



# Lets Code!

Fundamentals: Input and Output

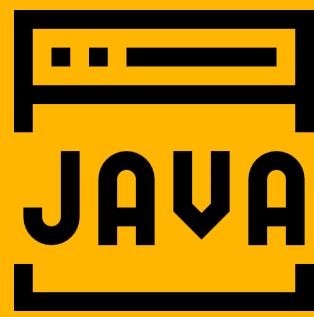
Instructor: Tariq Hook

You can find me on github @code-rhino

# Key Terms

- Scanner
- Formatting

- System



# Input and Output

# Reading Input

To read console input, you first construct a Scanner that is attached to System.in

```
Scanner in = new Scanner(System.in);  
  
System.out.print("What is your name? ");  
String name = in.nextLine();
```

```
String firstName = in.next(); //read a single word
```

# Reading Input

```
System.out.print("How old are you? ");
int age = in.nextInt();
```

Similarly, the nextDouble method reads the next floating-point number.

# Formatting Output

You can print a number  $x$  to the console with the statement `System.out.print(x)`. That command will print  $x$  with the maximum number of nonzero digits for that type.

```
double x = 10000.0 / 3.0; System.out.print(x);  
// prints 3333.333333333335
```

That is a problem if you want to display, for example, dollars and cents.

# Formatting Output

```
System.out.printf("%8.2f", x); // 3333.33
```

prints x with a field width of 8 characters and a precision of 2 characters.

# Formatting Output

```
System.out.printf("Hello, %s. Next year, you'll be %d", name, age);
```

Each of the format specifiers that start with a % character is replaced with the corresponding argument. The conversion character that ends a format specifier indicates the type of the value to be formatted

## Conversions for printf

# Formatting Output

In addition, you can specify flags that control the appearance of the formatted output.

```
System.out.printf("%,.2f", 10000.0 / 3.0); // 3,333.33
```

# Formatting Output

You can use the static `String.format` method to create a formatted string without printing it

```
String message = String.format("Hello, %s. Next year, you'll be %d", name, age);
```

## Flags for printf

# File Input and Output

To read from a file, construct a Scanner object

```
Scanner in = new Scanner(Paths.get("myfile.txt"), "UTF-8");
```

If the file name contains backslashes, remember to escape each of them with an additional backslash:  
"c:\\mydirectory\\myfile.txt".

# Scanner

```
import java.util.Scanner; // Import the Scanner class

class MyClass {
    public static void main(String[] args) {
        Scanner myObj = new Scanner(System.in); // Create a Scanner object
        System.out.println("Enter username");

        String userName = myObj.nextLine(); // Read user input
        System.out.println("Username is: " + userName); // Output user input
    }
}
```

If the file does not exist, it is created.

# Scanner Methods

Method	Description
nextBoolean()	Reads a boolean value from the user
nextByte()	Reads a byte value from the user
nextDouble()	Reads a double value from the user
nextFloat()	Reads a float value from the user
nextInt()	Reads a int value from the user
nextLine()	Reads a String value from the user
nextLong()	Reads a long value from the user
nextShort()	Reads a short value from the user

```
import java.util.Scanner;

class MyClass {
    public static void main(String[ ] args) {
        Scanner myObj = new Scanner(System.in);

        System.out.println("Enter name, age and salary:");
        // String input
        String name = myObj.nextLine();

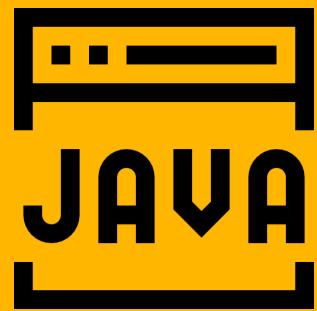
        // Numerical input
        int age = myObj.nextInt();
        double salary = myObj.nextDouble();

        // Output input by user
        System.out.println("Name: " + name);
        System.out.println("Age: " + age);
        System.out.println("Salary: " + salary);
    }
}
```

# Wrap Up

- Scanner
- Formatting

- System



# Keep Coding !!!

Clean Code is Happy Code