

シェルプログラミング 変数

[シェルプログラミング]

変数

- ・変数は宣言せずに使用
- ・すべて文字列として扱われる
- ・大文字小文字を区別
- ・変数の前に \$ をつけることで、値を取得できる
- ・変数に値を割り当てる場合以外 \$ を使う
- ・echo で内容を出力できる

ソース

```
msg=Hello  
echo $msg
```

結果

```
Hello
```

ユーザの入力

- ・read コマンドを使用すると、ユーザからの入力を変数に格納できる

ソース

```
echo your name ?  
read username  
echo your name is $username
```

結果

```
your name ?  
yagi  
your name is yagi
```

クォート

- ・二重引用符内の \$ 変数は値が展開される
- ・引用符内の \$ 変数は値が展開されない
- ・\ \$ は \$ をエスケープ

```
myvar="Hello"
```

ソース	出力
echo \$myvar	Hello
echo "\$myvar"	Hello

echo '\$myvar'	\$myvar
echo \\$myvar	\$myvar

環境変数

環境変数	説明
\$HOME	ホームディレクトリ
\$PATH	コマンド検索パス
\$PS1	コマンドプロンプト
\$IFS	内部フィールドセパレータ。シェルが入力を読むときに区切りに使用する文字
\$0	シェルスクリプトの名前
\$#	渡されたパラメータ数
\$\$	プロセス ID

• env

パラメータ変数

パラメータ変数	説明
\$1、\$2・・・	スクリプトに渡されたパラメータ
\$*	すべてのパラメータを1つの変数で表したもの \$IFS 使用
\$@	すべてのパラメータを1つの変数で表したもの \$IFS 使用しない

ソース (hoge)

```
IFS=' '
echo "$1"
echo "$2"
echo "$*"
echo "$@"
```

結果

```
$ sh hoge aaa bbb ccc
aaa
bbb
aaabbbccc
aaa bbb ccc
```

上記はこの本からの覚書。

非常にわかりやすく説明されている良書