JAVA 基本プログラム

■

```
class DisplayFloat {
   public static void main(String args[]) {
      float price;
      price = 45.35f;
      System.out.print("The price is ");
      System.out.println(price);
   }
}
```

■

```
public class OutputVariables {
   public static void main(String args[]) {
      char ch = 'X';
      short s = 456;
      double d = 123.009;
      System.out.println("ch is " + ch);
      System.out.println("s is " + s);
      System.out.println("d is " + d);
   }
}
```

■

```
public class OutputVariables {
   public static void main(String args[]) {
      char ch = 'X';
      short s = 456;
      double d = 123.009;
      System.out.println("ch is " + ch);
      System.out.println("s is " + s);
      System.out.println("d is " + d);
   }
}
```

■

```
class LincolnQuote {
   public static void main(String args[]) {
    String s = "Lincoln said: " +
      "¥"Four score and seven years ago¥"";
     System.out.println(s);
   }
}
```

■

```
 class Concatenation {
    public static void main(String args[]) {
       System.out.println("My book " +
          "will teach you " +
          "about Java programming");
    }
}
```

■

```
class StringVariables {
    public static void main(String args[]) {
     String s1 = "My book teaches ";
     String s2 = "you how to ";
     String s3 = "use Java";
      System.out.println(s1 + s2 + s3);
    }
}
```

■東京から大阪までを時速 75km **で走行した場合の所要時間を計算しなさい**
東京-大阪を 600km **とする。**

```
class DrivingTime {
   public static void main(String args[]) {
    float h =              ;
     System.out.println（
       "  東京から " +
        "大阪までの所要時間 "+h);
   }
```

}

配列

まず、配列の型を定義します。

```
int   ia[];
```

この状態では配列の領域は定義されていません。

```
varName=new type [size];
 ia=new int[10];
```

これによって、10 個の要素を持つ、int 型配列が作成されます。

■

```
class OneDimensionArray {

  public static void main(String args[]) {

    // 配列を宣言して領域を割り当てる
    int myarray[] = new int[4];

    // 要素を初期化する
    myarray[0] = 33;
    myarray[1] = 71;
    myarray[2] = -16;
    myarray[3] = 45;

    // 要素数を表示する
    System.out.println("myarray.length = " +
      myarray.length);

    // 要素を表示する
    System.out.println(myarray[0]);
    System.out.println(myarray[1]);
    System.out.println(myarray[2]);
    System.out.println(myarray[3]);
  }
}
```

■

```
class ArrayInitializer {

   public static void main(String args[]){

      // 配列を宣言し，領域を割り当て，初期化する
      int myarray[] = { 33, 71, -16, 45 };

      // 要素数を表示する
      System.out.println("myarray.length = " +
         myarray.length);

      // 要素を表示する
      System.out.println(myarray[0]);
      System.out.println(myarray[1]);
      System.out.println(myarray[2]);
      System.out.println(myarray[3]);
   }
}
```

■

```
class ArrayReference {

  public static void main(String args[]){

     // array1 を宣言し，領域を割り当てる
     float array1[] = new float[3];

     // array1 を初期化する
     array1[0] = -3.45f;
     array1[1] = 7.7f;
     array1[2] = 101.56f;

     // array2 を宣言し，領域を割り当てる
     float array2[] = new float[3];
```

```
    // array2 に array1 と同じ配列を参照させる
    array2 = array1;

    // array2 の要素を表示する
     System.out.println("array2:");
     System.out.println(array2[0]);
     System.out.println(array2[1]);
     System.out.println(array2[2]);

    //  要素を変更する
    array2[1] = 100;

    // array1 の要素を表示する
     System.out.println("array1:");
     System.out.println(array1[0]);
     System.out.println(array1[1]);
     System.out.println(array1[2]);

    // array2 の要素を表示する
     System.out.println("array2:");
     System.out.println(array2[0]);
     System.out.println(array2[1]);
     System.out.println(array2[2]);
  }
}
```

多次元配列

二次元またはそれ以上の次元の配列が利用できます。

```
type varName[][];
float fa[][];

varName=new type[size1][size2];
fa=new float[2][3];
class TwoDimensionArray {
```

```
  public static void main(String args[]){

    // 配列を宣言して領域を割り当てる
     int myarray[][] = new int[3][2];

    // 要素を初期化する
     myarray[0][0] = 33;
     myarray[0][1] = 71;
     myarray[1][0] = -16;
     myarray[1][1] = 45;
     myarray[2][0] = 99;
     myarray[2][1] = 27;

    // 要素数を表示する
     System.out.println("myarray.length = " +
        myarray.length);

    // 要素を表示する
     System.out.println(myarray[0][0]);
     System.out.println(myarray[0][1]);
     System.out.println(myarray[1][0]);
     System.out.println(myarray[1][1]);
     System.out.println(myarray[2][0]);
     System.out.println(myarray[2][1]);
  }
}
```

宿題

if **文の使い方**

for **ステートメント**（while, do, switch, continue, break **を含む**）

インクリメントとデクリメント演算子

バックスラッシュ・コード

■

```
class SquareRoot {
  public static void main(String args[]) {
    double d = Double.valueOf(args[0]).doubleValue();
    if(d < 0)
       System.out.println(Math.sqrt(-d) + "i");
    if(d >= 0)
       System.out.println(Math.sqrt(d));
  }
}
```

■

```
class Angle {
  public static void main(String args[]) {
     if(args.length > 0) {
       double angle =
          Double.valueOf(args[0]).doubleValue();
      double radians = angle * Math.PI/180;
       System.out.println("cosine: " + Math.cos(radians));
       System.out.println("sine: " + Math.sin(radians));
       System.out.println("tangent: " +
          Math.tan(radians));
     }
     else {
       System.out.println("Provide an angle in degrees " +
        "as command line argument");
     }
  }
}
```

■

```
public class ForDemo {
  public static void main(String args[]) {
      for(int num = 1; num < 11; num = num + 1)
         System.out.print(num + " ");
      System.out.println("terminating");
   }
}
```

■

```
public class ProductAndSum {
  public static void main(String args[]) {
      int sum = 0;
      int prod = 1;
      for(int num = 1; num < 6; num = num + 1) {
       sum = sum + num;
       prod = prod * num;
      }
      System.out.println("product and sum: " + prod + " " + sum);
   }
}
```

■

```
class CountLettersDigits {
  public static void main(String args[]) {
      int digits = 0;
      int letters = 0;
      for(int i = 0; i < args[0].length(); i = i + 1) {
        char ch = args[0].charAt(i);
        if(Character.isDigit(ch))
          digits += 1;
        else if(Character.isLetter(ch))
          letters += 1;
      }
      System.out.println("There are " + digits + " digits");
```

```
    System.out.println("There are " + letters + " letters");
  }
}
```

■

```
class CountLettersDigits2 {
  public static void main(String args[]){
    int digits = 0;
    int letters = 0;
    for(int i = 0; i < args[0].length(); i++) {
      char ch = args[0].charAt(i);
      if(Character.isDigit(ch))
        ++digits;
      else if(Character.isLetter(ch))
        ++letters;
    }
    System.out.println("There are " + digits +
      " digits");
    System.out.println("There are " + letters +
      " letters");
  }
}
```

■

```
class IncrementDecrement {
  public static void main(String args[]){
    int i;
    i = 0;
    System.out.println(++i);   // 1 を表示
    System.out.println(i++);   // 1 を表示
    System.out.println(i);     // 2 を表示
    System.out.println(--i);   // 1 を表示
    System.out.println(i--);   // 1 を表示
    System.out.println(i);     // 0 を表示
  }
}
```

■コマンドラインの引数

```
class Add2Integers {

  public static void main(String args[]) {

    // 最初の整数を取得する
     int i = Integer.parseInt(args[0]);

    // 次の整数を取得する
     int j = Integer.parseInt(args[1]);

    // 合計を表示する
     int sum = i + j;
     System.out.println("Sum is " + sum);
  }
}
```