

■Java プログラムの形式

Java プログラムは「クラス」の集合体

```
import 文

public class クラス名 {
    public static void main(String[] arg) {
        (ここが main メソッド)
        java の文 ;
        java の文 ;
        :
    }
}
```

- ①import文 → 他のクラスの機能を利用したいときに指定する
- ②クラス → プログラムを構成する単位 (クラス名=プログラム名)
- ③mainメソッド → プログラム本体を記述する部分

■プログラムの作成から実行まで

(1)プログラムの新規作成

Eclipse 画面 **ファイル** - **新規** - **クラス** を開き、クラス名を入力する

例 [Sample_1]

クラス名は大文字で始まる英数字にする。区切り記号は_ (アンダーバー) を用いる。

(2)プログラムの入力

```
public class Sample_1 {
    public static void main(String[] args){
        System.out.println("文教大学");
    }
}
```

← ソースプログラム
(大文字と小文字に注意！)

■画面に文字を表示する文(print)

System.out.println("メッセージ"); → 表示後に改行あり
System.out.print("メッセージ"); → 改行なし

(任意の位置で改行する場合)

System.out.print("¥n"); (¥n は改行の意味)

■定数と変数

整数	int	10進数	-2,147,483,648 ~ +2,147,483,647	
		8進数	数字の先頭が0で始まる (010 077 など)	
		16進数	数字の先頭が0xで始まる (0x10 0xFFFF など)	
実数	float	単精度	$-3.4 \times 10^{38} \sim +3.4 \times 10^{38}$	
	double	倍精度	$-1.7 \times 10^{308} \sim +1.7 \times 10^{308}$	
文字	char	シングルコーテーションで囲んだ単一文字	('a' '5' '文' など)	
文字列	String	ダブルコーテーションで囲んだ文字列	("bunkyo" "文教大学" など)	

■変数の利用

型名 変数名 ;
 (定義) int number;
 double d;
 char c;
 String str;

変数名のつけ方

- ・先頭は半角小文字
- ・英語表記または英語の略語が基本
- ・単語の区切りは大文字 (例 fieldName)

(代入) number=100;
 str="bunkyo";

(注) 右辺の値が左辺の変数に代入される

(変数の表示例)

変数のみを表示	System.out.println(number);
メッセージ付きで表示	System.out.println("数値="+number);
8進数で表示	System.out.println(Integer.toOctalString(number));
16進数で表示	System.out.println(Integer.toHexString(number));

+記号で文字列を連結できる

■キーボードからの入力

(文字列の入力)

```
import java.util.*; // java.util のインポート
public static void main(String[] args) {

    Scanner sc = new Scanner(System.in); // キーボード入力の準備 (スキャナクラス)
    System.out.print("文字列を入力してください-->");

    String keybd; // String 型変数の定義
    keybd=sc.nextLine(); // 入力した文字列の読み込み
}
```

文字列以外の値を処理したい場合は点線部分を次のようにする。

整数を入力する場合

```
int number;
number=sc.nextInt();
```

実数を入力する場合

```
double realNum;
realNum=sc.nextDouble();
```

1文字を入力する場合

```
String keybd;
keybd=sc.nextLine();

char c;
c=keybd.charAt(0); } (この2行を追加する)
```