**Javadoc Tutorial**

Table of Contents

[Basics 3](#_Toc62852528)

[Tags 4](#_Toc62852529)

[Example 5](#_Toc62852530)

[HTML Webpage 7](#_Toc62852531)

Javadoc is a documentation tool that comes built in with any IDE for Java. It helps us create professional documentation for our projects.

Javadoc comments are similar to comments within our code. They are written like this:

// some code  
  
/\*\*  
 \* This is a Javadoc comment.  
 \*/  
  
// some more code

JAVA

Notice that the only difference between a Javadoc comment and a multiline comment is that there are two \* symbols on the first line.

## Basics

The way Javadoc comments work is, if we want to add documentation for any part of the code, we should add it before the code.

For example, if we want to add documentation for a class, we add the text before the class. If we want to add documentation for a function, we add the text before the function.

The documentation for a class is generally used to describe the class and how it works. Thus, anyone looking at the code will be able to quickly understand what is going on in the class without going over the code.

The documentation for a function describes the work of the function as well as the parameters and return types of the function. For small functions, this is not always necessary, but it is always better to use Javadoc comments.

When writing Javadoc comments, it is easier to use an IDE. The benefit to is that a lot of the text will be autogenerated. For example, just writing /\*\* and pressing Enter before a function will automatically add the parameters and return type of the function to the Javadoc comment. We will just need to add text next to those to describe what the parameters are being used for.

Once Javadoc comments have been added, whenever we try to use any of the functions or create objects of the classes, the Javadoc documentation will be shown in a visually pleasing format in the autocomplete suggestions.

Note that suggestions are not shown by default while typing within the NetBeans IDE. To be able to see it, we need to press Ctrl + Space.

## Tags

An important feature of Javadoc is the availability of tags. In the default Javadoc created by the IDE when we make a new class, there may be a tag like @author, followed by the name of the author. There are many such tags, such as @params for (non-private) parameters to a function, @return for return types etc.

## Example

The following code is for an additional class created inside a project. Javadoc comments can, of course, be used anywhere in the project.

package untitledJavaApplication;  
import java.lang.Math;  
  
/\*\*  
 \* This is an example class for Javadoc.  
 \*  
 \* @author Me  
 \*/  
public class Point  
{  
 private int x, y;  
  
 /\*\*  
 \* Constructs and initializes a 2D point on (x, y) coordinate.  
 \*  
 \* @param ***x*** x coordinate  
 \* @param ***y*** y coordinate  
 \*/  
  
 public Point (int x, int y)  
 {  
 this.x = x;  
 this.y = y;  
 }

/\*\*  
 \* Calculates the straight line distance between this point and the other point given in the argument.  
 \*  
 \* @param ***other*** The other point to calculate the distance from  
 \* @return straight line distance between the two points  
 \*/  
 public double distance (Point other)  
 {  
 return Math.sqrt(Math.pow(this.x - other.x, 2) + Math.pow(this.y - other.y, 2));  
 }

/\*\*  
 \* Give access to the private attribute x.  
 \*  
 \* @return x coordinate of the object  
 \*/  
 public int getX()  
 {  
 return x;  
 }  
  
 /\*\*  
 \* Give access to the private attribute y.  
 \*  
 \* @return y coordinate of the object  
 \*/  
 public int getY()  
 {  
 return y;  
 }  
}

JAVA

## HTML Webpage

Most IDEs will also provide a way to view all of the documentation for a particular project as a webpage. The exact way to do this may vary from IDE to IDE.

In NetBeans, the webpage can be generated by going to the Run tab and clicking Generate Javadoc. This will open the generated webpage in the default browser. The files for the webpage can be found in the Javadoc subfolder inside the dist subfolder of the main project folder.