



Chapter 2

Array & String



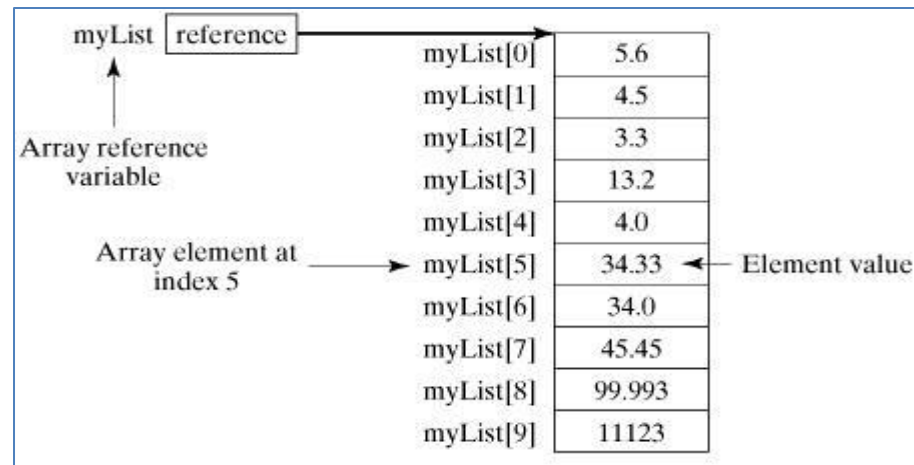
อาเรย์ 1 มิติ

- การประกาศอาเรย์
- `type array-variable[];`
 - `type [] array-variable;`
 - ตัวอย่าง:
 - `int table[];`
 - `char[] grades;`
- หลังจากการประกาศ ไม่มีการสร้างอาเรย์เกิดขึ้นจริง



อาเรย์ 1 มิติ (Cont.)

- การสร้างอาเรย์ทำได้ดังนี้
 - `array-variable = new type[size];`
 - ตัวอย่าง :
 - `table = new int[5];`
 - `grades = new char[60]`





อาเรย์ 1 มิติ (Cont.)

- การเข้าถึงอาเรย์
 - เริ่มต้นที่ตำแหน่ง 0
 - การเข้าถึง: `array-variable[index]`
 - ตัวอย่าง: `table[0] = 1;`

- การใส่ค่าเริ่มต้นให้อาเรย์

- ตัวอย่าง:

```
int monthDays[] =  
    {31,28,31,30,31,30,31,31,30,31,30,31};
```



อาเรย์ 1 มิติ (Cont.)

- การหาขนาดของอาเรย์
 - คำสั่ง `length`

- ตัวอย่าง:

```
int[] arr = {1,2,3} ;  
for (int i = 0 ; i < arr .length ; i++){  
    System.out.println(arr [i]);  
}
```

อาเรย์ 2 มิติ



- การประกาศ

- `int array[][];`

- การสร้าง

- `int array = new int[2][3];`

- การใส่ค่าเริ่มต้น

- `int array[][] = { {1, 2, 3}, {4, 5, 6} };`

อาณัติ 2 มิติ (Cont.)



```
class Array {  
    public static void main(String args[]) {  
        int array[][] = { {1, 2, 3}, {4, 5, 6} };  
        int i, j;  
        for(i=0; i< array.length; i++) {  
            for(j=0; j< array[i].length; j++)  
                System.out.print(array[i][j] + " ");  
            System.out.println();  
        }  
    }  
}
```



foreach

- การประกาศ

```
for (type var : array)
{
    statements using var;
}
```




foreach (Cont.)

- ตัวอย่าง

```
int[] arr = {1,2,2,3} ;
```

```
for (int eachArr : arr) {  
    System.out.println(eachArr) ;  
}
```

```
for (int[] u: uu) { for (int elem: u) { // Your individual element } }
```



foreach (array 2D)

- ตัวอย่าง

```
int[][] arr2D= { {1, 2, 3}, {4, 5, 6} };
```

```
for (int[] arr1D : arr2D) {  
    for (int elem: arr1D) {  
        System.out.print(elem)  
    }  
}
```



String

- String คือข้อมูลชนิดตัวอักษรที่ต่อกัน
- การประกาศ

```
String greeting = "Hello world!";
```

หรือ

```
char[] helloArray = { 'h', 'e', 'l', 'l', 'o', '.' };
```

```
String helloString = new String(helloArray);
```

```
System.out.println( helloString );
```



String length()

- เราสามารถหาความยาวของ String ด้วยเมธอด length()
- ตัวอย่าง

```
String palindrome = "Dot saw I was Tod";
```

```
int len = palindrome.length();
```

```
System.out.println( "String Length is : " + len );
```



String charAt()

- เราสามารถเข้าถึงตัวอักษรใน String โดยการอ้างตำแหน่งด้วยวิธีการผ่านเมธอด charAt

```
public char charAt(int index)
```

- ตัวอย่าง

```
public static void main(String args[]) {  
    String s = "Strings are immutable";  
    char result = s.charAt(8);  
    System.out.println(result);  
}
```



for and String

- เราสามารถใช้การวนซ้ำเพื่อเข้าถึงทุก element ของ String ได้โดยสามารถใช้ได้ทั้ง for
- ตัวอย่าง

```
public static void main(String args[]) {  
    String s = "Strings are immutable";  
    for(int i = 0; i < s.length();i++){  
        System.out.println(s.charAt(i));  
    }  
}
```



foreach and String

- สำหรับ foreach จะต้องใช้เมธอด toCharArray() เพื่อแปลงข้อมูลจาก string ไปเป็นอาร์เรย์ของ char ก่อน
- ตัวอย่าง

```
public static void main(String args[]) {  
    String s = "Strings are immutable";  
    for(char c : s.toCharArray()) {  
        System.out.println(c);  
    }  
}
```



String comparison

- การเปรียบเทียบค่าใน String ให้ใช้ เมธอด equals ()
- ตัวอย่าง

```
public static void main(String args[]) {  
  
    String  strA = "I'm String";  
    String  strB = "I'm String";  
    boolean result = strA.equals(strB);  
  
}
```




String comparison

- ใช้เปรียบเทียบข้อความสองข้อความว่าเหมือนกันหรือไม่ โดยไม่สนใจว่า
จะเป็นตัวพิมพ์เล็ก หรือว่าตัวพิมพ์ใหญ่ หากเหมือนกัน จะเป็นค่า true
หากต่างกันจะเป็นค่า false
- ตัวอย่าง

```
public static void main(String args[]) {  
  
    String strA = "I'm string";  
    String strB = "I'm String";  
    boolean result = strA.equalsIgnoreCase(strB);  
  
}
```



Concatenating Strings

- String สามารถต่อกันได้ โดยใช้เมธอด concat

— `string1.concat(string2);`

- ตัวอย่าง

`"My name is ".concat("Zara");`

- หรือ ใช้เครื่องหมาย +

`"Hello," + " world" + "!"`



Concatenating Strings (Cont.)

```
public static void main(String args[]) {  
    String string1 = "My name is ".concat("Zara");  
    System.out.println(string1);  
}
```

```
public static void main(String args[]) {  
    String string1 = "saw I was ";  
    System.out.println("Dot " + string1 + "Tod");  
}
```



String indexOf()

- ใช้สำหรับหาดำแหน่งของตัวอักษรที่เราต้องการว่าอยู่ตำแหน่งใดในข้อความ หากไม่เจอ จะ return เป็น -1 เช่น ใช้หา ตัวอักษร "e" ของคำว่า "Hello"
- ตัวอย่าง

```
String word = "Hello";  
int index = word.indexOf('e');
```



String lastIndexOf()

- ใช้สำหรับหาดำแหน่งของตัวอักษรตำแหน่งสุดท้ายที่เราต้องการว่าอยู่ตำแหน่งใดในข้อความ หากไม่เจอ จะ return เป็น -1 เช่น ใหาตัวอักษร "l" ของคำว่า "Hello" จะได้ตำแหน่ง ของ "l" ตัวที่สอง เนื่องจากเป็นตัวสุดท้าย
- ตัวอย่าง

```
String word = "Hello";
```

```
int index = word.lastIndexOf("l");
```



String replace()

- ใช้เปลี่ยนตัวอักษร โดยเปลี่ยนตัวอักษรตัวเก่า เป็นอักษรตัวใหม่ที่ต้องการ
 - เช่น ต้องการเปลี่ยน 'o' เป็น 'a' ใน "Hello"
- ตัวอย่าง

```
String word = "Hello";  
String newWord = word.replace('o', 'a');
```



String replace()

- ใช้เปลี่ยนตัวอักษร หรือข้อความใหม่ สามารถเปลี่ยนแค่ตัวอักษรเดียว หรือเปลี่ยนทั้งคำได้ เช่น ต้องการเปลี่ยนคำว่า "World" ไปเป็น "Java"
- ตัวอย่าง

```
String word = "Hello World";  
String newWord = word.replace("World", "Java");
```



String Split()

- ใช้แบ่งคำตัวอักษรด้วย Regex ให้อยู่ในรูปแบบ array
- เช่นข้อความ "J,A,V,A " เราจะแบ่งคำไปเป็น array ด้วยตัว ", " ทุกครั้งที่มันเจอ ", " มันก็จะทำการตัดคำทั้งหมดก่อนเจอ ", " ไปไว้ใน array แล้วก็หาตัวต่อไปเจออีก ก็จะตัดคำไปใส่ array อีกซึ่ง array จะมีค่าดังนี้ {"J", "A", "V", "A"}



String Split()

- ตัวอย่าง

```
String[] value= "Hello,World".split(",");  
  
for(int i = 0; i < value.length; i++) {  
    System.out.println(value[i]);  
}
```



String Split()

- ใช้เพื่อต้องการตัดคำบางส่วนออกจาก String โดย index คือตำแหน่งที่จะเริ่มทำการตัด

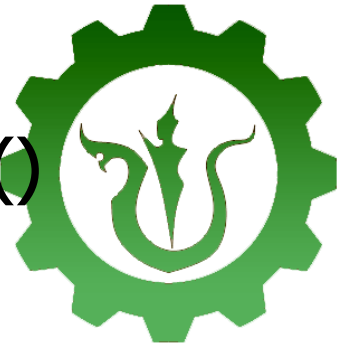
ตัวอย่าง

```
String word = "JavaWorld".substring(4);
```

ผลลัพธ์

World

String toLowerCase() and toUpperCase()



- แปลง String เป็นตัวพิมพ์เล็ก พิมพ์ใหญ่

ตัวอย่าง

```
String word = "JavaWorld".toLowerCase();
```

หรือ

```
String word = "JavaWorld".toUpperCase();
```



String trim()

- ไข่ไก่ตัดของว่างต่าง ๆ ด้านหน้าและด้านหลังของข้อความทิ้งไป เช่น " Hello Word " มีทั้งของว่างด้านและด้านหลังเลย เมื่อไข่ trim มันก็จะทำการตัดของว่างก่อน Hello และของว่างหลัง World ทิ้ง
- โดยของว่างระหว่างคำ จะไม่มีผล

ตัวอย่าง

```
String word = "Java World".trim();
```



Type Conversions

- การแปลงข้อมูลในภาษา Java เพื่อที่จะแปลงข้อมูลจากประเภทหนึ่งไปยังอีกประเภทหนึ่ง เพื่อนำไปใช้ในสถานการณ์ต่างๆ ในการเขียนโปรแกรม
 - การแปลงข้อมูลจากตัวเลขเป็น String
 - แปลงข้อมูลจาก String เป็นตัวเลข
- Type Casting
- Type Wrapper Classes



Type Casting

- รูปแบบ

`newType variableName = (newType) valueToCovert`

- `newType` คือประเภทข้อมูลที่ต้องการแปลงไปเป็น โดยจะต้องเป็นประเภทข้อมูลแบบ primitive data type เท่านั้น
 - `char short int float double` ยกเว้น `boolean`
 - สามารถเปลี่ยนข้อมูลประเภทเดิมยังประเภทใหม่ โดยสามารถทำได้ทั้งตัวแปรและ literals;

Type Casting (Cont.)



```
public class TypeConversions {
    public static void main (String[] args) {
        float f = 5.0f;
        long newF = (long) f;
        System.out.println("Cast float to long = " + newF);

        System.out.println("Cast double to int = " + (int) 123.345);

        System.out.println("Cast int to float = " + (float) 100);

        float n1 = 10.12f;
        System.out.println("n1 is float = " + n1);
        System.out.println("Cast double to float = " + (double) n1);

        double n2 = 10.12;
        System.out.println("n2 is double = " + n1);
        System.out.println("Cast float to double = " + (float) n2);
    }
}
```



Type Wrapper Classes

- Java คลาสของ primitive data type นั้นจะมีทั้งหมด 8 คลาส

Simple Types

int
byte
short
long
float
double
char
boolean

Type Wrapper Classes

Integer
Byte
Short
Long
Float
Double
Character
Boolean

Type Wrapper Classes (Cont.)



- โดยในแต่ละคลาสจะมีเมธอดในการทำงานที่แตกต่างกันออกไป โดยส่วนมากแล้วเมธอดเหล่านี้จะใช้กับข้อมูลที่เป็น string เป็นส่วนใหญ่

```
public class PrimitiveTypeClass {
    public static void main (String[] args) {
        // Convert from string
        System.out.println("Covert string to int = " + Integer.parseInt("15"));
        System.out.println("Covert string to short = " + Short.parseShort("15"));
        System.out.println("Covert string to long = " + Long.parseLong("15"));

        System.out.println("Covert string to float = " + Float.parseFloat("11.54f"));
        System.out.println("Covert string to double = " + Double.parseDouble("11.54"));

        // Convert to string
        System.out.println("Covert int to string = " + Integer.toString(10));
        System.out.println("Covert int to base 16 = " + Integer.toHexString(10));
        System.out.println("Covert int to base 8 = " + Integer.toOctalString(10));
        System.out.println("Covert int to base 2 = " + Integer.toBinaryString(10));
    }
}
```