

บทที่ 1 ความรู้เบื้องต้นของภาษาจาวา

บรรยายโดย ผศ.ดร.ธราวิเชษฐ์ ธิติจรูญโรจน์

คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

หัวข้อ

- การรับค่าผ่านทางคีย์บอร์ดและแสดงผล
- ตัวแปร (Variable)
- ตัวดำเนินการ (Operation)

หัวข้อ



- การรับค่าผ่านทางคีย์บอร์ดและแสดงผล
- ตัวแปร (Variable)
- ตัวดำเนินการ (Operation)

การแสดงผลทางจอภาพ



- การแสดงผลผ่านทางจอภาพ

- `System.out.print(...)` ;

การพิมพ์ข้อความภายในเครื่องหมายคำพูด “...” ออกทางจอภาพ

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

การพิมพ์ข้อความภายในเครื่องหมายคำพูด “...” ออกทางจอภาพ จากนั้นจะ

ขึ้นบรรทัดใหม่

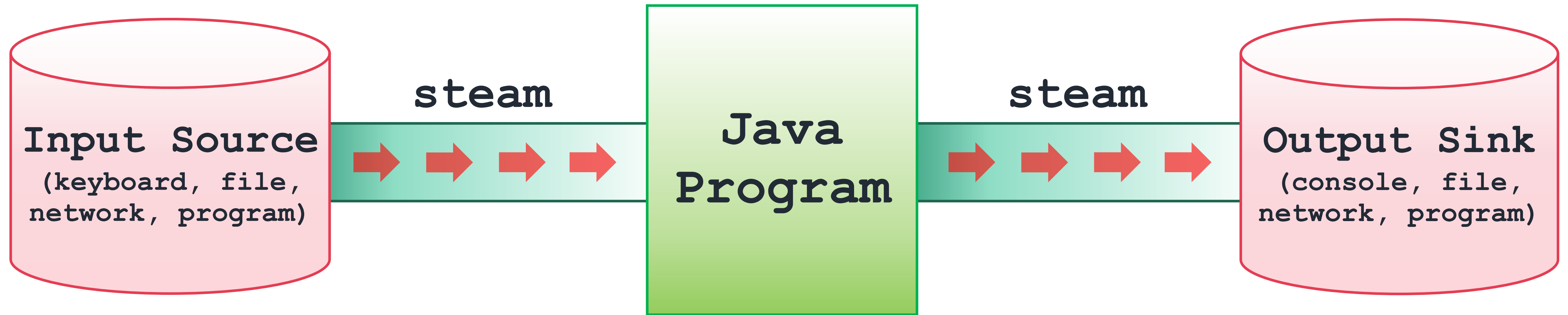
การแสดงผลทางจอภาพ



```
public class MyCode01 {  
    public static void main(String[] args) {  
        System.out.print("line 1");  
        System.out.print("line 2");  
        System.out.println("line 3");  
        System.out.println("line 4");  
    }  
}
```

```
public class MyCode02 {  
    public static void main(String[] args) {  
        System.out.println("line 1");  
        System.out.print("line 2");  
        System.out.println("line 3");  
        System.out.print("line 4");  
    }  
}
```

โครงสร้างช่องทางการสื่อสาร



ประกอบด้วย 3 ส่วนหลัก ได้แก่ ต้นทาง (input source) ช่องทางการแลกเปลี่ยนข้อมูล (stream) และปลายทาง (output sink)

การรับค่าผ่านทางคีย์บอร์ด

- การรับค่าผ่านทางคีย์บอร์ด

นำเข้า Package (ชุดคำสั่งที่จำเป็น)

- `import java.util.*;` หรือ
- `import java.util.Scanner;`

การสร้าง Object (ต่อเชื่อม)

- `Scanner sc = new Scanner(System.in);`

โดยที่ **sc** คือ object ที่ทำหน้าที่เป็นตัวกลางในการรับค่าจากคีย์บอร์ดมาสู่โปรแกรม

การรับค่าผ่านทางคีย์บอร์ด

- การรับค่าผ่านทางคีย์บอร์ด

กรณีต้องการรับค่าเป็น String

- `String str = sc.nextLine();`

กรณีต้องการรับค่าเป็น Integer

- `int num = sc.nextInt();`

กรณีต้องการรับค่าเป็น Float

- `float num = sc.nextFloat();`

กรณีต้องการรับค่าเป็น Double

- `double num = sc.nextDouble();`

การรับค่าผ่านทางคีย์บอร์ด

```
import java.util.*;
public class MyCode03 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        String str = sc.nextLine();
    }
}
```

```
import java.util.*;
public class MyCode03 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int num1 = sc.nextInt();
        double num2 = sc.nextDouble();
    }
}
```

การรับค่าผ่านทางคีย์บอร์ด

```
import java.util.*;
public class MyCode05 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int num = sc.nextInt();
        String str = sc.nextLine();
    }
}
```



```
import java.util.*;
public class MyCode05 {
    public static void main(String[] args) {
        Scanner sc1 = new Scanner(System.in);
        String str = sc1.nextLine();
        int num = sc1.nextInt();
    }
}
```



```
import java.util.*;
public class MyCode05 {
    public static void main(String[] args) {
        Scanner sc1 = new Scanner(System.in);
        Scanner sc2 = new Scanner(System.in);
        int num = sc1.nextInt();
        String str = sc2.nextLine();
    }
}
```



หัวข้อ

- การรับค่าผ่านทางคีย์บอร์ดและแสดงผล
- ตัวแปร (Variable)
- ตัวดำเนินการ (Operation)

ตัวแปร



ตัวแปร คือ สิ่งที่ใช้สำหรับการจัดเก็บข้อมูลหรือสารสนเทศ เพื่อนำมา (1) จัดการหรืออ้างอิงการใช้งานในภายหลัง นอกจากนี้ ยังเป็นการ (2) ติดป้ายบ่งบอกว่าสิ่งที่เก็บคืออะไร หรือ (3) เป็นการให้ความหมายของสิ่งที่จัดเก็บ

ตัวแปร

age

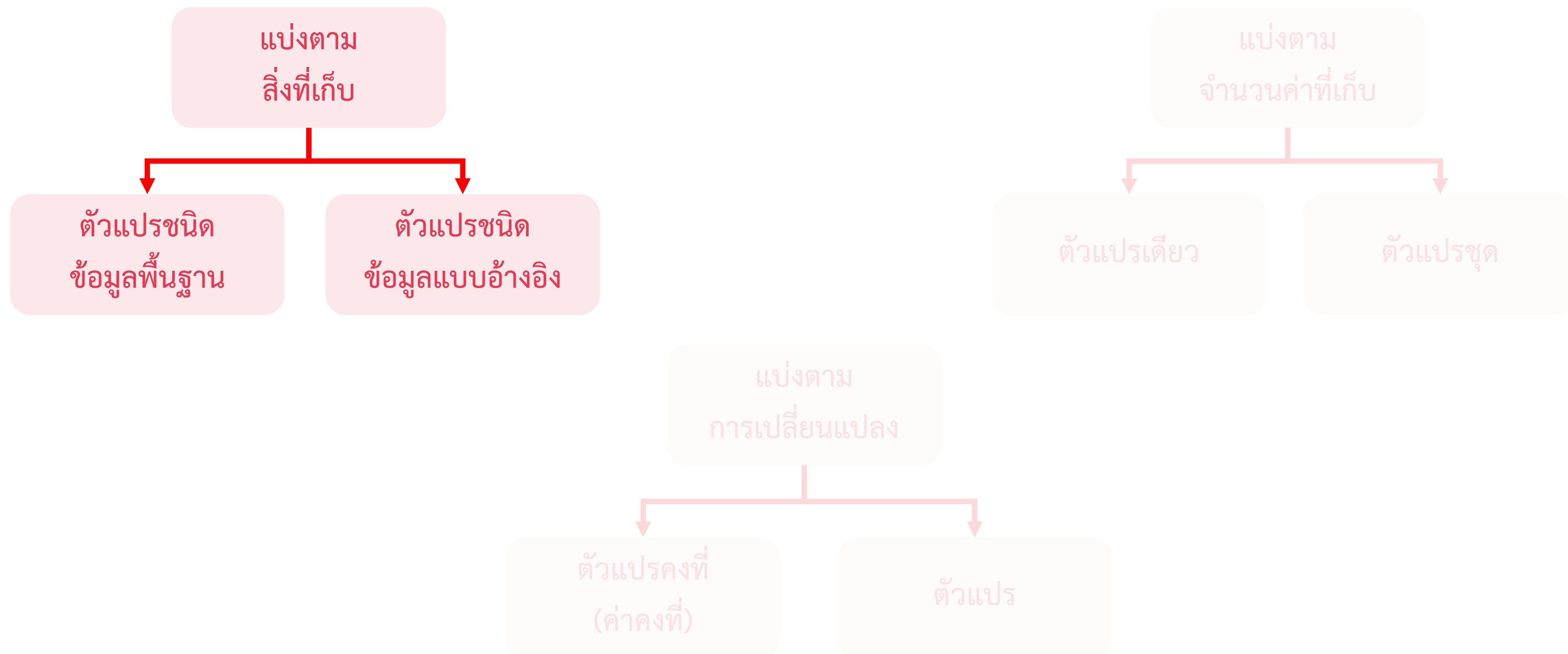
57

ตัวแปร คือ สิ่งที่ใช้สำหรับการจัดเก็บข้อมูลหรือสารสนเทศ เพื่อนำมา (1) จัดการหรืออ้างอิงการใช้งานในภายหลัง นอกจากนี้ ยังเป็นการ (2) ติดป้ายบ่งบอกว่าสิ่งที่เก็บคืออะไร หรือ (3) เป็นการให้ความหมายของสิ่งที่จัดเก็บ

ตัวแปร

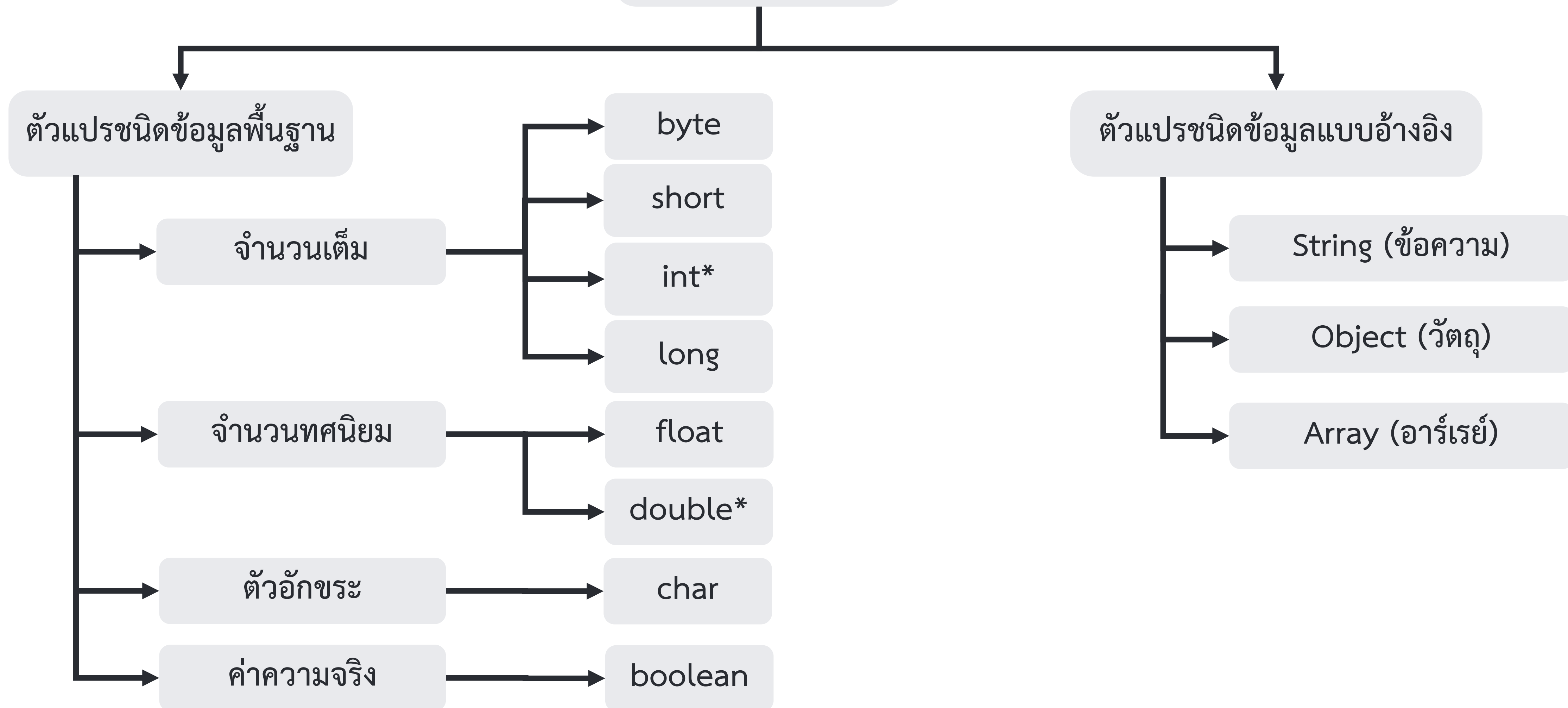


ตัวแปร



ตัวแปร

แบ่งตามสิ่งที่เก็บ



ขนาดและช่วงค่าของชนิดข้อมูลแบบพื้นฐาน



ชนิดข้อมูล	ขนาด (Bit/Byte)	ช่วงค่า	ค่าเริ่มต้น
boolean	1 / Not Precisely define	true หรือ false	false
char	16 / 2	16 bit Unicode character '\u0000' ถึง '\uFFFF'	\u0000
byte	8 / 1	-128 ถึง +127	0
short	16 / 2	-32,768 ถึง +32,767	0
int	32 / 4	-2^{31} ถึง $+2^{31}-1$	0
long	64 / 8	-2^{63} ถึง $+2^{63}-1$	0L
float	32 / 4	IEEE 754 floating point standard -3.40E+38 ถึง +3.40E+38	0.0f
double	64 / 8	IEEE 754 floating point standard -1.80E+308 ถึง +1.80E+308	0.0d หรือ 0.0

ข้อมูลชนิดตัวเลขจำนวนเต็ม



ข้อมูลค่าคงที่ของเลขจำนวนเต็มที่เป็นชนิด long **อาจจะ**มีตัวอักษร L หรือ LL ต่อท้าย**หรือไม่ก็ได้** อาทิเช่น

21

หมายถึง เลข**ฐานสิบ**ที่มีค่าเป็น 2 ซึ่งเป็นข้อมูลชนิด long

077L

หมายถึง เลข**ฐานแปด**ที่เป็นข้อมูลชนิด long

0xBAACL

หมายถึง เลข**ฐานสิบหก**ที่เป็นข้อมูลชนิด long

ข้อมูลชนิดตัวเลขทศนิยม



ข้อมูลชนิดตัวเลขทศนิยมที่มีชนิดข้อมูลเป็น float จะมีตัวอักษร F หรือ f ต่อท้าย อาทิเช่น

2.718F หรือ 3.14f

ข้อมูลชนิดตัวเลขทศนิยมที่มีชนิดข้อมูลเป็น double จะมีตัวอักษร D หรือ d ต่อท้าย อาทิเช่น

2.718D

(โดยทั่วไปไม่จำเป็นต้องใส่ตัวอักษร D เพราะข้อมูลชนิดตัวเลขทศนิยมจะกำหนดให้เป็น double อยู่แล้ว)

ข้อมูลชนิดตรรกะ

ข้อมูลชนิดตรรกะมีข้อมูลค่าคงที่อยู่ที่ 2 ค่า คือ

`true` และ `false`

ตัวอย่าง

```
boolean flag = true;
```

เป็นการกำหนดตัวแปรที่ชื่อว่า `flag` ให้มีชนิดข้อมูลเป็น `boolean` โดยกำหนดให้มีค่าเป็น `true`

ข้อมูลชนิดตัวอักษร

ข้อมูลชนิดตัวอักษรใช้เพื่อแสดงตัวอักษรหนึ่งตัว ซึ่งถูกเก็บอยู่ในรูปของมาตรฐาน Unicode ขนาด 16 บิต โดยจะมีค่าตั้งแต่ ‘\u0000’ ถึง ‘\uFFFF’ ตัวอย่างเช่น

```
char letter = '\u0041';
```

เป็นการประกาศตัวแปรที่ชื่อว่า letter ให้เป็นข้อมูลชนิด char โดยมีค่าเป็น \u0041 ซึ่งมีค่าเท่ากับตัวอักษร A หรืออาจจะเขียนเป็นตัวอักษรโดยตรงเลย เช่น

```
char letter = 'A';
```

เป็นการประกาศตัวแปรที่ชื่อว่า letter ให้เป็นข้อมูลชนิด char โดยมีค่าเป็นตัวอักษร A เช่นเดียวกับคำสั่งก่อนหน้านี้

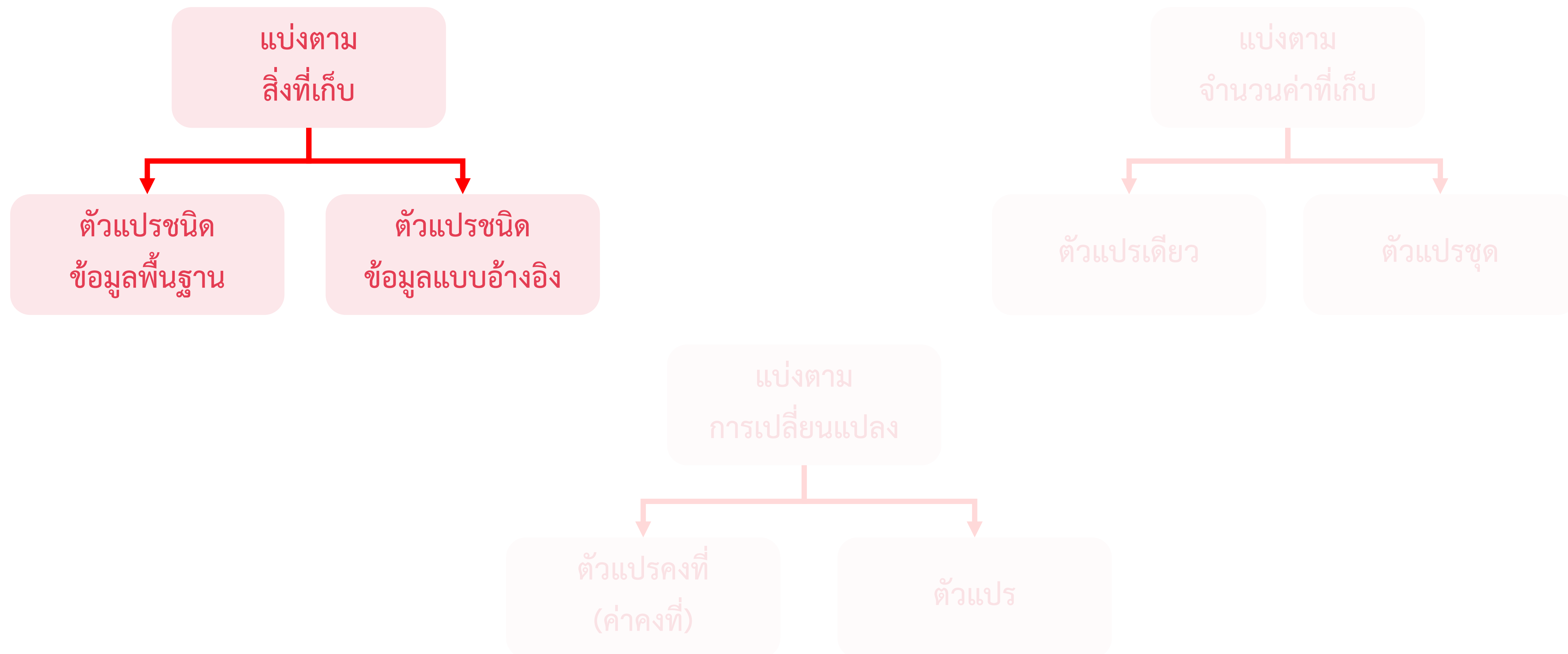
Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
0	00	Null	32	20	Space	64	40	@	96	60	`
1	01	Start of heading	33	21	!	65	41	A	97	61	a
2	02	Start of text	34	22	"	66	42	B	98	62	b
3	03	End of text	35	23	#	67	43	C	99	63	c
4	04	End of transmit	36	24	\$	68	44	D	100	64	d
5	05	Enquiry	37	25	%	69	45	E	101	65	e
6	06	Acknowledge	38	26	&	70	46	F	102	66	f
7	07	Audible bell	39	27	'	71	47	G	103	67	g
8	08	Backspace	40	28	(72	48	H	104	68	h
9	09	Horizontal tab	41	29)	73	49	I	105	69	i
10	0A	Line feed	42	2A	*	74	4A	J	106	6A	j
11	0B	Vertical tab	43	2B	+	75	4B	K	107	6B	k
12	0C	Form feed	44	2C	,	76	4C	L	108	6C	l
13	0D	Carriage return	45	2D	-	77	4D	M	109	6D	m
14	0E	Shift out	46	2E	.	78	4E	N	110	6E	n
15	0F	Shift in	47	2F	/	79	4F	O	111	6F	o
16	10	Data link escape	48	30	0	80	50	P	112	70	p
17	11	Device control 1	49	31	1	81	51	Q	113	71	q
18	12	Device control 2	50	32	2	82	52	R	114	72	r
19	13	Device control 3	51	33	3	83	53	S	115	73	s
20	14	Device control 4	52	34	4	84	54	T	116	74	t
21	15	Neg. acknowledge	53	35	5	85	55	U	117	75	u
22	16	Synchronous idle	54	36	6	86	56	V	118	76	v
23	17	End trans. block	55	37	7	87	57	W	119	77	w
24	18	Cancel	56	38	8	88	58	X	120	78	x
25	19	End of medium	57	39	9	89	59	Y	121	79	y
26	1A	Substitution	58	3A	:	90	5A	Z	122	7A	z
27	1B	Escape	59	3B	;	91	5B	[123	7B	{
28	1C	File separator	60	3C	<	92	5C	\	124	7C	
29	1D	Group separator	61	3D	=	93	5D]	125	7D	}
30	1E	Record separator	62	3E	>	94	5E	^	126	7E	~
31	1F	Unit separator	63	3F	?	95	5F	_	127	7F	□

ข้อมูลชนิดตัวอักขระ



อักขระ	Unicode	ความหมาย
' \b '	' \u000B '	Backspace
' \t '	' \u0009 '	Tab
' \n '	' \u000A '	New line
' \r '	' \u000D '	Return
' \\ '	' \u005C '	Backslash
' \' '	' \u0027 '	Single quote
' \" '	' \u0022 '	Double quote

ตัวแปร

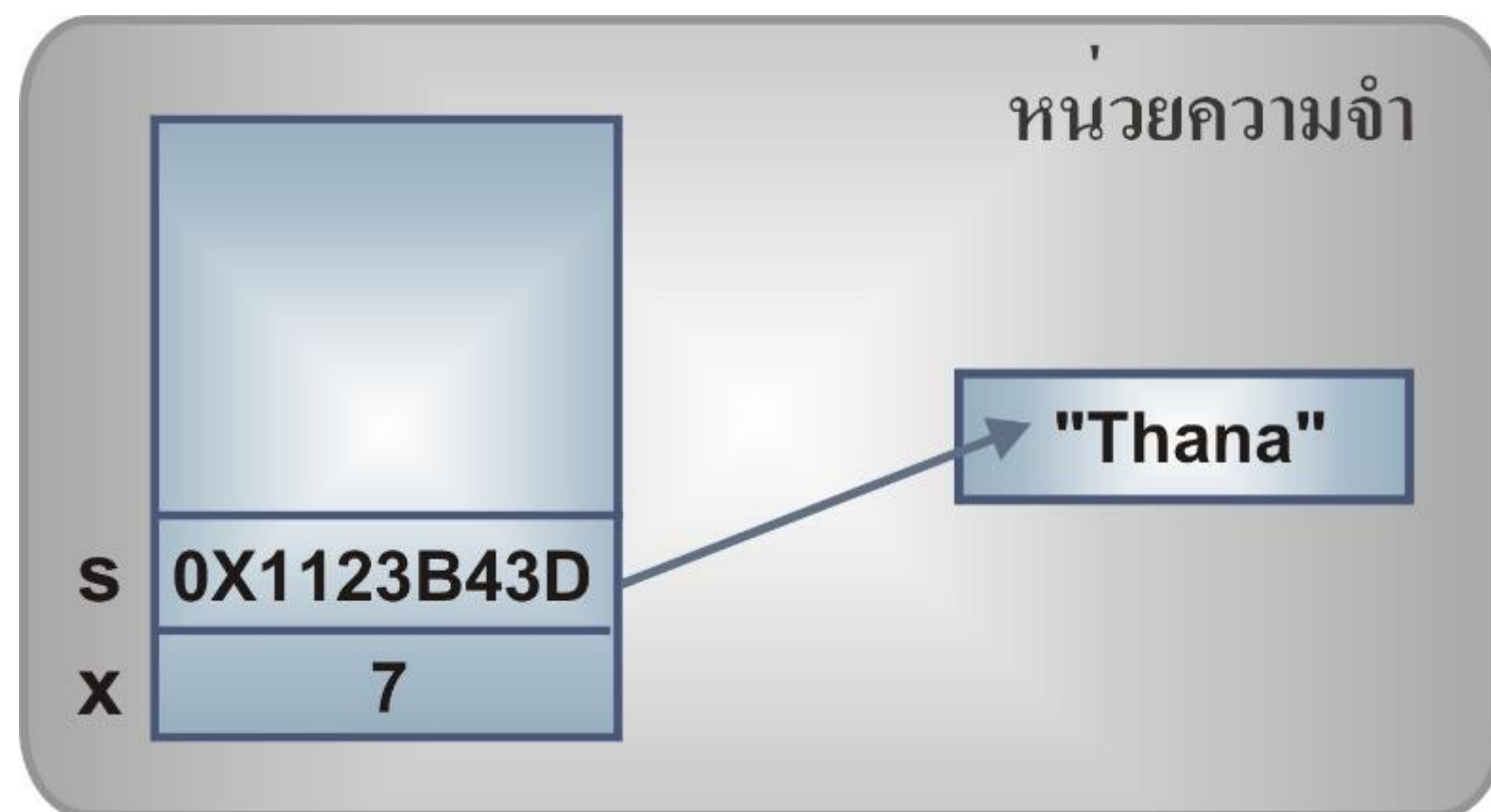


ชนิดข้อมูลแบบอ้างอิง

ชนิดข้อมูลแบบอ้างอิง คือ ออปเจ็คในภาษาจาวา โดยแบ่งออกเป็นสองแบบคือ

- ชนิดข้อมูลที่เป็นคลาส
- ชนิดข้อมูลที่เป็นอະเรย์

ชนิดข้อมูลแบบอ้างอิงจะมีวิธีการเก็บข้อมูลในหน่วยความจำที่แตกต่างจากการเก็บข้อมูลของชนิดข้อมูลแบบพื้นฐานอาทิเช่น คำสั่ง



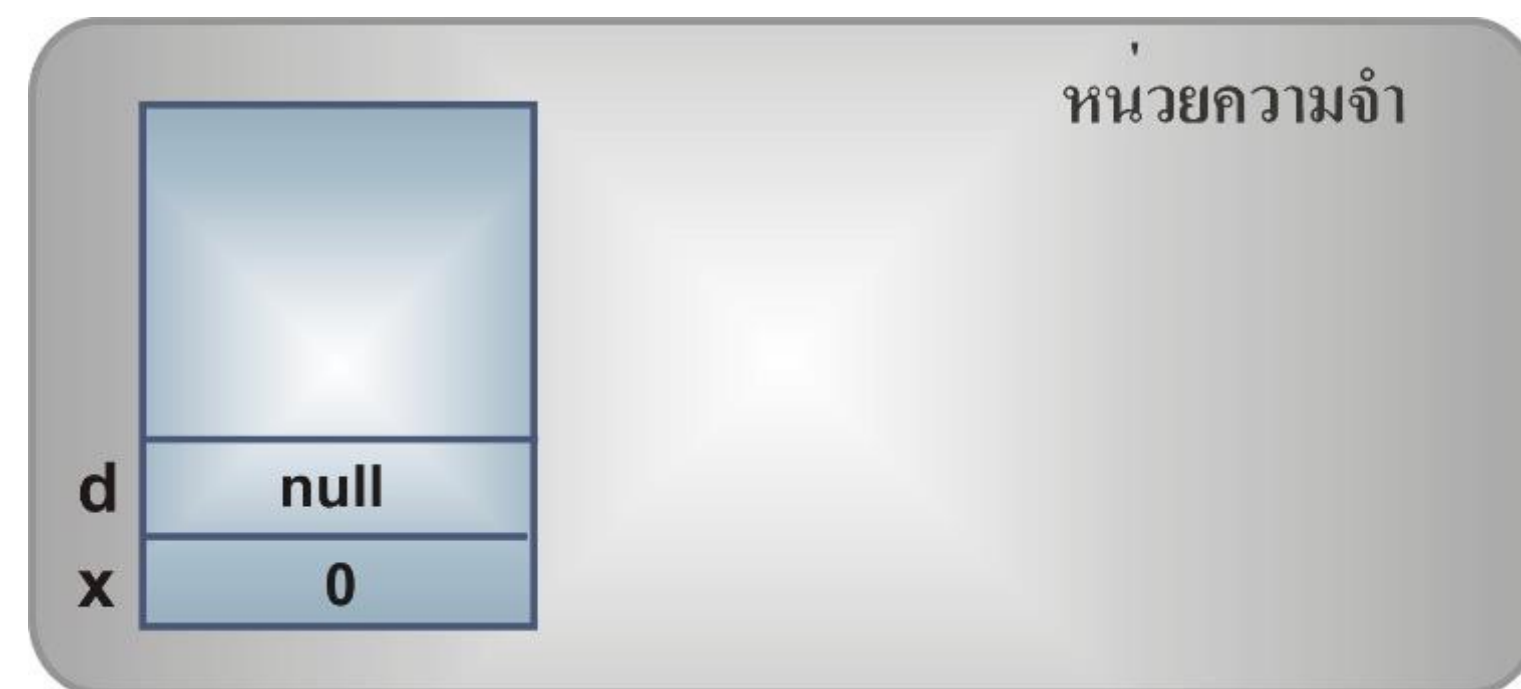
```
int x = 7;  
String s = new String("Thana");
```

ชนิดข้อมูลแบบอ้างอิง

อาทิเช่นคำสั่ง

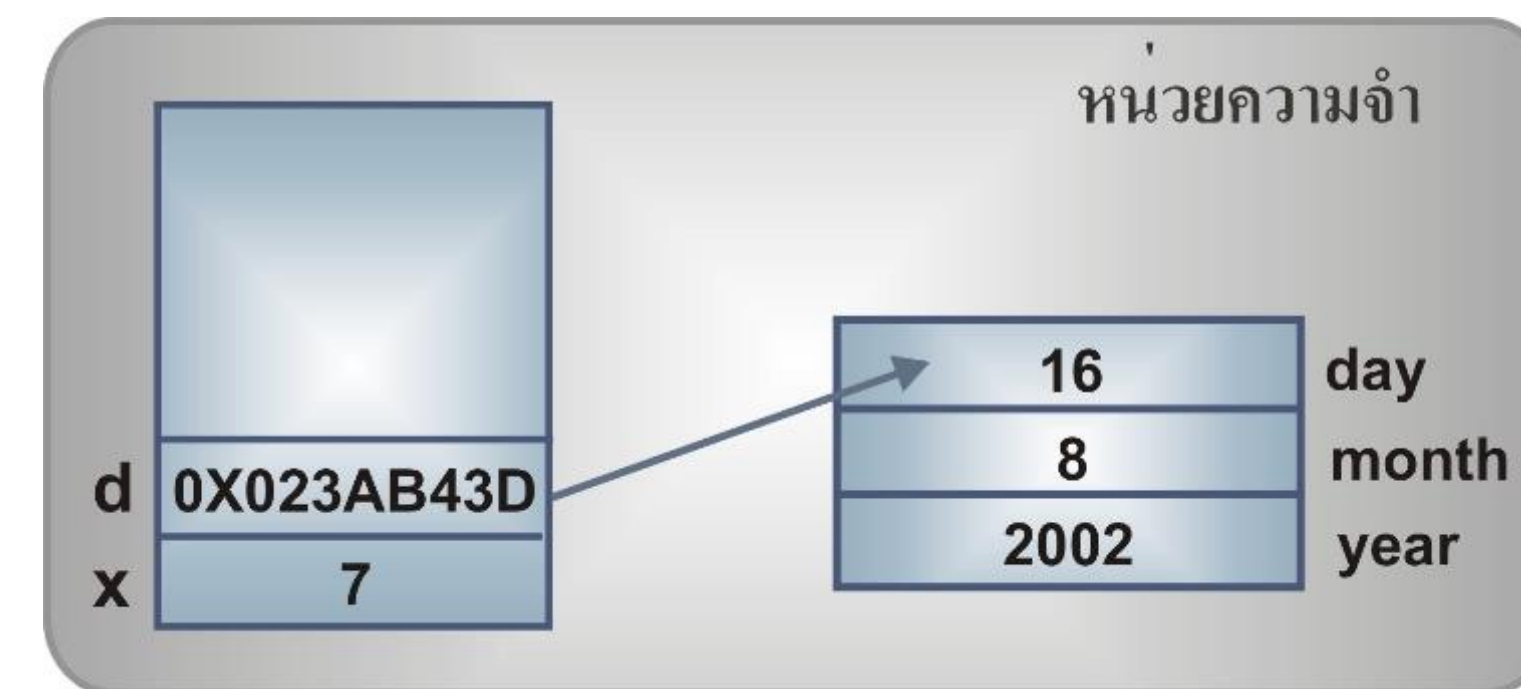
```
int x;
```

```
Date d;
```



```
x = 7 ;
```

```
d = new Date (16, 8, 2002) ;
```



คลาส String



String เป็นคลาสที่กำหนดไว้ใน Java ตัวแปรที่มีชนิดข้อมูลเป็น String ก็คือออปเจ็คชนิดหนึ่ง ซึ่ง String มีข้อแตกต่างจากออปเจ็คทั่ว ๆ ไปดังนี้

- String เป็นออปเจ็คที่มีค่าคงที่ข้อมูลซึ่งก็คือข้อความใดๆ ที่อยู่ภายในเครื่องหมาย double quote (“ ”) ตัวอย่างเช่น

“This is a java course”

- String เป็นออปเจ็คที่สามารถสร้างหรือกำหนดค่าได้โดยไม่ต้องใช้คำสั่ง new ตัวอย่างเช่น เราสามารถใช้คำสั่ง

```
String s = "Thana";
```

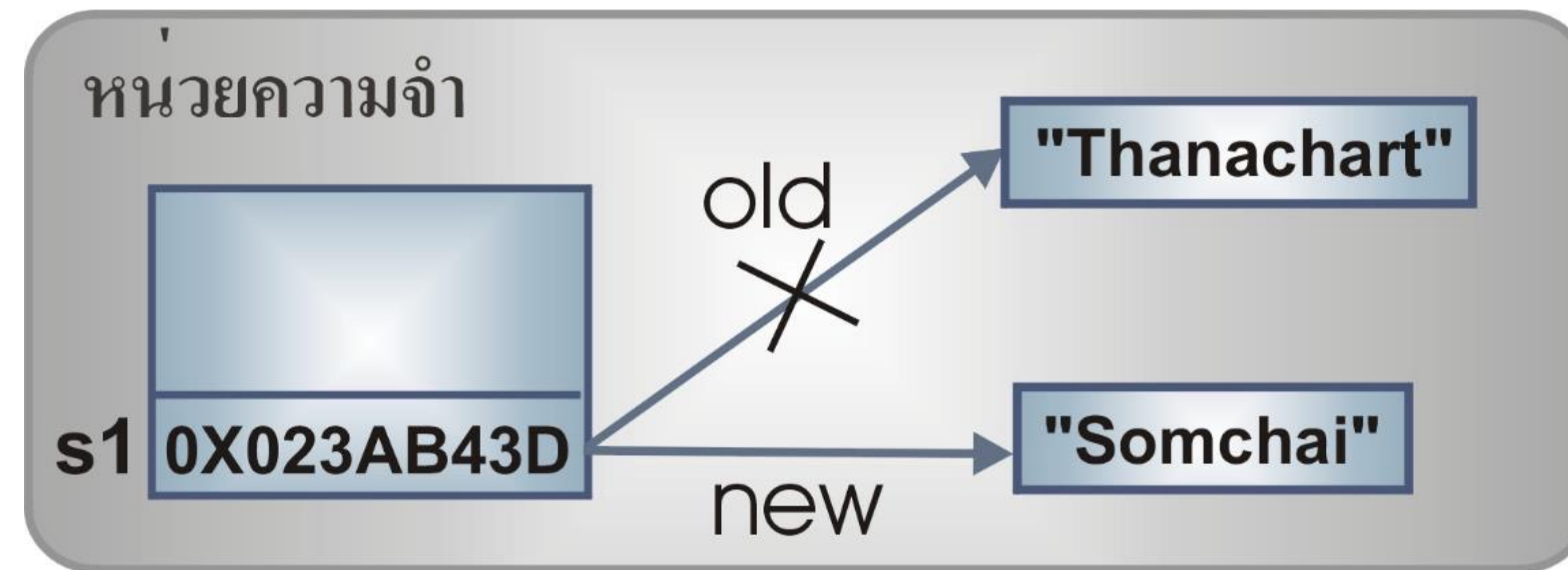
- โดยไม่จำเป็นที่จะต้องใช้คำสั่ง

```
String s = new String("Thana");
```

ตัวอย่างโปรแกรม

```
public class StringDemo {  
    public static void main(Stirng args[]){  
        String s1 = "Thana";  
        String s2 = "Thana";  
        String s3 = new String("Thana");  
    }  
}
```

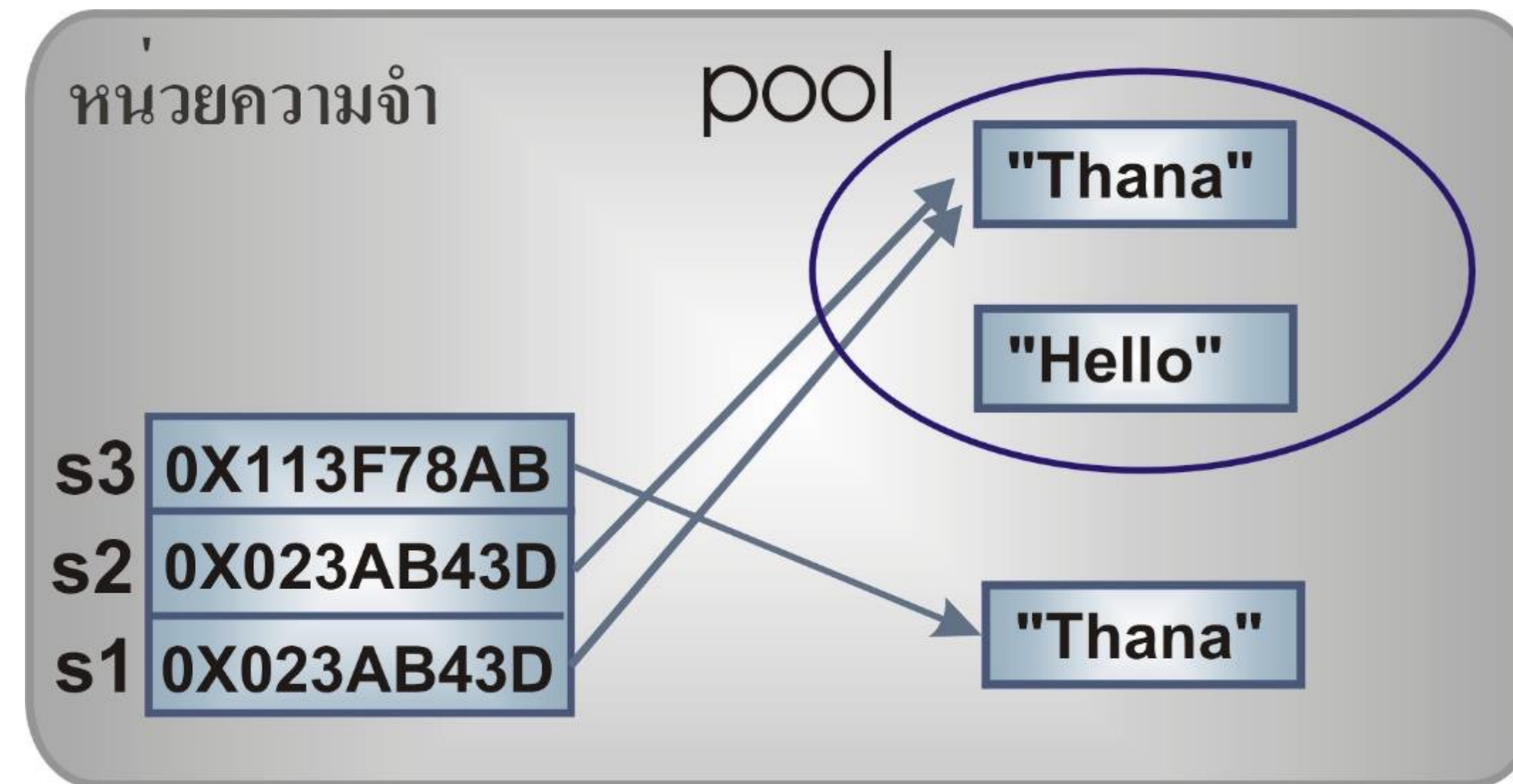
คำอธิบายโปรแกรม



ตัวอย่างเช่น คำสั่ง

```
String s1;  
s1 = "Thanachart";  
s1 = "Somchai";
```

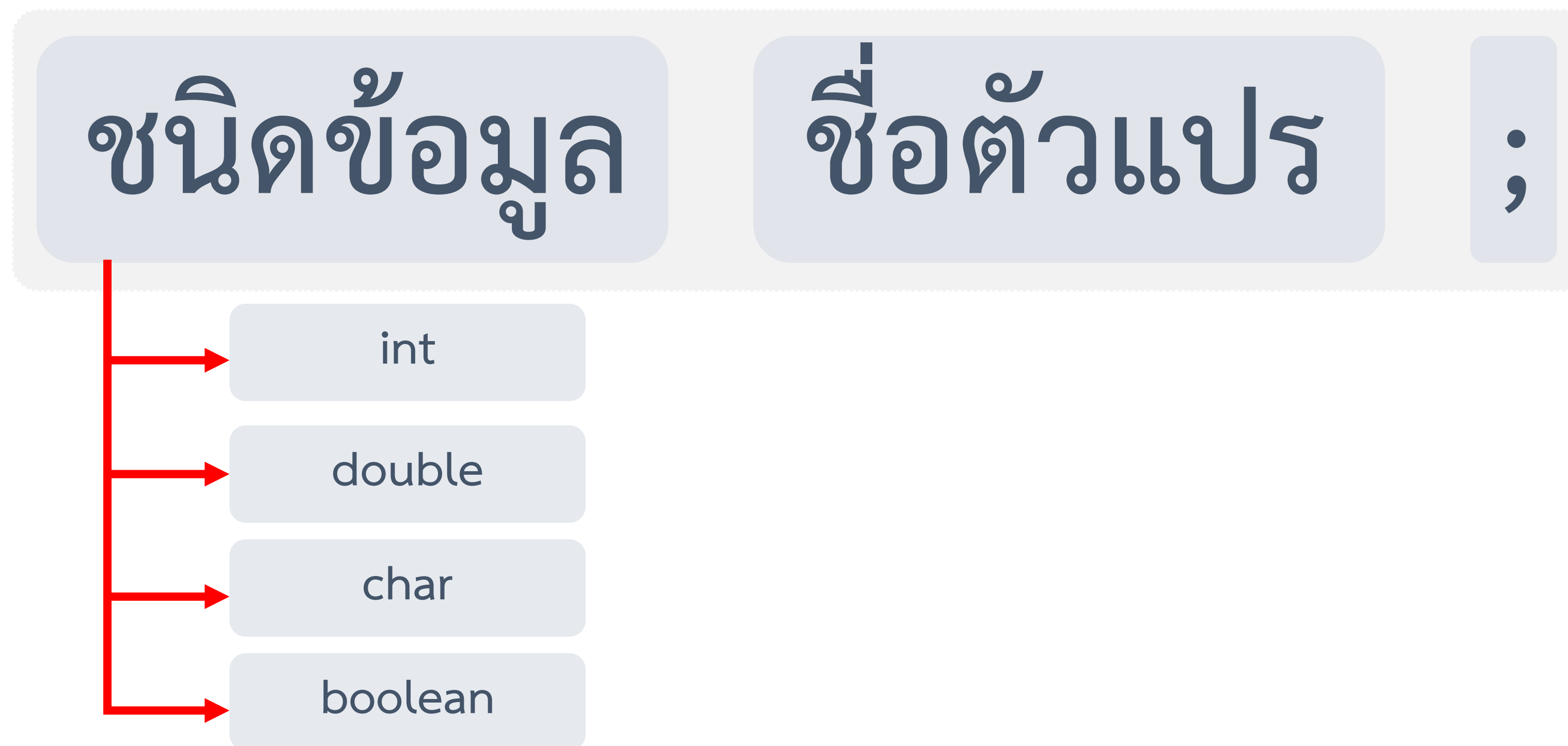
คำอธิบายโปรแกรม



- ▶ กรณีที่ ไม่ใช้คำสั่ง new ภาษายาวจะพิจารณาจาก String Pool ว่ามีข้อความเดิมอยู่หรือไม่
- ▶ กรณีที่ ใช้คำสั่ง new ภาษายาวจะสร้างข้อความใหม่และจองเนื้อที่ในหน่วยความจำเสมอ

ตัวแปร

- วิธีการประกาศตัวแปร



หลักการตั้งชื่อที่ปฏิบัติกันโดยทั่วไป

จะต้องเป็นไปตามกฎการตั้งชื่อนี้

- ต้องขึ้นต้นด้วยอักขระ A-Z, a-z, _ หรือ \$ เท่านั้น
- กรณีที่ประกอบไปด้วยตัวอักขระมากกว่าหนึ่งตัว ตัวอักขระหลังจากตัวแรกนั้นจะต้องเป็นตัวอักขระข้างต้น หรือเป็นตัวเลข 0 ถึง 9 เท่านั้น
- ต้องไม่ตรงกับคีย์เวิร์ด
- ในภาษาจาวาเป็น **case sensitive** ทำให้ myVariable แตกต่างจาก MyVariable

คีย์เวิร์ดที่ใช้ในภาษาจาวา

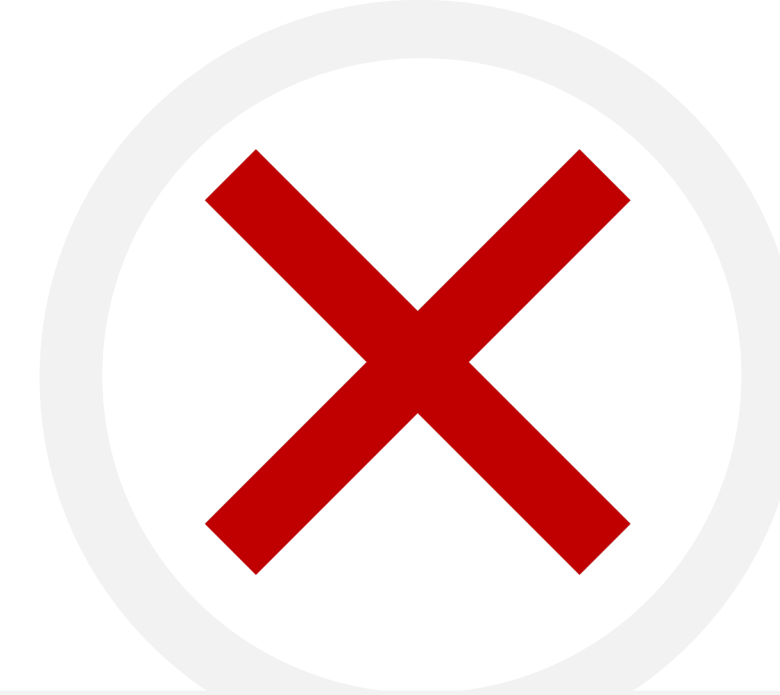


<code>abstract</code>	<code>continue</code>	<code>for</code>	<code>new</code>	<code>switch</code>
<code>assert</code>	<code>default</code>	<code>goto</code>	<code>package</code>	<code>synchronized</code>
<code>boolean</code>	<code>do</code>	<code>if</code>	<code>private</code>	<code>this</code>
<code>break</code>	<code>double</code>	<code>implements</code>	<code>protected</code>	<code>throw</code>
<code>byte</code>	<code>else</code>	<code>import</code>	<code>public</code>	<code>throws</code>
<code>case</code>	<code>enum</code>	<code>instanceof</code>	<code>return</code>	<code>transient</code>
<code>catch</code>	<code>extends</code>	<code>int</code>	<code>short</code>	<code>try</code>
<code>char</code>	<code>final</code>	<code>interface</code>	<code>static</code>	<code>void</code>
<code>class</code>	<code>finally</code>	<code>long</code>	<code>strictfp</code>	<code>volatile</code>
<code>const</code>	<code>float</code>	<code>native</code>	<code>super</code>	<code>while</code>

ตัวอย่างของการตั้งชื่อในภาษาจาวา



- `MyVariable`
- `_MyVariable`
- `$x`
- `This_is_also_a_variable`
- `isMoney`
- `stu2`



- `My Variable`
- `9pns`
- `a+c`
- `Hello 'World`
- `public`

หลักการตั้งชื่อที่ปฏิบัติกันโดยทั่วไป



ประเภท	หลักการ	ตัวอย่างเช่น
Class	<ul style="list-style-type: none">จะขึ้นต้นด้วยตัวอักษรพิมพ์ใหญ่แล้วตามด้วยตัวอักษรพิมพ์เล็กหรือตัวเลข โดยจะใช้ตัวอักษรพิมพ์ใหญ่เฉพาะอักษรนำของแต่ละคำที่ตามมาในชื่อควรเป็นคำนาม	Sample, HelloWorld, MyBigBoss, Student, GraduateStudent เป็นต้น
Variable or Attribute	<ul style="list-style-type: none">จะขึ้นต้นด้วยตัวอักษรพิมพ์เล็ก โดยจะใช้ตัวอักษรพิมพ์ใหญ่เฉพาะอักษรนำของแต่ละคำที่ตามมาในชื่อควรเป็นคำนามหรือเป็นชื่อสั้น ๆ	x, i, name, id, gpa, firstName, myFirstName เป็นต้น
Method	<ul style="list-style-type: none">จะใช้หลักการเดียวกับการตั้งชื่อ Variable or Attributeควรเป็นคำกริยา	getName, setName, showDetails, getMyFirstName เป็นต้น
Constant	<ul style="list-style-type: none">จะใช้ตัวอักษรพิมพ์ใหญ่ทั้งหมด และจะแยกคำโดยใช้เครื่องหมาย _ (underscore)ควรเป็นคำนาม	ID, MIN_GPA เป็นต้น

ตัวแปร

- วิธีการประกาศตัวแปร



ตัวแปร



- วิธีการประกาศตัวแปร

ชนิดข้อมูล

ชื่อตัวแปร

;

- ตัวอย่างเช่น

```
int age ;  
double weight ;  
char grade ;
```

ตัวแปร

- วิธีการกำหนดค่าตัวแปร

ชื่อตัวแปร = ค่า ;

- ตัวอย่างเช่น

```
age = 10 ;  
weight = 120.5 ;  
grade = 'B' ;
```

ตัวแปร

- วิธีการประกาศและกำหนดค่าตัวแปร

ชนิดข้อมูล ชื่อตัวแปร = ค่า ;

- ตัวอย่างเช่น

```
int age = 10 ;  
double weight = 120.5 ;  
char grade = 'B' ;
```

ตัวแปร



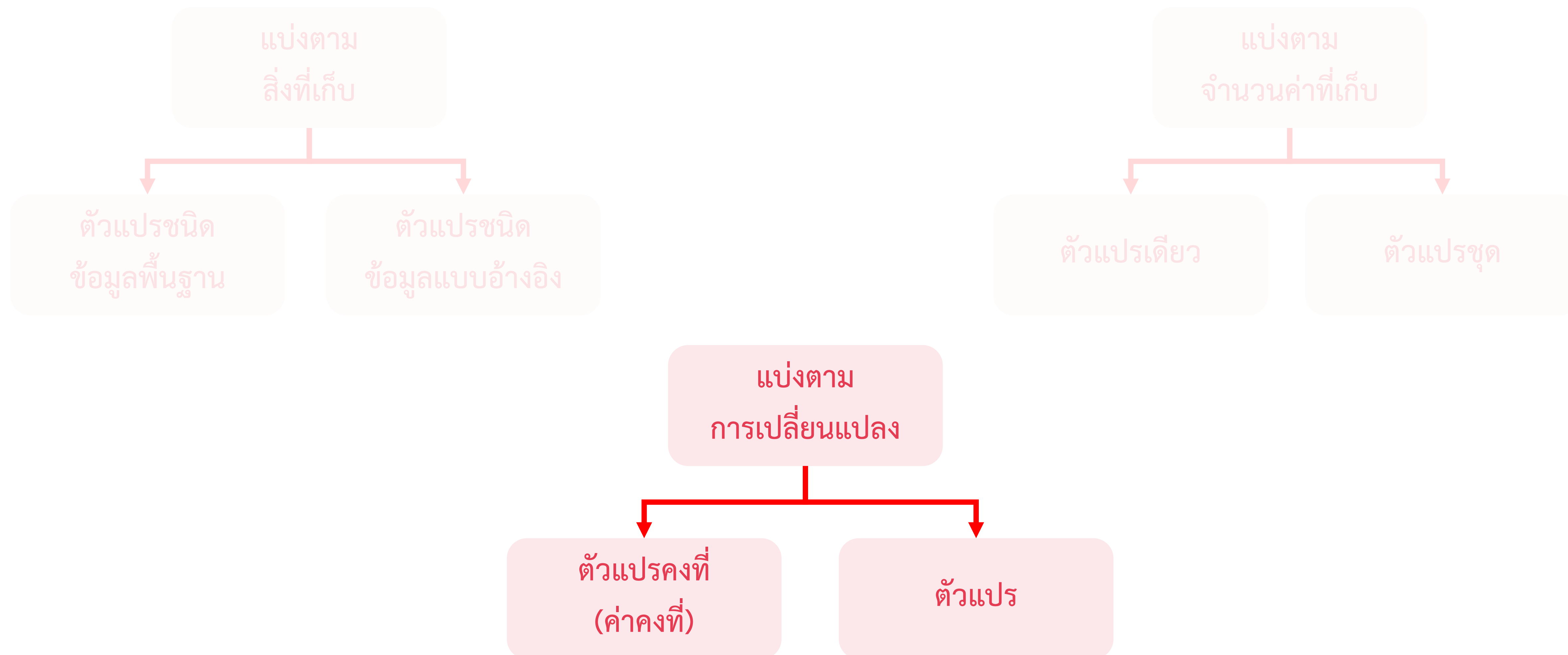
- ตัวอย่างเช่น

```
int amount;  
double x = 1, y;  
float price, wholeSalePrice;  
double radius = 3.14;  
char c = 'a';  
double y = x+4*3;  
amount = 121+14;
```


ตัวอย่างโปรแกรม

```
public class VariableAssignDemo {  
    public static void main(String args[]) {  
        int x = 5, y;  
        boolean b1;  
        float z = 3.414f;  
        double w;  
        y = 4;  
        b1 = (x > y); ➡ b1 = true;  
        w = x * 3.2; ➡ w = 16.0;  
  
        System.out.println("x = " + x + " y = " + y);  
        System.out.println("b1 = " + b1);  
        System.out.println("z = " + z + " w = " + w);  
    }  
}
```

ตัวแปร



ตัวแปรค่าคงที่



- วิธีการประกาศและกำหนดค่า

final ชนิดข้อมูล ชื่อตัวแปร = ค่า ;

- ตัวอย่างเช่น

```
final int age = 10 ;
```

```
final int age ;  
age = 10 ;
```

ตัวอย่างโปรแกรม

```
public class ConstantDemo {  
    public static void main(String args[]) {  
        final int MAXIMUM = 10;  
        final double MIN_GPA;  
  
        System.out.println("Maximum is " + MAXIMUM);  
        MIN_GPA = 2.00;  
  
        System.out.println("Minimum GPA is " + MIN_GPA);  
        // MIN_GPA = 3.00;    //illegal  
    }  
}
```

หัวข้อ

- การรับค่าผ่านทางคีย์บอร์ดและแสดงผล
- ตัวแปร (Variable)
- ตัวดำเนินการ (Operation)

ตัวดำเนินการในภาษาจาวา

- กำหนดค่า (Assignment Operator)

เครื่องหมายเท่ากับ (=)

- คอมเมนต์ (Comment Operator)

//, /* ... */, /** ... */

- คณิตศาสตร์ (Arithmetic Operator)

+, -, *, /, %

- การคำนวณและการกำหนดค่า (Compound Operator)

+=, -=, *=, /=, %=

- การเพิ่มลดค่า (Increment & Decrement Operator)

++, --

- เชื่อมข้อความ (Concatenate Operator)

เครื่องหมายบวก (+)

ตัวดำเนินการในภาษาจาวา

- กำหนดค่า (Assignment Operator)

เครื่องหมายเท่ากับ (=)

- คอมเมนต์ (Comment Operator)

//, /* ... */, /** ... */

- คณิตศาสตร์ (Arithmetic Operator)

+, -, *, /, %

- การคำนวณและการกำหนดค่า (Compound Operator)

+=, -=, *=, /=, %=

- การเพิ่มลดค่า (Increment & Decrement Operator)

++, --

- เชื่อมข้อความ (Concatenate Operator)

เครื่องหมายบวก (+)

ตัวดำเนินการ



คอมเมนต์เขียนไว้เพื่อ

- อธิบายโปรแกรม
- ให้ผู้อ่านเข้าใจโปรแกรมง่ายขึ้น
- ช่วยทำให้การแก้ไขและปรับปรุงโปรแกรมเป็นไปได้ง่ายขึ้น

ภาษาจาวากำหนดรูปแบบของการเขียนคอมเมนต์ไว้สามรูปแบบดังนี้

- `// This is a comment`

คอมเมนต์สำหรับข้อความบรรทัดเดียว

- `/* This is also a comment */`

คอมเมนต์สำหรับข้อความตั้งแต่หนึ่งบรรทัดขึ้นไป

- `/** This is a comment for documentation */`

คอมเมนต์สำหรับข้อความที่ต้องการสร้างเป็นไฟล์เอกสารที่เป็นไฟล์ประเภท HTML

ตัวดำเนินการ



- ความแตกต่างระหว่าง `//` และ `/*...*/`

```
public class App00 {  
    public static void main(String[] args) {  
        int a = 10; // b = 20, c = 5;  
        /*  
        System.out.println(10 + 20 + 30);  
        System.out.println(a + 20 + c);  
        System.out.println(a + b + "30")  
        System.out.println("a" + b + c);  
        System.out.println(a + "20" + c);  
        System.out.println("a" + b + c);  
        */  
        System.out.println("I'm Bank");  
    }  
}
```

ตัวอย่างโปรแกรม

```
/* This program is to show  
   how to write comments */  
public class CommentDemo {  
    // Main method  
    public static void main(String args[]) {  
        /** This is a comment for documentation */  
        System.out.println("Document");  
    }  
}
```

ตัวดำเนินการในภาษาจาวา

- กำหนดค่า (Assignment Operator)

เครื่องหมายเท่ากับ (=)

- คอมเมนต์ (Comment Operator)

//, /* ... */, /** ... */

- คณิตศาสตร์ (Arithmetic Operator)

+, -, *, /, %

- การคำนวณและการกำหนดค่า (Compound Operator)

+=, -=, *=, /=, %=

- การเพิ่มลดค่า (Increment & Decrement Operator)

++, --

- เชื่อมข้อความ (Concatenate Operator)

เครื่องหมายบวก (+)

ตัวดำเนินการทางคณิตศาสตร์



เครื่องหมาย	ความหมาย	ตัวอย่างนิพจน์
+	บวก	$a+b$
-	ลบ	$a-b$
*	คูณ	$a*b$
/	หาร	a/b
%	เศษจากการหาร	$a\%b$

ตัวอย่างโปรแกรม

```
public class Main {  
    public static void main(String[] args) {  
        int a = 10 + 7;  
        System.out.println("line 1: "+a);  
  
        a = 10 - 7;  
        System.out.println("line 2: "+a);  
  
        a = 10 * 7;  
        System.out.println("line 3: "+a);  
  
        a = 10 / 2;  
        System.out.println("line 4: "+a);  
    }  
}
```

ตัวดำเนินการในภาษาจาวา

- กำหนดค่า (Assignment Operator)

เครื่องหมายเท่ากับ (=)

- คอมเมนต์ (Comment Operator)

//, /* ... */, /** ... */

- คณิตศาสตร์ (Arithmetic Operator)

+, -, *, /, %

- การคำนวณและการกำหนดค่า (Compound Operator)

+=, -=, *=, /=, %=

- การเพิ่มลดค่า (Increment & Decrement Operator)

++, --

- เชื่อมข้อความ (Concatenate Operator)

เครื่องหมายบวก (+)

ตัวดำเนินการทางคณิตศาสตร์แบบย่อ

```
int a = 0;
```

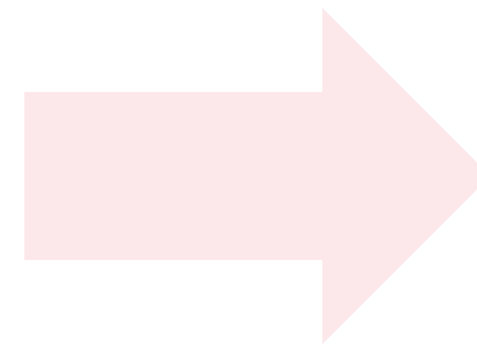
```
a = a + 3;
```

```
a = a - 3;
```

```
a = a * 3;
```

```
a = a / 3;
```

```
a = a % 3;
```



```
int a = 0;
```

```
a += 3;
```

```
a -= 3;
```

```
a *= 3;
```

```
a /= 3;
```

```
a %= 3;
```

ตัวอย่างโปรแกรม

```
public class App05 {  
    public static void main(String[] args) {  
        int a = 10;  
        System.out.println("line 1: "+a);  
  
        a += 4*3;  
        System.out.println("line 2: "+a);  
  
        a += ++a;  
        System.out.println("line 3: "+a);  
  
        a += ++a*2;  
        System.out.println("line 4: "+a);  
    }  
}
```

ตัวดำเนินการในภาษาจาวา

- กำหนดค่า (Assignment Operator)
เครื่องหมายเท่ากับ (=)
- คอมเมนต์ (Comment Operator)
//, /* ... */, /** ... */
- คณิตศาสตร์ (Arithmetic Operator)
+, -, *, /, %
- การคำนวณและการกำหนดค่า (Compound Operator)
+=, -=, *=, /=, %=
- การเพิ่มลดค่า (Increment & Decrement Operator)
++, --
- เชื่อมข้อความ (Concatenate Operator)
เครื่องหมายบวก (+)

ตัวดำเนินการเพิ่มค่าและลดค่า

```
int a = 0;
```

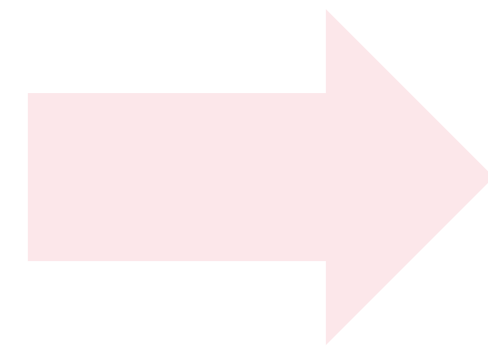
```
a = a + 1;
```

```
a = a - 1;
```

```
a = a * 1;
```

```
a = a / 1;
```

```
a = a % 1;
```



```
int a = 0;
```

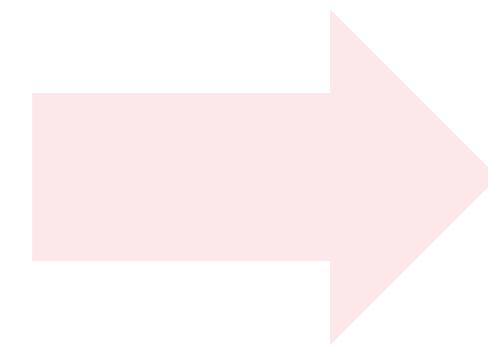
```
a += 1;
```

```
a -= 1;
```

```
a *= 1;
```

```
a /= 1;
```

```
a %= 1;
```



```
int a = 0;
```

```
a++ or ++a;
```

```
a-- or --a;
```

ตัวดำเนินการเพิ่มค่าและลดค่า

- ความแตกต่างระหว่าง **a++** และ **++a**

```
public class App01 {  
    public static void main(String[] args) {  
        int a = 10;  
        System.out.println("line 1: "+a);  
        System.out.println("line 2: "+ (a++));  
        System.out.println("line 3: "+a);  
    }  
}
```

```
public class App02 {  
    public static void main(String[] args) {  
        int a = 10;  
        System.out.println("line 1: "+a);  
        System.out.println("line 2: "+ (++a));  
        System.out.println("line 3: "+a);  
    }  
}
```

ตัวดำเนินการเพิ่มค่าและลดค่า

- ความแตกต่างระหว่าง **a--** และ **--a**

```
public class App03 {  
    public static void main(String[] args) {  
        int a = 10;  
        System.out.println("line 1: "+a);  
        System.out.println("line 2: "+ (a--));  
        System.out.println("line 3: "+a);  
    }  
}
```

```
public class App04 {  
    public static void main(String[] args) {  
        int a = 10;  
        System.out.println("line 1: "+a);  
        System.out.println("line 2: "+ (--a));  
        System.out.println("line 3: "+a);  
    }  
}
```

ตัวอย่างโปรแกรม

```
public class IncrementDemo {  
    public static void main(String args[]) {  
        int x;  
        int y;  
  
        x = 5;  
        y = x++;  
  
        System.out.println("x = "+x+" y = "+y) ;  
  
        y = ++x;  
  
        System.out.println("x = "+x+" y = "+y) ;  
    }  
}
```


ลำดับความสำคัญของตัวดำเนินการ

ลำดับ	เรียงจาก	ตัวดำเนินการ
1	ขวาไปซ้าย (R to L)	++, --, +, -, ~, !, ()
2	ซ้ายไปขวา (L to R)	*, /, %
3	ซ้ายไปขวา (L to R)	+, -
4	ขวาไปซ้าย (R to L)	=, +=, -=, *=, /=, %=, <<=, >>=, >>>=, &=, ^=, =

ตัวอย่าง



ให้นักศึกษาคำนวณหาผลลัพธ์ตามลำดับดังนี้

$$x = 2 + 3 * 4 - (7 + 2);$$

ขั้นที่ 1 คำนวณหาผลลัพธ์ค่า $7 + 2$ ทำให้ได้

$$x = 2 + 3 * 4 - 9$$

ขั้นที่ 2 คำนวณหาผลลัพธ์ค่า $3 * 4$ ทำให้ได้

$$x = 2 + 12 - 9$$

ขั้นที่ 3 คำนวณหาผลลัพธ์ค่า $2 + 12$ ทำให้ได้

$$x = 14 - 9$$

ขั้นที่ 4 คำนวณหาผลลัพธ์ค่า $14 - 9$ ทำให้ได้

$$x = 5$$

ตัวดำเนินการในภาษาจาวา

- กำหนดค่า (Assignment Operator)

เครื่องหมายเท่ากับ (=)

- คอมเมนต์ (Comment Operator)

//, /* ... */, /** ... */

- คณิตศาสตร์ (Arithmetic Operator)

+, -, *, /, %

- การคำนวณและการกำหนดค่า (Compound Operator)

+=, -=, *=, /=, %=

- การเพิ่มลดค่า (Increment & Decrement Operator)

++, --

- เชื่อมข้อความ (Concatenate Operator)

เครื่องหมายบวก (+)

ตัวดำเนินการที่ใช้กับข้อมูลชนิด String

String เป็นออเปอเรเตอร์ที่มีตัวดำเนินการที่ใช้ในการเชื่อมข้อความสองข้อความเข้าด้วยกัน โดยใช้เครื่องหมาย + อาทิเช่น

```
String s1 = "Hello" + " World";
```

ตัวดำเนินการในการเชื่อมข้อความสามารถใช้เชื่อมข้อมูลชนิด String กับตัวถูกดำเนินการที่เป็นชนิดข้อมูลอื่น ๆ ภาษาจาวาจะแปลงชนิดข้อมูลดังกล่าวให้เป็นชนิด String โดยอัตโนมัติ อาทิเช่น คำสั่ง

```
String s1 = "This";
```

```
String s2 = s1 + " is a test ";
```

```
String s3 = s1+4;
```

ตัวดำเนินการที่ใช้กับข้อมูลชนิด String

ความแตกต่างระหว่าง + สำหรับ ตัวเลข และ String

```
public class App00 {  
    public static void main(String[] args) {  
        int a = 10, b = 20, c = 5;  
        System.out.println(10 + 20 + 30);  
        System.out.println(a + 20 + c);  
        System.out.println(a + b + "30")  
        System.out.println("a" + b + c);  
        System.out.println(a + "20" + c);  
        System.out.println("a" + b + c);  
  
        System.out.println("I'm " + (b + c));  
        System.out.println("I'm " + b + c);  
  
    }  
}
```

ตัวดำเนินการ



- ตัวดำเนินการที่มีการ**คืนค่า**

ตัวดำเนินการ	ประกอบด้วย
คณิตศาสตร์	+, -, *, /, %
การเปรียบเทียบ	< , > , <= , >= , == , !=
ตรรกศาสตร์	!, &&, & , ,
เชื่อมต่อข้อความ	+

→ int / double

} boolean

→ string

- ตัวดำเนินการที่**ไม่มีการคืนค่า**

ตัวดำเนินการ	ประกอบด้วย
กำหนดค่า	=
คอมเมนต์	//, /* ... */ , /** ... */
การคำนวณและการกำหนดค่า	+=, -=, *=, /=, %=
การเพิ่มลดค่า	++, --

ตัวดำเนินการ



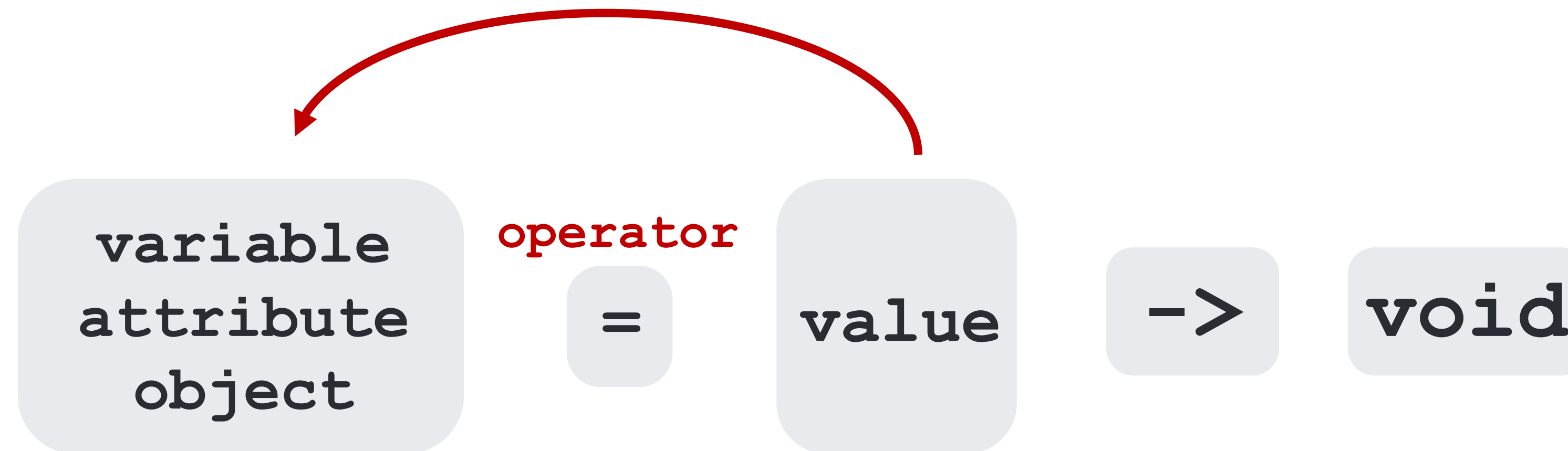
- ตัวดำเนินการที่มีการ**คืบค่า**

ตัวดำเนินการ	ประกอบด้วย
คณิตศาสตร์	+, -, *, /, %
การเปรียบเทียบ	< , > , <= , >= , == , !=
ตรรกศาสตร์	!, &&, & , ,
เชื่อมข้อความ	+

- ตัวดำเนินการที่**ไม่มีการคืบค่า**

ตัวดำเนินการ	ประกอบด้วย
กำหนดค่า	=
คอมเมนต์	//, /* ... */ , /** ... */
การคำนวณและการกำหนดค่า	+=, -=, *=, /=, %=
การเพิ่มลดค่า	++, --

ตัวดำเนินการที่ไม่มีค่าคืนค่า



ตัวดำเนินการ



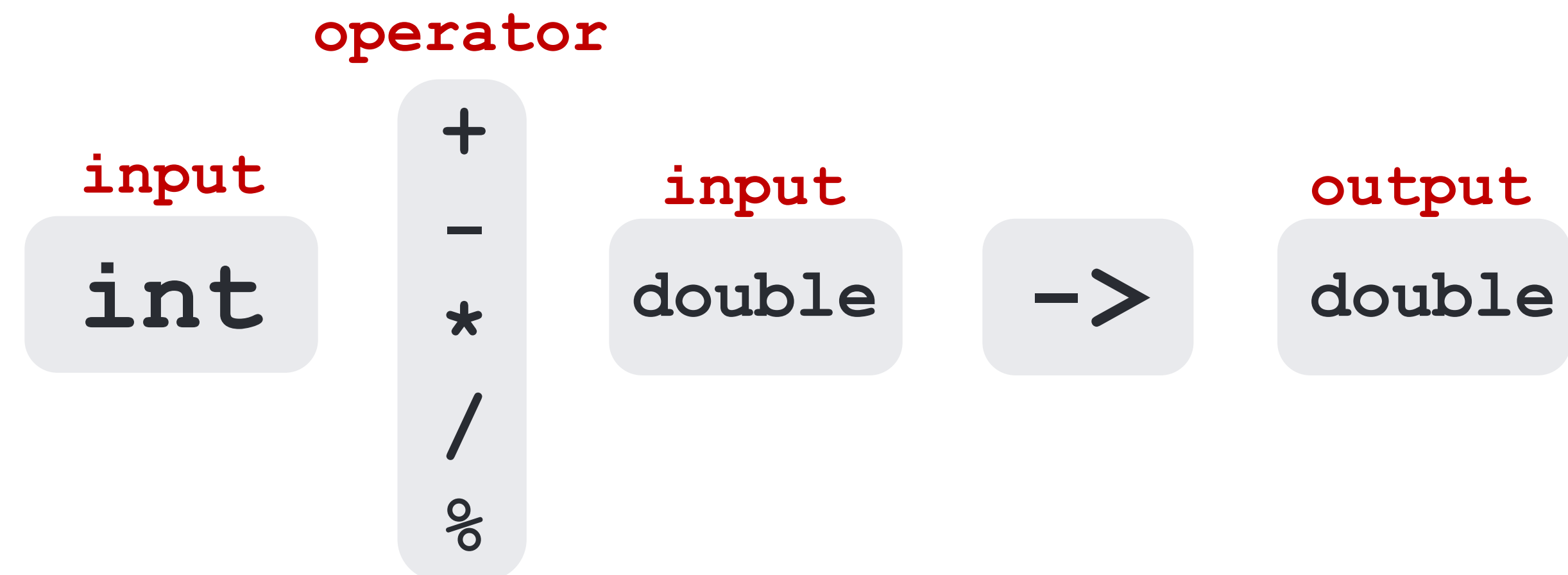
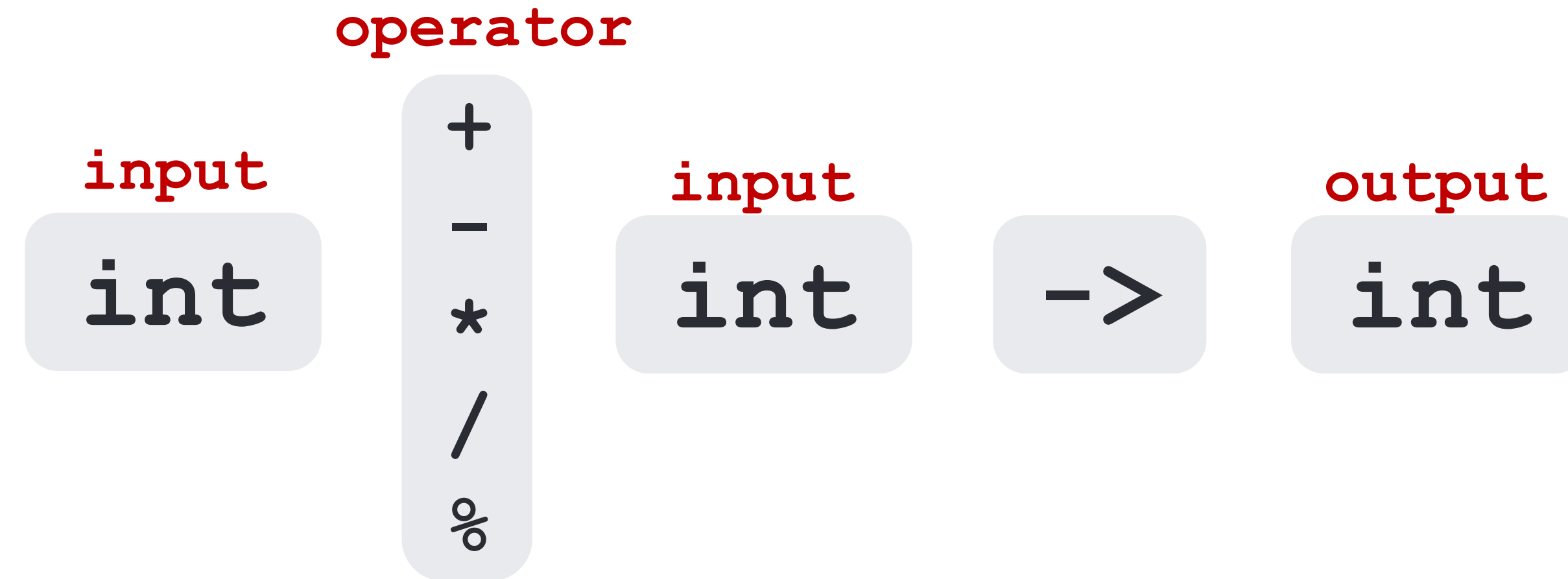
- ตัวดำเนินการที่มีการ**คืนค่า**

ตัวดำเนินการ	ประกอบด้วย
คณิตศาสตร์	+, -, *, /, %
การเปรียบเทียบ	< , > , <= , >= , == , !=
ตรรกศาสตร์	!, &&, & , ,
เชื่อมข้อความ	+

- ตัวดำเนินการที่**ไม่มีการคืนค่า**

ตัวดำเนินการ	ประกอบด้วย
กำหนดค่า	=
คอมเมนต์	//, /* ... */ , /** ... */
การคำนวณและการกำหนดค่า	+=, -=, *=, /=, %=
การเพิ่มลดค่า	++, --

ตัวดำเนินการทางคณิตศาสตร์



ตัวดำเนินการทางคณิตศาสตร์

7

```

public class Main {
    public static void main(String[] args) {
        int a = 1 , b = 2;
        double c = 1.0, d = 2.0;

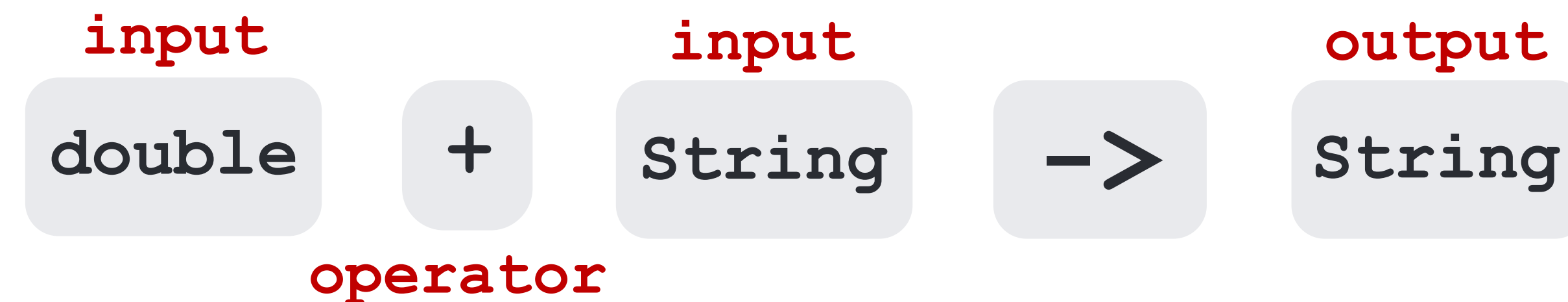
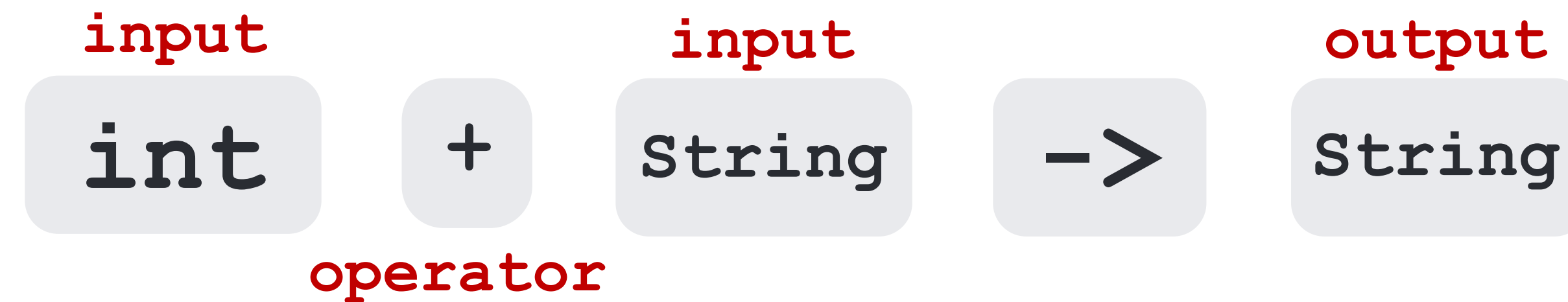
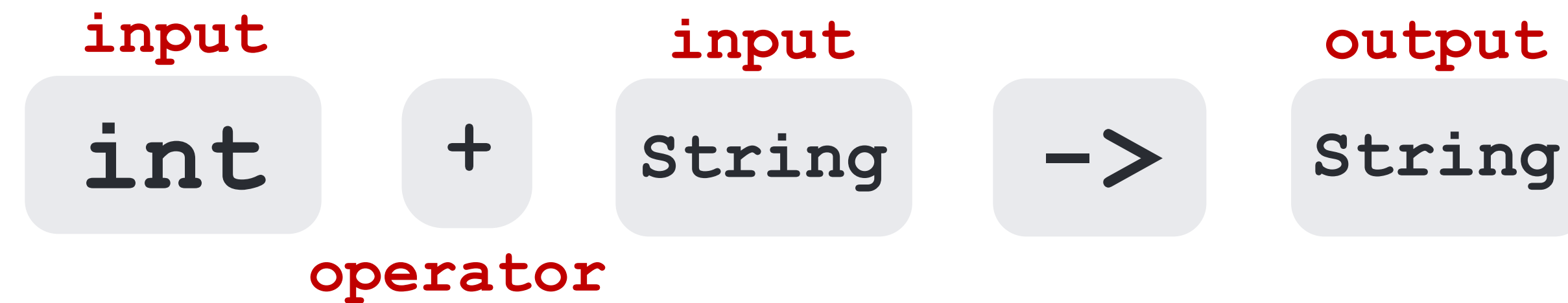
        System.out.println("===== int & int =====");
        System.out.println("a + b = " + (a+b));
        System.out.println("a - b = " + (a-b));
        System.out.println("a * b = " + (a*b));
        System.out.println("a / b = " + (a/b));
        System.out.println("a % b = " + (a%b));
        System.out.println("===== int & double =====");
        System.out.println("c + b = " + (c+b));
        System.out.println("c - b = " + (c-b));
        System.out.println("c * b = " + (c*b));
        System.out.println("c / b = " + (c/b));
        System.out.println("c % b = " + (c%b));
        System.out.println("===== double & double =====");
        System.out.println("c + d = " + (c+d));
        System.out.println("c - d = " + (c-d));
        System.out.println("c * d = " + (c*d));
        System.out.println("c / d = " + (c/d));
        System.out.println("c % d = " + (c%d));
    }
}
  
```

run:

```

===== int & int =====
a + b = 3
a - b = -1
a * b = 2
a / b = 0
a % b = 1
===== int & double =====
c + b = 3.0
c - b = -1.0
c * b = 2.0
c / b = 0.5
c % b = 1.0
===== double & double =====
c + d = 3.0
c - d = -1.0
c * d = 2.0
c / d = 0.5
c % d = 1.0
  
```

ตัวดำเนินการเชื่อมข้อความ

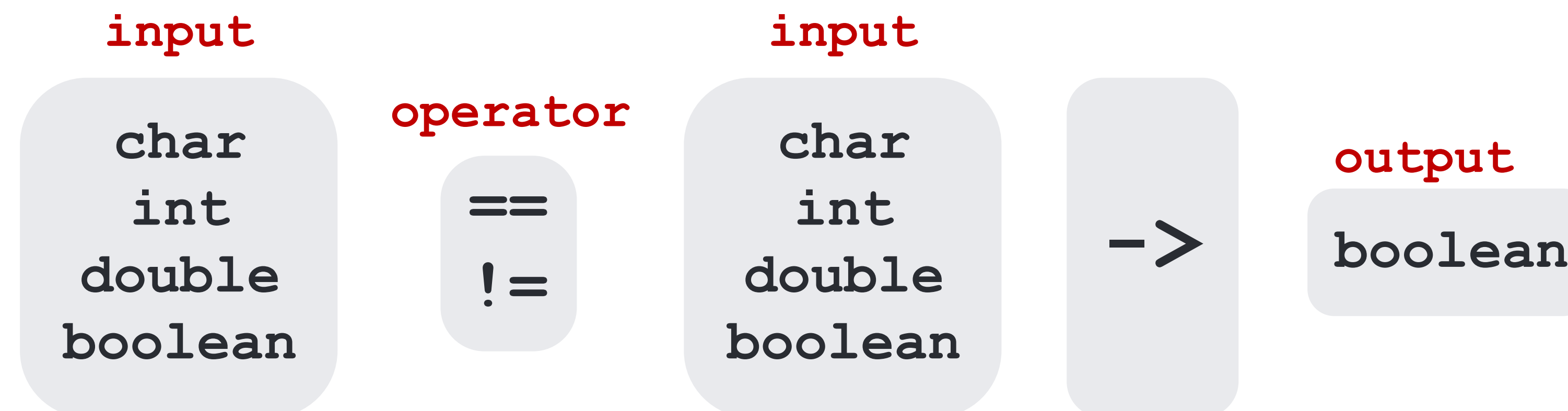
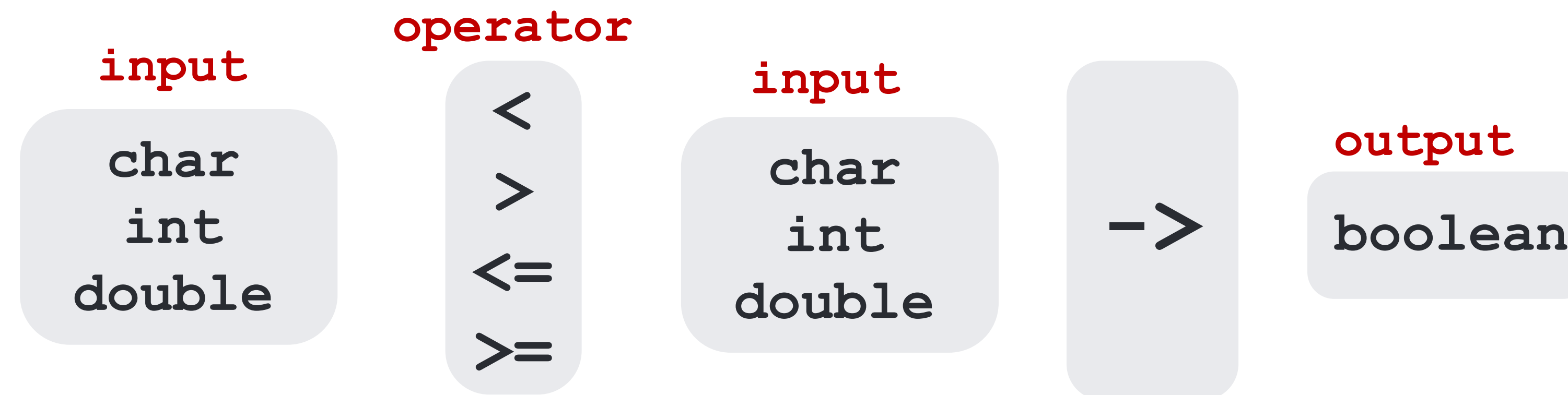


ตัวดำเนินการที่ใช้กับข้อมูลชนิด String

ความแตกต่างระหว่าง + สำหรับ ตัวเลข และ String

```
public class App00 {  
    public static void main(String[] args) {  
        int a = 10, b = 20, c = 5;  
        System.out.println(10 + 20 + 30); 60  
        System.out.println(a + 20 + c); 35  
        System.out.println(a + b + "30"); 3030  
        System.out.println("a" + b + c); "a205"  
        System.out.println(a + "20" + c); 10205  
System.out.println("a" + b + c);  
  
        System.out.println("I'm " + (b + c));  
        System.out.println("I'm " + b + c);  
  
    }  
}
```

ตัวดำเนินการเปรียบเทียบ

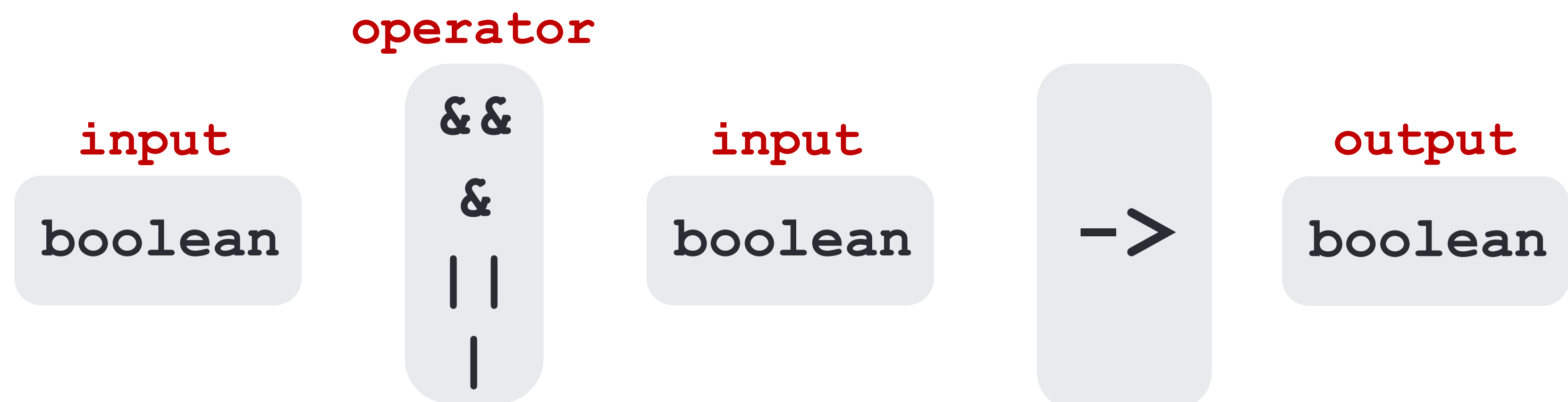


ตัวดำเนินการการเปรียบเทียบ

```
public class Main {  
    public static void main(String[] args) {  
  
        System.out.println(1 == 1);  
        System.out.println('a' > 64);           // เปรียบเทียบรหัส ASCII Code  
        System.out.println('a' != 'b');  
        System.out.println(true == false);  
  
    }  
}
```

run:
true
true
true
false

ตัวดำเนินการตรรกศาสตร์



ตัวดำเนินการตรรกศาสตร์

```
public class Main {  
    public static void main(String[] args) {  
  
        boolean a = true, b = false;  
  
        System.out.println("a && b = " + (a && b));  
        System.out.println("a || b = " + (a || b));  
        System.out.println("a & b = " + (a & b));  
        System.out.println("a | b = " + (a | b));  
        System.out.println("!a = " + (!a));  
  
    }  
}
```

run:
a && b = false
a || b = true
a & b = false
a | b = true
!a = false

ตัวดำเนินการ



- ตัวดำเนินการที่มีการ**คืนค่า**

ตัวดำเนินการ	ประกอบด้วย
คณิตศาสตร์	+, -, *, /, %
การเปรียบเทียบ	< , > , <= , >= , == , !=
ตรรกศาสตร์	!, &&, & , ,
เชื่อมข้อความ	+

- ตัวดำเนินการที่**ไม่มีการคืนค่า**

ตัวดำเนินการ	ประกอบด้วย
กำหนดค่า	=
คอมเมนต์	//, /* ... */ , /** ... */
การคำนวณและการกำหนดค่า	+=, -=, *=, /=, %=
การเพิ่มลดค่า	++, --