

# **JAVA BEGINNER**

**VARIABLE & DATA TYPE**



Abstract of New Technology

Website <http://www.antkh.com>

Tel 010 / 016 66 66 53

Prepared By Tum Sakal

Tel 087 36 31 30

Mail [tumsakal.ts@gmail.com](mailto:tumsakal.ts@gmail.com)

Facebook [sak kal](#)

# VARIABLE

- **Variable** គឺជាអញ្ញតិ រឺជាអថេរ ដែលប្រើសំរាប់រក្សាទុកទិន្នន័យ រឺ តំណាងអោយទិន្នន័យណាមួយ។
- **Variable** រក្សាទុកទិន្នន័យមិនអចិន្ត្រៃយ៍ រឺបណ្តោះអាសន្ន
- **Variable** មួយរក្សាទុកទិន្នន័យបានតែមួយ

# SYNTAX

- Uninitialized Variable Declaration
  - `data_type name;`
  - `data_type name, name2, ...;`
- Initialized Variable Declaration
  - `data_type name = value;`
  - `data_type name1 = value1, name2 = value2, ...;`

# DATA TYPE

- Primitive Type:

- Integer: byte, short, int, long
- Floating-point: float, double
- Character: char
- Boolean: boolean

# INTEGER

## Data Type

- byte
- short
- int
- long
- **Example:**
- `int number = 1234;`

## Size

- 1 byte
- 2 bytes
- 4 bytes
- 8 bytes

## Range

- 128 to 127
- $-2^{15}$  to  $2^{15} - 1$
- $-2^{31}$  to  $2^{31} - 1$
- $-2^{63}$  to  $2^{63} - 1$

# FLOATING-POINT

## Data Type

- float
- double

## Size

4 bytes  
8 bytes

## Range

$3.4e-38$  to  $3.4e+38$   
 $1.7e-308$  to  $1.7e+308$

## Example:

float money = 1000f;

double money = 10000d;

# CHARACTER

## Data Type

- char

## Size

2bytes

## Range

0 to 65,536

## Escaped Character:

**\n**    new line

**\r**    carriage return

**\', \", \\**

**\t**    tab

**\b**    backspace

## Example:

```
char letter = 'a';
```



# BOOLEAN

## Data Type

- boolean

## Size

1 byte

## Range

true/false

## Example:

```
boolean condition = true;
```

# ASSIGN VALUE

- ផ្តល់តម្លៃឱ្យ variable តាមរយៈសញ្ញា “=”

double money = 1000D;

money = 3200D;

- អ្វីដែលស្ថិតពីក្រោយសញ្ញា = គឺជាតម្លៃដែលត្រូវផ្តល់ទៅឱ្យ variable

# NAMING

- មិនអាចផ្ដើមដោយលេខ
- អាចជាអក្សរពី a-z ឬ A-Z ឬលេខ 0-9
- គួរដាក់ឈ្មោះឱ្យមានន័យគ្រប់គ្រាន់
- មិនត្រូវមានផ្ទុកសញ្ញាពិសេស ដូចជា \*, -, /, +, (, ), &, #, @,...
- មិនត្រូវមានឈ្មោះជាន់ keyword ដូចជា int, float, bool, return,...
- មិនអាចដកឃ្លា តែអាចជំនួសដោយសញ្ញា “\_” បាន
- គ្រប់ variable មានលក្ខណៈ Case-Sensitive
- មិនអាចមានឈ្មោះដូចគ្នាក្នុង block តែមួយ

# OUTPUT

- `Java.lang.System.out.print(value/variable);`
- `Java.lang.System.out.println(value/variable);`
- `Java.lang.System.out.printf(format, args...);`

# Input

## java.util.Scanner

គឺជា Class ដែលអាចចាប់យកទិន្នន័យពីអ្នកប្រើប្រាស់បានតាមរយៈ Keyboard

### Return

- byte
- short
- int
- long

### Method

nextByte()  
nextShort()  
nextInt()  
nextLong

### Return

float  
double  
  
boolean

### Method

nextFloat()  
nextDouble()  
  
nextBoolean()

- `Java.util.Scanner sc = new java.util.Scanner(System.in);`
- `byte b = sc.nextByte();`
- `short sh = sc.nextShort();`
- `int l = sc.nextInt();`
- `long lg = sc.nextLong();`
- `float fl = sc.nextFloat();`
- `double db = sc.nextDouble();`
- `char c = sc.next().charAt(0);`
- `boolean bl = sc.nextBoolean();`

# MATH OPERATOR

- Arithmetic Operator:  $+, -, *, /, \%$
- Assignment Operator:  $+=, -=, *=, /=, \%=, =$
- ប្រមាណវិធីត្រូវបានអនុវត្តន៍ពីឆ្វេងមកស្តាំ។ ប្រមាណវិធី ( $/, *, \%$ ) មានអាទិភាពស្មើគ្នា តែប្រមាណវិធីទាំងបីនេះមានអាទិភាពខ្ពស់ជាងប្រមាណវិធី ( $+, -$ ) ។
- ចំពោះសញ្ញា  $/$  ផលចែកនឹងទទួលបានចំនួនគត់ បើសិនជាយើងធ្វើប្រមាណវិធី ចំនួនគត់ និងចំនួនគត់។

# MATH METHOD

## java.lang.Math:

Math ជា class មួយដែលមានផ្ទុកនូវ Static method ជាច្រើនសំរាប់ការគណនាបែបគណិតវិទ្យា

### Return

### Method

double

ceil(variable)

double

floor(variable)

double

round(variable)

Type

min(a, b)

Type

max(a, b)

### Return

### Method

double

pow(a, b)

double

sqrt(variable)

double

cos(variable)

double

random()

double

PI

$a^b$

3.14



# RANDOM NUMBER

- `double rd = Math.random();`  
`0.0D <= rd < 1.0D`
- `Double rd2 = Math.random() * 100D;`  
`0.0D <= rd < 100.0D`
- `Double rd3 = (Math.random() * (max - min)) + min;`  
`min <= rd < max`

# WIDENING

- **Widening** គឺជាការបំប្លែងទិន្នន័យ នៃប្រភេទមួយ ទៅកាន់ប្រភេទមួយទៀត ដោយស្វ័យប្រវត្តិ។
- លក្ខណៈចំណាំ
  - ប្រភេទទិន្នន័យស្រដៀងគ្នា
  - ពីរតួចទៅធំ

# WIDENING LIST

- char -> int, to long, to float, to double.
- byte -> short, to int, to long, to float, to double.
- short -> int, to long, to float, to double.
- int -> long, to float, to double.
- long -> float, to double.
- float -> double.

# NARROWING

- **Narrowing** ផ្ទុយពី Widening មានន័យថាអ្នកត្រូវសរសេរកូដបំប្លែងទិន្នន័យដោយខ្លួនឯង។
- **Casting** អាចត្រូវបានប្រើ ដើម្បីបំប្លែងទិន្នន័យ នៃប្រភេទមួយ ទៅកាន់ប្រភេទមួយទៀត។

*(target\_data\_type) variable*

# NARROWING LIST

- char -> byte, short
- short -> byte, char
- int -> byte, short, char
- long -> byte, short, int, char
- float -> byte, short, int, long, char
- double -> byte, short, int, long, float, char

# PRIMITIVE WRAPPER CLASS

- **Wrapper Class** ជាប្រភេទ class ដែលខ្ចប់តម្លៃនៃប្រភេទ Primitive នៅក្នុងខ្លួនវា។

- java.lang.\*:

Integer: Byte, Short, Integer, Long

Floating-Point: Float, Double

Character: Character

Boolean: Boolean

# PRIMITIVE WRAPPER CLASS

- Wrapper Class ត្រូវបានប្រើប្រាស់ជាចំណាត់ថ្នាក់នៅក្នុងពពួក Generic API របស់ Java ដែលមានដូចជា Collection API និង Reflection API ។
- Wrapper Class ជំនួយការយឺត និងចំណាយ Memory Space ច្រើនជាងប្រភេទ Primitive ។

# AUTO-BOXING

- **Boxing:** ខ្ចប់តម្លៃនៃ Primitive ទៅក្នុង Wrapper Class

```
int i = 100;
```

```
Integer number = i;
```

- **Unboxing:** ពន្លាតតម្លៃនៃ Primitive ចេញពី Wrapper Class

```
Integer number = 100;
```

```
int value = number;
```



# EXTRACTING INTEGRAL

## Return

byte

short

int

long

float

double

## Method

byteValue()

shortValue()

intValue()

longValue()

floatValue()

doubleValue()