Java Basic Program

Module 1

1. Introduction part, Languages Programing, Java history, JDK, “hello World”

2. Project, package, Class, method

3. Variables. Keyboard input

4. Variables and data types

5. Consultation

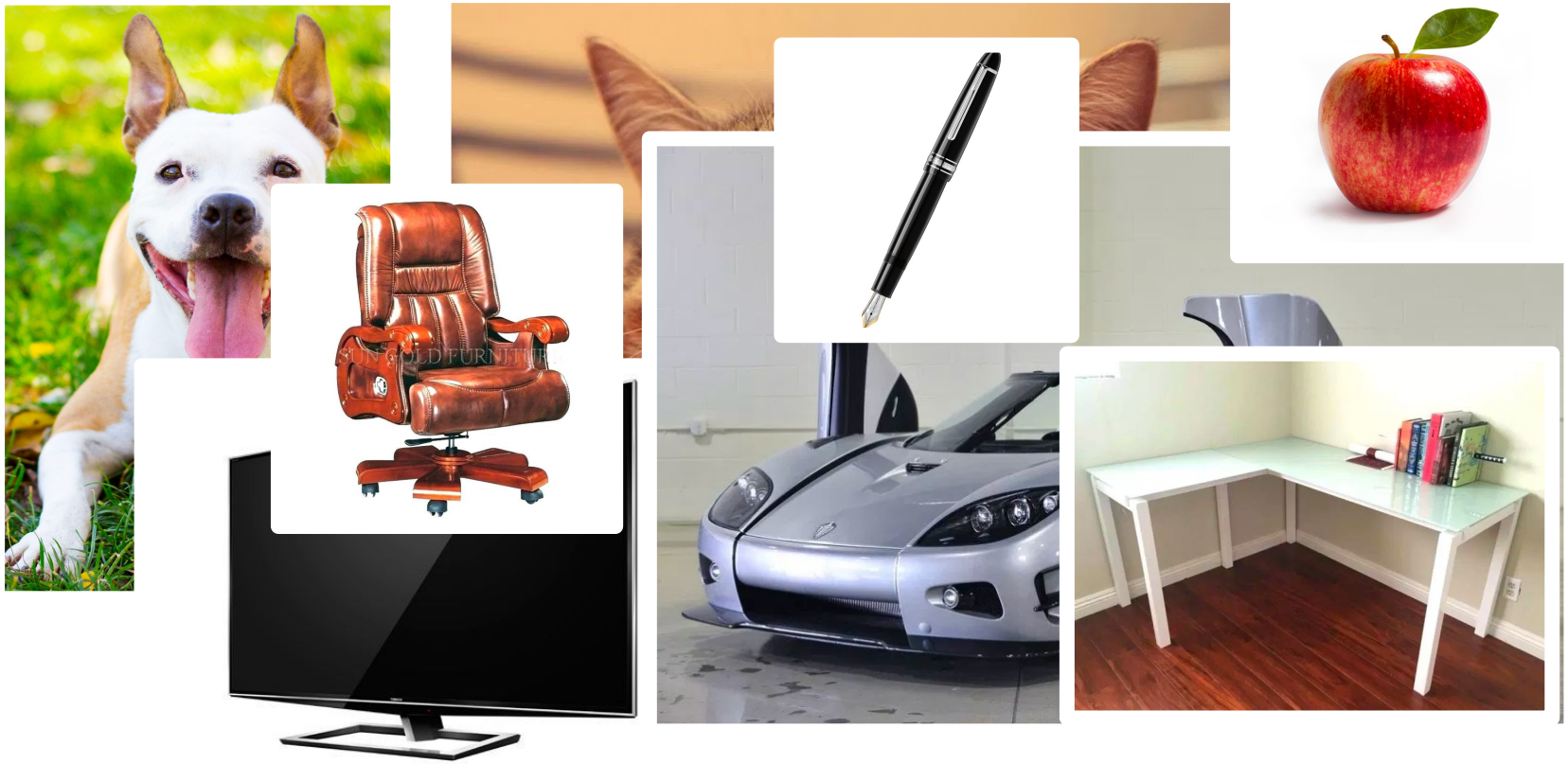
INTRODUCTION TO

JAVA

OBJECT-ORIENTED PRINCIPLES OVERVIEW

WHAT IS AN OBJECT?

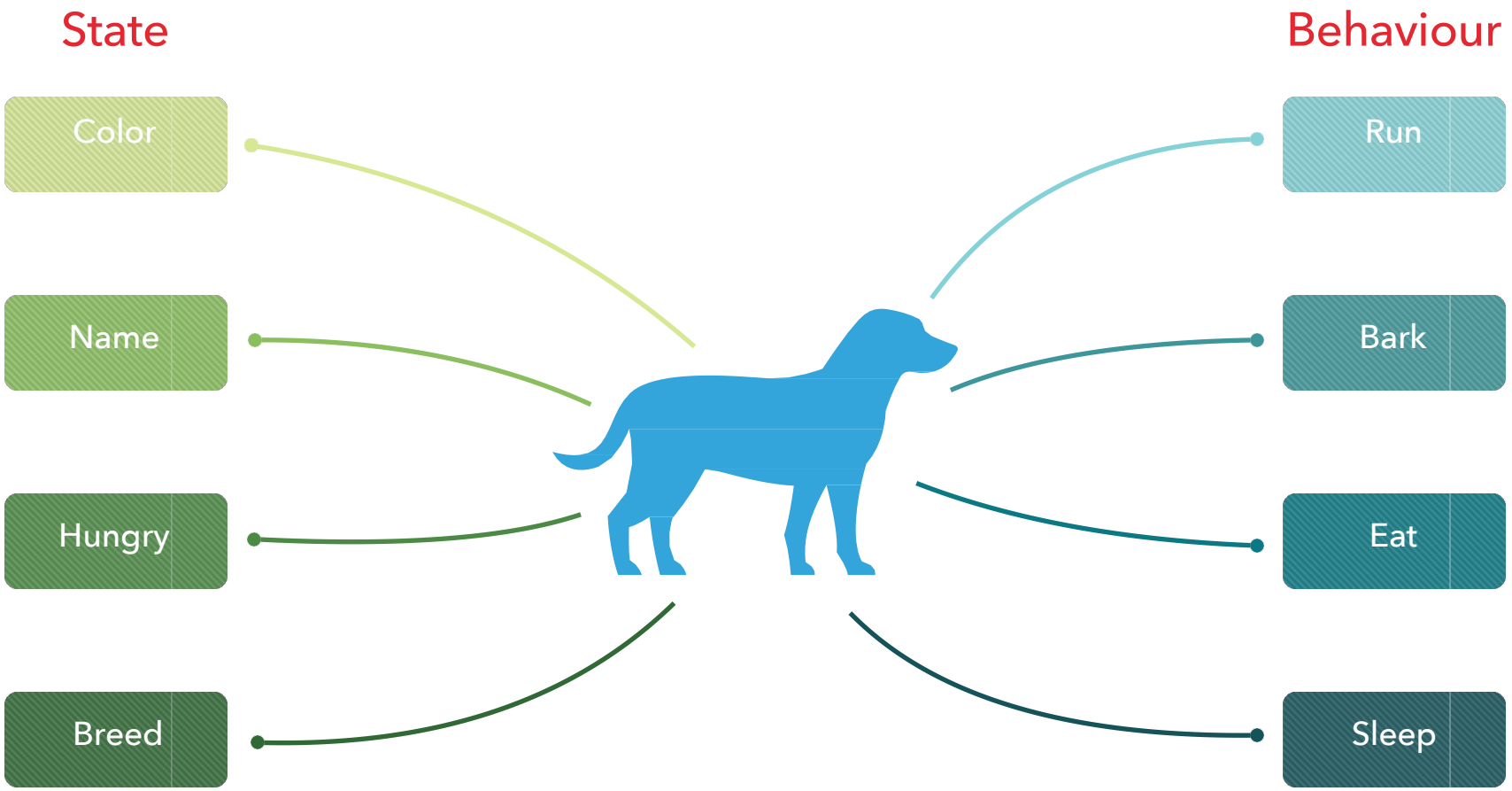
- Just look around... Everything is an object!



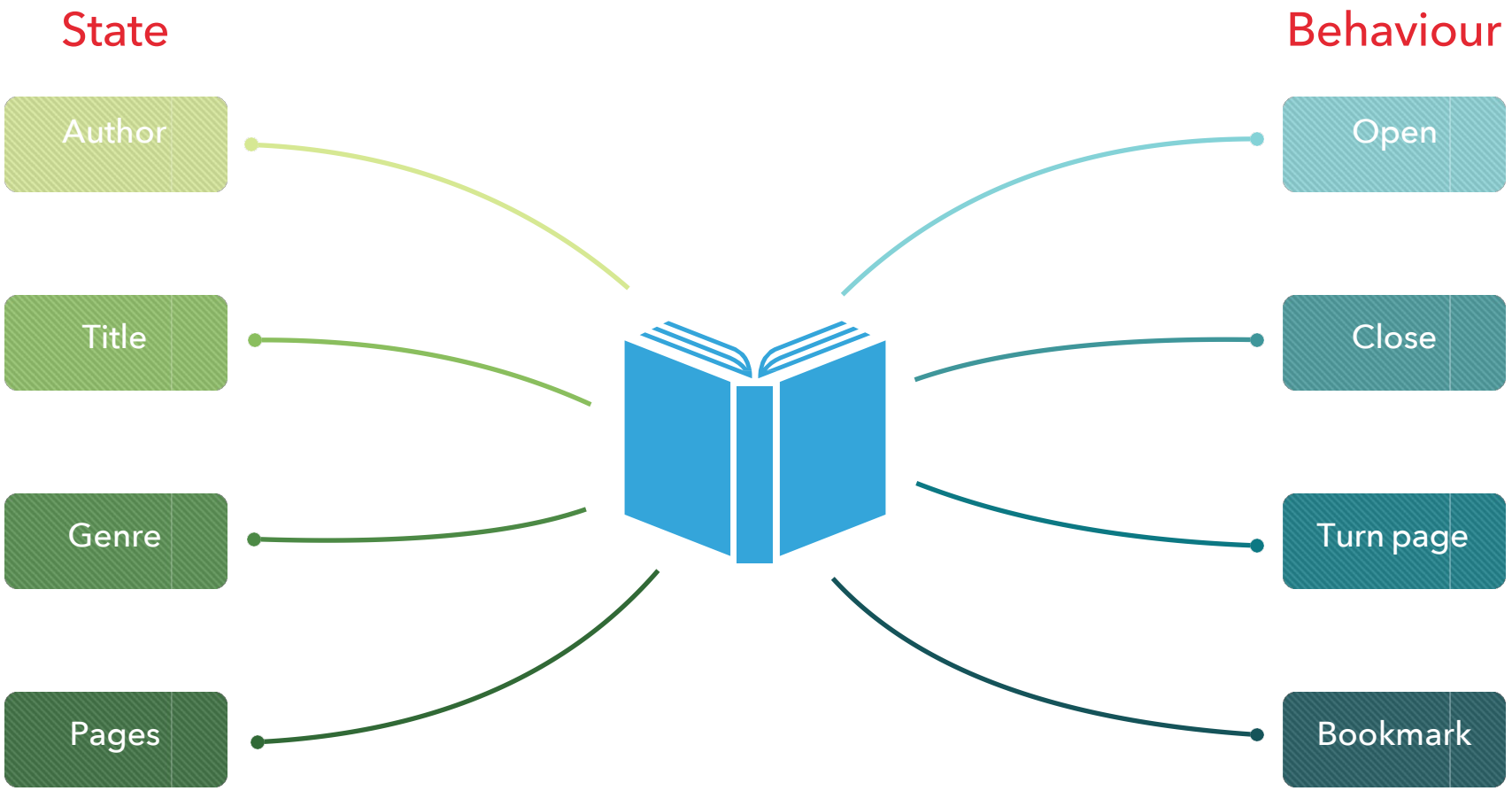
CHARACTERISTICS

- Every object share two characteristics:
  - Object has some sort of state
  - Object might have behaviour

OBJECT BREAKDOWN: DOG



OBJECT BREAKDOWN: BOOK



INTRODUCTION TO JAVA

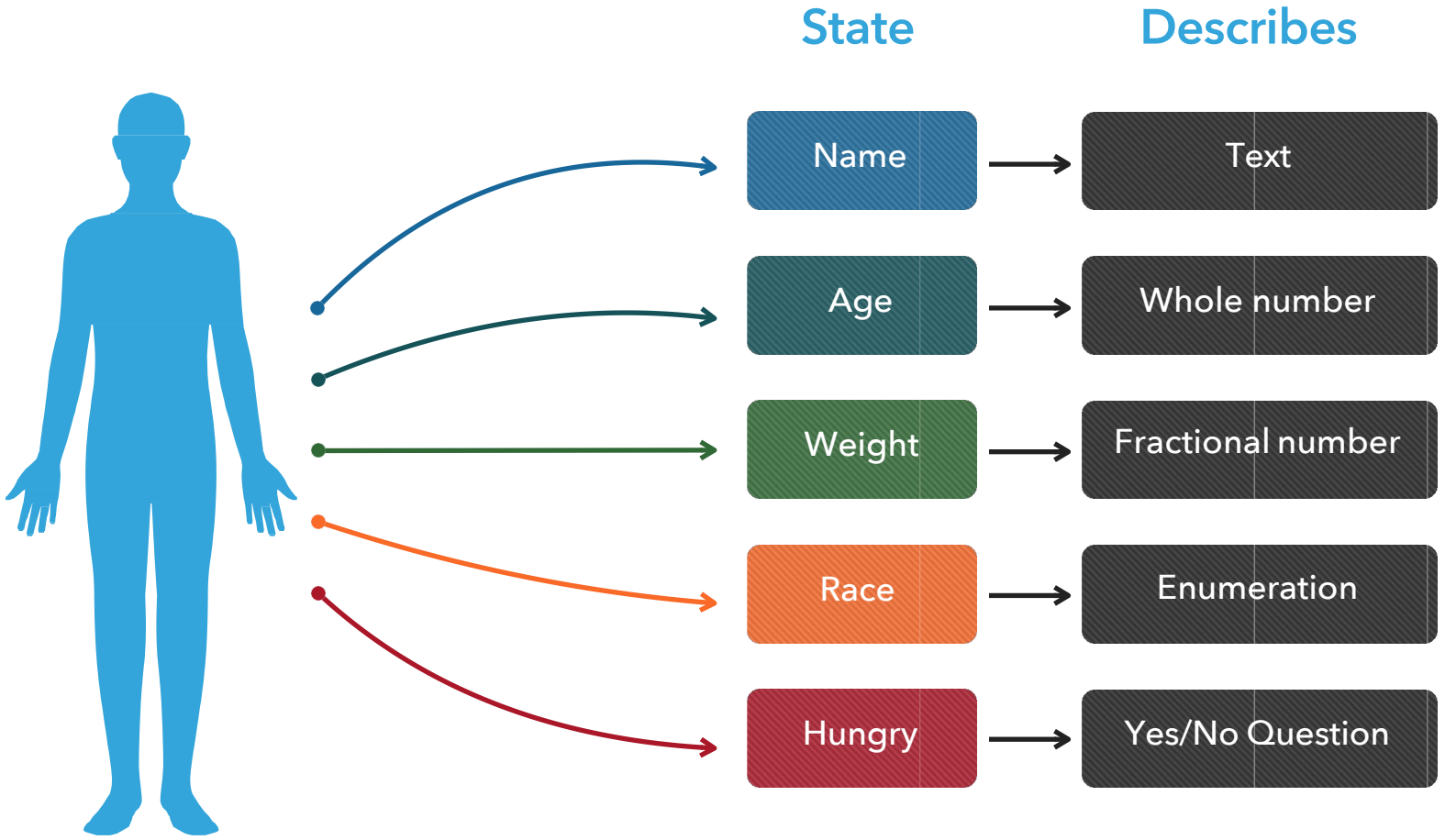
LESSON

VARIABLES
OVERVIEW

VARIABLES DEFINITION

- Variable is a named placeholder that
  - Stores data
  - Describes what type of data you can store
  - Describes size or amount of data it can store

VARIABLES RELATIONSHIP WITH OBJECTS



DATA TYPE CATEGORIES

- **Primitive** values
  - Integer: *byte, short, int, long* (e.g. 3, 7, 42, 2018)
  - Fractional: *float, double* (e.g. 3.1415, 2.7, 19.0)
  - Logical: *boolean* (true or false)
  - Textual: *char* (e.g. a, b, c, x, y, z)
- Reference values
  - Everything else

PRIMITIVE DATA TYPES IN DEPTH: INTEGER

Name	Assignable Values	Space
byte	-128 ... 127	1 byte
short	-32,768 ... 32,767	2 bytes
int	-2^{31} ... $2^{31}-1$	4 bytes
long	-2^{63} ... $2^{63}-1$	8 bytes

PRIMITIVE DATA TYPES IN DEPTH: FLOATING POINT

Name	Precision	Space
float	Single	4 bytes
double	Double	8 bytes

PRIMITIVE DATA TYPES IN DEPTH: LOGICAL

Name	Assignable Values	Space
boolean	true / false	1 byte

PRIMITIVE DATA TYPES IN DEPTH: TEXTUAL

Name	Assignable Values	Space
char (unicode)	0 ('\u0000') ... 65535 ('\uffff')	2 bytes

DEFINING VARIABLES IN JAVA CODE

VARIABLE DECLARATION IN JAVA: SYNTAX

- Variable declaration without value assignment

```
type name;
```

- Variable declaration **with** value assignment

```
type name = value;
```

VARIABLE DECLARATION IN JAVA: EXAMPLE

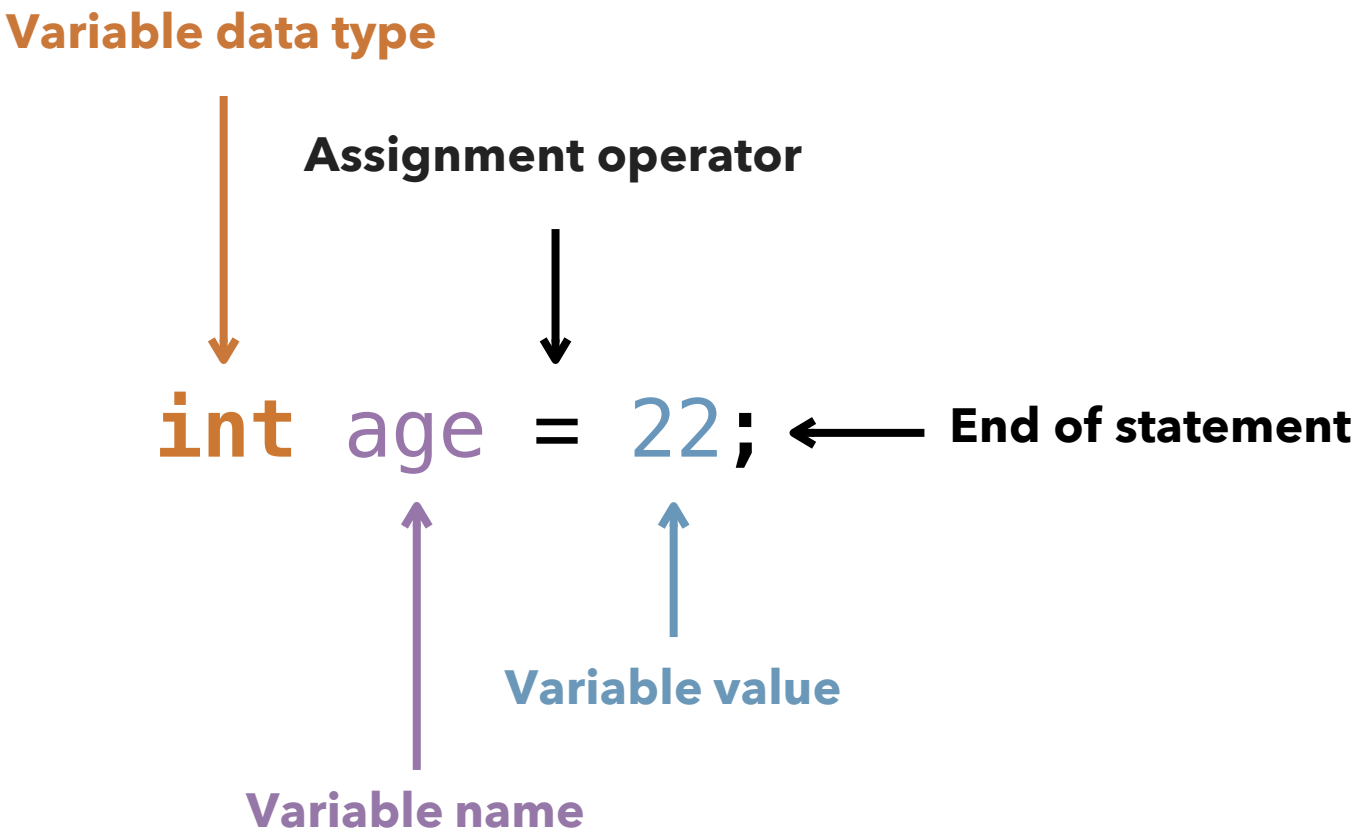
- Variable declaration without value assignment

```
int age;
```

- Variable declaration **with** value assignment

```
int age = 22;
```

VARIABLE DECLARATION BREAKDOWN



MORE EXAMPLES

```
byte numberOfWheels = 4;
short selfEsteem = -1;
int studentsGraduated = 1001;
long height = 80;
float pie = 3.14f;
double weight = 70.5;
boolean hungry = true;
char lastLetterOfTheAlphabet = 'Z';
```

NAMING RULES

- Any variable is allowed to **start** with
  - Letters (A-Z)
  - Special characters ('$' - dollar, '_' - underscore)
- Any variable name is allowed to **contain**
  - Alphanumeric characters (A-Z, 0-9)
  - Special characters ('$' - dollar, '_' - underscore)
- Variable name is **case-sensitive**
- Java language keywords[1] or reserved words **cannot** be used as variable name

[1] List of keywords can be found at https://docs.oracle.com/javase/tutorial/java/nutsandbolts/_keywords.html

NAMING DOS

- Single-worded name should be **lowercase**
- Multi-worded name should
  - First word **lowercase**
  - **Subsequent words start with** capital letters
  - No intervening spaces or punctuation
- Explains the purpose of variable

NAMING DON'TS

- Starting variable name with $ or _ is highly discouraged
- Avoid using $ anywhere in the variable name

NAMING DOS AND DON'TS EXAMPLES

- Please, do
  - size, xCoordinate, skinColor, currentDayOfTheWeek
- Please, don't
  - _counter, $bankBalance, Timestamp, 7daysOfTheWeek, !variableName, *notPointer

ARITHMETIC OPERATORS

ARITHMETIC OPERATORS OVERVIEW

Operator	Operation
+	Addition
-	Subtraction
/	Division
*	Multiplication
%	Remainder

OPERATORS BREAKDOWN: ADDITION

Integer numbers

```
int a = 10;
int b = 30;
int result = a + b;
```

```
result == 40
```

Fractional numbers

```
double x = 1.5;
double y = 2.7;
double result = x + y;
```

```
result == 4.2
```

OPERATORS BREAKDOWN: SUBTRACTION

Integer numbers

```
int a = 30;
int b = 20;
int result = a - b;
```

```
result == 10
```

Fractional numbers

```
double x = 5.4;
double y = 1.6;
double result = x - y;
```

```
result == 3.8
```

OPERATORS BREAKDOWN: MULTIPLICATION

Integer numbers

```
int a = 2;
int b = 4;
int result = a * b;
```

```
result == 8
```

Fractional numbers

```
double x = 2.5;
double y = 6.4;
double result = x * y;
```

```
result == 16.0
```

OPERATORS BREAKDOWN: DIVISION

Integer numbers

```
int a = 10;
int b = 5;
int result = a / b;
```

```
result == 2
```

Fractional numbers

```
double x = 18.0;
double y = 4.8;
double result = x / y;
```

```
result == 3.75
```

OPERATORS BREAKDOWN: REMAINDER

Integer numbers

```
int a = 9;
int b = 6;
int result = a % b;
```

```
result == 3
```

Fractional numbers

```
double x = 10.0;
double y = 4.5;
double result = x % y;
```

```
result == 1.0
```

TRICKY QUESTIONS

Type for division result is integer?

```
int a = 10;
int b = 4;
int result = a / b;

---

result == ?
```

Type for division result is double?

```
int x = 10;
int y = 4;
double result = x / y;

---

result == ?
```

TYPE CONVERSION: CASTING

- Operations with widening result require explicit type conversion (cast)

```
int    x  =  10;
int    y  =  4;
double result = x / (double) y;

result == ?
```

COMMON MISTAKES AND PITFALLS OVERVIEW

OVERVIEW

1. Missing terminator sign ';'
2. Incorrect spelling
    1. Class name
    2. Package name
    3. Variable name
3. Code placement outside of the body
4. Missing quotes or misplacement

(1) MISSING TERMINATOR SIGN: THE CODE

```
public class ForgotSemicolonAgain {

    public static void main(String[] args) {
        System.out.println("Oops.. I did it again")
    }

}
```

(1) MISSING TERMINATOR SIGN: THE FIX

```
public class ForgotSemicolonAgain {

    public static void main(String[] args) {
        System.out.println("Oops.. I did it again");
    }

}
```

(2.1) BAD CLASS SPELLING: THE CODE

```
public class sizeMatters {

    public static void main(String[] args) {
        system.out.println("Sorry, it does");
    }

}
```

(2.1) BAD CLASS SPELLING: THE FIX

```
public class SizeMatters {

    public static void main(String[] args) {
        System.out.println("Sorry, it does");
    }

}
```

(2.2) BAD PACKAGE SPELLING: THE CODE

```
package lv.javajava.lessons.HOMEWork;
```

(2.2) BAD PACKAGE SPELLING: THE CODE

```
package lv.javajava.lessons.homework;
```

(3) CODE PLACEMENT OUTSIDE OF THE BODY: THE CODE

```
public class AttentionPlease {

    System.out.println("Hide and seek");
    public static void main(String[] args) {
    }

}
```

(3) CODE PLACEMENT OUTSIDE OF THE BODY: THE FIX

```
public class AttentionPlease {

    public static void main(String[] args) {
        System.out.println("Hide and seek");
    }

}
```

(4) QUOTES MISPLACEMENT: THE CODE

```
public class NoSleepNoFocus {

    public static void main(String[] args) {
        System.out.println(I wanna coffee);
        System.out.println("So bad);
    }

}
```

(4) QUOTES MISPLACEMENT: THE FIX

```
public class NoSleepNoFocus {

    public static void main(String[]args)  {
        System.out.println("I wanna coffee");
        System.out.println("Sobad");
    }

}
```