

# Hello World

## Print Line

`System.out.println()` can print to the console:

- `System` is a class from the core library provided by Java
- `out` is an object that controls the output
- `println()` is a method associated with that object that receives a single argument

```
System.out.println("Hello, world!");  
// Output: Hello, world!
```

## Comments

Comments are bits of text that are ignored by the compiler. They are used to increase the readability of a program.

- Single line comments are created by using `//` .
- Multi-line comments are created by starting with `/*` and ending with `*/` .

```
// I am a single line comment!
```

```
/*  
And I am a  
multi-line comment!  
*/
```

## main() Method

In Java, every application must contain a `main()` method, which is the entry point for the application. All other methods are invoked from the `main()` method. The signature of the method is `public static void main(String[] args) { }` . It accepts a single argument: an array of elements of type `String` .

```
public class Person {  
  
    public static void main(String[] args) {  
  
        System.out.println("Hello, world!");  
  
    }  
  
}
```

## Classes

A *class* represents a single concept.

A Java program must have one class whose name is the same as the program filename.

In the example, the `Person` class must be declared in a program file named **Person.java**.

```
public class Person {  
  
    public static void main(String[] args) {  
  
        System.out.println("I am a person, not  
a computer.");  
  
    }  
  
}
```

## Compiling Java

In Java, when we compile a program, each individual class is converted into a **.class** file, which is known as byte code.

The JVM (Java virtual machine) is used to run the byte code.

```
# Compile the class file:  
javac hello.java  
  
# Execute the compiled file:  
java hello
```

## Whitespace

Whitespace, including spaces and newlines, between statements is ignored.

```
System.out.println("Example of a statement");  
  
System.out.println("Another statement");  
  
// Output:  
// Example of a statement  
// Another statement
```

## Statements

In Java, a statement is a line of code that executes a task and is terminated with a `;`.

```
System.out.println("Java Programming 🐍");
```