

# Perl

[Ruby]

## ■コメント

- ・先頭に # をつける

```
#comment
```

## ■変数

```
$age = 35;          # $ではじめるとスカラ変数
print "my age is $age \n";  # ダブルクォート文字列
print 'my age is $age ', "\n"; # シングルクォート文字列 変数展開されない
```

| 宣言 | 変数   |
|----|------|
| \$ | スカラ  |
| @  | 配列   |
| %  | ハッシュ |

## ■数値演算

算術演算

```
$n1 = 2;
$n2 = 10;
print $n1 + $n2 , "\n";      # 加算
print $n1 ** $n2 , "\n";    # べき乗
```

文字列演算

```
$n1 = 2;
$n2 = 10;
print $n1 , $n2 , "\n";      # 連結 (,)
print $n1 x $n2 , "\n";      # 繰り返し (x)
```

論理演算

```
$b1 = true;
$b2 = false;
print $b1 && $b2 , "\n";      # 論理積 ( and )
print $b1 || $b2 , "\n";      # 論理和 ( or )
print !$b1 , "\n";            # 否定 ( not )
```

## ■配列

リスト配列

```
@week = ("sun", "mon", "tue", "wed", "thu", "fri", "sat"); # 宣言
print "$week[0] $week[1] $week[2] $week[3]\n";          # アクセス
$week_cnt = @week;
print "$week_cnt\n";                                     # 配列の個数
```

| 操作      | 内容          |
|---------|-------------|
| push    | 配列の末尾へ要素を追加 |
| pop     | 配列の末尾の要素を削除 |
| unshift | 配列の先頭へ要素を追加 |
| shift   | 配列の先頭の要素を削除 |
| reverse | 配列の順序を逆順    |
| sort    | 配列の順序をソート   |

## マップ配列

```
%eng_num = ("1", "one", "2", "two", "3", "tree");
print "$eng_num{'1'}\n"; # 取得

%alpha = (                                     # 別の宣言方法
    "a" => "A",
    "b" => "B",
    "c" => "C",
);
print "$alpha{'b'}\n";

$lang{"1"} = "C"; # さらに別の宣言
$lang{"2"} = "C++";
$lang{"3"} = "Java";
print "$lang{'3'}\n";
```

| 操作     | 内容           |
|--------|--------------|
| keys   | すべてのキーを取り出す  |
| values | すべての値を取り出す   |
| each   | 1組のキーと値を取り出す |
| delete | 特定の要素を取り除く   |

eg.

```
@alpha_keys = keys(%alpha);
```

## ■制御

### 選択

```
$cnd = 1;
if ($cnd == 1) {
    print "1";
} elsif ($cnd == 2) { # elsif じゃない
    print "2";
}
```

```

    } else {
        print "else";
    }

```

繰り返し

for

```

for ($i=0; $i<5; $i++) {
    print "$i\n";
}

```

while

```

$check = 0;
while (true) {
    print "$check\n";
    if (++$check > 4) {
        last;          # break じゃない
    }
}

```

## ■ファイル操作

読込

```

open IN, "<c:\\test.txt";
while ($line = <IN>) {
    print $line;
}
close IN;

```

書出し

```

open OUT, ">c:\\out.txt";
print OUT "Hello!";
close OUT;

```

## ■サブルーチン

```

print add(5, 6);

sub add {
    ($n1, $n2) = @_;      # パラメータの受取
    return ($n1 + $n2);   # 戻値
}

```

ローカル変数、My 変数のスコープ

| 種類    | スコープ                               |
|-------|------------------------------------|
| local | 変数を宣言したサブルーチンから呼び出したサブルーチンでも参照できる。 |

|    |                               |
|----|-------------------------------|
| my | いわゆる自動変数。宣言したサブルーチンからのみ参照できる。 |
|----|-------------------------------|

## ■正規表現

```
$s = "30/Jan/2007:02:39:34 +0900";
# 一致 =
if ($s =~ /([0-9]{4}):[0-9]{2}:[0-9]{2}/) {
    print "$1\n";
}
# 不一致 !=
if ($s !~ /abc/) {
    print "no match\n";
}
```

## ■書式

```
printf("%05d $line", $lin_num);
```

### 型指定

| 記号 | 内容    |
|----|-------|
| s  | 文字列   |
| d  | 整数    |
| f  | 浮動小数点 |

### フラグ

| フラグ | 内容       |
|-----|----------|
| -   | 左詰め      |
| 0   | 左側を 0 埋め |