STRING

```
public class StringExample {
   public static void main(String[] args) {
      String greeting = "Hello" ;

      // length, charAt
      for ( int i = 0 ; i < greeting.length() ; i ++ )
         System.out.println(greeting.charAt(i)) ;

      // substring (int beginIndex, int endIndex) : [beginIndex .. endIndex-1]
      String hel = greeting.substring(0, 3) ;
      System.out.println(hel);    // Hel

      // concatenation
      String language = "Java !" ;
      String msg = greeting + " " + language ;
      System.out.println("Welcome to " + msg) ;    // Welcome to Hello Java !

      // equality, use equals; DO NOT USE ==
      if ( greeting.equals("hello"))
         System.out.println("Exactly same!") ;
      if ( greeting.equalsIgnoreCase("hello"))
         System.out.println("Same when case ignored") ;       // this executed
      // comparison
      if ( greeting.compareTo(language) < 0 )
         System.out.println(greeting + " comes before " + language) ;  // this executed
      else if ( greeting.compareTo(language) > 0 )
         System.out.println(greeting + " comes after " + language) ;
      else
         System.out.println(greeting + " equals with " + language) ;
```

```
        // replacement
        String greeting2 = greeting.replace('l', 'L') ;
        System.out.println("The original string: " + greeting + " After replacement: " + greeting2) ;

        // indexOf, lastIndexOf
        System.out.println(greeting.indexOf('l')) ;       // 2
        System.out.println(greeting.lastIndexOf('l')) ; // 3
        System.out.println(greeting.indexOf('L')) ;       // -1
        System.out.println(greeting.indexOf("lo")) ;    // 3

        // startsWith, endsWith
        System.out.println(greeting.startsWith("He"));              // true
        System.out.println(greeting.startsWith("he"));              // false
        System.out.println(greeting.endsWith("lo"));                // true
        System.out.println(greeting.startsWith("hlo"));             // false

        // split(Strig regex)
        String line = "first : second : third";
        String[] items1 = line.split(":");
        System.out.println(Arrays.asList(items1));        // [first ,  second ,  third]
        String[] items2 = line.split("₩₩s*:₩₩s*");
        System.out.println(Arrays.asList(items2));        // [first, second, third]

        // toLowerCase, toUpperCase, trim

        // join Since Java 8
        System.out.println(String.join("-", "I", "Love", "Java"));    // I-Love-Java
    }
}
```

Splitting String

```
public class StringSplitExample {
   public static void main(String[] args) {
      String message1 = "Hello₩tWorld₩t₩tI₩t₩tLove₩tJava";

      String[] words11 = message1.split("₩t");
      for ( int i=0; i < words11.length; i++ )
         System.out.println(i + ": [" + words11[i] + "]");

      String[] words12 = message1.split("₩t+");
      for ( int i=0; i < words12.length; i++ )
         System.out.println(i + ": [" + words12[i] + "]");

      String message2 = "Hello₩tWorld₩nLove  Java";

      String[] words21 = message2.split("₩₩s");
      for ( int i=0; i < words21.length; i++ )
         System.out.println(i + ": [" + words21[i] + "]");

      String[] words22 = message2.split("₩₩s+");
      for ( int i=0; i < words22.length; i++ )
         System.out.println(i + ": [" + words22[i] + "]");
   }
}
```

```
0: [Hello]
1: [World]
2: []
3: [I]
4: []
5: [Love]
6: [Java]
```

```
0: [Hello]
1: [World]
2: [I]
3: [Love]
4: [Java]
```

```
0: [Hello]
1: [World]
2: [Love]
3: []
4: [Java]
```

```
0: [Hello]
1: [World]
2: [Love]
3: [Java]
```

Conversion between Number and String

```
public class NumberBetweenString {
   public static void main(String[] args) {
      // 1) String ==> Number
      String intString = "100", floatString = "1.234F" ;

      // valueOf() return Wrapper object
      int a = Integer.valueOf(intString);          // Auto unboxing: Integer -> int
      float b = Float.valueOf(floatString);        // Auto unboxing: Float -> float
      System.out.println( a + " " + b ) ;          // 100 1.234

      // or use parseXXX()
      a = Integer.parseInt(intString) ;
      b = Float.parseFloat(floatString) ;

      // 2) Number ==> String
      Integer intValue = 100 ;
      String strI = intValue.toString() ;
      System.out.println(strI) ; // 100

      float f = 1.234F ;
      String strF = Float.valueOf(f).toString() ;
      System.out.println(strF) ; // 1.234
   }
}
```

valueOf(): better space and time performance

Formatting String

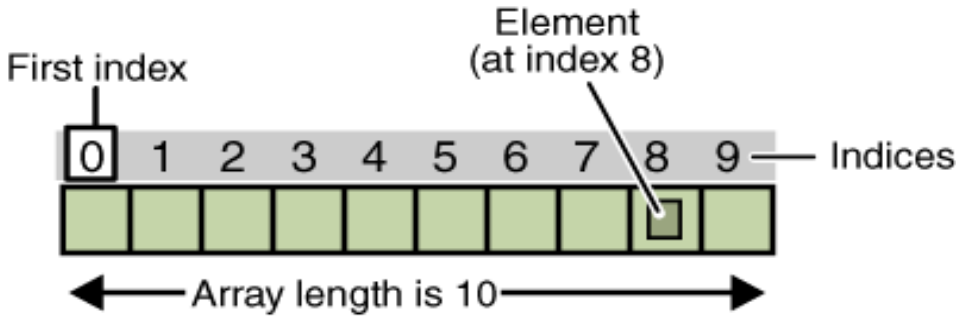
```
public class StringFormat {
    public static void main(String[] args) {
        String str1 = String.format("%d", 101);                  // Integer value
        String str2 = String.format("|%15d|", 101);              // length and right-justified
        String str3 = String.format("|%-15s|", "Hello, Java");   // left-justified
        String str4 = String.format("|%015f|", 101.00);          // leading zeros
        String str5 = String.format("|%15.2f|", 101.00);

        String str6 = String.format("%x", 101);                  // Hexadecimal value

        System.out.println(str1);
        System.out.println(str2);
        System.out.println(str3);
        System.out.println(str4);
        System.out.println(str5);
        System.out.println(str6);
    }
}
```

```
101
|            101|
|Hello, Java     |
|00000101.000000|
|          101.00|
65
```

ARRAY

Arrays

- int [] intArray = new int[10] ;
  - An array of size 10
  - Index starts at 0.




```
class ArrayExample1 {
    public static void main(String[] args) {
        int [] ia = {0, 1, 2, 3} ;
        for (int i = 0; i < ia.length; i++)
            System.out.println(ia[i]);
    }
}
```

Arrays: An Example

```
import java.util.Random;

public class ArrayExample2 {
   public static void main(String[] args) {
      Random oRandom = new Random() ;

      int [] ia = new int[101];
      for ( int i = 0; i < ia.length; i++ ) {
         ia[i] = oRandom.nextInt(100) ;          // [ 0 .. 100 )
         System.out.println(ia[i]) ;
      }

      int sum = 0;
      for ( int v : ia ) // Enhanced for loop(for each loop): array and Collection
         sum += v;
      System.out.println(sum);
   }
}
```

Copying Arrays

- Shallow copy

```
int [] smallPrimes = {2, 3, 5, 7, 11, 13} ;
int [] luckyNumbers = smallPrimes ;
luckyNumbers[5] = 12 ; // now smallPrimes[5] is also 12
```

- Deep copy: System.arraycopy(from, fromIndex, to, toIndex, count) ;

```
class ArrayCopy {
   public static void main(String args[]) {
      int [] smallPrimes = {2, 3, 5, 7, 11, 13} ;
      int [] luckyNumbers = {1001, 1002, 1003, 1004, 1005, 1006, 1007};
      System.arraycopy(smallPrimes, 2, luckyNumbers, 3, 4) ;
      for ( int v : luckyNumbers )
         System.out.print(v + " ") ;                    // 1001 1002 1003 5 7 11 13
   }
}
```

Arrays Class

- java.util.Arrays class provides useful array operations.

```
public class ArraysExample {
   public static void main(String[] args) {
      int[] array1 = new int[10];
      for(int i = 0; i < array1.length; i++) array1[i] = i;
      System.out.println(Arrays.binarySearch(array1, 7)); // 7

      int[] array2 = Arrays.copyOf(array1, 10);   // truncating or padding with zeros (if necessary)
      for (int v: array2) System.out.print(v + " "); // 0 1 2 3 4 5 6 7 8 9
      System.out.println(Arrays.equals(array1, array2)); // true

      int[] array3 = Arrays.copyOfRange(array1, 2, 5);          // [from .. to )
      System.out.println();
      for (int v: array3) System.out.print(v + " "); // 2 3 4
      System.out.println(Arrays.equals(array1, array3)); // false

      int[] array4 = new int[5];
      Arrays.fill(array4, 7);
      System.out.println();
      for (int v: array4) System.out.print(v + " "); // 7 7 7 7 7

      System.out.println();
      System.out.println(Arrays.asList("Hello", "Java"));
   }
}
```

int [] array vs int array []

- Q: int [] ia and int ia [] are same ?
- A: They are different! Use int [] ia rather than int ia[].

```
class ArrayInit {
    public static void main(String args[]) {
        int[] a1 = {10, 20, 30}, a2 = {100, 200, 300} ;
        int a3[] = new int[10], a4 = a1 ;
        // ERROR: incompatible types, found: int[], required: int
    }
}
```

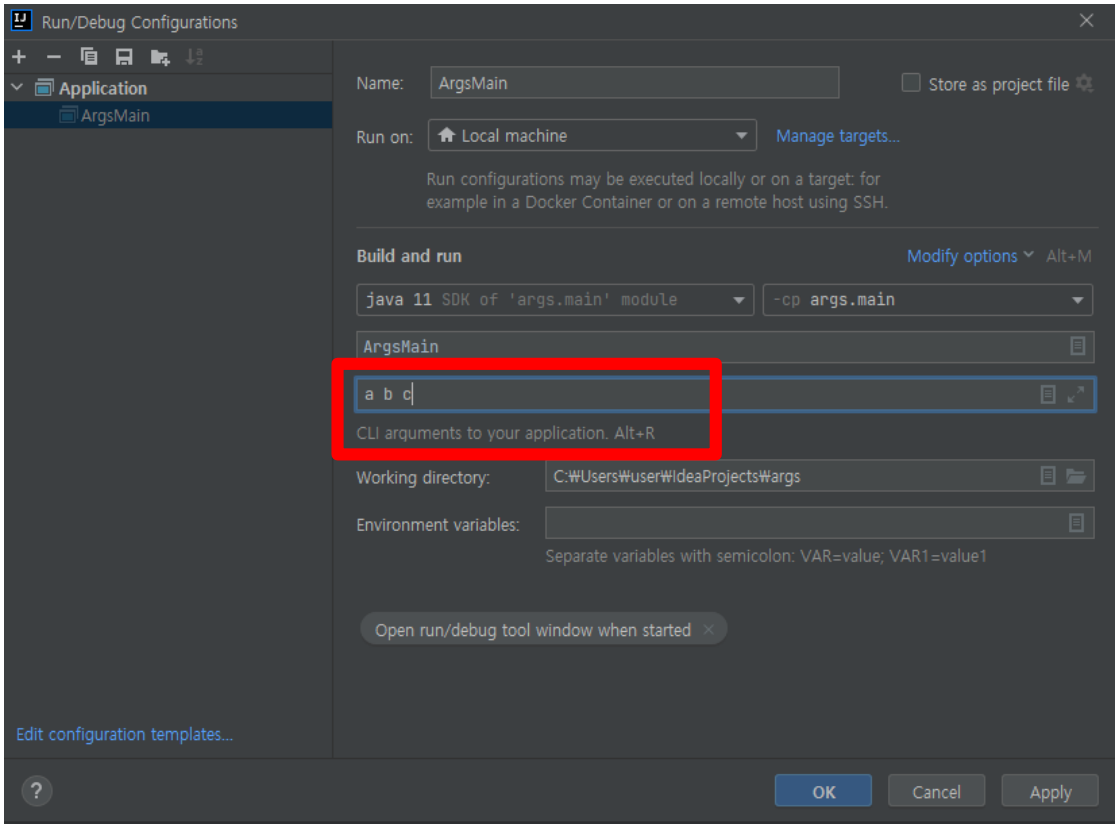
Program Arguments in IntelliJ

❖ Run – Edit Configurations…

```
public class ArgsMain {
    public static void main(String[] args) {
        for (String arg : args)
            System.out.println(arg);
    }
}
```

실행결과

a
b
c



Constants

- You can use the keyword **final** to denote a constantness for local variable and parameter

```
import java.util.Scanner;
public class FinalVariableParameter {
   public static void main(String[] args) {
      Scanner scanner = new Scanner(System.in) ;
      final int n = scanner.nextInt() ;
      scanner.close() ;
      // n = 200 ; final local variable cannot be assigned!
      System.out.printf("Factorial of " + n + ": %,20d", factorial(n)) ;
   }
   public static long factorial(final int v) {
      // v = 100 ; final local variable cannot be assigned!
      long result = 1 ;
      for ( int i = 2 ; i <= v ; i ++ ) result *= i ;
      return result ;
   }
}
```