード
 - 電子メール転送などに利用される
 - シフトJISコード (SJIS)
 - PC (Windows, Mac)などで広く利用される
 - EUC (Extended UNIX Code)
 - 多くのUNIX OSで使用されている

文字コード: Unicode

- 1つの文字コードで多国語を表現可能にした文字コード
 - 2byte(16bit)コード
 - 世界の主要な言語をカバーしている
 - しかし, 2byteでは最大65536文字しか収録できない
(日本語と中国語などで, コードの使い回しがある)
 - 様々な文字符号化(エンコード)方式がある
 - 演習環境(OS X)ではUTF-8が使用されている

文字化け

- 本来の文字コードとは異なる文字コードで読んだために、(読めない)文字に置き換えれる現象
 - Windows上で作成したファイルをUNIXに転送した場合などに発生する
 - 例: Windows上でSJISで作ったファイルを, UNIXシステム上でUTF-8のファイルとして開いた場合

プログラミング入門及び演習
Bクラス

SJIS

?v???O???~???O?????y?щ??K
B?N???X

UTF-8

改行コード

- 改行を表わすために用いられる文字コード
 - OSによって異なる

	コード表記	16進表記
UNIX	<LF>	0A
Windows	<CR> <LF>	0D 0A
昔のMac OS	<CR>	0D

CR: Carriage Return (復帰コード)
LF: Line Feed (改行コード)

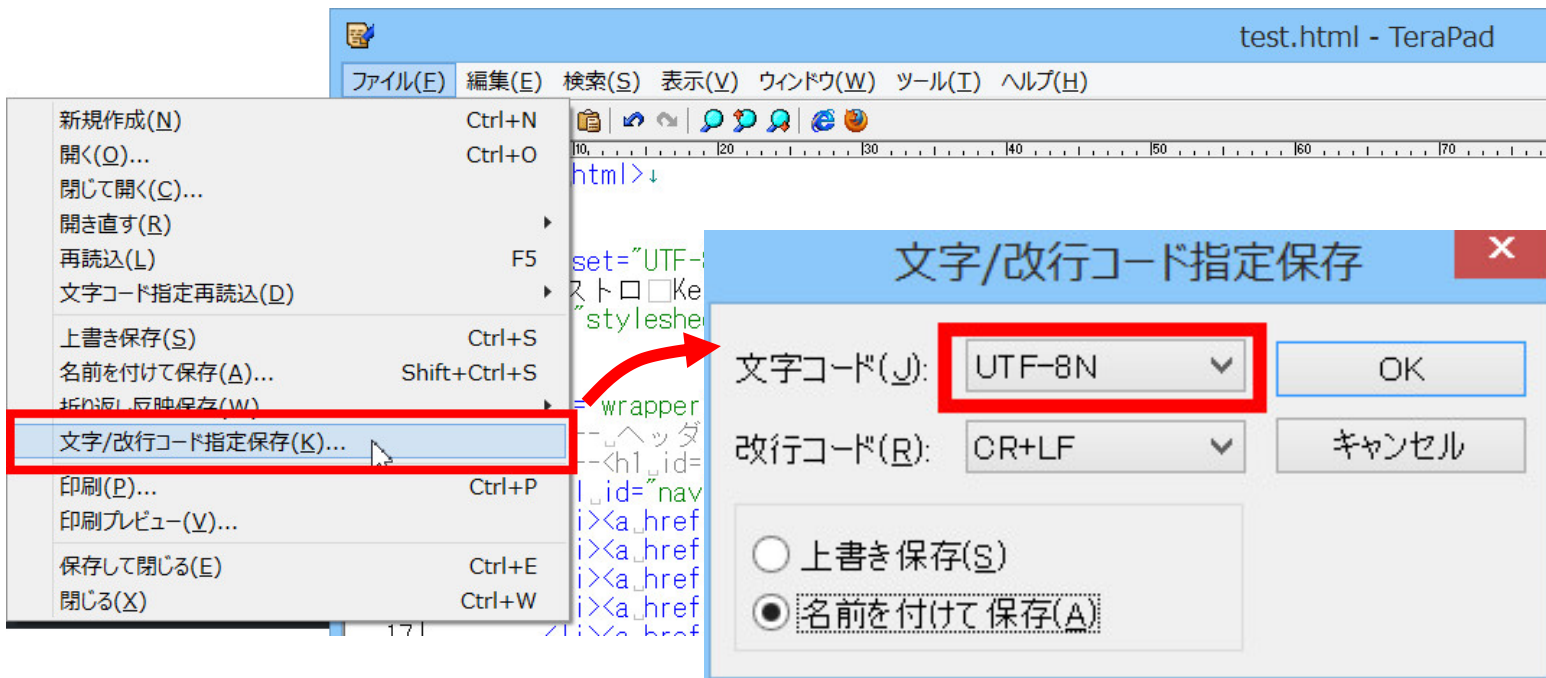
- OS間でファイルを転送する場合、改行コードも変換する必要がある
 - 演習環境(OS X)の改行コードは<LF>

情報工学科の授業で必要となるソフトウェア一覧

- 今後4年間の演習ですべて必要
 1. TeraTerm: 端末ソフト (フルインストール)
 2. WinSCP: ファイル転送 (標準インストール)
 - 3. TeraPad: 文字コード変換ができるテキストエディタ**
- 今回はTeraPadのインストール
 - 以下のURLからダウンロード
<http://www.cse.ce.nihon-u.ac.jp/download>

TeraPadで文字コードを指定して保存

- ファイルメニューから“文字コード指定保存”
 - 最近のパソコンやサーバはほぼUTF-8 (UTF-8N)
 - TeraPadで保存するときはUTF-8Nで



練習1

- 第7回でを使用したexample_utf.txt
(example_utf2.txtでも良い)を使って以下を確認
 - ① 文字コード指定保存でShift-JISを選択し,
ファイル名をexample_sjis.txtとして保存
 - ② 保存後, 画面右下に“UTF-8”と書かれたステータスが
“SJIS”に変わっていることを確認
 - ③ WinSCPでホームディレクトリに転送
 - ④ TeraTermでログインし,
catまたはmoreコマンドでexample_sjis.txtの中身を確認

本日の内容

- 文字コード
- シェル

シェル

- ユーザが入力する色々なコマンドを受け付け、解釈し、実行するソフトウェア
 - ログインすると自動的に起動(黒画面上にコマンドプロンプト)
 - 多くのシェルがある
 - Windows: ファイルエクスプローラ
 - Mac OS: Finder
 - UNIX: sh(シェル), **csch (Cシェル)**, bash



演習環境はCシェル

Cシェル

- 今回確認するシェルの機能
 - ①コマンドライン編集(カーソルの移動)
 - ②ファイル名の指定
 - ③ヒストリ機能(コマンド履歴)
 - ④エイリアス機能(別名ファイルの登録)

①Cシェル: コマンドライン編集(カーソルの移動)

- コマンドライン: コマンドを入力する行
 - CTRL(Control)キーを押しながら下表の操作
 - Emacsと同じキーバインド

C-h	カーソルの1つ前の文字を削除	C-d	カーソルのある位置の文字を削除(DeleteのD)
C-a	行の先頭に移動(AheadのA)	C-e	行の最後に移動(EndのE)
C-b	1文字前に移動(BackのB)	C-f	1文字後に移動(ForwardのF)
C-k	カーソルから右を削除(KillのK)	C-u	1行全体を削除

②Cシェル: ファイル名の指定

- ファイル指定を容易にする特殊文字(ワイルドカード)を用いた省略形
 - ? : 任意の(空でない)1文字に整合
 - * : (空も含む)すべての文字列に整合
 - ~ : ホームディレクトリを表わす



他にも[] や {} など(「UNIXとC」p.61参照)

例: ファイル名の指定 - 特殊文字「？」

- 1文字のみのワイルドカード

例: 以下のように, 1文字のみ任意のファイルを選択できる

- ?に対応する文字列として, 1, 2

```
tetsuo@cse-ssh[61]: ls -l
total 76
-rwxr-xr-x  1 tetsuo  faculty    8824  5  6 09:35 a.out*
-rw-r--r--  1 tetsuo  faculty   11957  5  6 09:35 file1  ←
-rw-r--r--  1 tetsuo  faculty    7264  5  6 09:35 file10
-rw-r--r--  1 tetsuo  faculty    8950  5  6 09:35 file2  ←
-rw-r--r--  1 tetsuo  faculty     319  5  6 09:35 prog1.c
tetsuo@cse-ssh[62]: ls -l file?
-rw-r--r--  1 tetsuo  faculty   11957  5  6 09:35 file1
-rw-r--r--  1 tetsuo  faculty    8950  5  6 09:35 file2
```

例: ファイル名の指定 - 特殊文字「*」

- 1文字以上任意の文字列のワイルドカード

例: 以下のように, 任意の文字列のファイルを選択できる

- *に対応する文字列として, 1, 2, 10

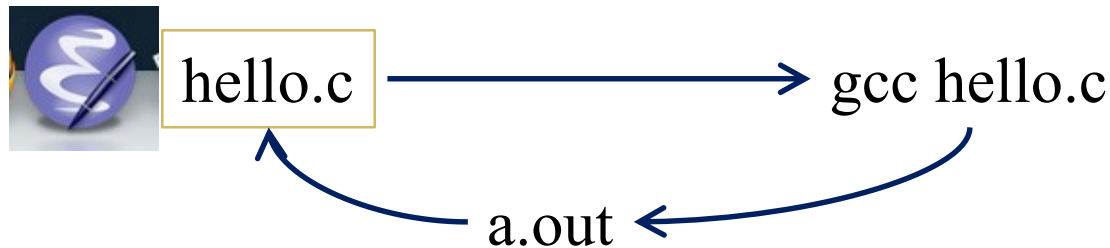
```
tetsuo@cse-ssh[63]: ls -l
total 76
-rwxr-xr-x  1 tetsuo  faculty    8824  5  6 09:35 a.out*
-rw-r--r--  1 tetsuo  faculty   11957  5  6 09:35 file1
-rw-r--r--  1 tetsuo  faculty    7264  5  6 09:35 file10
-rw-r--r--  1 tetsuo  faculty    8950  5  6 09:35 file2
-rw-r--r--  1 tetsuo  faculty     319  5  6 09:35 prog1.c
tetsuo@cse-ssh[64]: rm file*
remove file1? y
remove file10? y
remove file2? y
tetsuo@cse-ssh[65]: ls
a.out*          prog1.c
```

ワイルドカードの例

- *.txt
 - 末尾に.txtがつくファイル・ディレクトリ
- xyz*
 - xyzで始まるすべてのファイル・ディレクトリ
- h30/*
 - h30ディレクト直下にあるすべてのファイルとディレクトリ
- test?.txt
 - testの後に任意の一文字がきて, その後.txtがつくファイルすべて
 - test1.txt testa.txt test-.txt マッチ
 - testaa.txt test.txt マッチしない

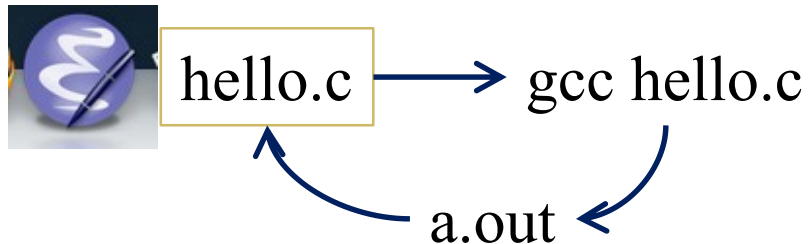
③Cシェル: ヒストリ機能

- 入力したコマンドを記憶する機能
 - 使用するには, 環境設定が必要
 - ~/ .history にコマンドの履歴が保存される
 - 過去に使ったコマンドを再度実行する際に便利
 - 例:
「hello.c をテキストエディタで編集, gccでコンパイル,
コンパイルしてできたa.outを実行し, また編集」を繰り返す



コマンド: history

- history
 - 機能: コマンドの履歴リストの処理
 - 形式
 - history 記憶されているすべての履歴を表示する
 - history *n* 直近*n*個の履歴を表示する
 - 例
 - history 7



```
sekizawa@cse-gw[103]: history 7
 97  1:03  gcc hello.c
 98  1:03  ./a.out
 99  1:05  emacs hello.c
100  1:05  gcc hello.c
101  1:05  ./a.out
102  1:05  clear
103  1:06  history 7
sekizawa@cse-gw[104]:
```


Cシェル: ヒストリ機能の利用

- ヒストリの参照

!!	直前に実行されたコマンド
! <i>n</i>	ヒストリ番号 <i>n</i> のコマンド
! <i>-n</i>	<i>n</i> 個前のコマンド
! <i>文字列</i>	先頭が指定した <i>文字列</i> で直近のコマンド
! <i>?文字列?</i>	指定した <i>文字列</i> を含む直近のコマンド
!*	直前に実行されたコマンドのすべての引数
!\$	直前に実行されたコマンドの最後の引数

Tips: ヒストリ機能

- 矢印キー(↑と↓)を押すだけで,
記憶されているコマンド履歴の呼び出しが可能
- 呼び出したコマンド履歴は編集可能

④別名機能(alias)

- コマンドに別名(alias)を付ける機能
 - 頻繁に使うコマンド, 長いコマンドの短縮形の登録に利用
 - alias コマンドで設定
- 例
 - 端末画面の消去は, UNIXではclear, Windowsのコマンドプロンプト(cmd)ではcls.
 - UNIXでclsをclearの別名として登録しておけば, 環境の差を考えず, 慣れたコマンドで実行できる.

コマンド: alias

- alias
 - 機能:
コマンド(群)の別名または省略形の生成と削除
 - 形式
 - alias ...alias一覧の表示
 - alias 別名 コマンド ...コマンドを別名で定義する
 - 例
 - alias cls clear
cls という名のコマンドを, clear の別名として定義する

練習2

設定済みの別名をaliasで確認せよ

- aliasを用いて, historyコマンドに省略名としてhisを登録してみよ
- 登録した別名を実際にコマンド実行し, 確認せよ
- 登録した別名を `unalias his` で削除せよ

UNIXコマンド操作の小テスト

- 制限時間30分
- 授業資料参照可, PC使用可
 - PCで実行を確認してから, 答えを解答欄に記入する
- 問題用紙を配布します