CODEGYM

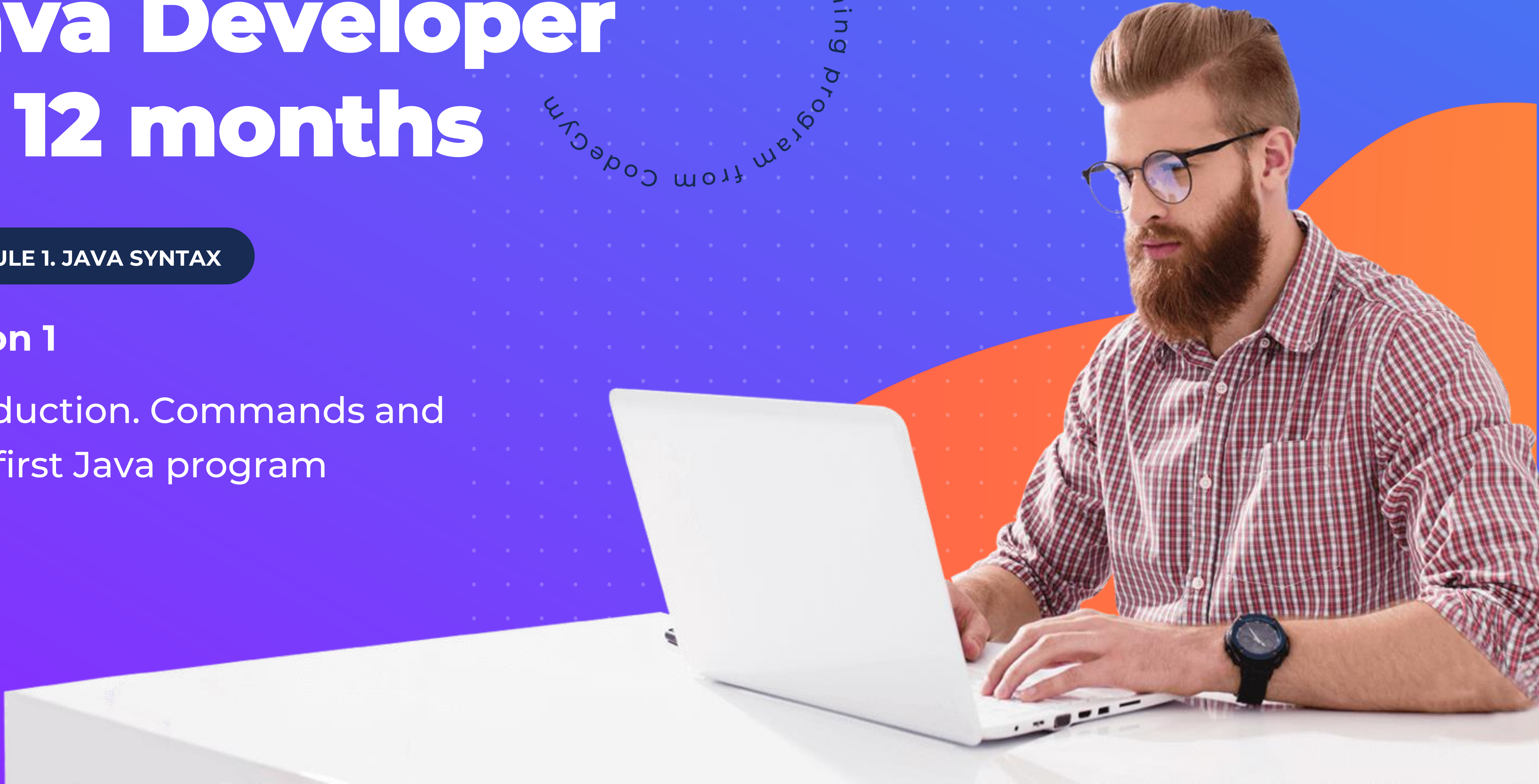
Mentor-supported training program from CodeGym

Java Developer in 12 months

MODULE 1. JAVA SYNTAX

Lesson 1

Introduction. Commands and your first Java program



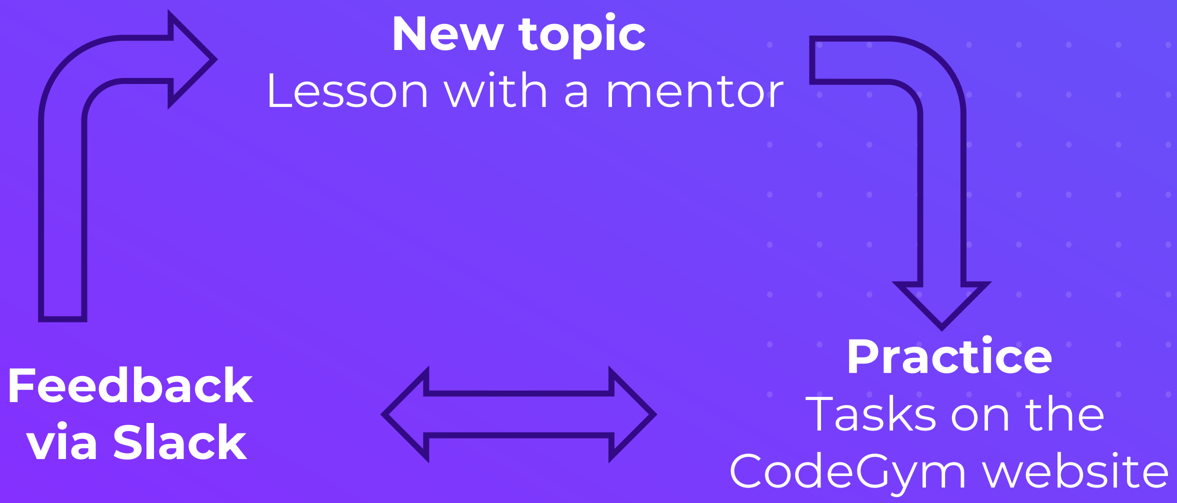
Lesson plan

- Learning program
- Java's advantages, fields of application
- Program structure, main method
- Console output
- Variables declaration
- Basic mathematical operations with int variables
- Increment, decrement
- Comments in code

CODEGYM

How the course progresses

- Lessons with a mentor 2 times a week.
- Practice: tasks are automatically verified on the website.
- Feedback via Slack.
- Analysis of tasks in the next lesson, as needed.



CODEGYM

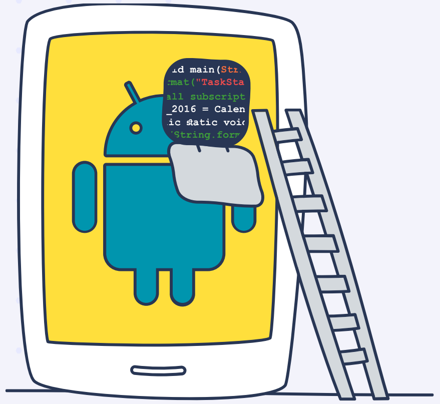
Java's advantages, fields of application

Advantages:

- Cross-platform.
- Automatic memory management
- Speed (JIT compiler)
- Backward compatibility
- Object orientation
- Static typing (fail fast)
- Code as documentation
- Lots of open source libraries and frameworks
- Big community

Fields of application

- Web applications
- Financial (banking) programs
- Android apps
- Desktop applications, development tools
- Embedded systems

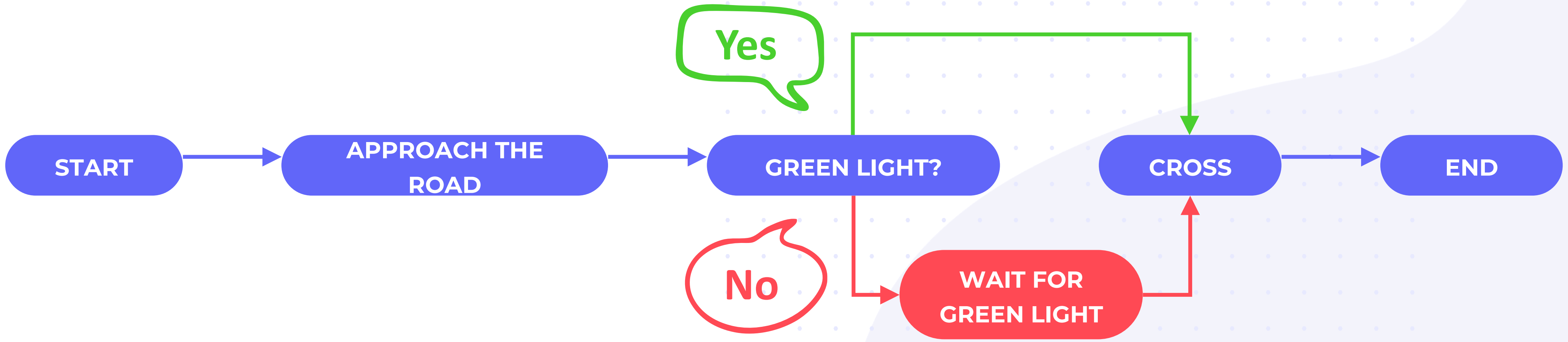


Your FIRST Java program

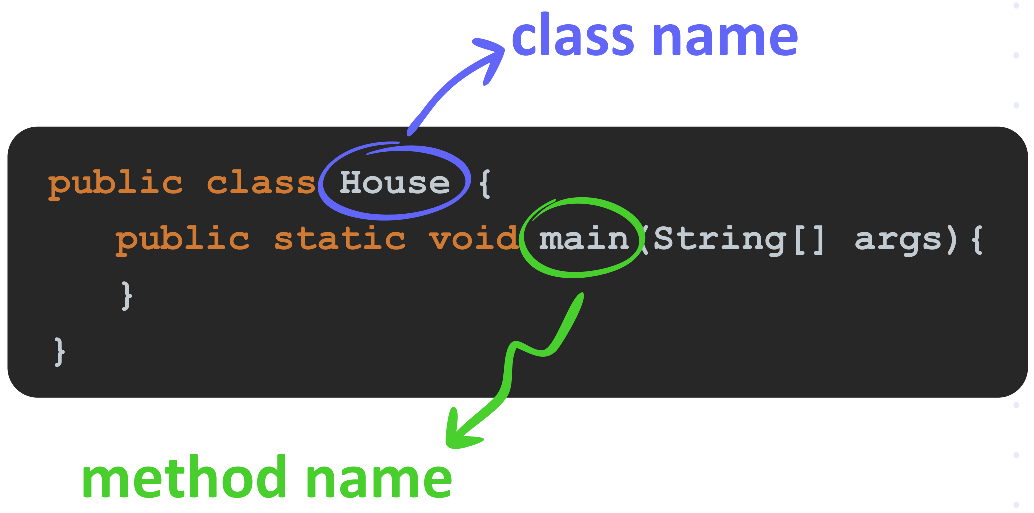
MODULE 1. JAVA SYNTAX

CODEGYM

Structure of a typical program



main() method



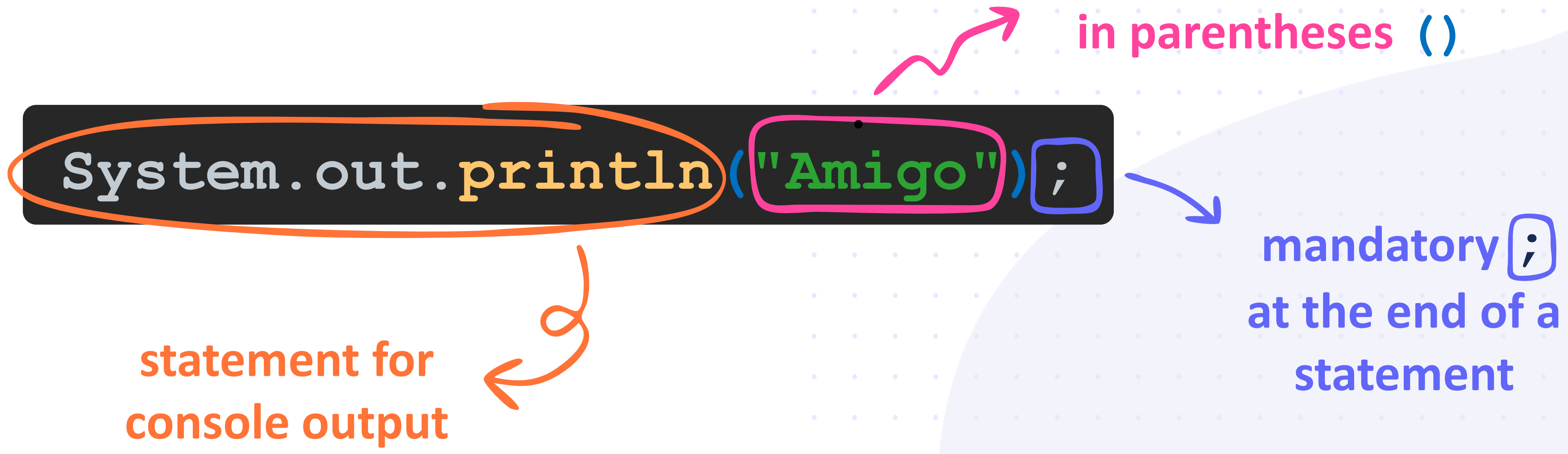
```
public class House {
    public static void main(String[] args){
    }
}
```

- A minimal program must have at least one class
- The class must have at least one method: main
- Declaring the main() method

```
public static void main(String[] args)
```

Statements for console output

Parameter of the `println()` method



```
System.out.println("Amigo");
```

CODEGYM

Commands for console output

Differences between `println()` and `print()`

Statements	What will be displayed

`System.out.println("Amigo");`
`System.out.println("IsThe");`
`System.out.println("Best");`
	Amigo
	IsThe
	Best

`System.out.print("Amigo");`
`System.out.println("IsThe");`
`System.out.print("Best");`
	AmigoIsThe
	Best

`System.out.print("Amigo");`
`System.out.print("IsThe");`
`System.out.print("Best");`
	AmigoIsTheBest

subsequent text will be on a new line

```
System.out.println()
```

```
System.out.print()
```

subsequent text will be displayed on the same line

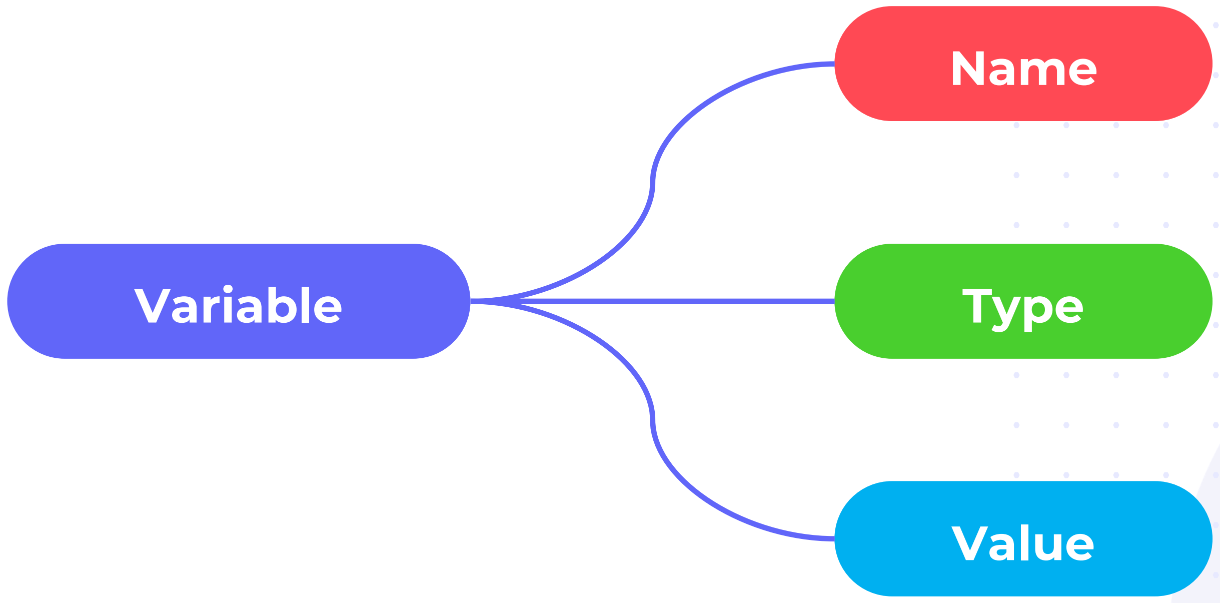
CODEGYM

VARIABLES IN JAVA

MODULE 1. JAVA SYNTAX

What are variables

A variable is a memory area for storing data



used to distinguish one variable from another

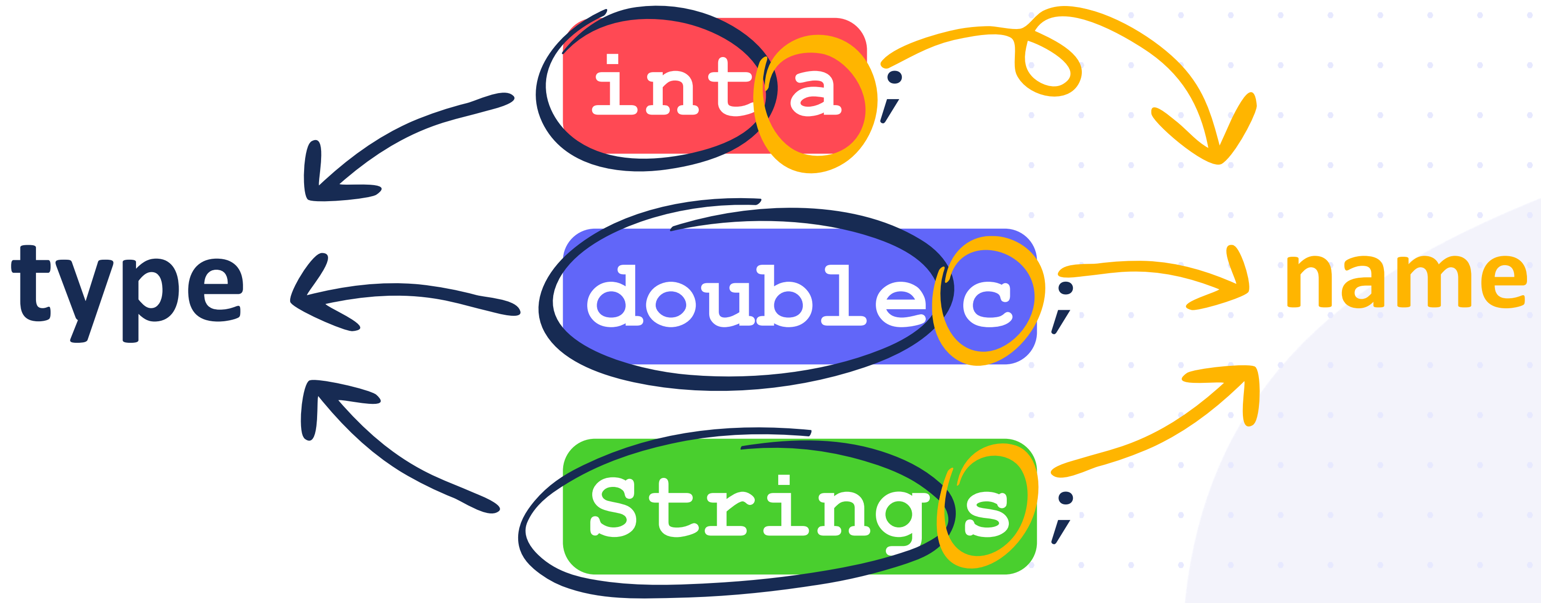
determines the type of values/data that can be stored in it

some object or the data stored in the variable

CODEGYM

Creating a variable

To create a variable, a command like this is used: `type name ;`



`int`	integers
`String`	text
`double`	fractional numbers

CODEGYM

Features of declaring variables

Statement	Explanation
`String s;`	A `String` variable named `s` is created. This variable can store text.
`int x;`	An `int` variable named `x` is created. This variable can store integers.
`int a, b, c;` `int d;`	`int` variables named `a, b, c, and d` are created These variables can store integers.

- You cannot create two variables with the same name in the same method.
- In different methods, you can.

- The name of a variable cannot contain spaces or special characters such as +, -, etc. It is best to use only Latin letters and numerals.

- In Java it matters whether you write uppercase or lowercase letters. `int a` is not the same as `Int a`.

Assignment operator

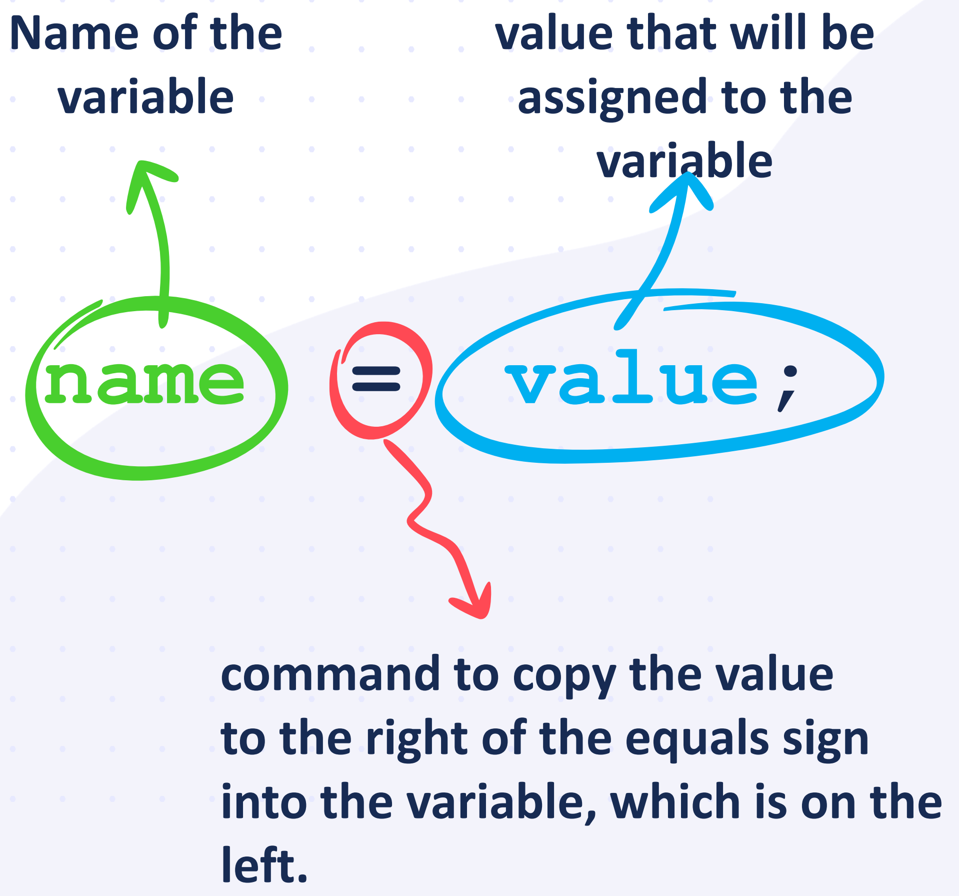
Code	Description

`int i;`
`int a, b;`
`int x;`
	The i variable is created
	The aand b variables are created
	The x variable is created

`i = 3;`	The i variable is set to the value 3.

`a = 1;`
`b = a + 1;`
	The a variable is set to the value 1.
	The b variable is set to the value 2.

`x = 3;`
`x = x + 1;`
	The x variable is set to the value 3.
	On the next line, the value of x is increased by 1. x is now 4.


Name of the variable
value that will be assigned to the variable
name
=
value;
command to copy the value to the right of the equals sign into the variable, which is on the left.

CODEGYM

The int type: whole numbers

`int` is short for `Integer` and is a type for storing integers.

Value range:

from `-2,147,483,648` to `2,147,483,647`

(-2 billion) *(+2 billion)*

32 bits are used for storing an `int` variable

Creating an int variable

```
int name ;
```

declaring an `int` variable

```
int name1, name2, name3;
```

shorthand for creating multiple variables of the same type

The case of the letters matters.

That means the commands `int color` and `int Color` will declare two different variables.

`int` is a special keyword for the integer type and it must be written in lowercase.

CODEGYM

Assigning values

Put a value into an `int` variable:

```
name = value;
```

Shorthand for creating and initializing a variable

```
int name = value;
```

Examples

`int a;`
`a = 5;`	

`int b;`
`b = 2*1000*1000*1000;`	

`int c;`
`c = -10000000;`	

`int d;`
`d = 3000000000;`
	This code won't compile, because `3,000,000,000` is greater than the maximum possible value for an `int`, which is `2,147,483,647`

Evaluating integer expressions

In Java the = symbol is an operator that assigns to the variable on the left of the = sign the calculated value of the expression to the right of the = sign.

The right side of an assignment operator (equal sign) can be any expression — any combination of numbers, variables, and mathematical operators (+, –, *, /), as well as `spaces`.

Examples

`int a = (2 + 2) * 2;`	The value of the variable will be 8
`int b = (6 - 3) / (9 - 6);`	The value of the variable will be 1

`int a = 1;`	The value of the variable a will be 1
`int b = 2;`	The value of the variable b will be 2
`int c = a * b + 2;`	The value of the variable c will be 4

CODEGYM

Division of integers

In Java, dividing an **integer** by an **integer** always results in an **integer**.

The result of division is rounded down.

Example

Statement	Result of division	Note
`int a = 5 / 2;`	2.5	The value of the variable a will be 2
`int b = 20 / 3;`	6.3333(3)	The value of the variable b will be 6
`int c = 6 / 5;`	1.2	The value of the variable c will be 1
`int d = 1 / 2;`	0.5	The value of the variable d will be 0

% operator — remainder of division of integers

Example

Statement	Result of division	Note
`int a = 5 % 2;`	2 with a remainder of 1	The value of the variable a will be 1
`int b = 20 % 4;`	5 with a remainder of 0	The value of the variable b will be 0

```
(a % 2) == 0
```

for checking odd or even

CODEGYM

Increment and decrement

Increment increases a variable by one

```
a++;
```

Decrement decreases a variable by one

```
a--;
```

Example

`int x = 5;`	The value of the variable x will be 5
`x++;`	The value of the variable x will be 6
`x++;`	The value of the variable x will be 7
`x++;`	The value of the variable x will be 8
`x++;`	The value of the variable x will be 9
`x++;`	The value of the variable x will be 10
`int x = 5;`	The value of the variable x will be 5
`x--;`	The value of the variable x will be 4
`x--;`	The value of the variable x will be 3
`x--;`	The value of the variable x will be 2
`x--;`	The value of the variable x will be 1
`x--;`	The value of the variable x will be 0
`x--;`	The value of the variable x will be -1

Comments in Java

Method No. 1. The beginning of the comment is indicated by a pair of symbols (/*), and the end – by (*/).

```
public class Home {
   public static void main (String[] args) {
      /*
      Now we'll display the phrase
      'Amigo is the Best'
      */
      System.out.print("Amigo ");
      System.out.print("is the ");
      System.out.print("Best");
   }
}
```

when the program is compiled, the compiler omits everything between the symbols /* and */

Method No. 2. Using //

```
public class Home {
   public static void main (String[] args) {
      System.out.print("Amigo ");
      System.out.print("is the ");
      // This is also a comment
      System.out.print("Best");
   }
}
```

when the program is compiled, all the comments are skipped

CODEGYM

Homework

MODULE 1. JAVA SYNTAX

Complete Level 1, 2

MODULE 1. JAVA SYNTAX

CODEGYM

Answers to questions

MODULE 1. JAVA SYNTAX