UNIT 1: Primitive Types

Variable - Store data in RAM (random access memory)

no spaces

Assignment operator

semicolon

```
type  name  =  value;
```

Camel Case

int, Double, Boolean, String

↳ ex) 8, 12.1, 6.0, true, "Hello"

Integer - Whole number (int)

Double - Number with decimal places (double)

Boolean - True / false (boolean)

} primitive data types

String - Text values (String) Capital → reference data type

Example)

```
int myInteger = 8;
double myDouble = 7.6;
boolean isAwesome = true;
String myName = "Ajay";
```

Constant - Variable whose value cannot be changed

$$\text{final } \overbrace{\text{type name = value;}}^{\text{variable initialization}}$$

Ex) final int hoursInADay = 24;

Chaning value of variables

```
int myInteger = 7;
myInteger = 8;
```

use assignment operator "=" and place new value.

☆ `System.out.println(myInteger);` >>> 8

↳ displays the value of variable.

Ex)

```
boolean flag = true;
flag = false;
```

```
String myGrade = "B";
myGrade = "A";
```

Arithemetic Operators

+ addition

− subtraction

* Multiplication

/ Division

% Modulo
↳ Remainder

12 % 5

ex)

$$\begin{array}{r} 2 \\ 5\overline{)12} \\ -10 \\ \hline \boxed{2} \end{array}$$

12%5 → $\underline{\underline{2}}$

Ex)

```
int sum = 8+7;
System.out.println(sum);
```

>>> 15

```
double diff = 17.0 - 7.0;
System.out.println(diff); >>> 10.0
```

int Double

```
double product = 7 * 2.0;
System.out.println(product); >>> 14.0
```

```
double quotient = 3.0/2;
System.out.println(quotient); >>> 1.5
```

int int

```
int quotientTwo = 3/2;
System.out.println(quotientTwo); >>> 1
```

* Integer division TRUNCATES

1 % 2

```
int modulo = quotientTwo % 2;
System.out.println(modulo); >>> 1
```

$$\begin{array}{r} 0 \\ 2\overline{)1} \\ -0 \\ \boxed{1} \end{array}$$

Modifying values with math

```
int x = 7;
x = x + 1;
System.out.println(x); >>> 8
```

```
x += 2;
System.out.println(x); >>> 10
```

+= -= *= /= %= } ← Compound assignment operators

Increment Operators

++ --

```
var x= 8;
x++;
System.out.println(x);  >>> 9
```

Casting Variables

Casting - Converting one type into another

int ⟷ double

```
int d = (int) 7.8;
System.out.println (d);  >>> 7
```

→ Integer Always truncates!

```
double i = (double)6;
System.out.println (i);  >>> 6.0
```

Min + Max Values → Integer

Integer → 4 bytes of RAM

Integer.MIN_VALUE → -2,147,483,648

Integer.MAX_VALUE → 2,147,483,647

Lower b/c count 0

Long stores bigger numbers.