

JAVA / LESSON 02 OF 40

Variable Boxes in Your Inventory

Give the data a name and track a small adventure

Java 11+ | 90 minutes



Create a changing console player status bar

The player name in the welcome screen is always written as Alex. If another player comes, every name must be found again. Variables give us a better way: put data in a named location, read it, and update it. What we need to understand in this lesson is not only "how to write variables", but also "what exactly does it save at this moment".

Before you start: It is recommended to complete Lesson 1 first

Read the full lesson and copy the code:

<https://teacher-ma.de/minecraft-java-lesson.php?lesson=2&lang=en>

Long code lines may wrap at the > margin marker. The marker is not part of the code.

In this lesson

- 01. tag helps us find the current value
- 02. Statement, initialization, reading
- 03. Assignment: Calculate the right side first, then write the left side
- 04. One output is a snapshot
- 05. Update health values along the timeline
- 06. Good names let the code explain itself
- 07. Variable troubleshooting: first distinguish what kind of problem it is
- 08. Design a changing player status bar
- 09. Complete work and output verification
- 10. Continue practicing with predictions

01 / 10

tag helps us find the current value



Storage rooms rely on labels to find items; programs rely on variable names to read data.

Steve put the bread into the unlabeled box, but opened the cobblestone box during dinner time. Variable names solve a similar problem: letting the program find the data it needs. It is not the data itself, but the name used to access the data.

Let's start with two types: `int` holds integers, and `String` holds text. Health points and item quantities are suitable for integers; player names are suitable for text. More types will be discussed in the next lesson.

The life recovery and item changes in this lesson are practice rules. The output is a snapshot of the state in the console and does not automatically turn into a game interface, nor is it automatically saved after the program ends.

02 / 10

Statement, initialization, reading

Read `int health = 20;`

Element	Explanation
<code>int</code>	Type · Save integer
<code>health</code>	Variable name · Find the current value
<code>=</code>	Assignment number · Write the value on the right
<code>20</code>	Initial value · The value of the first deposit
<code>;</code>	Semicolon · End this statement

Complete program · `PlayerStatus.java`

```
public class PlayerStatus {  
    public static void main(String[] args) {  
        String playerName = "Alex";  
        int health = 20;  
        int bread = 3;  
        System.out.println(playerName + " Life=" + health);  
        System.out.println("Bread.=" + bread);  
    }  
}
```

Output

```
Alex Life=20  
Bread.=3
```

When `health` is not quoted, the value of the variable is read; after `"health"` is quoted, it is just a fixed text. Change the player name to Steve and the bread to 5. Guess the two lines of output first and then run it.

Assignment: Calculate the right side first, then write the left side



Assignment is like a transfer: prepare the result first, and then put it back at the designated location.

three actions of bread = bread - 1

Step	Explanation
1. Read Old Values	bread is 3
2. Calculate the right side	3 - 1 gets 2
3. Write back to the left	bread becomes 2

Snippet: place inside main and run separately

```
int bread = 3;
System.out.println("Before eating=" + bread);
bread = bread - 1;
System.out.println("After eating=" + bread);
int diamonds = 0;
diamonds = diamonds + 3;
System.out.println("diamond=" + diamonds);
```

Output

```
Before eating=3
After eating=2
diamond=3
```

= is an update, not a mathematical equation. Write `int bread = 3;` when creating it for the first time, and write `bread = 2;` when updating the same variable later. Do not declare it again in the same scope.

04 / 10

One output is a snapshot

Name is not equal to content: four current data

Variable name	Type	Current value
playerName	String	"Alex"
health	int	20
bread	int	3
diamondCount	int	0

The line of text previously output will not change with the variable. If you want to show the new status, you need to output it again. When the program is re-run, the initialization statements are also re-executed.

Snippet: place inside main and run separately

```
int chestA = 5;
int chestB = chestA;
chestA = 9;
System.out.println("A=" + chestA);
System.out.println("B=" + chestB);
```

Output

```
A=9
B=5
```

What is copied here is the int value: when performing the assignment, store the current 5 into chestB. Afterwards, only chestA is changed, chestB will not be automatically synchronized. Objects and references will be studied specifically in later courses.

05 / 10

Update health values along the timeline

The same variable, four moments

Step	Explanation
1. Birth	health = 20
2. Injury	health - 4 Get 16
3. Eating.	health + 3 Get 19
4. Restore initial value	health = 20

Snippet: place inside main and run separately

```
int health = 20;
System.out.println("Birth=" + health);
health = health - 4;
System.out.println("Injury=" + health);
health = health + 3;
System.out.println("Eating.=" + health);
health = 20;
System.out.println("Restore initial value=" + health);
int arrows = 10;
arrows += 5;
arrows -= 2;
System.out.println("Arrow=" + arrows);
```

Output

```
Birth=20
Injury=16
Eating.=19
Restore initial value=20
Arrow=13
```

The last 20 comes from explicit assignment, and the program does not automatically limit the upper limit of life. For int variables in this lesson, += and -= are abbreviations for writing back after corresponding additions and subtractions; learn the complete writing method first, and then use abbreviations.

06 / 10

Good names let the code explain itself

Do a check on named street signs

Example	Judgment
playerName	Clear, use camelCase
diamondCount	Explain that this is the number of diamonds
2player	Error: cannot start with a number
player health	Error: Name cannot have spaces
class	Error: This is a Java keyword
x	Can run, but the purpose is unclear

When starting to learn, it is recommended to start with a lowercase English letter, and use camelCase for multiple words: the first word is lowercase, and the first letter of subsequent words is capitalized. Java is case-sensitive, health and Health are not the same name.

Snippet: place inside main and run separately

```
int playerLevel = 8;
int torchCount = 24;
String petName = "Snow";
System.out.println(playerLevel + " / " + torchCount);
System.out.println(petName);
```

Output

```
8 / 24
Snow
```

Change x, n, and thing to names that readers don't have to guess. Rules are just the bottom line; allowing people to understand the purpose is the value of naming.

07 / 10

Variable troubleshooting: first distinguish what kind of problem it is



Examining a cause and then observing the result is more effective than tearing down the entire building.

Deliberate typo: three different problems

```
int 2player = 10;
int bread = 3;
int bread = 2;
int diamonds;
System.out.println(diamonds);
```

- 2player starts with a number and changes to playerLevel.
- Repeated declaration of bread in the same scope: remove the second int when updating.
- Local variable diamonds has no value yet: explicitly assign value before reading.

Snippet: place inside main and run separately

```
int playerLevel = 10;
int bread = 3;
bread = 2;
int diamonds = 0;
System.out.println(playerLevel + " / " + bread + " / " + diamonds);
```

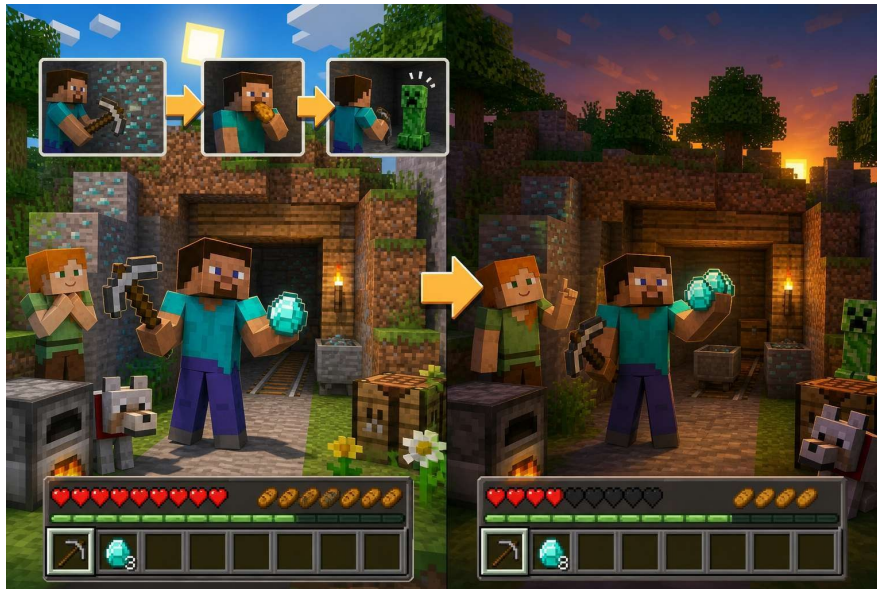
Output

```
10 / 2 / 0
```

Check the type again: `int health = "20";` If the text is given to an integer variable, the compilation will fail. Lesson 3 will expand on this issue.

08 / 10

Design a changing player status bar



In the same adventure, events change data; new output records the changed state.

First draw the working route of the status bar

Step	Explanation
1. Create variables	name, life Bread, diamonds
2. Birth Snapshot	First output the initial value
3. Event Update	Injury, supply With mining
4. Output again	Check new value

This work does not require input, looping or conditional judgment. Just use the declarations, assignments, and outputs you've already learned to observe data changing with events. Before each update, write down the expected values on paper.

- Birth: 20 lives, 3 bread, 0 diamonds.
- Injured: Life minus 4, others unchanged.

- Supply and mining: life plus 3, bread minus 1, and diamonds plus 3.

09 / 10

Complete work and output verification

Complete program · AdventureStatus.java

```
public class AdventureStatus {  
    public static void main(String[] args) {  
        String playerName = "Alex";  
        int health = 20;  
        int bread = 3;  
        int diamondCount = 0;  
        System.out.println("Birth | " + playerName  
            + " | Life=" + health + " | Bread.=" + bread  
            + " | diamond=" + diamondCount);  
        health = health - 4;  
        System.out.println("Injury | " + playerName  
            + " | Life=" + health + " | Bread.=" + bread  
            + " | diamond=" + diamondCount);  
        health = health + 3;  
        bread = bread - 1;  
        diamondCount = diamondCount + 3;  
        System.out.println("Replenishment and mining | " + playerName  
            + " | Life=" + health + " | Bread.=" + bread  
            + " | diamond=" + diamondCount);  
    }  
}
```

Output

```
Birth | Alex | Life=20 | Bread.=3 | diamond=0  
Injury | Alex | Life=16 | Bread.=3 | diamond=0  
Replenishment and mining | Alex | Life=19 | Bread.=2 | diamond=3
```

Only change the player name, all three lines of the name should be changed. Change the damage to 2, and the health points to 20, 18, and 21; the occurrence of 21 is not an error in Java, but that your practice rules do not set an upper limit.

10 / 10

Continue practicing with predictions



Pet wolves can also have their own status records: name, life and number of items.

The ability map of the second lesson

Step	Explanation
1. Statement	Type and name
2. Initialization	First assignment
3. Read	Current value
4. Update	Calculate and write back
5. Explanation	Compare before and after snapshots

Select a task

- Basics: Declare the quantities of five items and output a backpack list.
- Advanced: Start with 0 iron ingots, get 8, use up 3, and get 2 more; the three outputs should be 8, 5, 7.
- Creation: Design the pet wolf status bar to simulate feeding and consuming bones.

School selection: two boxes exchange contents

Snippet: place inside main and run separately

```
int leftChest = 12, rightChest = 30;  
int temporary = leftChest;  
leftChest = rightChest;  
rightChest = temporary;  
System.out.println(leftChest + " / " + rightChest);
```

Output

```
30 / 12
```

temporary Keep the original 12 first. The next lesson will introduce us to more types and explain why integers, decimals, true and false, and literals cannot be mixed randomly.