

Java Method Parameters

Topics : [JAVA](#)

Written on [April 07, 2023](#)

In Java, a method parameter is a variable that is passed to a method when the method is called. The parameter allows the method to receive data or information from the caller, which the method can then use to perform its functionality.

Java methods can have zero or more parameters, depending on the requirements of the method. Parameters are defined in the method declaration, enclosed in parentheses, following the method name. Each parameter is declared with a type and a name, separated by a comma.

For example, consider the following method that takes two parameters of type int:

Example :

```
public void add(int x, int y) {  
    int sum = x + y;  
    System.out.println("The sum is: " + sum);  
}
```

In this example, the method `add` takes two parameters `x` and `y`, both of type `int`. The method adds the two parameters and prints the result to the console.

When calling a method, the caller must provide values for each of the method parameters. These values are passed to the method as arguments, in the same order as the parameter list. For example:

```
add(8, 9);
```

This call to the `add` method passes the values `8` and `9` as arguments for the `x` and `y` parameters, respectively.

Java also supports optional parameters using the `varargs` syntax, which allows a method to accept a variable number of arguments. These are defined by placing an ellipsis `...` after the parameter type, and the `varargs` parameter must be the last parameter in the list.

Output will be The sum is: 17

Example :

```
import java.util.*;
```

```
public class Main {  
    public static void main(String[] args) {  
        Main m = new Main();  
  
        m.add(11,22);  
    }  
    public void add(int x, int y) {  
        int sum = x + y;  
        System.out.println("The sum is: " + sum);  
    }  
}
```

Output :

```
The sum is: 33
```

© Copyright **Aryatechno**. All Rights Reserved. Written tutorials and materials by [Aryatechno](#)