



Array and String in Java

Benjamas Panyangam
Computer Science, CMU

เนื้อหา

อาร์เรย์ (Array)

- อาร์เรย์ 1 มิติ
- อาร์เรย์หลายมิติ
- Ragged Array
- คลาส ArrayList

สายอักขระ(String)

ข้อมูลชนิดอาร์เรย์

- ❁ ตัวแปรประเภทอาร์เรย์ (Array)
 - กลุ่มข้อมูลประเภทเดียวกันหลาย ๆ ตัว
 - มีการระบุขนาดเพื่อจองเนื้อที่หน่วยความจำ

ข้อมูลคะแนนการเรียน 7 วิชาของนักเรียนคนหนึ่ง

หากใช้ตัวแปรเดี่ยว ต้องใช้ 7 ตัวแปร

คือ `int s1, s2, s3, s4, s5, s6, s7;`

	วิชาที่ 1	วิชาที่ 2	วิชาที่ 3	วิชาที่ 4	วิชาที่ 5	วิชาที่ 6	วิชาที่ 7
คะแนน	77	89	80	90	68	70	75
	s1	s2	s3	s4	s5	s6	s7

ข้อมูลชนิดอาร์เรย์

❁ ตัวแปรอาร์เรย์ จะใช้เพียง 1 ตัวแปร เก็บคะแนนทั้ง 7 วิชา

- ข้อมูลแต่ละตัวเรียกว่า element
- การอ้างถึงแต่ละ element จะใช้ index หรือ subscript ซึ่งเป็นตัวชี้ตำแหน่งข้อมูล โดยใช้เครื่องหมาย []

	วิชาที่ 1	วิชาที่ 2	วิชาที่ 3	วิชาที่ 4	วิชาที่ 5	วิชาที่ 6	วิชาที่ 7
คะแนน	77	89	80	90	68	70	75
	s[0]	s[1]	s[2]	s[3]	s[4]	s[5]	s[6]

ตัวแปรอาร์เรย์ 1 มิติ

- ❁ อาร์เรย์ของภาษาจาวาจะเก็บข้อมูลประเภทเดียวกันต่อเนื่องกันไป
- ❁ รูปแบบการประกาศตัวแปร มี 2 รูปแบบ

1) เครื่องหมาย [] อยู่หน้าชื่อตัวแปรอาร์เรย์
รูปแบบ *type* [] *var_name* ;
เช่น *int* [] *s* ;

2) เครื่องหมาย [] อยู่หลังชื่อตัวแปรอาร์เรย์
รูปแบบ *type* *var_name* [] ;
เช่น *int* *s* [] ;

ตัวแปรอาร์เรย์ 1 มิติ

- ❁ หลังการประกาศตัวแปร เมื่อจะใช้งานตัวแปรอาร์เรย์ ต้องจองเนื้อที่หน่วยความจำด้วยคำสั่ง **new**

รูปแบบ

```
type [ ] var_name;
```

```
var_name = new type[size];
```

โดยที่ **size** เป็นจำนวนสมาชิกของอาร์เรย์ที่ต้องการประกาศ

ตัวอย่าง

```
int [ ] s;
```

```
s = new int [7];
```

ตัวแปรอาร์เรย์ 1 มิติ

- ❁ สามารถจองหน่วยความจำพร้อมกับการประกาศตัวแปรได้

รูปแบบ

```
type var_name = new type[size];
```

ตัวอย่าง

```
int[ ] s = new int[7];
```

ตัวแปรอาร์เรย์ 1 มิติ

❁ หลังใช้คำสั่ง **new** จะมีการกำหนดค่าเริ่มต้นให้โดยอัตโนมัติ ดังนี้

➤ กรณีอาร์เรย์เก็บข้อมูลประเภทตัวเลข

ค่าเริ่มต้นจะเป็น 0

เช่น `int [] x = new int[3];`

int

0
0
0

➤ กรณีอาร์เรย์เก็บข้อมูลตรรกะ (boolean) ค่าเริ่มต้นจะเป็น false

การอ้างอิงสมาชิกในตัวแปรอาร์เรย์

❁ ตัวอย่างการอ้างอิงสมาชิกของตัวแปรอาร์เรย์

```
x[0] = 77;           //นำค่า 77 ใส่ในเซลล์ช่องที่ 0
x[6] = 75;           //นำค่า 75 ใส่ในเซลล์ช่องที่ 6
```

```
int [ ] x;
x = new int[7];
```

```
x[4] = x[0] + x[1]; //ช่องที่ 4 เก็บผลบวกช่องที่ 0 และ 1
```

```
int i = 2;
x[i+2] = 90;        //ใช้ index ในรูปนิพจน์ได้
```

```
//รับข้อมูลจากแป้นพิมพ์เก็บในอาร์เรย์ช่องที่ i (ในที่นี้ i = 2)
x[i] = scanVar.nextInt();
```

77
0
0
167
90
0
75

```
for (int k =0; k < 7; k++) //แสดงข้อมูลในตัวแปรอาร์เรย์ช่องที่ 0 ถึง 6
    System.out.println(x[k]);
```

```
x[-1] = 100; หรือ x[7] = 99; //ทำไม่ได้ เพราะ อ้างอิง index ได้
                          //แค่ 0 ถึง 6
```

การประกาศตัวแปรอาร์เรย์ พร้อมกับกำหนดค่าเริ่มต้น

รูปแบบ

type [] var_name = { value1,value2,value3,...};

ตัวอย่าง

int [] score = {77, 80, 90, 68,75};

77	score[0]
80	score[1]
90	score[2]
68	score[3]
75	score[4]

String [] color = { "red", "green", "blue"};

ตัวอย่างที่ 1

โปรแกรมสำหรับรับค่าเลข 5 จำนวน ที่มีค่าอยู่ระหว่าง 0-100 เพื่อหาค่าสูงสุด ค่าต่ำสุด และพิมพ์ทุกคะแนนที่สูงกว่าค่าเฉลี่ย

```
import java.util.Scanner;
class maxminaverage {
    public static void main(String[ ] args) {
        Scanner keyboard = new Scanner(System.in);
        double [ ] score = new double [ 5 ];
        double min =100, max=0,sum = 0,avg;
        for (int k = 0; k < 5; k++){
            System.out.print("score [ "+ (k+1) + " ] = " );
            score[ k ] = keyboard.nextDouble();
            if (score [k] < min) min = score[k];
            if (score [k] > max) max = score[k];
            sum = sum + score[k];
        }// end for
```

ตัวอย่าง INPUT

```
score [ 1 ] = 10
score [ 2 ] = 20
score [ 3 ] = 5
score [ 4 ] = 30
score [ 5 ] = 40
```

```
avg = sum / 5;
System.out.println("Average score is " + avg);
System.out.print("Score(s) which are more than average:
");
for (int k = 0; k < 5; k++)
    if (score[k] > avg)
        System.out.print(score [ k] + " ");
System.out.println();
System.out.println("Maximum score is " + max);
System.out.println("Minimum score is " + min);
} // end main
} // end class
```

ผลการทำงาน

```
Average score is 21.0
Score(s) which are more than average : 30.0
40.0
Maximum score is 40.0
Minimum score is 5.0
```

การอ้างถึงข้อมูลในอาร์เรย์โดยไม่ใช้ index ผ่านคำสั่ง for

รูปแบบ **for** (*type var_name : array_name*)
statement(s);

```
char [ ] grade = {'A', 'D', 'A', 'B'};  
//แสดงค่าทีละ element  
for(char value : grade)  
    System.out.println(value);  
//แก้ไขค่าในแต่ละ element  
for(int k =0; k < grade.length; k++)  
    grade[k] = (char)((int) grade[k] + 10);  
//แสดงค่าทีละ element อีกครั้ง  
for(char value : grade)  
    System.out.println(value);  
//แสดงทั้งอาร์เรย์  
System.out.println(grade);
```

ผลการทำงาน

A
D
A
B
K
N
K
L
KNKL

ตัวแปรอาร์เรย์หลายตัว

- ❁ ใช้เครื่องหมาย , คั่นเช่นเดียวกับการประกาศตัวแปรเดียว
`int [] A,B; //ประกาศตัวแปร A และ B เป็นตัวแปรอาร์เรย์`
`A = new int [10];`
`B = new int [100];`

หรือ

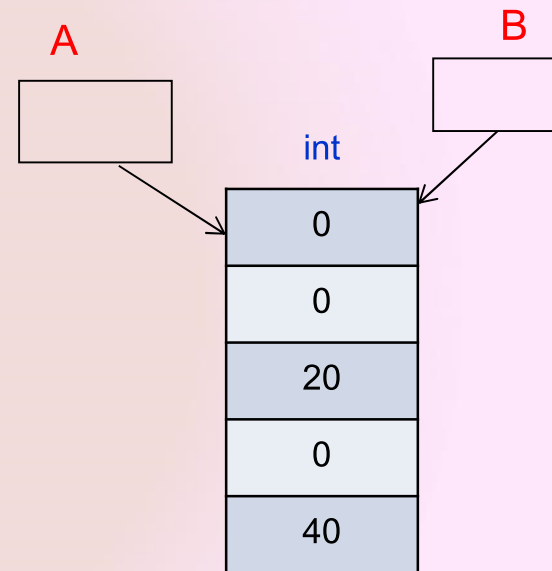
`int [] A = new int [10] , B = new int [100];`

- ❁ **ระวัง!** กรณีประกาศโดยใช้เครื่องหมาย [] หลังชื่อตัวแปร
`int A[], B;` ตัวแปร A เป็นอาร์เรย์
ตัวแปร B ไม่ได้เป็นอาร์เรย์ เป็นแค่ int
กรณีที่ต้องการให้เป็นอาร์เรย์ทั้งคู่
อาจใช้ `int a[], b[];`

ตัวแปรอาร์เรย์หลายตัว

❁ ประกาศตัวแปรอาร์เรย์ 2 ตัวชี้ไปยังตำแหน่งหน่วยความจำเดียวกัน

```
int [ ] A, B;  
A = new int [5];  
B = A;  
A[2] = 20;  
B[4] = 40;
```



สามารถใช้งานหน่วยความจำ 5 ช่องนี้ ผ่านตัวแปรได้ทั้ง 2 ตัว

ตัวอย่างที่ 2

ผลการทำงาน

```
class multiArray{  
    public static void main(String[ ] args) {  
        int [ ] x = { 10,20, 30, 40 };  
        int [ ] y = x; int k;  
        for (k=0; k < 4; k++)  
            System.out.print(y[ k] + " ");  
  
        for (k = 0; k < 4; k++)  
            x[k] = x[k] + 10;  
  
        for (k=0; k < 4; k++)  
            System.out.print(x[ k] + " ");  
  
        for (k=0; k < 4; k++)  
            System.out.print(y[ k] + " ");
```

10 20 30 40

20 30 40 50

20 30 40 50


```
int [ ] a = { 1,2, 3, 4 };  
y = a;  
for (k=0; k < 4; k++) {  
    x[k] = y[k] * k;  
    System.out.print(x[ k] + " "); }  

```

0 2 6 12

```
for (k=0; k < 4; k++)  
    System.out.print(y[ k] + " ");  

```

1 2 3 4

```
for (k=0; k < 4; k++)  
    System.out.print(a[ k] + " ");  
} // end main  
} // end class
```

1 2 3 4

ตัวแปรอาร์เรย์ 2 มิติ

- ❁ เก็บข้อมูลประเภทเดียวกันหลาย ๆ ตัว
- ❁ แต่มองข้อมูลในลักษณะตารางที่ประกอบด้วยแถวและคอลัมน์

เช่น ตารางเก็บคะแนน 4 วิชาของนักเรียน 3 คน โดยแต่ละแถวเป็นคะแนนของแต่ละคน และคอลัมน์จะเป็นคะแนนสอบแต่ละวิชา

	วิชาที่ 1	วิชาที่ 2	วิชาที่ 3	วิชาที่ 4
คะแนนนักเรียนคนที่ 1	77	89	80	90
คะแนนนักเรียนคนที่ 2	90	70	60	80
คะแนนนักเรียนคนที่ 3	80	67	82	70

❏ ประกาศตัวแปรอาร์เรย์ 2 มิติชื่อ **score** เก็บค่าคะแนนของนักเรียน 3 คน 4 วิชา `int [][] score = new int[3][4];`₁₈

ตัวแปรอาร์เรย์ 2 มิติ



รูปแบบการประกาศและจองหน่วยความจำ

```
type [ ][ ] array_name; หรือ type array_name [ ][ ];  
array_name = new type [row_size][column_size];
```

```
type [ ][ ]array_name = new type [row_size][column_size];  
เช่น double [ ][ ] rain = new double[3][4];  
//เก็บเลขจำนวนจริงทั้งหมด 12 ค่า
```

โดยที่

type

ประเภทข้อมูลที่ต้องการจัดเก็บในอาร์เรย์

array_name

ชื่อตัวแปรอาร์เรย์

row_size

จำนวนแถวของสมาชิกในอาร์เรย์

column_size

จำนวนคอลัมน์ของสมาชิกในอาร์เรย์

ตัวแปรอาร์เรย์ 2 มิติ : การอ้างสมาชิก

```
int [ ][ ] score = new int[3][4];
```

❁ ต้องใช้ index 2 ค่า ดังตัวอย่าง

```
// กำหนดคะแนนวิชาที่ 3 ของนักเรียนคนที่ 1 เป็น 80  
score[0][2] = 80;
```

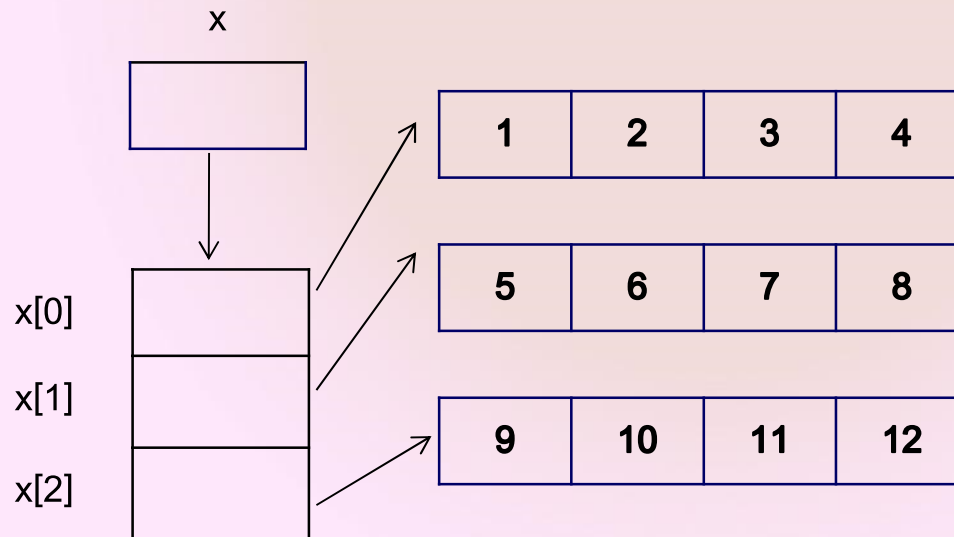
```
// กำหนดคะแนนวิชาที่ 2 ของนักเรียนคนที่ 2 เป็น 70  
score[1][1] = 70;
```

	วิชาที่ 1	วิชาที่ 2	วิชาที่ 3	วิชาที่ 4
คะแนนนักเรียนคนที่ 1	0	0	80	0
คะแนนนักเรียนคนที่ 2	0	70	0	0
คะแนนนักเรียนคนที่ 3	0	0	0	0

ตัวแปรอาร์เรย์ 2 มิติแบบกำหนดค่าเริ่มต้น

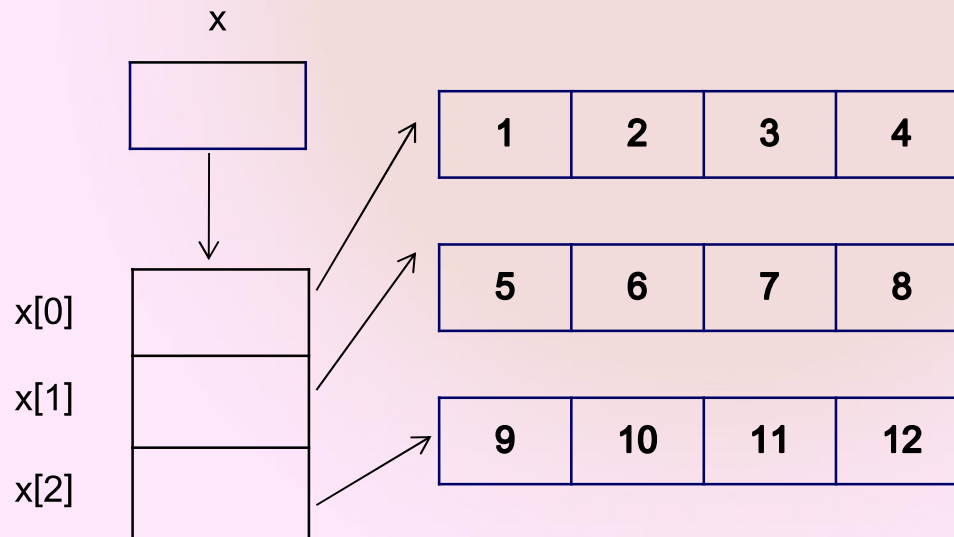
❁ ประกาศตัวแปร x อาร์เรย์ขนาด 3 แถว 4 column
`int [][] x = {{1,2,3,4}, {5,6,7,8}, {9,10,11,12}};`

❏ ตัวแปร x จะมีลักษณะการจองหน่วยความจำดังรูป



ตัวแปรอาร์เรย์ 2 มิติ : length method

- ❁ **x.length** จะได้จำนวนแถวซึ่งมีค่าเท่ากับ 3
- ❁ แต่ถ้ากำหนด **index 1** มิติ จะได้จำนวนคอลัมน์ในแต่ละแถว เช่น **x[0].length** จะได้ค่าเท่ากับ 4 เช่นเดียวกับ **x[1].length** และ **x[2].length**



ตัวอย่างที่3

ผลการทำงาน

1--> red
2 -->green
3 -->blue

```
class showcolor{  
    public static void main(String[ ] args) {  
        String [ ] color = { "red", "green", "blue"};  
  
        for (int k =0; k < color.length; k++)  
            System.out.println((k+1) + " --> " +color[k]);  
    }//end main  
}//end class
```

Ragged Arrays

❁ อาร์เรย์ 2 มิติที่มีจำนวนคอลัมน์ในแต่ละแถวไม่เท่ากัน
เช่น `int [][] R = new int[3][];` // ยังไม่ระบุจำนวนคอลัมน์

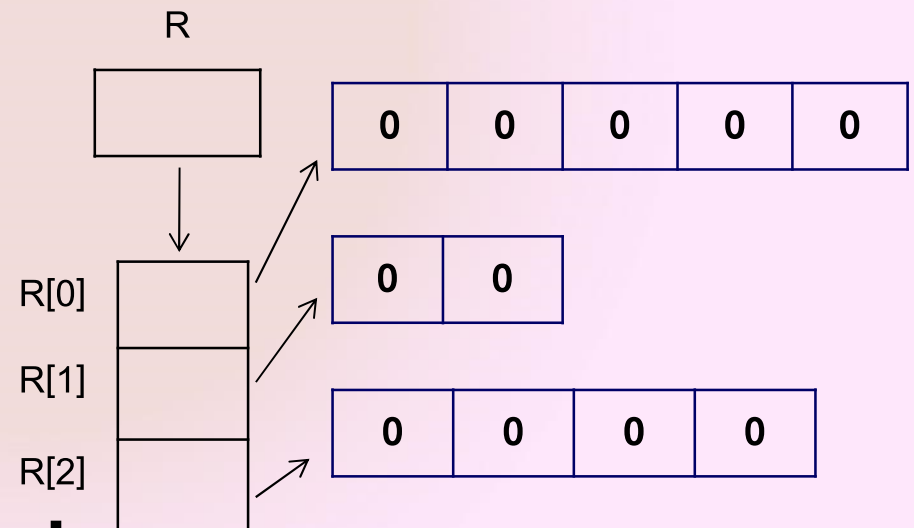
❁ หากกำหนดจำนวนคอลัมน์ในแต่ละแถว ดังนี้

`R[0] = new int [5];`

`R[1] = new int [2];`

`R[2] = new int [4];`

❌ ตัวแปร R จะมีลักษณะ
การจองหน่วยความจำดังรูป



ตัวอย่างที่4

```
int [][]x = {{1,2},{3,4,5},{6,7}};  
System.out.println(x[0][0]);  
System.out.println(x[1][2]);  
System.out.println(x[2][1]);
```

ผลการทำงาน

1
5
7

คลาส ArrayList



ตัวแปรอาร์เรย์

- ก่อนใช้งานต้องจองเนื้อที่หน่วยความจำ → รู้ขนาด
- หลังใช้คำสั่ง `new` ขนาดของอาร์เรย์จะไม่สามารถเปลี่ยนแปลงได้



ตัวแปรที่สร้างจากคลาส ArrayList

- ใช้งานตัวแปรได้โดยไม่ต้องกำหนดขนาดก่อน
- ต้อง `import` คลาสนี้ในโปรแกรมด้วยคำสั่ง
`import java.util.ArrayList;`

ตัวอย่าง Methods ภายในคลาส ArrayList

- ❁ **add(int index, Object obj)**
นำ object ไปเก็บในอาร์เรย์ในตำแหน่ง index ที่ระบุ
- ❁ **remove(int index)**
นำ object ที่ตำแหน่ง index ออกจากอาร์เรย์
- ❁ **indexOf(Object obj)**
คืนตำแหน่ง index แรกที่พบค่า object ที่ค้นหา
- ❁ **lastIndexOf(Object obj)**
คืนตำแหน่ง index สุดท้ายที่พบค่า object ที่ค้นหา
- ❁ **isEmpty()** ตรวจสอบว่า object ว่างหรือไม่
- ❁ **clear()** ล้างข้อมูลทั้งหมดใน ArrayList
- ❁ **size()** หาขนาดของอาร์เรย์
- ❁ **get(int index)** คืนค่า object ในตำแหน่ง

คลาส ArrayList : ตัวอย่างคำสั่ง

```
ArrayList L = new ArrayList(); //ประกาศอาร์เรย์ L (ขนาด 0 ช่อง)
L.add("Benjamas");             //เพิ่มสมาชิกในอาร์เรย์ L (3 ช่อง)
L.add("Siriporn");
L.add("Krich");
System.out.println(L);         //แสดงสมาชิกทุกตัวในตัวแปร L
                                //คือ [Benjamas, Siriporn, Krich]
int a = L.size();               //a เก็บค่า 3 (ขนาดของอาร์เรย์)
L.remove(1);                    //นำคำว่า Siriporn ออกจากอาร์เรย์

L.add(100);                     //เพิ่มสมาชิกในอาร์เรย์ L (1 ช่อง)
int b = L.indexOf("Krich");     //หาตำแหน่งของ Krich ในอาร์เรย์
                                // ซึ่งคือค่า 1
```

Benjamas	Krich	100
----------	-------	-----

คลาส ArrayList : ตัวอย่างคำสั่ง

System.out.println(b);	//พิมพ์ค่า b คือ 1
System.out.println(L);	//แสดงสมาชิกทุกตัวในตัวแปร L
	//คือ [Benjamas, Krich, 100]
String s = (String) L.get(b);	//นำค่าในตำแหน่งที่ 1 (Krich) เก็บใน s
int x = (int) L.get(2);	//นำค่าในตำแหน่งที่ 2 (100) เก็บใน x
L.remove(b);	//นำคำว่า Krich ออกจากอาร์เรย์
System.out.println(x);	//พิมพ์ค่า x คือ 100
System.out.println(L);	//แสดงสมาชิกทุกตัวในตัวแปร L
	//คือ [Benjamas, 100]

สายอักขระหรือสตริง (String)

❁ ต้องสร้างออบเจกต์จาก **คลาส String** เช่น

➤ **String myFriend;**

myFriend = new String("Matinee");

➤ **String myname = new String("Benjamas");**

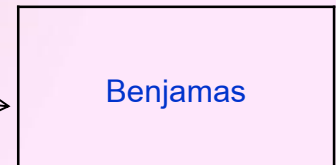
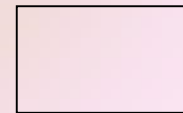
❁ หรือกำหนดแบบย่อได้ ดังตัวอย่าง

➤ **String myFriend;**

myFriend = "Matinee";

➤ **String myname = "Benjamas";**

myname



การสร้างตัวแปรประเภทสตริง

❁ มี **Constructor Method** ในการสร้างตัวแปรสตริงได้หลาย

ลักษณะ

- **String()** เช่น **String S;**
- **String(String value)** เช่น **String S = new String("ABC");**
- **String(char[] charArrayName)**
- **String(char[] charArrayName,int offset, int count)**
- **String(byte[] byteArrayName);**
- **String (byte[] byteArrayName,int offset, int count)**
- ตัวอย่างเช่น

```
byte [] byteArray= {97,99,101,103};  
String s1 = new String(byteArray); //s1 เก็บ "aceg"
```

```
char [] charArray={'a','b','z','y','x'};  
String s2 = new String(charArray,1,3); // s เก็บ "bzy"
```

Methods พื้นฐานเกี่ยวกับสตริง

- ❁ **char charAt(int pos)** ส่งค่ากลับเป็นอักขระในตำแหน่งที่กำหนด
- ❁ **int length()** ส่งค่ากลับเป็นความยาวของสตริง
- ❁ **String trim()** ส่งค่ากลับเป็นสตริงที่ตัดช่องว่างด้านหน้าและหลังออก
- ❁ **String toUpperCase(), toLowerCase()**
ส่งค่ากลับเป็นข้อความสตริงที่เป็นตัวพิมพ์ใหญ่/ตัวพิมพ์เล็กทั้งหมด

```
String a = " Hello ".toLowerCase();  
System.out.println(a);  
a = a.trim();  
String b = a.toUpperCase();  
System.out.println(b+a);
```

ผลการทำงาน

```
hello  
HELLOhello
```


Methods เกี่ยวกับการค้นหาสตริง

- ❁ **int indexOf(String s)** ค้นหาตำแหน่งที่ปรากฏสตริง s โดยพบเป็นตำแหน่งแรกในสตริง
- ❁ **int indexOf(String s, int start)** ค้นหาตำแหน่งที่ปรากฏสตริง s โดยพบเป็นตำแหน่งแรก แต่ให้เริ่มต้นค้นหาตั้งแต่ตำแหน่ง start เป็นต้นไป
- ❁ **int indexOf(char c)** ค้นหาตำแหน่งที่ปรากฏอักขระ c เป็นตำแหน่งแรกในสตริง
- ❁ **int indexOf(char c, int start)** ค้นหาตำแหน่งที่ปรากฏอักขระ c เป็นตำแหน่งแรกในสตริง แต่ให้เริ่มต้นค้นหาตั้งแต่ตำแหน่ง start เป็นต้นไป

เช่น **String s1 = "Thailand is", s2 = "is";**

a = s1.indexOf(s2); // a มีค่าเป็น 9

a = s1.indexOf('a'); // a มีค่าเป็น 2

a = s1.indexOf('i',4) // a มีค่าเป็น 9

Methods เกี่ยวกับการค้นหาสตริง

- ❁ **int lastIndexOf(String s)** ค้นหาตำแหน่งที่ปรากฏสตริง s โดยพบเป็นตำแหน่งสุดท้ายในสตริง
- ❁ **int lastIndexOf(String s, int final)** ค้นหาตำแหน่งที่ปรากฏสตริง s โดยพบเป็นตำแหน่งสุดท้าย แต่ให้ค้นหาถึงตำแหน่ง final
- ❁ **int lastIndexOf(char c)** ค้นหาตำแหน่งที่ปรากฏอักขระ c เป็นตำแหน่งสุดท้ายในสตริง
- ❁ **int lastIndexOf(char c, int final)** ค้นหาตำแหน่งที่ปรากฏอักขระ c เป็นตำแหน่งสุดท้ายในสตริง แต่ให้ค้นหาถึงตำแหน่ง final

เช่น **String s1 = "Thailand is a", s2 = "is";**

a = s1.lastIndexOf("Land"); // a มีค่าเป็น -1 กรณีหาไม่พบ

a = s1.lastIndexOf('a'); // a มีค่าเป็น 12

a = s1.lastIndexOf('a',10); // a มีค่าเป็น 5

Methods เกี่ยวกับการค้นหาสตริง



startsWith(String s)

คืนค่าจริง (true) ถ้าสตริงเริ่มต้นด้วยสตริง s



endsWith(String s)

คืนค่าจริง (true) ถ้าส่วนท้ายของสตริงเป็นค่าสตริง s



replace(char oldChar, newChar)

แทนที่อักขระ oldChar ในสตริงด้วยอักขระ newChar

เช่น `String s1 = "Thailand is", s2 = "is";`

`a = s1.endsWith(s2); // a มีค่าเป็น true`

`s1 = s1.replace('a', 'A'); // s1 มีค่าเป็น "ThAilAnd is",`

Methods เกี่ยวกับการคืนค่ามาเป็นสตริง

❁ **String concat(String s)**

นำสตริง s ไปเชื่อมต่อกับสตริงที่กำหนด(ที่เรียกใช้ method นี้)

เช่น **String s1 = "Thailand";**

s1= s1.concat(" is"); //s1 มีค่าเป็น "Thailand is"

❁ **String valueOf(int x) , String valueOf(long x)**

String valueOf(double x) , String valueOf(float x)

แปลงค่าเลขจำนวนเต็มหรือเลขจำนวนจริงในตัวแปร x ให้เป็นสตริง

เช่น **s1=String.valueOf(10);**

❁ **String.valueOf(boolean b)**

แปลงค่าของ b ให้เป็นสตริง "true" หรือ "false "

เช่น **s1=String.valueOf(isMan);**

Methods เกี่ยวกับการตัดสตริง

❁ **String substring(int start)**

คืนค่าข้อความในสตริง โดยเริ่มต้นตั้งแต่ตำแหน่ง **start**

❁ **String substring(int start, int end)**

คืนค่าข้อความในสตริง โดยเริ่มต้นตั้งแต่ตำแหน่ง **start** ไปจนถึงตำแหน่ง **end-1**

เช่น **String s1 = "Thailand is a";**

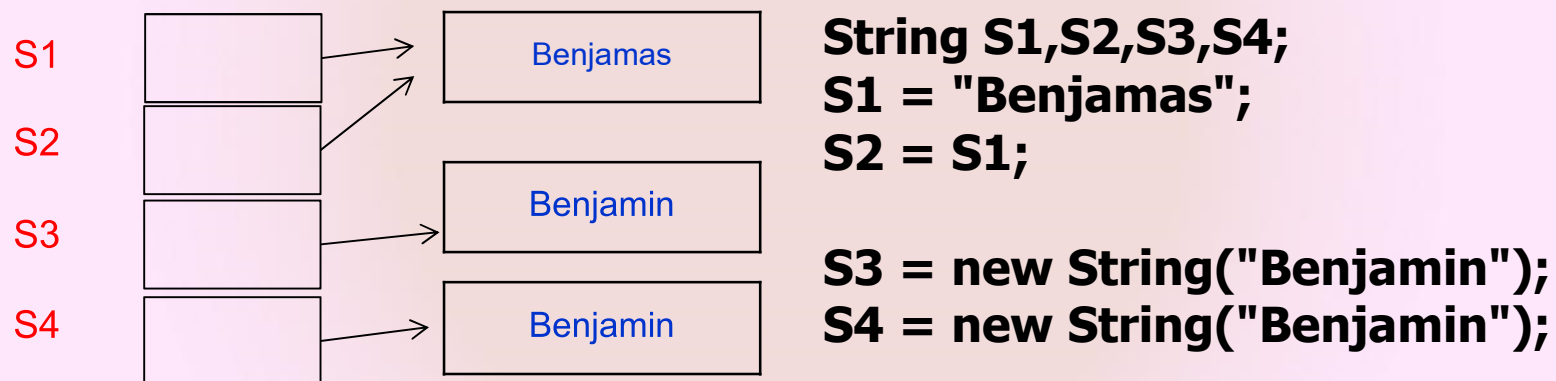
System.out.println(s1.substring(4)); //แสดง "land is a"

System.out.println(s1.substring(4,7)); //แสดง "lan"

Methods เกี่ยวกับการเปรียบเทียบสตริง

❁ เครื่องหมาย **==** ที่จะคืนค่าจริง ถ้าสตริงทั้งสองชี้ไปยังค่าข้อมูล เดียวกัน address เดียวกัน

❁ **equals**(String s) จะคืน true ถ้าสตริง s และสตริงที่เรียกใช้ method นี้ชี้ไปยังข้อความเดียวกัน



```
System.out.println(S1==S2);           //true
System.out.println(S1.equals(S2));     //true
System.out.println(S3==S4);           //false
System.out.println(S3.equals(S4));     //true
```

ตัวอย่างที่ 5

```
String S1,S2,S3,S4;  
S1 = "Benjamas"; S2 = S1;  
System.out.println(S1==S2);           //true  
System.out.println(S1.equals(S2));    //true
```

```
S2="Matinee";  
System.out.println(S1==S2);           //false  
System.out.println(S1.equals(S2));    //false
```

```
S2="Benjamas";  
System.out.println(S1==S2);           //true  
System.out.println(S1.equals(S2));    //true
```

ตัวอย่างที่ 5 (ต่อ)

```
S3="Benjamas";
```

```
System.out.println(S1==S3);      //true
```

```
System.out.println(S1.equals(S3)); //true
```

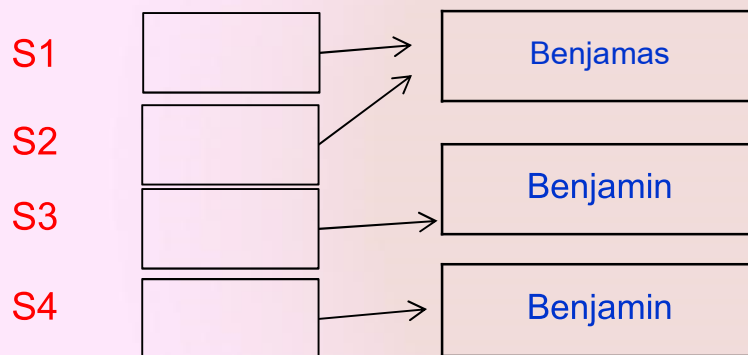
```
S4= new String("Benjamas");
```

```
System.out.println(S1==S4);      //false
```

```
System.out.println(S1.equals(S4)); //true
```


Methods เกี่ยวกับเปรียบเทียบสตริง

❁ **compareTo(String s)** หน้าที่เหมือนคำสั่ง equals แต่คืนค่าเป็นความต่างของรหัส ASCII ของอักขระตำแหน่งแรกที่แตกต่างกันจากทั้งสองสตริง



```
String S1,S2,S3,S4;  
S1 = "Benjamas";  
S2 = S1;  
S3 = new String("Benjamin");  
S4 = new String("Benjamin");
```

```
System.out.println(S1.compareTo(S2));      0  
System.out.println(S3.compareTo(S4));      0  
System.out.println(S1.compareTo(S3));     -8  
System.out.println(S3.compareTo(S1));      8  
(ความแตกต่างระหว่างค่า ASCII ของอักขระ i และ a)
```