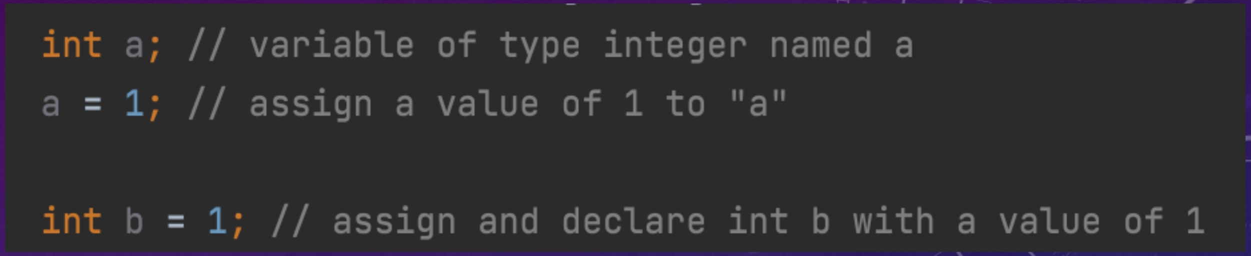
VARIABLES AND OPERATIONS

VARIABLES



```
int a; // variable of type integer named a
a = 1; // assign a value of 1 to "a"

int b = 1; // assign and declare int b with a value of 1
```

- Stores data of different types
- Data can be recalled by using the variable name
- Declare a variable with the type and the name
    - Save a space for a variable of type ___ and give it the name ___
- Give variables values with the assignment operator (=)
- Able to declare and assign in one step

PRIMITIVE DATA TYPES

- boolean – most basic data type, stores true or false
- int (integer) – stores a whole number from -2.15 billion to 2.15 billion
- float – stores a floating point number (i.e. decimal point “floats” around); IEEE 32 bit float
- double – stores a floating point number **double** the capacity/precision of a float; IEEE 64 bit float
- byte – stores a whole number from -128 to 128
- short – store a whole number from -32,768 to 32,767
- long – stores a whole number from -2^{63} to -2^{63}-1
- char – stores a Unicode character surrounded by ‘single quotes'
    - Unicode is a standard for making emojis, symbols, and characters from many different languages

ASSIGNING VALUES

- Floats need to be suffixed with an “F” or “f”
    - Otherwise, Java will think it’s a double
- Longs need to be suffixed with an “L” or “l”
    - Otherwise, Java will think it’s an int
    - Should not use “l”, might think it is a 1 (one)
- Bytes and shorts don’t need to be suffixed with anything
- YOU MUST DECLARE VARIABLES BEFORE ASSIGNING THEM

STRINGS

- Store text data
- Not a primitive data type, but can be used as one
- Created by surrounding text in “double quotes”
- Can be combined by “adding” two strings together
    - “hello “ + “world” = “hello world”

NUMERIC OPERATORS

- Addition (+)
- Subtraction (-)
- Multiplication (*)
- Division (/)
- Modulus (%)
  - Remainder
- Java follows order of operations
  - Parentheses, multiplication, division, modulus, addition, subtraction

Operation	Symbol	Example
Addition	+	5 + 6 = 11
Subtraction	-	10 - 8 = 2
Multiplication	*	9 * 3 = 27
Division	/	4 / 2 = 2
Modulus	%	7 % 2 = 1

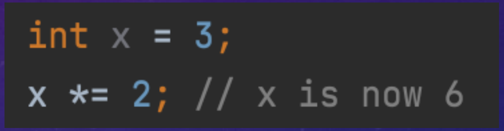
INTEGER VS FLOATING POINT DIVISION

- If both operands are integers, the decimals are truncated
  - Example: 5/3 = 1
  - Both numbers are integers so the decimal part goes away
- If at least ONE of the operands is a floating point (float or double), the decimals are NOT truncatetd
  - Example 5.0/3 = 1.66666667
  - One number has a decimal so the decimal part stays

AUGMENTED ASSIGNMENT OPERATORS

- Just like numeric operators but takes value from variable, does operation, and stores result back in variable

- Multiply x by 2 to get 6
- Store 6 into x



```
int x = 3;
x *= 2; // x is now 6
```

INCREMENT AND DECREMENT OPERATORS

- Add or subtract 1 from the variable
- ++x, x++, --x, x--
- Can be used in expressions or standalone
- Pre-increment increments and uses the new value
- Post-increment uses the old value then increments
- Pre/post doesn’t matter if not using in an expression

```
int c = 5;
int d = 3;
d = ++c; // d = 6, c = 6; add one to c and use the new value of c
```

```
int e = 5;
int f = 3;
f = e++; // f = 5, e = 6; add one to e and use the old value of e
```

INCREMENT AND DECREMENT OPERATORS CONT'D

Name	Symbol	Definition	Example (x = 1)
Pre-increment	++x	Adds 1 to x and uses the new value	y = ++x y is now 2
Post-increment	x++	Adds 1 to x and uses the old value	y = x++ y is now 1
Pre-decrement	--x	Subtracts 1 from x and uses the new value	y = --x y is now 0
Post-decrement	x--	Subtracts 1 from x and uses the old value	y = x-- y is now 1

BOOLEAN OPERATIONS

- AND operator – outputs true if both inputs are true
- OR operator – outputs true if either input is true
- NOT operator – outputs the opposite of the input
- Order of operations
    1. NOT
    2. AND
    3. OR

TRUTH TABLES

AND Truth Table

A	B	A && B
true	true	true
true	false	false
false	true	false
false	false	false

OR Truth Table

A	B	A || B
true	true	true
true	false	true
false	true	true
false	false	false

Not Truth Table

A	!A
true	false
false	true

COMPARISON OPERATORS

Operator	Definition	Example
==	Equals	1 == 1 is true
!=	Not equal	3 != 2 is true
>=	Greater than or equal to	2 >= 5 is false
<=	Less than or equal to	5 <= 5 is true
>	Greater than	8 > 9 is false
<	Less than	7 < 5 is false