

モバイルシステム演習

Processing

- データ型と変数 -

株式会社GOCCO.

watanabe@gocco.co.jp

ログを出力してみよう

```
println("ログを出力する");
```

数值

宣言

型 変数名, 変数名2, 変数名3…;

型 変数名;

型 変数名 = 初期値;

整数

```
int a, b;  
int c;  
a = 20;  
b = 5;  
c = a + b;  
println(c);
```

少数

```
float d, e, f;  
d = 2.5;  
e = 1.2;  
f = d * e;  
println(f);
```

boolean

宣言

型 変数名, 変数名2, 変数名3…;

型 変数名;

型 変数名 = 初期値(true / false);

```
boolean bool1, bool2;  
bool1 = true;  
bool2 = !bool1;  
println(bool1);  
println(bool2);
```

unicode

宣言

型 変数名, 変数名2, 変数名3…;

型 変数名;

型 変数名 = '初期値';

```
char chr = 'A'; // aをそのまま代入
char chr2 = 0x0041; // 0x0041を指定
char chr3 = 65; // 65を指定
println(chr);
println(chr2);
println(chr3);
```

// NOTE: Javaの文字コードはUnicode。

// 「"」 シングルクォートで囲う。

// LINK : https://ja.wikipedia.org/wiki/Unicode%E4%B8%80%E8%A6%A7_0000-0FFF

文字列

宣言

型 変数名, 変数名2, 変数名3…;

型 変数名;

型 変数名 = 初期値;

```
// String str1_error = 0x0041;  
// println(str1);  
String str = "名古屋文理大学";  
println(str);
```

// NOTE: String型は、複数の文字を格納できるデータ型です。

// 1文字でも2文字でもそれ以上でも大丈夫です。char型の時は、文字を「」で囲みましたが、

// String型の場合は文字列を「」で囲むという違いがあります。

// char型の時は文字コードを直接代入できましたが、String型ではこれはいけません。

配列

宣言

型[] 変数名, 変数名2, 変数名3…;

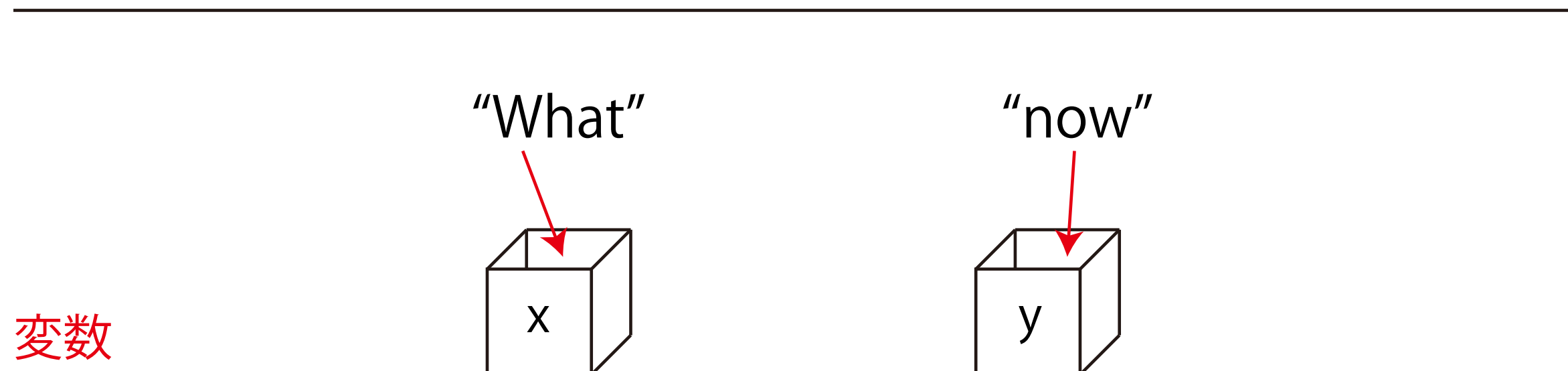
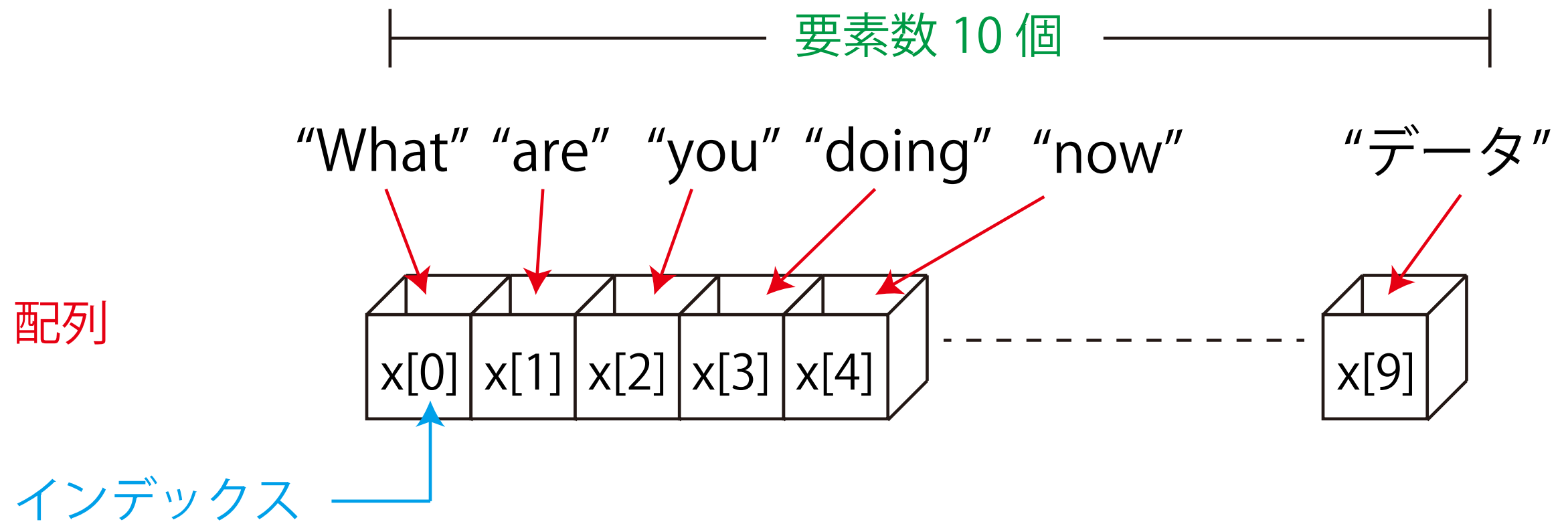
型[] 変数名;

型[] 変数名 = 初期値;

```
int[] a, b;  
a = new int[10];  
b = new int[5];  
println("a配列の要素数は" + a.length + "個");  
  
// 初期値を設定した場合の宣言  
String[] strArr = new String[] {"abc", "def"};  
String[] strArr2 = {"ghi", "jkl"};  
println(strArr[0] + strArr2[1]);  
  
// 初期値を設定しない場合はデフォルト値が入る  
int[] c = new int[3];  
String[] d = new String[2];  
println(c[0]);  
println(d[0]);
```

配列 - Array -

変数と配列の違い



変数と配列の違い

配列

```
String[] strArr; //宣言(int型)  
strArr = new String[3]; //作成 3は要素の数  
strArr[0]="What"; //代入
```

変数

```
String str = ""; //宣言  
str="What"; //代入
```

配列

宣言

型[] 変数名, 変数名2, 変数名3…;

型[] 変数名;

型[] 変数名 = 初期値;

// NOTE:

// String[] ss;

// ○ ss = new String[]{ "abc", "def" };

// × ss = { "abc", "def" }; ←この書き方は出来ない

// ○ int[] a = { 1, 2, 3};

// ○ a = new int[]{ 4, 5, 6};

// × a = { 7, 8, 9 };