

2015年11月24日

---

# プログラミング入門

## 配列と文字列

---

横浜国立大学

倉光君郎

# 配列

---

配列 (array) とは、値が並んだデータ構造である。

# 配列と型

---

T型の値が並んだ配列は、T[] 型で型付けされる。

```
int[] a;    // int 型の配列
```

# 配列の初期化

---

配列は、1,2,3,4,5 のように値の並びで初期化できる。

```
int[] a = {1,2,3,4,5};
```

# 空の配列

---

配列は、要素数10個の空配列は次のとおり作ることができる。

```
int[] a = new int[10]; // Java
```

言語風

```
int[10] a; // C言語風
```

# 配列の要素

---

配列の要素は、 $a[i]$  のように添え字でアクセスできる。 $i$  は、0 から始まる。

$a = \{1, 2, 3\};$

$a[0]$

$a[1]$

# 配列の要素

---

配列の要素は、 $a[i]$  のように添え字で代入できる。 $i$  は、0 から始まる。

```
a = {1, 2, 3};
```

```
a[0] = 2;
```

```
a[1] = 3;
```

# 配列の要素

---

配列の要素数を越えた要素にはアクセスできない。

```
a = {1,2,3};
```

```
a[3];
```



# 配列の要素数

---

配列の要素数は、size() メソッドでえることができる。

```
a.size()
```

# メソッドとは

---

メソッドとは、オブジェクト指向プログラミングのメッセージの概念を関数風に扱えるようにしたものである。

配列は、オブジェクトであり、 `a.size()` のように書き、オブジェクト `a` に `size()` というメッセージを送って評価している。

`a.size()`

基本的に、`size(a)` と同じであるが、メソッドは

a.size() のように書く。

# 文字列

---

文字列とは、コンピュータ上でテキストを表す値のこと。String 型で型付けられている。"abc" のようにダブルクォートで囲む。

"abc"

"hello,world"

特殊な文字は、次のようなエスケープシーケンスで指定する。

<code>\n</code>	// 改行
<code>\t</code>	// タブ
<code>\"</code>	// ダブルクオート
<code>\\</code>	// バックスラッシュ

文字列は、文字の配列である。配列と同様に  $n$  番目の文字を取り出すことができる。

# 文字と文字コード

---

文字は、コンピュータ内では文字コードで表現されている。

文字コードは、ASCII コード, UNICODE コードが存在する。

Konoha は、Java とおなじく、UNICODE で char 型で型付けられている。

# printf (C 言語風)

printf は、プログラムの実行中の変数の状態を示す関数である。整数  $x$  の変数の値を見たいときは、次のように書く

```
printf("x=%d\n", x);
```

# println (Java 風)

println() は、プログラムの実行中の変数の状態を示す。整数  $x$  の変数の値を見たいときは、次のように書く

```
println("x=" + x);
```



# println (スクリプト風)

print は、プログラムの実行中の変数の状態を示す。整数  $x$  の変数の値を見たいときは、次のように書く

```
println(' 'x=${x}' '');
```

どれでもよい。