



# Java Method Parameters

[< Previous](#)[Next >](#)

## 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 **fname** 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

```
public class Main {  
  
    System.out.println(fname + " Refsnes");  
}  
  
public static void main(String[] args) {
```

```
    }  
}  
// Liam Refsnes  
// Jenny Refsnes  
// Anja Refsnes
```

Try it Yourself »

When a **parameter** is passed to the method, it is called an **argument**. So, from the example above: **fname** is a **parameter**, while **Liam**, **Jenny** and **Anja** are **arguments**.

## Multiple Parameters

You can have as many parameters as you like:

### Example

```
public class Main {  
  
    System.out.println(fname + " is " + age);  
}  
  
public static void main(String[] args) {  
    myMethod("Liam", 5);  
    myMethod("Jenny", 8);  
    myMethod("Anja", 31);  
}  
}  
  
// Liam is 5  
// Jenny is 8  
// Anja is 31
```

Try it Yourself »

Note that when you are working with multiple parameters, the method call must have the same number of arguments as there are parameters, and the arguments must be passed in the same order.

## 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

```
public class Main {  
    static int myMethod(int x) {  
  
    }  
  
    public static void main(String[] args) {  
        System.out.println(myMethod(3));  
    }  
}  
// Outputs 8 (5 + 3)
```

Try it Yourself »

This example returns the sum of a method's **two parameters**:

### Example

```
public class Main {  
    static int myMethod(int x, int y) {  
        return x + y;  
    }  
  
    public static void main(String[] args) {  
        System.out.println(myMethod(5, 3));  
    }  
}  
// Outputs 8 (5 + 3)
```

Try it Yourself »

You can also store the result in a variable (recommended, as it is easier to read and maintain):

## Example

```
public class Main {  
    static int myMethod(int x, int y) {  
        return x + y;  
    }  
  
    public static void main(String[] args) {  
        int z = myMethod(5, 3);  
        System.out.println(z);  
    }  
}  
// Outputs 8 (5 + 3)
```

Try it Yourself »

---

ADVERTISEMENT

## A Method with If...Else

It is common to use `if...else` statements inside methods:

### Example

```
public class Main {  
  
    // Create a checkAge() method with an integer variable called age  
    static void checkAge(int age) {  
  
        // If age is less than 18, print "access denied"  
        if (age < 18) {  
            System.out.println("Access denied - You are not old enough!");  
  
            // If age is greater than, or equal to, 18, print "access granted"  
        } else {  
            System.out.println("Access granted - You are old enough!");  
        }  
  
    }  
  
    public static void main(String[] args) {  
        checkAge(20); // Call the checkAge method and pass along an age of 20  
    }  
}
```

```
}  
  
// Outputs "Access granted - You are old enough!"
```

[Try it Yourself »](#)

---

## Test Yourself With Exercises

### Exercise:

Add a `fname` parameter of type `String` to `myMethod`, and output "John Doe":

```
static void myMethod(           ) {  
    System.out.println(         + " Doe");  
}  
  
public static void main(String[] args) {  
    myMethod("John");  
}
```

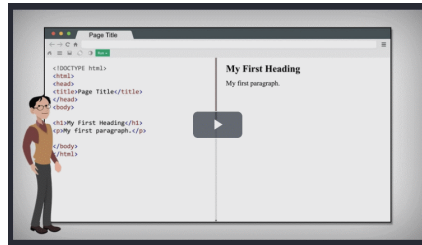
[Submit Answer »](#)

[Start the Exercise](#)

## ADVERTISEMENT

### NEW

We just launched  
W3Schools videos



Explore now

COLOR PICKER



Get certified  
by completing  
a Java  
course today!



Get started

CODE GAME





Play Game

ADVERTISEMENT

ADVERTISEMENT

ADVERTISEMENT



Report Error

Spaces

Pro

Buy Certificate

## Top Tutorials

[HTML Tutorial](#)

[CSS Tutorial](#)

[JavaScript Tutorial](#)

- [How To Tutorial](#)
- [SQL Tutorial](#)
- [Python Tutorial](#)
- [W3.CSS Tutorial](#)
- [Bootstrap Tutorial](#)
- [PHP Tutorial](#)
- [Java Tutorial](#)
- [C++ Tutorial](#)
- [jQuery Tutorial](#)

## Top References

- [HTML Reference](#)
- [CSS Reference](#)
- [JavaScript Reference](#)
- [SQL Reference](#)
- [Python Reference](#)
- [W3.CSS Reference](#)
- [Bootstrap Reference](#)
- [PHP Reference](#)
- [HTML Colors](#)
- [Java Reference](#)
- [Angular Reference](#)
- [jQuery Reference](#)

## Top Examples

- [HTML Examples](#)
- [CSS Examples](#)
- [JavaScript Examples](#)
- [How To Examples](#)
- [SQL Examples](#)
- [Python Examples](#)
- [W3.CSS Examples](#)
- [Bootstrap Examples](#)
- [PHP Examples](#)
- [Java Examples](#)
- [XML Examples](#)
- [jQuery Examples](#)

## Get Certified

- [HTML Certificate](#)
- [CSS Certificate](#)
- [JavaScript Certificate](#)
- [Front End Certificate](#)
- [SQL Certificate](#)
- [Python Certificate](#)
- [PHP Certificate](#)
- [jQuery Certificate](#)
- [Java Certificate](#)
- [C++ Certificate](#)
- [C# Certificate](#)
- [XML Certificate](#)

W3Schools is optimized for learning and training. Examples might be simplified to improve reading and learning. Tutorials, references, and examples are constantly reviewed to avoid errors, but we cannot warrant full correctness of all content. While using W3Schools, you agree to have read and accepted our terms of use, cookie and privacy policy.

Copyright 1999-2022 by Refsnes Data. All Rights Reserved.  
W3Schools is Powered by W3.CSS.

