

ISBN978-4-7973-5735-6

C0055 ¥2200E



9784797357356



1920055022002

定価 本体2,200円 +税

UNIXシェルスクリプトコマンドブック

# UNIXシェルスクリプト コマンドブック

山下哲典 著

第2版

★ ★ ★ ★ ★  
sh ★ bash

Bourne Shell, Bourne Again Shell



山下  
哲典

007.67

台東区立 中央図書館  
52465911



114301559

SoftBank Creative

# CONTENTS

## 第1章 シェルの基本機能

1.1	変数●値を表現する入れ物	2
	変数を作成する	3
	変数の値を参照する	5
1.2	標準入力とファイルディスクリプタ●入出力の仕組み	7
1.3	リダイレクション●出力先/入力元を変更する仕組み	10
	ファイルの標準入出力を変更する	12
	標準入力と標準エラー出力を同じファイルに向ける	15
	ファイルディスクリプタ番号を指定する	18
1.4	ヒアドキュメント●標準入力をスクリプト内部で生成する仕組み	19
	シェルスクリプトでヒアドキュメントを使う	21
	ヒアドキュメントの記述に関する注意点	23
	シェルスクリプトでメッセージを出力する	25
1.5	パイプ●標準出力をそのまま別のプロセスの標準入力へ渡す仕組み	27
	複数のコマンドの標準入出力をつなぐ	28
	標準エラー出力をパイプに渡す	29
1.6	メタキャラクタと特殊文字●特殊な意味をもつ文字	31
	特殊文字の意味を無効化する	33
	文字列を区切る(スペースとタブ)	34
	シェルスクリプト内でコメントを記述する(#)	35
	変数を参照する(\$)	36
	直後の1文字をエスケープする(\)	37
	コマンドを区切る、複数のコマンドを1行にまとめる(;, \)	38
	任意の1文字に該当する(?)	39
	任意の文字列に該当する(*)	40
	候補内の1文字に該当する([ ])	42
	候補内の文字列に該当する({ })	44
	バックグラウンドジョブを制御する(&)	45
	[ ]内の候補の否定、先頭の文字を指定する(^)	47
	ユーザのホームディレクトリを表す(~)	48

	コマンドの実行結果に置き換える(`, `')	49
	コマンド履歴を参照する, [ ]内の候補の否定(!)	52
1.7	クォーテーション●メタキャラクタの意味を無効にする仕組み	55
	文字列を表す(" ")	56
	変数などを展開せずに文字列を表す(' ')	57
	特殊文字を無効化する(\)	58
	クォーテーションを使い分ける	59
	クォーテーションを文字として扱う	65
1.8	コマンドセパレータ●コマンドを区切る仕組み	66
	複数のコマンドを1行で実行する(:)	67
	左項のコマンドが正常終了のときに右項のコマンドを実行する(&&)	68
	左項のコマンドが異常終了のときに右項のコマンドを実行する(   )	69
1.9	グルーピング●複数のコマンドをまとめる仕組み	71
	コマンドをグルーピングする(() )	72
	コマンドをグルーピングする({ })	73
	()と{}を使い分ける	75
1.10	位置パラメータ●コマンド行を分解してそれぞれの値を保持する変数	76
	位置パラメータを表示する	77

## 第2章 制御文とtestコマンド

2.1	制御文●制御構造のための文	80
2.2	if文●条件分岐	81
	数値を比較して分岐する	83
	コマンドの実行結果で分岐する	87
	ファイルの種別を調べる	89
	if文をネストする	90
	if文を&&と   に置き換える	91
	if文の結果をファイルに出力する	92
	if文を読みやすく記述する	94
2.3	for文●ループ文	95

ワードリストを順番に表示する .....	97
コマンドの実行結果や変数の値をワードリストにする .....	99
シェルスクリプトの引数をワードリストにする .....	101
for文を使ったサンプルスクリプト .....	103
for文の結果をファイルに出力する .....	106
for文内で操作した変数をfor文の外で利用する .....	107
for文をネストする .....	108
for文を読みやすく記述する .....	109
for文をC言語風に記述する .....	110
<b>2.4 while文●ループ文</b> .....	<b>112</b>
条件を満たしているならコマンドを繰り返す .....	113
無限ループを使う .....	116
ファイルに結果を出力する .....	119
ファイルの内容を行単位で読み込む .....	121
while文を使ったサンプルスクリプト .....	124
while文内で操作した変数をwhile文の外で利用する .....	126
while文をネストする .....	127
while文を読みやすく記述する .....	129
<b>2.5 case文●パターンによる分岐</b> .....	<b>130</b>
パターンで処理を分岐する .....	132
さまざまなパターンで文字列をチェックする .....	134
キーボードからの入力で処理を分ける .....	135
echoコマンドの改行なし出力をOSにより判定する .....	137
シェルスクリプトのオプションを処理する .....	139
case文を読みやすく記述する .....	144
<b>2.6 until文●ループ文</b> .....	<b>146</b>
条件を満たすまでループを繰り返す .....	147
<b>2.7 select文●メニューリストによる分岐</b> .....	<b>148</b>
メニュー画面からユーザに選択させる .....	149
メニュー画面から実行するコマンドを選択する .....	152
メニューのリストではなく番号で処理する .....	154

引数をメニュー項目にする .....	156
メニューをマルチカラム出力する .....	157
select文を読みやすく記述する .....	158
<b>2.8 testコマンド●条件の判定</b> .....	<b>159</b>
制御文の条件に使う .....	161
数値を比較する .....	163
文字列を比較する .....	166
ファイルの形式や情報を調べる .....	170
複雑な条件判定を行う .....	173
数値と文字列の扱いの違い .....	178

## 第3章 シェルの組み込みコマンド

<b>3.1 .コマンド●カレントシェルでのシェルスクリプトの展開／実行</b> .....	<b>180</b>
カレントシェル上でスクリプトを実行する .....	181
シェル関数をスクリプトに読み込む .....	183
変数や関数を定義したファイルをどこに配置するか .....	185
<b>3.2 :コマンド●何もせずに真(終了コード0)を返す、変数の値のチェック</b> .....	<b>186</b>
何もしないことを明示的に示す .....	187
条件を常に真にする .....	188
拡張的な変数展開を出力なしで処理する .....	189
<b>3.3 breakコマンド●ループ文からの脱出</b> .....	<b>192</b>
ループ文を抜ける .....	193
入力が正しいときに無限ループを抜ける(不当な入力を許さない) .....	196
<b>3.4 cd, chdirコマンド●カレントディレクトリの変更</b> .....	<b>198</b>
cdコマンドに関する環境変数(CDPATH, HOME) .....	199
サブシェル内でディレクトリを移動する .....	200
シンボリックリンクしたディレクトリに移動する .....	201
<b>3.5 continueコマンド●ループ内の処理を中断しループの先頭からの再開</b> .....	<b>202</b>
ループ内の処理を中断してループの先頭へ戻る .....	203
<b>3.6 declare, typesetコマンド●変数の宣言</b> .....	<b>206</b>

	配列や数値用の変数の設定する .....	207
3.7	<b>echo コマンド</b> ●標準出力へ文字列の出力 .....	209
	System V系とBSD系のechoコマンドの違い .....	210
	echoコマンドの実行結果を調べて改行しない方法を判別する .....	212
	メッセージ出力後に改行しないechoコマンドを関数にする .....	215
	メッセージ出力後に改行しないechoコマンドを関数にする .....	218
3.8	<b>eval コマンド</b> ●引数をシェルのコマンドとして実行,ひとまとめにした変数の展開 .....	219
	evalコマンドを使いこなす .....	221
	環境変数を設定する .....	222
	位置パラメータの最後の値を取り出す .....	223
3.9	<b>exec コマンド</b> ●カレントシェルの置き換え .....	224
	カレントシェルの別プロセスに置き換える .....	225
	シェルをリダイレクトする .....	227
	while文への入力をファイルからもらう .....	232
3.10	<b>exit コマンド</b> ●シェル/シェルスクリプトの終了 .....	233
	任意の終了コードでスクリプトを終了する .....	234
	オプションや引数に問題があったときにヘルプメッセージを出力して終了する .....	236
	割り込みを受けたときに終了する .....	237
3.11	<b>export コマンド</b> ●シェル変数のエクスポート .....	238
	シェル変数の値を環境変数にする .....	239
	スクリプト内の変数を他のスクリプトに受け渡す .....	242
3.12	<b>getopts コマンド</b> ●シェルスクリプトでのオプション処理 .....	244
	シェルスクリプトのオプションを処理する .....	248
3.13	<b>hash コマンド</b> ●コマンドのハッシュテーブルの操作 .....	249
	コマンドのハッシュテーブルを操作する .....	250
3.14	<b>kill コマンド</b> ●プロセスへのシグナルの送出 .....	251
	プロセスにシグナルを送る,シグナルの一覧を表示する .....	252
	代表的なシグナルの意味 .....	254
	該当のプロセスを見つけてシグナルを送る .....	258
3.15	<b>let コマンド</b> ●算術式による数値計算 .....	261
	letコマンドをwhile文の条件にする .....	262
	letコマンドを(( ))で代用する .....	262

3.16	<b>local コマンド</b> ●シェル関数内でしか有効でないローカル変数の生成 .....	263
	ローカル変数と通常の変数との違い .....	264
3.17	<b>printf コマンド</b> ●文字列の整形 .....	266
	ファイル/etc/passwdを整形する .....	267
3.18	<b>pwd コマンド</b> ●現在のワーキングディレクトリの表示 .....	268
	ファイルやディレクトリの絶対パス名を表示する .....	270
3.19	<b>read コマンド</b> ●キーボードからの入力を変数に代入 .....	272
	入力文字列の数とreadコマンドの引数の数が異なるとき .....	274
	readコマンドの文字列区切り .....	275
	ユーザの応答で処理を分岐する .....	276
	定型ファイルのフィールドを抜き取って編集する .....	278
	ファイルの内容に行番号をつけて出力する .....	281
3.20	<b>readonly コマンド</b> ●変数の読み出し専用化 .....	283
	読み出し専用の変数を書き換えたようにみせる .....	284
3.21	<b>return コマンド</b> ●関数からの戻り値の設定 .....	286
	関数の結果と関数の戻り値を使い分ける .....	287
	数値かどうかを判定する .....	289
3.22	<b>set コマンド</b> ●位置パラメータのセット,変数の値の一覧表示,シェルオプションのセット .....	293
	位置パラメータをセットする .....	295
	変数の設定状況を表示する .....	298
	シェルオプションをセットする .....	299
3.23	<b>shift コマンド</b> ●位置パラメータの移動 .....	301
	引数をずらしながら処理する .....	302
	shiftコマンドを使ってシェルスクリプトのオプションと引数を分ける .....	304
3.24	<b>test コマンド</b> ●数値や文字列などの比較 .....	306
3.25	<b>trap コマンド</b> ●シグナルを受けたときの処理の設定 .....	307
	シグナルを受けたときの処理を設定する .....	308
	スクリプト終了時の処理を設定する .....	310
	trapコマンドの引数のクォーテーションを使い分ける .....	312
	trapコマンドを設定するタイミング .....	313
	trap処理を関数にする .....	316



3.26 typeコマンド●組み込みコマンドか外部コマンドかなどのチェック .....	318
3.27 umaskコマンド●ファイル作成時のファイルモードのデフォルトの決定 .....	319
umask コマンドを使いこなす .....	320
umask 値を表示する .....	321
3.28 unsetコマンド●変数や関数の未定義化 .....	322
配列を削除する .....	325
3.29 waitコマンド●プロセス終了の待機 .....	327
負荷によらずに一定時間待つ .....	329

## 第4章 変数と関数

4.1 拡張的な変数展開●変数の値の有無による設定／展開 .....	332
未定義の変数と未設定の変数の違い .....	333
変数の値がないときに代替表示する( <code>\${variable:-word}</code> ) .....	334
変数の値がないときに代入する( <code>\${variable:=word}</code> ) .....	335
変数の値がないときにメッセージを出力する( <code>\${variable:?word}</code> ) .....	337
変数の値がないときにメッセージを出力する( <code>\${variable:?word}</code> ) .....	339
変数の値があるときに別の値を代入する( <code>\${variable:+word}</code> ) .....	340
4.2 bashで追加された拡張的な変数展開●変数の値の編集／展開 .....	341
変数の値を切り出す .....	341
( <code>\${variable:offset}</code> , <code>\${variable:offset:length}</code> ) .....	341
合致する変数名をすべて表示する .....	343
( <code>\${!prefix*}</code> ) .....	343
変数の値の文字数を表示する .....	344
( <code>\${#variable}</code> ) .....	344
変数の値を文字列指定(前方一致)で削除する .....	345
( <code>\${variable#word}</code> , <code>\${variable##word}</code> ) .....	345
変数の値を文字列指定(後方一致)で削除する .....	348
( <code>\${variable%word}</code> , <code>\${variable%%word}</code> ) .....	348
変数の値を文字列指定で置き換える .....	349
( <code>\${variable/pattern/string}</code> , <code>\${variable//pattern/string}</code> ) .....	349
4.3 特殊なシェル変数●特別な意味をもっている変数 .....	350

直前のコマンドの終了コードを知る( <code>\$?</code> ) .....	351
シェルのプロセスIDを知る( <code>\$\$_</code> ) .....	353
直前のバックグラウンドジョブのプロセスIDを知る( <code>\$!</code> ) .....	354
カレントシェルの動作オプションを知る( <code>\$-</code> ) .....	355
4.4 <code>\$*</code> と <code>\$@</code> ●位置パラメータの一括取得 .....	356
すべての位置パラメータが代入されている変数としての違い .....	357
位置パラメータの値にスペースが含まれているときの違い .....	358
位置パラメータに何も値が入っていないときの違い .....	361
" <code>\$*</code> "や" <code>\$@</code> "を <code>\${variable:+word}</code> と組み合わせる .....	362
4.5 組み込み変数●シェルがあらかじめ定義する変数 .....	363
ホームディレクトリを参照する( <code>HOME</code> ) .....	367
コマンドのサーチパスをセットする( <code>PATH</code> ) .....	368
文字列の区切りを変更する( <code>IFS</code> ) .....	370
言語を設定する( <code>LANG</code> ) .....	372
4.6 シェル関数●使い回しのきくルーチン .....	373
シェル関数を作る .....	375
シェル関数を利用したサンプルスクリプト .....	377
ユーザ名／グループ名を取り出す関数を作る .....	380

## 第5章 シェルスクリプトのデバッグ

5.1 シェルオプションを用いたデバッグ●コマンドの展開の逐一表示 .....	384
シェルスクリプトを実行せずに構文チェックを行う( <code>-n</code> ) .....	385
実行するコマンド行を表示する( <code>-v</code> , <code>-i</code> ) .....	389
実行直前のコマンド行を表示してループ内の処理や変数の値を確認する( <code>-x</code> ) .....	391
スクリプトの一部だけを詳細表示する .....	394
5.2 echoコマンドを用いたデバッグ●変数の状態のチェック .....	396
変数の値をチェックする .....	397
変数の特別な処理をチェックする .....	399
関数などで変数を受け渡す場合のクォーテーションの指定忘れをチェックする .....	404
制御文をチェックする .....	408

5.3	変数やメタキャラクタのデバッグ●変数やメタキャラクタの扱いでの注意点 .....	410
	値のない変数をチェックする .....	411
	メタキャラクタの展開をチェックする .....	413
	引数に指定したファイルが存在するかチェックする .....	418
5.4	シェルスクリプトの処理速度●while文の弱点 .....	420

## 第6章 サンプルシェルスクリプト

6.1	サムネイル画像の自動作成●シェル関数の利用 .....	428
6.2	自動FTPと自動telnet●システムによる違いを組み込んだスクリプトの作成 .....	447

索引 .....	459
----------	-----

# 第 1 章

## シェルの基本機能

本章で記述しているのはシェルを使う上で最も基本的な機能や概念に関する内容です。スクリプトを作成するだけでなくUNIXシステムを使いこなすにも必須の内容です。