

JAVA METHOD

A Java **method** is a collection of statements that are grouped together to perform an operation. When you call the `System.out.println()` method, for example, the system actually executes several statements in order to display a message on the console.

Creating Method

Considering the following example to explain the syntax of a method –

Syntax

```
public static void methodName() {  
    // body  
}
```

Here,

- **public static** – modifier
- **void** return type
- **methodName** – name of the method
- **inside the ()** – list of parameters

Method definition consists of a method header and a method body. The same is shown in the following syntax –

Syntax

```
modifier returnType nameOfMethod (Parameter List) {  
    // method body  
}
```

The syntax shown above includes –

- **modifier** – It defines the access type of the method and it is optional to use.
- **returnType** – Method may return a value.
- **nameOfMethod** – This is the method name. The method signature consists of the method name and the parameter list.
- **Parameter List** – The list of parameters, it is the type, order, and number of parameters of a method. These are optional, method may contain zero parameters.
- **method body** – The method body defines what the method does with the statements.

Call a Method

To call a method in Java, write the method's name followed by two parentheses **()** and a semicolon;

In the following example, `myMethod()` is used to print a text (the action), when it is called:

Example No. 1

```
Online Java Compiler IDE
For Multiple Files, Custom Library and File Read/Write, use our new - Advanced Java IDE

1 public class Main {
2     public static void myMethod() {
3         System.out.println("XYZ Company");
4         System.out.println("8900 U.S. Hwy 14");
5         System.out.println("Crystal Lake, IL 60014");
6         System.out.println("First Java application");
7     }
8     public static void main(String[] args) {
9
10        myMethod();
11    }
12 }
```

```
XYZ Company
8900 U.S. Hwy 14
Crystal Lake, IL 60014
First Java application
```

Example No. 2

```
1 public class ExampleNumber {
2
3     public static void main(String[] args) {
4         multiply (5,10);
5         multiply (10,20);
6         multiply (30,15);
7     }
8
9
10    public static void multiply(int a, int b ) {
11        System.out.println(a*b);
12    }
13
14 }
```

```
50
200
450
```

Parameters and Arguments

Information can be passed to methods as parameter. **Parameters** act as variables inside the method.

Parameters are specified after the method name, inside the parentheses. You can add as many parameters as you want, just separate them with a comma.

The following example has a method that takes a String called **Subject** as parameter. When the method is called, we pass along a first name, which is used inside the method to print the full name

Example No. 3

```
1 public class Main {  
2     static void myMethod(String Subject) {  
3         System.out.println(Subject + " Programming Language ");  
4     }  
5  
6     public static void main(String[] args) {  
7         myMethod("Java");  
8         myMethod("Phyton");  
9         myMethod("Arduino");  
10    }  
11 }  
12
```

Java Programming Language
Phyton Programming Language
Arduino Programming Language

Return Values

The **void** keyword, used in the examples above, indicates that the method should not return a value. If you want the method to return a value, you can use a primitive data type (such as int, char, etc.) instead of void, and use the return keyword inside the method:

Example No. 4

```
1 public class Main {  
2     static int myMethod(int x) {  
3         return 54 + x;  
4     }  
5  
6     public static void main(String[] args) {  
7         System.out.println(myMethod(17));  
8     }  
9 }
```

71

Example No. 5

```
1 public class Main {  
2  
3     // method defined  
4     private static int getSquare(int x){  
5         return x * x;  
6     }  
7  
8     public static void main(String[] args) {  
9         for (int i = 1; i <= 5; i++) {  
10  
11             // method call  
12             int result = getSquare(i);  
13             System.out.println("Square of " + i + " is: " + result);  
14         }  
15     }  
16 }
```

```
Square of 1 is: 1  
Square of 2 is: 4  
Square of 3 is: 9  
Square of 4 is: 16  
Square of 5 is: 25
```

A Method with If...Else

Example No. 6

```
1  
2 public class Main {  
3  
4     // Create a checkAge() method with an integer variable called age  
5     public static void checkAge(int age) {  
6  
7  
8  
9         if (age < 18) {  
10             System.out.println("Access denied - You are not in a legal age !");  
11  
12  
13         } else {  
14             System.out.println("Access granted - You are in a legal age!");  
15         }  
16  
17     }  
18  
19     public static void main(String[] args) {  
20         checkAge(20);  
21     }  
22 }
```