# An Introduction to Processing

## Variables, Data Types & Arithmetic Operators

Lecturer: Caio Fonseca

Waterford Institute *of* Technology
INSTITIÚID TEICNEOLAÍOCHTA PHORT LÁIRGE

Department of Computing and Mathematics
http://www.wit.ie/

# Introduction

- Data types are used in programs to create named locations in the computer's memory.
- These will contain values while a program is running.
- This process is known as **declaring.**
- These named locations are called **variables.**

# Topics List

- Variables.
- Assignment statement.
- Data Types.
- Java's Primitive Data Types
  - Whole numbers.
  - Decimal numbers.
  - Others.
- Arithmetic operators.

# Topics List

- Variables.
- Assignment statement.
- Data Types.
- Java's Primitive Data Types
  - Whole numbers.
  - Decimal numbers.
  - Others.
- Arithmetic operators.

# Variables

In Programming, variables:

- are created (defined) in your programs
- are used to store data (whose value can change over time)
- have a data type
- have a name
- are a VERY important programming concept

# Variable Names…

- Are case-sensitive.
- Begin with either:
  - a **letter (preferable)**,
  - the dollar sign "$", or
  - the underscore character "_".
- Can contain letters, digits, dollar signs, or underscore characters.
- Can be any length you choose.



https://docs.oracle.com/javase/tutorial/java/nutsandbolts/variables.html

# Variable Names…

- Must not be a **keyword or reserved word** e.g. int, void, while, etc.
- Cannot contain white spaces.
- Variables must not have the same name as a variables already declared (must be unique).
- Variables must be declared before they can be used.

# Variable Names should be Carefully Chosen

- Use full words instead of cryptic abbreviations e.g.
  - variables named age and salary are much more intuitive than abbreviated versions, such as a and s.

- If the name consists of:
  - only one word, spell that word in all lowercase letters e.g. name.
  - more than one word, capitalise the first letter of each subsequent word e.g. yourAge and weeklyWage.



https://docs.oracle.com/javase/tutorial/java/nutsandbolts/variables.html

# Topics List

- Variables.
- Assignment statement.
- Data Types.
- Java's Primitive Data Types
  - Whole numbers.
  - Decimal numbers.
  - Others.
- Arithmetic operators.

# Assignment Statement

- Values are stored in variables via assignment statements:

| Syntax | `variable = expression;` |
|---|---|
| Example | `yourAge = 25;` |

- A variable stores a **single** value, so any previous value is lost.
- Assignment statements work by taking the value of what appears on the right-hand side of the operator and copying that value into a variable on the left-hand side.

# Assignment Statement

- Assignments allow values to be put into variables.
- They are written in Java with the use of the equality symbol (=).
- This symbol is known as **the assignment operator**.
- Simple assignments take the following form:
  **variableName = value;**

```
age = 21;
weeklySalary = 565.62;
```

# Topics List

- Variables.
- Assignment statement.
- Data Types.
- Java's Primitive Data Types
  - Whole numbers.
  - Decimal numbers.
  - Others.
- Arithmetic operators.

# Data Types

- In Java, when we define a variable, we **have** to give it a data type.

- The data type defines the **kinds of values** (data) that can be stored in the variable e.g.
    - - 456
    - 2
    - 45.7897
    - I Love Programming
    - S
    - true

- The data type also determines the operations that may be performed on it.

# Data Types

- Java uses two kinds of data types:
  - Primitive types
  - Object types

- We are only looking at Primitive types now; we will cover Object types in the next module.

# Topics List

- Variables.
- Assignment statement.
- Data Types.
- Java's Primitive Data Types
  - Whole numbers.
  - Decimal numbers.
  - Others.
- Arithmetic operators.

# Java's Primitive Data Types

- Java programming language supports eight primitive data types.

- A primitive type is predefined by the language and is named by a **reserved keyword**.

- A primitive type is highlighted red when it is typed into the PDE e.g.

```
int age;
boolean flag;
float number;
double number1;
```

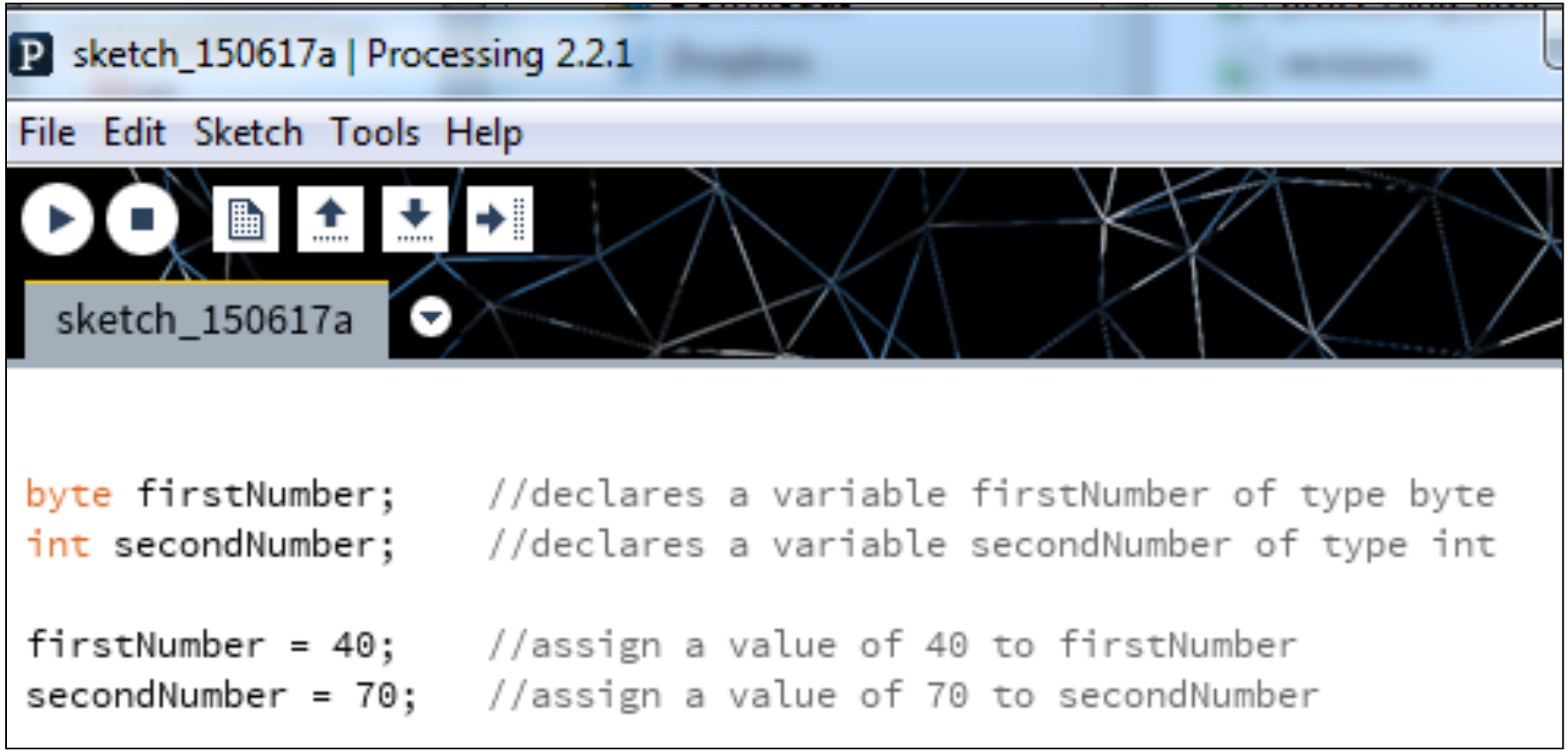
# Topics List

- Variables.
- Assignment statement.
- Data Types.
- Java's Primitive Data Types
  - Whole numbers.
  - Decimal numbers.
  - Others.
- Arithmetic operators.

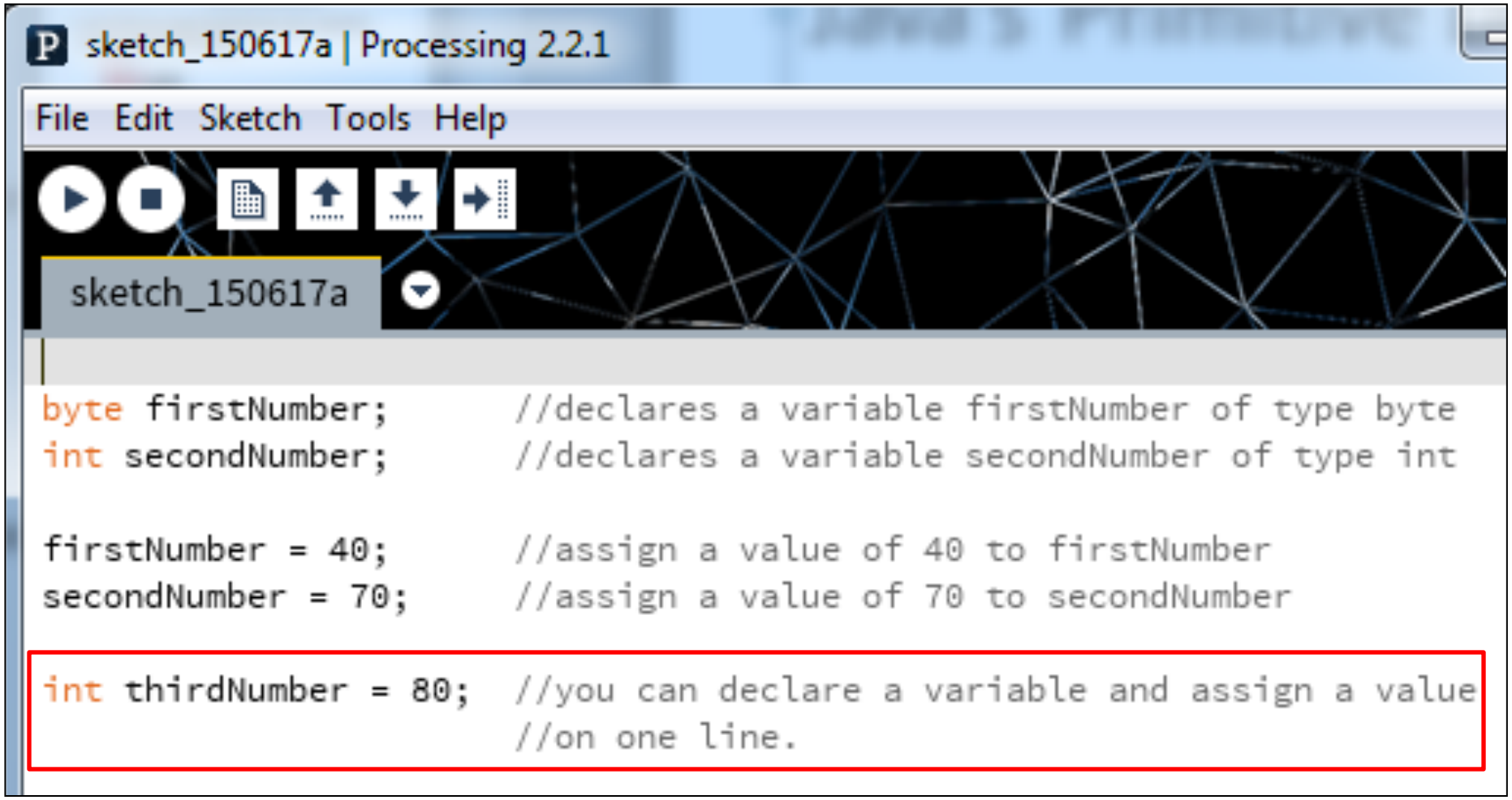
# Java's Primitive Data Types (Whole Numbers)

| Type | Byte-size | Minimum value (inclusive) | Maximum value (inclusive) | Typical Use |
|---|---|---|---|---|
| byte | 8-bit | -128 | 127 | Useful in applications where memory savings apply. |
| short | 16-bit | -32,768 | 32,767 | |
| int | 32-bit | -2,147,483,648 | 2,147,483,647 | Default choice. |
| long | 64-bit | -9,223,372,036,854,775,808 | 9,223,372,036,854,775,807 | Used when you need a data type with a range of values larger than that provided by int. |

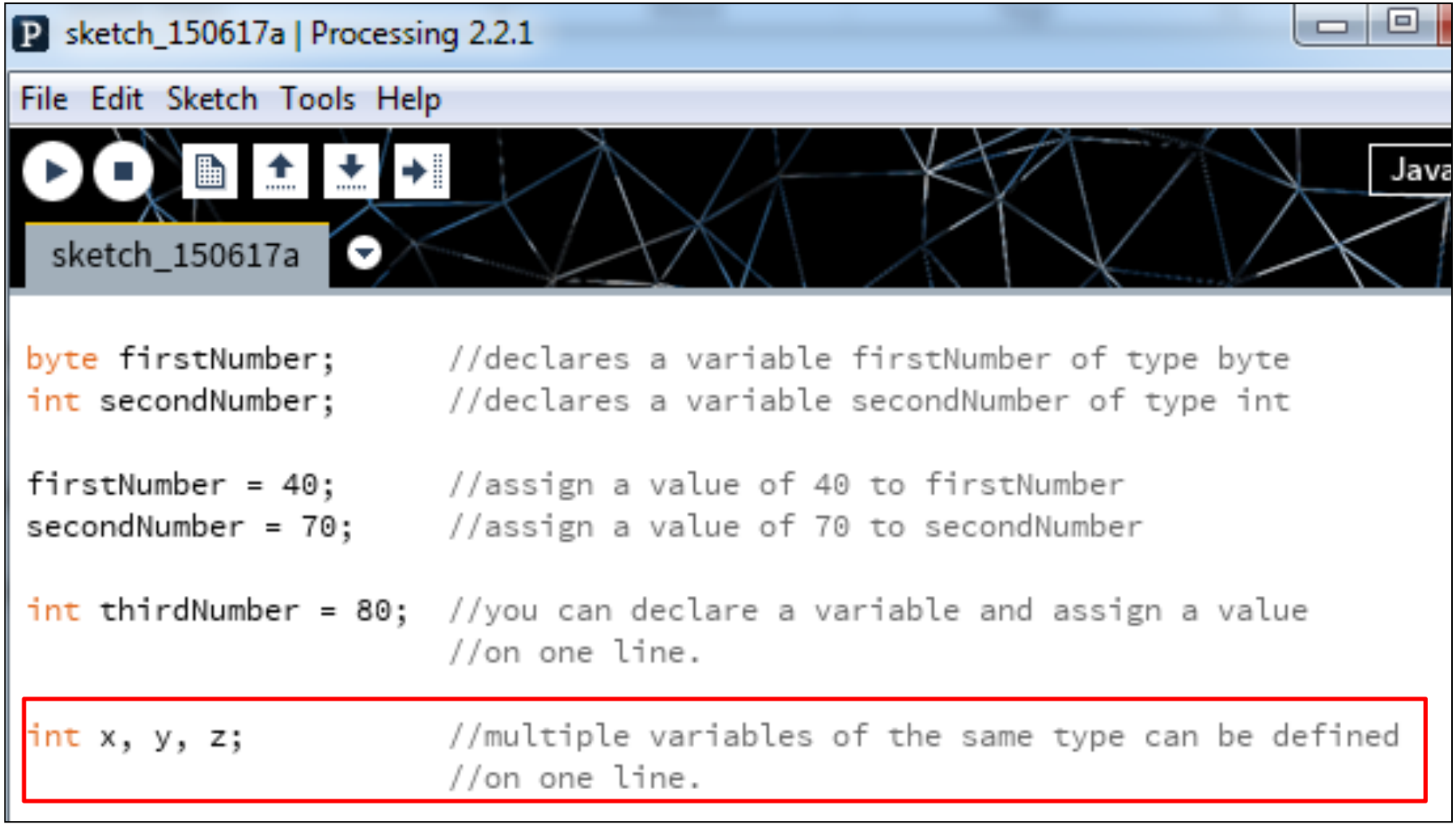
# Declaring Variables of a Specific Type



# Declaring Variables of a Specific Type



# Declaring Variables of a Specific Type



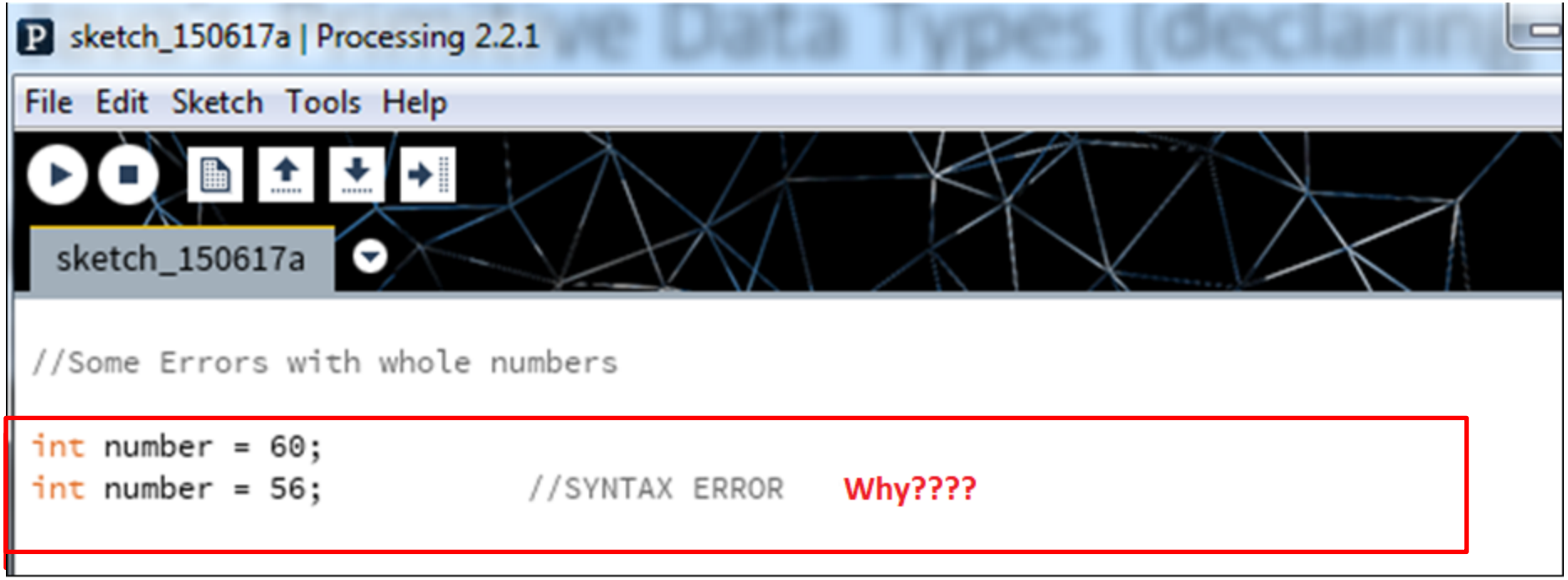
# Declaring Variables - Some Errors


Data types are case sensitive.

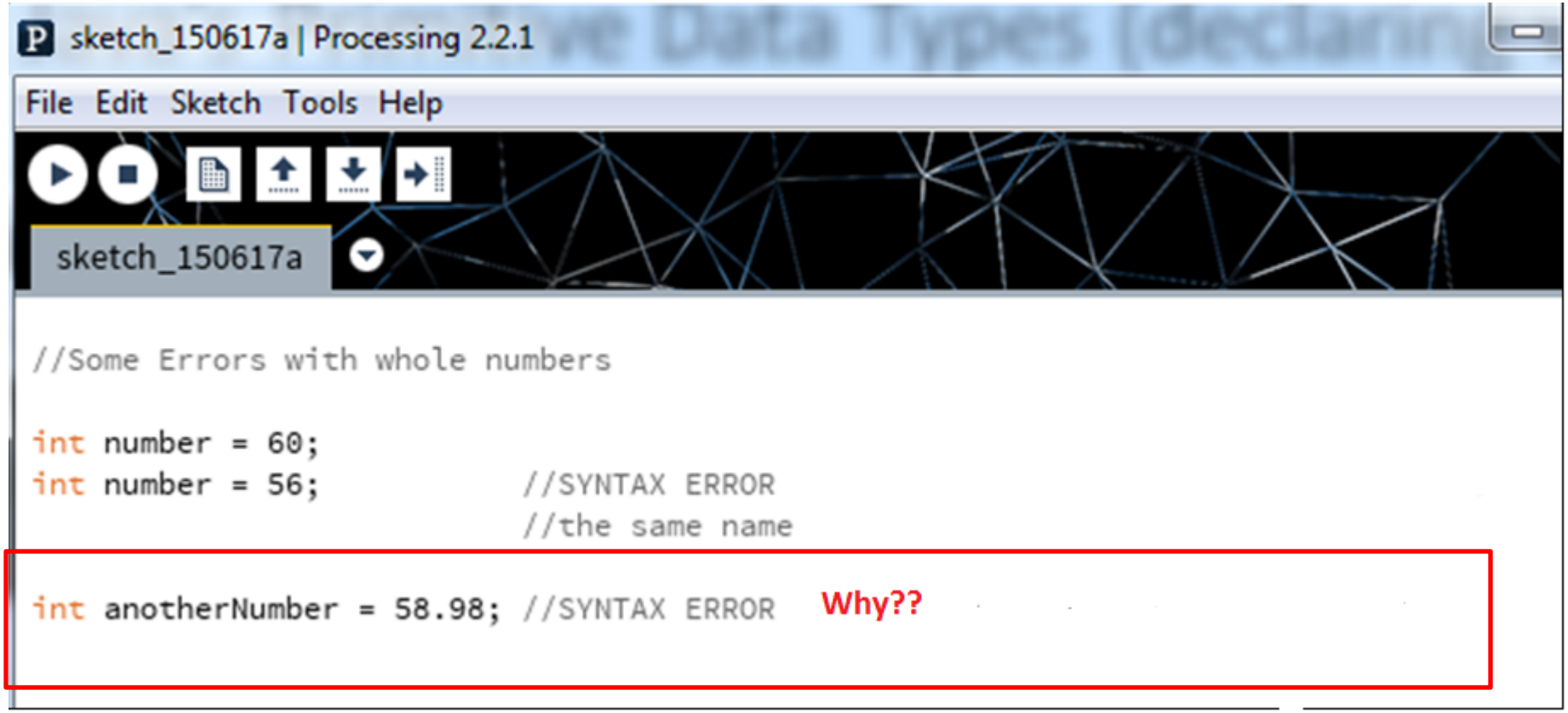
Int is not valid.

int is valid.

# Declaring Variables - Some Errors

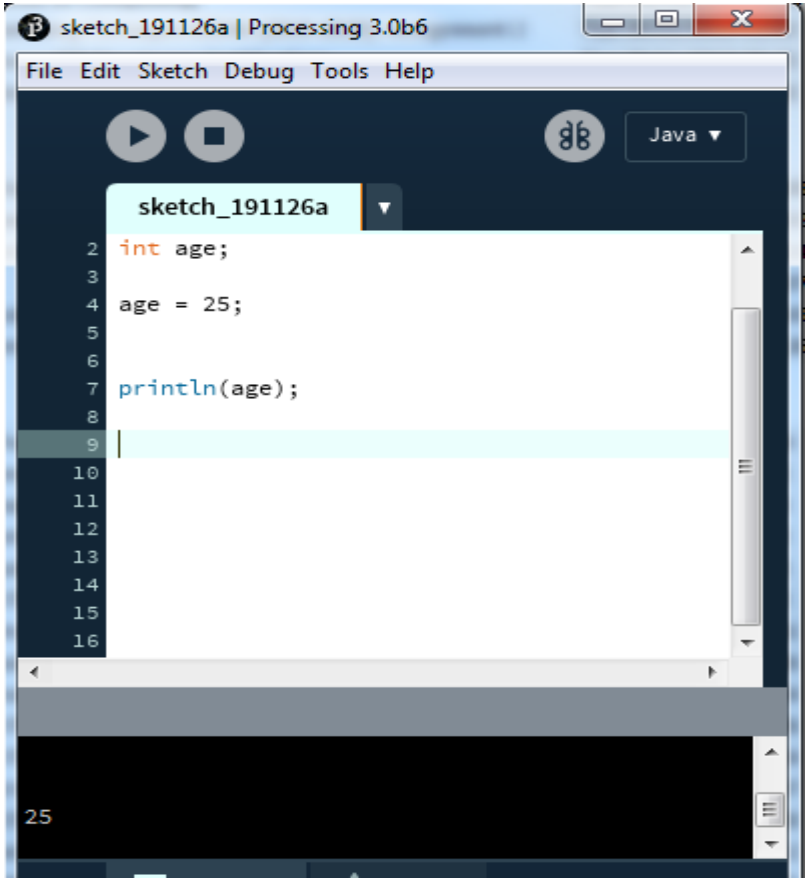


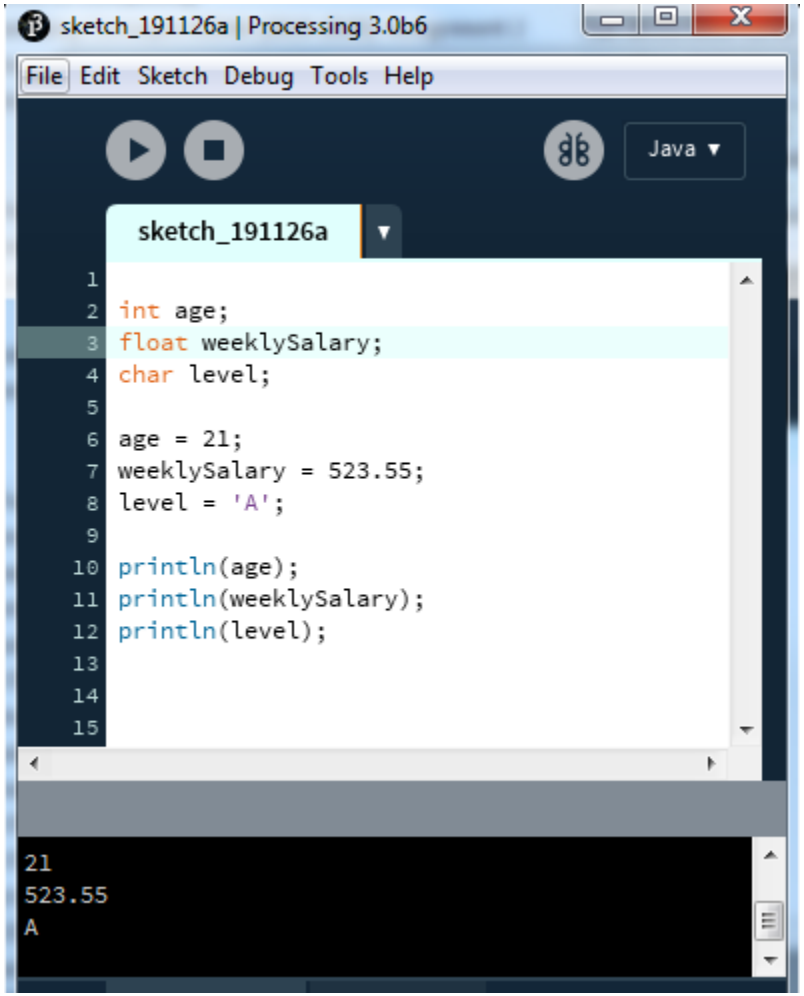
# Declaring Variables - Some Errors



# Printing Variables

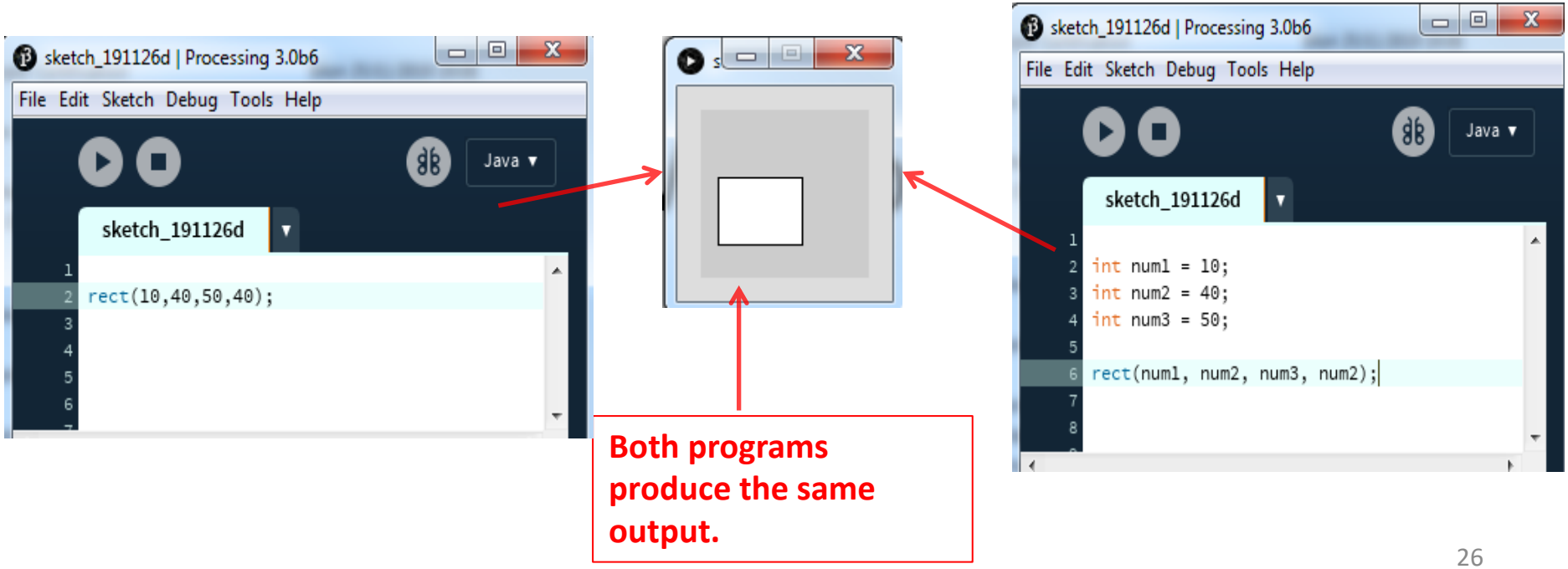
- You can print out the value of a variable to the screen using print or println.



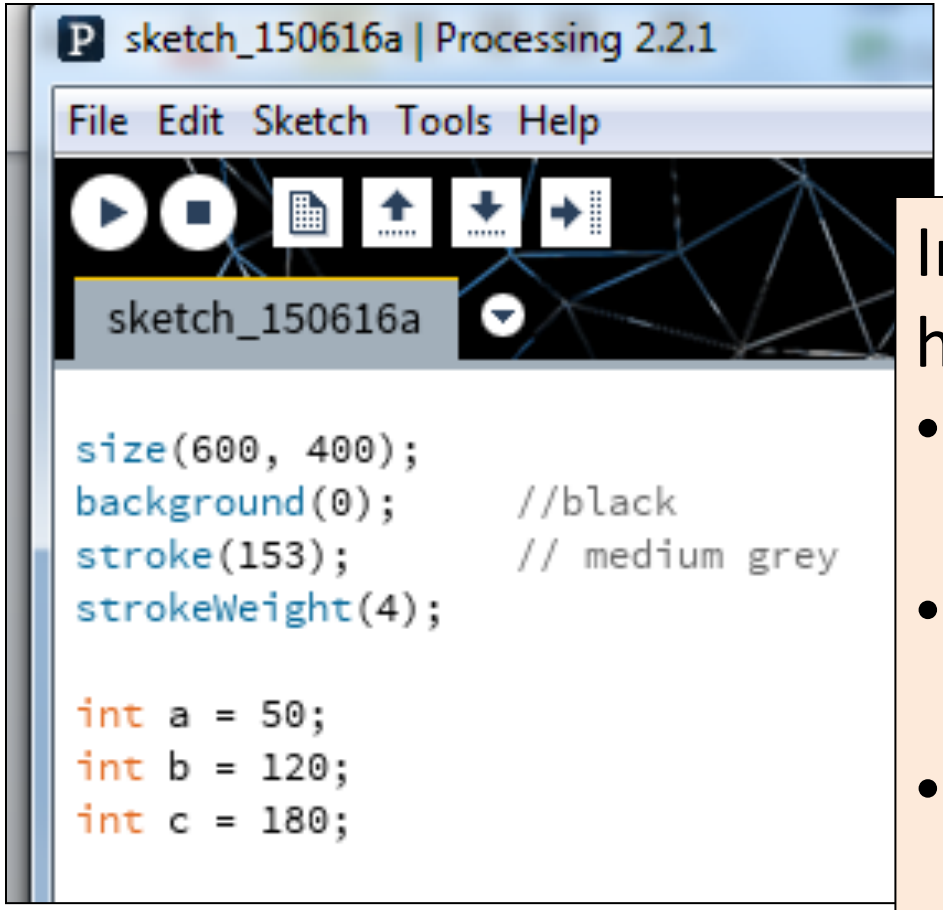



# Java's Primitive Data Types: int Example 1

- We can now use variables instead of hard values into our drawing.
- For example when drawing a rectangle we have used rect(10,40,50,40) we can now use variables.



# Java's Primitive Data Types: int Example 2




In this example, we have:

- defined three variables (a, b and c)
- that can hold whole numbers (int).
- and are set with a starting value.

Based on the Processing Example: Basics → Data → Variables

# Java's Primitive Data Types: int Example 2

sketch_150616a | Processing 2.2.1

File Edit Sketch Tools Help

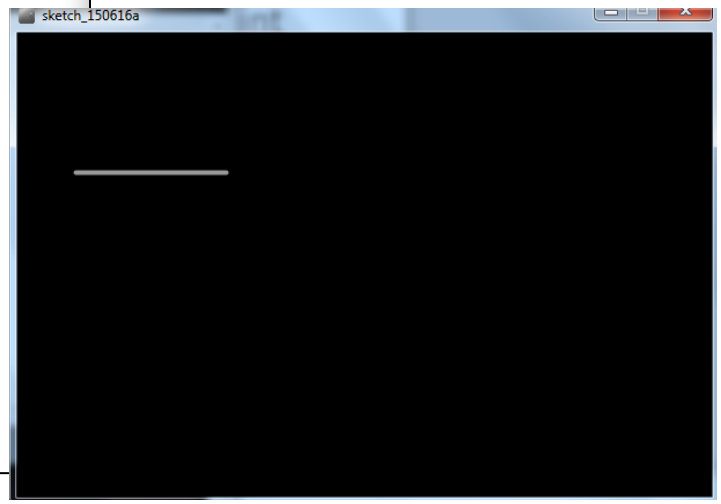
sketch_150616a

```
size(600, 400);
background(0);      //black
stroke(153);        // medium grey
strokeWeight(4);

int a = 50;
int b = 120;
int c = 180;

line(a, b, c, b);
```

We can pass the defined variables as values to functions.




Based on the Processing Example: Basics → Data → Variables

# Java's Primitive Data Types: int Example 2

```
size(600, 400);
background(0);      //black
stroke(153);        // medium grey
strokeWeight(4);

int a = 50;
int b = 120;
int c = 180;

line(a, b, c, b);
```

Could we have used the byte data type instead of int? Why?

| Type | Minimum value (inclusive) | Maximum value (inclusive) |
|---|---|---|
| byte | -128 | 127 |
| short | -32,768 | 32,767 |
| int | -2,147,483,648 | 2,147,483,647 |
| long | -9,223,372,036,854,775,808 | 9,223,372,036,854,775,807 |

Based on the Processing Example: Basics → Data → Variables

# Topics List

- Variables.
- Assignment statement.
- Data Types.
- Java's Primitive Data Types
  - Whole numbers.
  - Decimal numbers.
  - Others.
- Arithmetic operators.

# Java's Primitive Data Types (decimal numbers)

| Type | Byte-size | Minimum value (inclusive) | Maximum value (inclusive) | Typical Use |
|---|---|---|---|---|
| float | 32-bit | *Beyond the scope of this lecture .*<br><br>*There is also a loss of precision in this data-type that we will cover in later lectures.* | | Useful in applications where memory savings apply.<br><br>Default choice when using **Processing**. |
| double | 64-bit | | | Default choice when programming **Java apps.** |

# Java's Primitive Data Types: float example

sketch_150618a | Processing 2.2.1

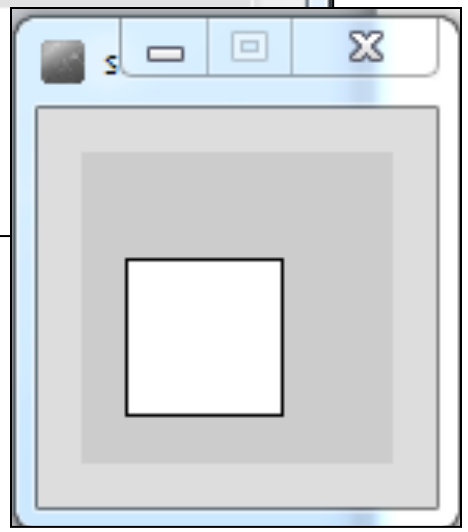
File Edit Sketch Tools Help

Java

sketch_150618a

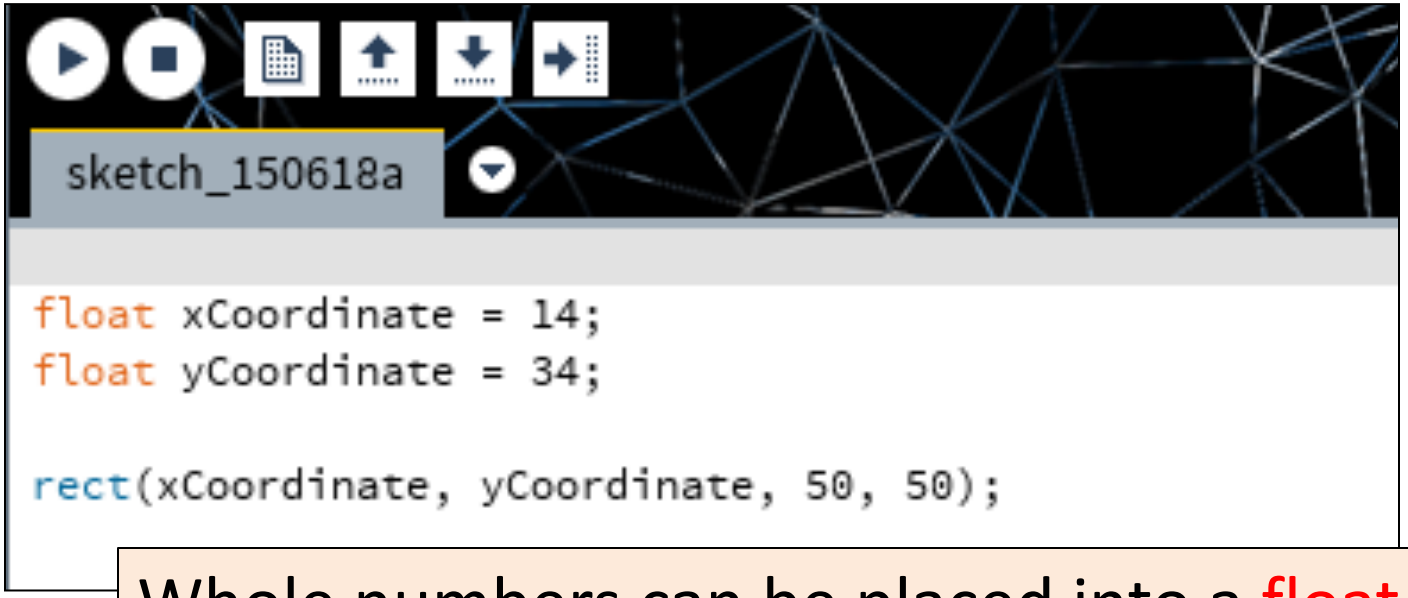
```
float xCoordinate = 14.65;
float yCoordinate = 34.43;

rect(xCoordinate, yCoordinate, 50, 50);
```



We can pass the defined variables as values to functions.

# Java's Primitive Data Types: float example




Whole numbers can be placed into a float variable.

Q: Why?

# Passing variables as arguments: some errors

sketch_150618a

```
double xCoordinate = 14.65;
double yCoordinate = 34.43;

rect(xCoordinate, yCoordinate, 50, 50);
```

The method rect(float, float, float, float) in the type PApplet is not applicable for the arguments (double, double, int, int)

We changed the data type of our variables from float to double.

When we try to run the code, we get this syntax error.

What's wrong?

# Passing variables as arguments: some errors

From: https://processing.org/reference/rect_.html

**Syntax** `rect(a, b, c, d)`

| **Parameters** | | |
|---|---|---|
| | a | float: x-coordinate of the rectangle by default |
| | b | float: y-coordinate of the rectangle by default |
| | c | float: width of the rectangle by default |
| | d | float: height of the rectangle by default |

```
double xCoordinate = 14.65;
double yCoordinate = 34.43;

rect(xCoordinate, yCoordinate, 50, 50);
```

The method rect(float, float, float, float) in the type PApplet is not applicable for the arguments (double, double, int, int)

# Topics List

- Variables.
- Assignment statement.
- Data Types.
- Java's Primitive Data Types
  - Whole numbers.
  - Decimal numbers.
  - Others.
- Arithmetic operators.

# Java's Primitive Data Types (others)

| Type | Byte-size | Minimum value (inclusive) | Maximum value (inclusive) | Typical Use |
|---|---|---|---|---|
| char | 16-bit | '\u0000' (or 0) | '\uffff' (or 65,535). | Represents a Unicode character. |
| boolean | 1-bit | n/a | | Holds either **true** or **false** and is typically used as a flag. |

http://en.wikipedia.org/wiki/List_of_Unicode_characters

# Java's Primitive Data Types (others)

- char is the data type to hold any single character from the keyboard
  - i.e. a or B or € or * or 1 or 2

- We declare this type as follows:
  - char grade;

- We assign a value to a char using single quotes
  - grade = 'A';
  - grade = 'a';
  - grade = '1';

# Java's Primitive Data Types (others)

- boolean is used to represent a value of true or false.
- We declare this type as follows:
  - boolean isTrue;
- We assign a value to a boolean as follows:
  - isTrue = true;
  - isTrue = false;
- We often use boolean values to test certain conditions (we will look at this in greater detail in later sections).

# Java's Primitive Data Types (default values)

| Data Type | Default Value |
|---|---|
| byte | 0 |
| short | 0 |
| int | 0 |
| long | 0L |
| float | 0.0f |
| double | 0.0d |
| char | '\u0000' |
| boolean | false |

http://docs.oracle.com/javase/tutorial/java/nutsandbolts/datatypes.html

# Topics List

- Variables.
- Assignment statement.
- Data Types.
- Java's Primitive Data Types
  - Whole numbers.
  - Decimal numbers.
  - Others.
- Arithmetic operators.

# Variables and Arithmetic Operators

| The arithmetic operators of Java | |
|---|---|
| **Operation** | **Java operator** |
| addition | + |
| subtraction | - |
| multiplication | * |
| division | / |
| remainder | % |

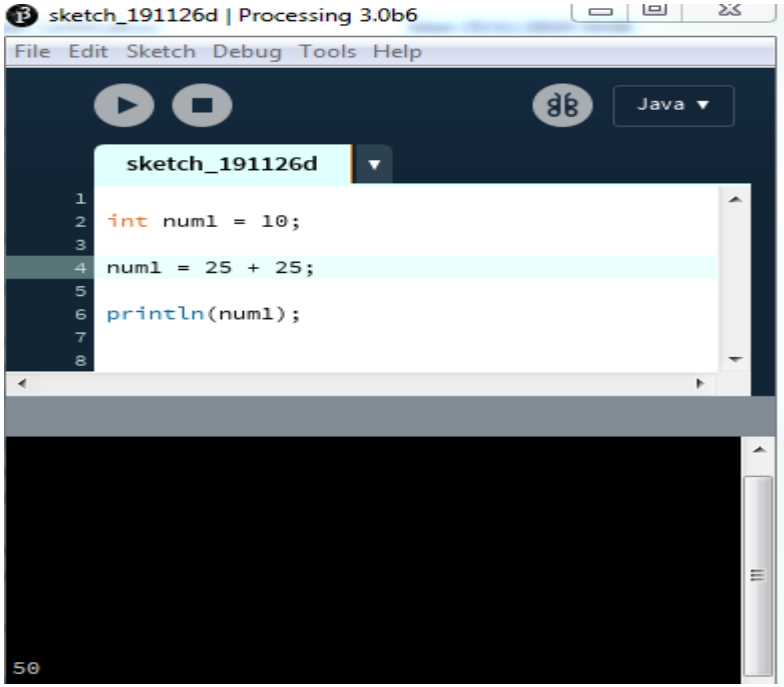
Where did we see this??

We will look at this in more detail later.

# Variables and Arithmetic Operators

- These are used when you want to do some calculation to number or to your variables. These are very similar to how you calculate in maths.

```
int num1;
num1 = 25 + 25;
println(num1);
```




- Terms on the right-hand side of assignment operators (like 25 + 25) are referred to as **expressions**.

# Variables and Arithmetic Operators

- When doing any calculations in Java you must follow the order of evaluation.
  - Brackets
  - Multiplication
  - Division
  - Addition
  - Subtraction

- (**B**rackets and **M**y **D**ear **A**unt **S**ally)(BOMDAS)

# Variables and Arithmetic Operators

- Example 1

  int answer;

  answer = 20 + 20 / 10 + 5;

- Example 2

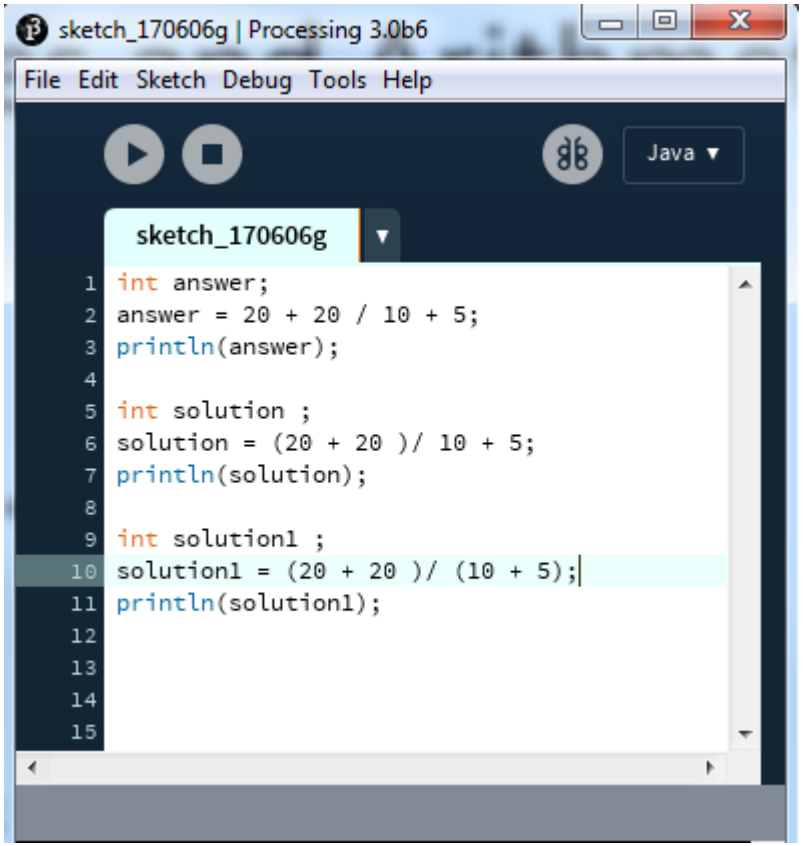
  int solution ;

  solution = (20 + 20 )/ 10 + 5;

- Example 3

  int solution1 ;

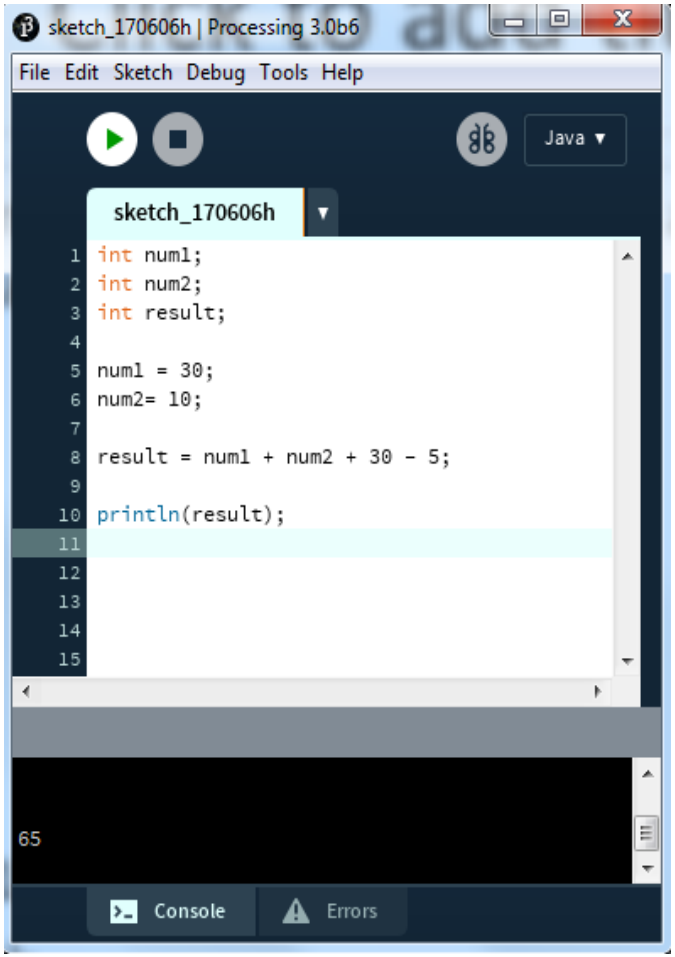
  solution1 = (20 + 20 )/ (10 + 5);



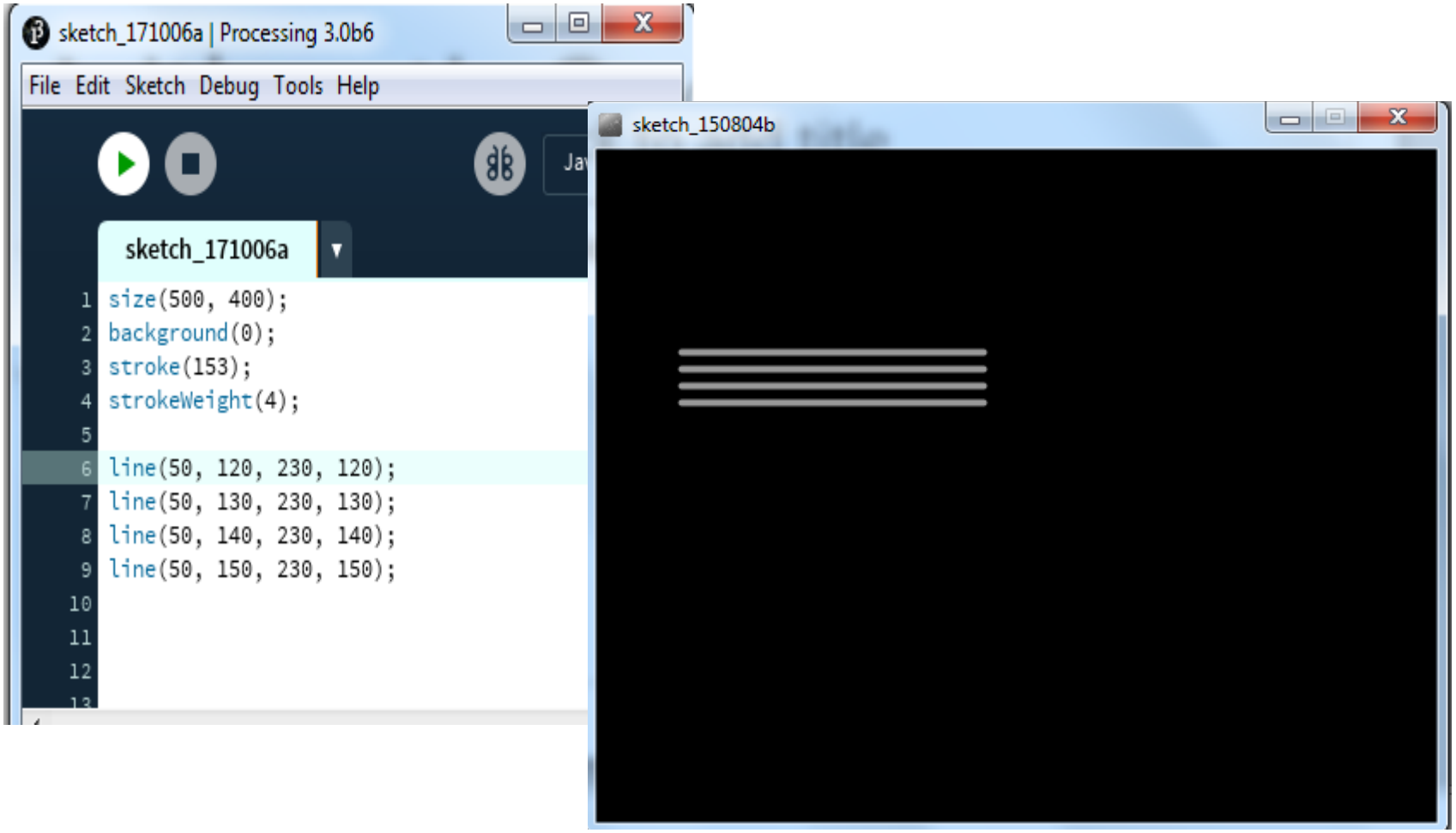
# Variables and Arithmetic Operators

- The expression on the right-hand side of an assignment statement can itself contain variable names;

```
int num1;
int num2;
int result;
num1 = 30;
num2= 10;
result = num1 + num2 + 30 - 5;
println(result);
```



# Arithmetic Operators: Example 1



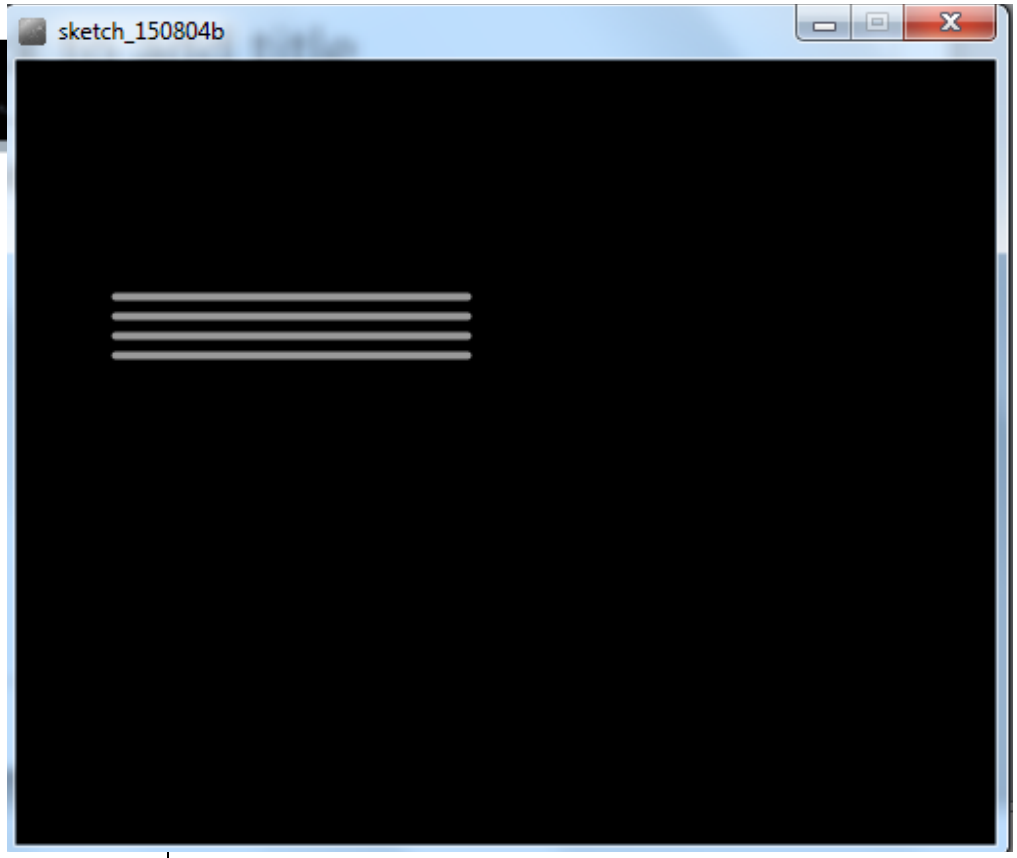
# Arithmetic Operators: Example 1

sketch_150804b

```
size(500, 400);
background(0);
stroke(153);
strokeWeight(4);

int a = 50;
int b = 120;
int c = 180;

line(a, b, a+c, b);
line(a, b+10, a+c, b+10);
line(a, b+20, a+c, b+20);
line(a, b+30, a+c, b+30);
```




Based on the Processing Example: Basics → Data → Variables

# Arithmetic Operators: Example 2

sketch_150804b

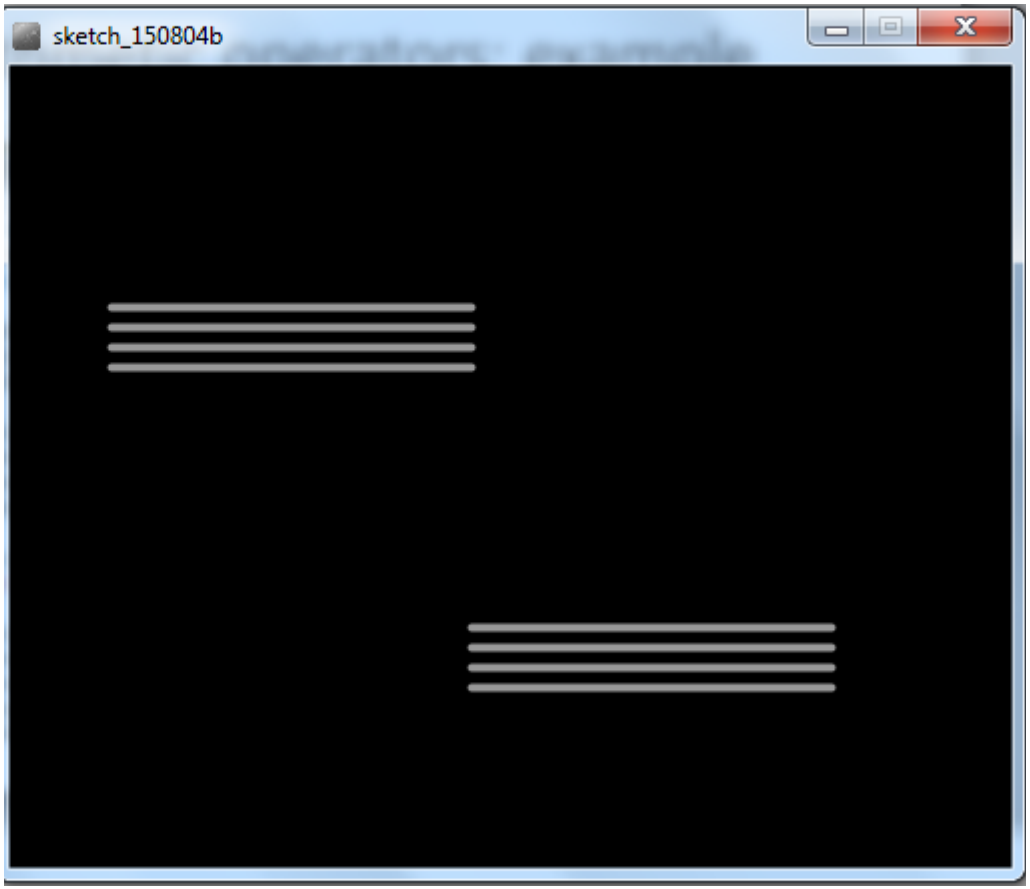
```
size(500, 400);
background(0);
stroke(153);
strokeWeight(4);

int a = 50;
int b = 120;
int c = 180;

line(a, b, a+c, b);
line(a, b+10, a+c, b+10);
line(a, b+20, a+c, b+20);
line(a, b+30, a+c, b+30);

a = a + c;
b = height-b;

line(a, b, a+c, b);
line(a, b+10, a+c, b+10);
line(a, b+20, a+c, b+20);
line(a, b+30, a+c, b+30);
```




Based on the Processing Example: Basics → Data → Variables

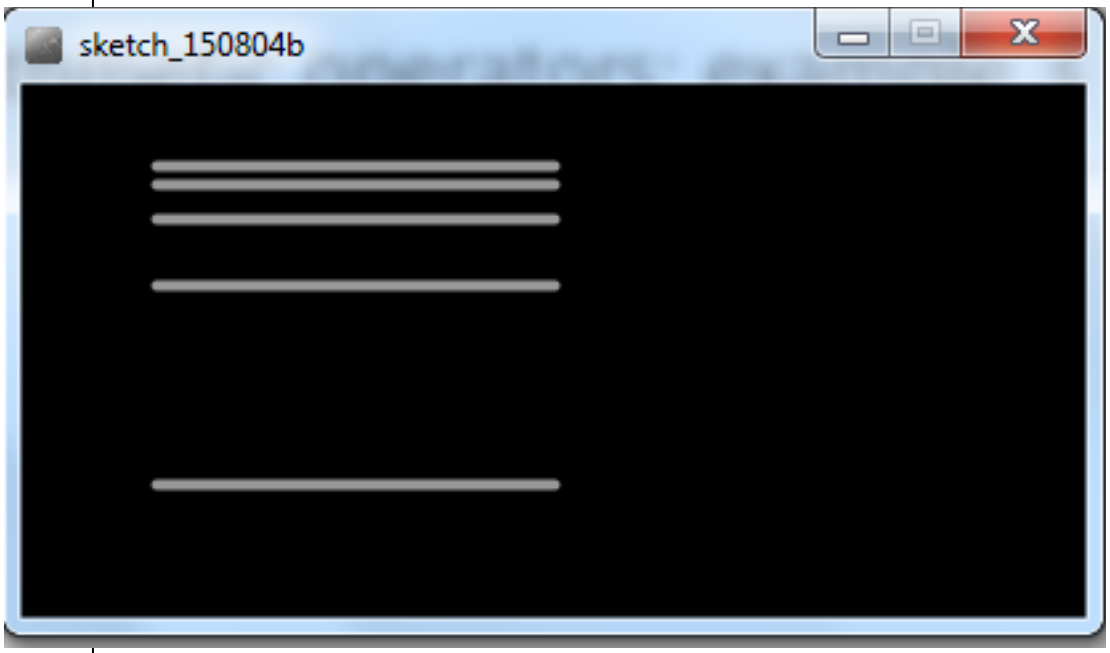
# Arithmetic Operators: Example 3

sketch_150804b

```
size(400, 200);
background(0);
stroke(153);
strokeWeight(4);

int a = 50;
int b = 1500;
int c = 4;

line(a, b/10, a*c, b/10);
line(a, b/20, a*c, b/20);
line(a, b/30, a*c, b/30);
line(a, b/40, a*c, b/40);
line(a, b/50, a*c, b/50);
```

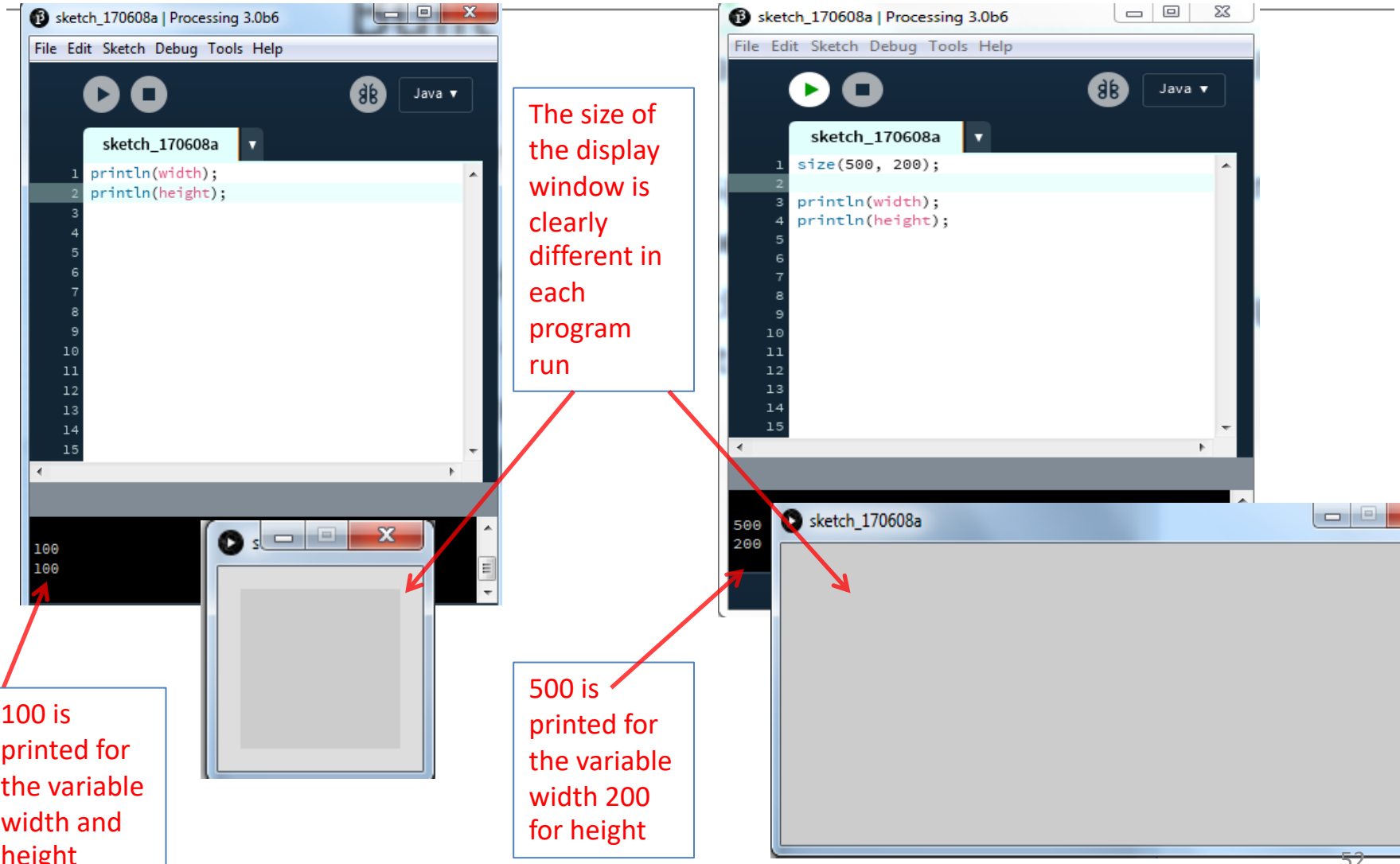



Based on the Processing Example: Basics → Data → Variables

# Built in Variables

- Processing has built in variables for storing commonly used data.
- Width and height of the display window are stored in variable called **width** and **height**.
- If the program does not include size () width and height both are automatically set to 100.

# Built in Variables



# A Special Shorthand

- If a variable x has been declared as an int, then the instruction for incrementing x (adding 1) would be:

```
x = x + 1;
```

- This is so common that there is a special shorthand for this instruction:

```
x++;
```

- The '++' is therefore known as the increment operator;

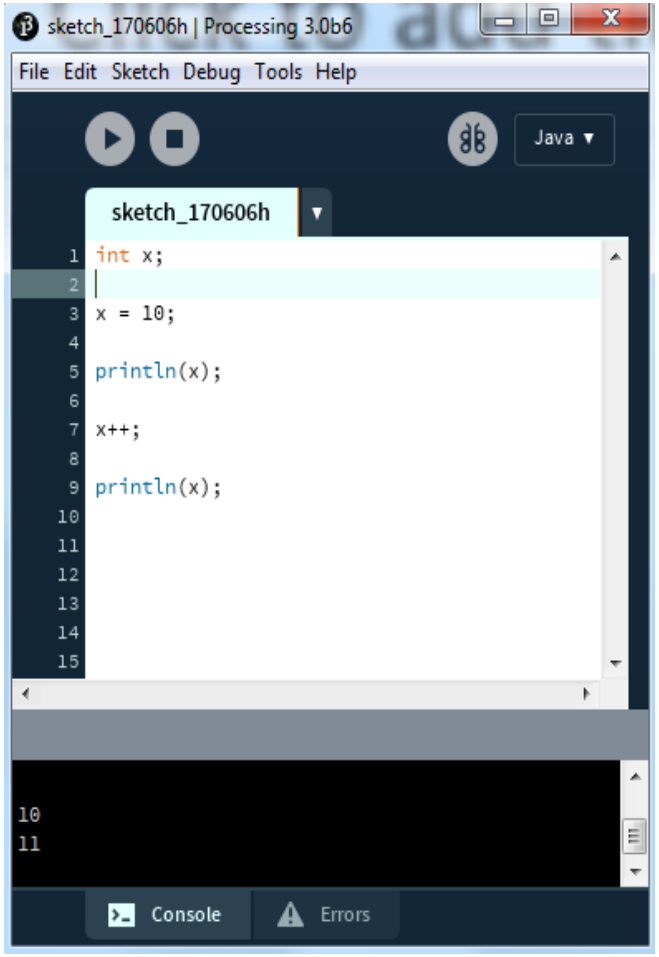
- Similarly there exists a decrement operator, '--':

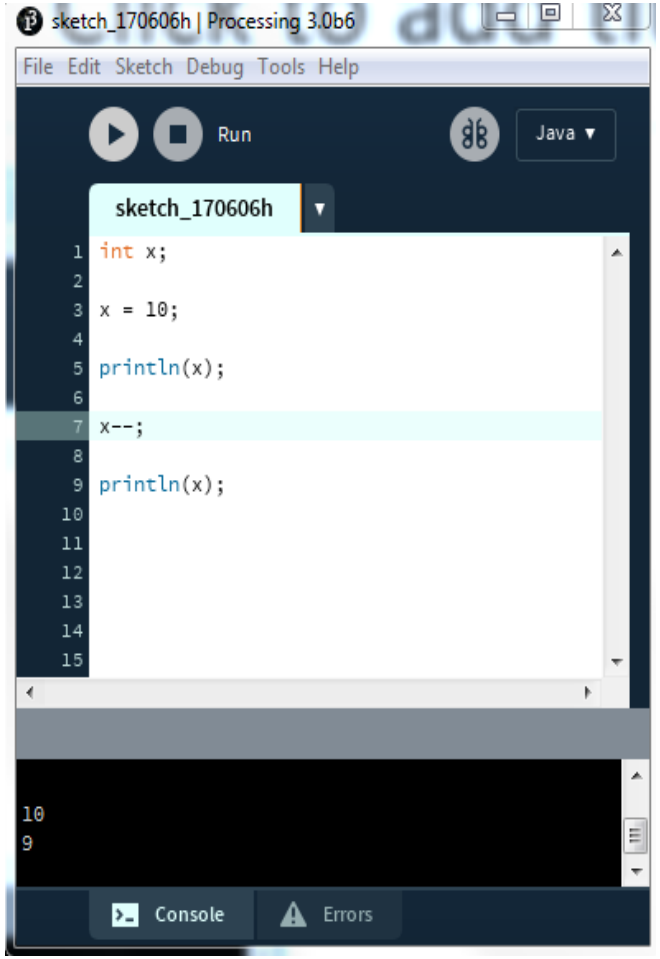
```
x--;
```

is shorthand for:

```
x = x - 1;
```

# A Special Shorthand



# Questions?




Produced by:

Dr. Siobhán Drohan
Mairead Meagher
Sinéad Walsh

Waterford Institute *of* Technology
INSTITIÚID TEICNEOLAÍOCHTA PHORT LÁIRGE

Department of Computing and Mathematics
http://www.wit.ie/