

**May the Force of Javadoc
be with You**

Clare So

`socm@mcmaster.ca`

What is Javadoc?

- A utility of Java that generates [API documentation](#) as HTML pages from [Java source file](#)
- A standard for documenting Java classes
- Included in Java JDK/SDK from Sun Microsystems

Why Javadoc?

- Your friend needs to reuse your wonderful Java code
- You don't want him/her to see your implementation
- Let's give him/her the class file
- Your friend asks you, "How to use your Java class?"
- We need documentation!

What does Javadoc Look Like?

Package **Class** [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: NESTED | FIELD | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

DETAIL: FIELD | [CONSTR](#) | [METHOD](#)

Class Circle

java.lang.Object
└ **Circle**

public class **Circle**
extends java.lang.Object

Circle is a class representing the mathematical concept of circle. This class includes the following operations:

- Increase the size
- Decrease the size
- Report the circumference

This class is written for the Java tutorials of SE2AA4 (Winter 07).

Constructor Summary

[Circle](#)()

The diameter of the circle is 10cm by default.

[Circle](#)(double d)

How to Make Javadoc?

- Write comments in your code following a specific format
- Run javadoc
- Voila! You have the documentation of your Java class

Circle Class Revisited

- The beginning of Circle.java now contains

```
/**
    Circle is a class representing the mathematical concept of circle.
    This class includes the following operations:
    <ul>
        <li>Increase the size</li>
        <li>Decrease the size</li>
        <li>Report the circumference</li>
    </ul>
    <p>
    This class is written for the Java tutorials of SE2AA4 (Winter 07).

    @author Clare So

    */
```

Format of Javadoc

- Comments must be included using `/** */`
(Note the **two asterisk** at the beginning of the comment)
- Some HTML tags are recognized
 - `<p>` – paragraph
 - `` – unordered list
 - `` – list item
- `@` indicates a Javadoc tag
(Don't panic. We will talk about Javadoc tags in a moment!)

Circle Class Revisited

- Here we include the description of the constructors in the comments

```
public class Circle {  
  
    private double diameter; // The diameter of the circle  
  
    /**  
     * The diameter of the circle is 10cm by default.  
     */  
    public Circle() { diameter = 10.0; }  
  
    /**  
     * The circle can be initialized of other sizes.  
     *  
     * @param d diameter of the new circle  
     */  
    public Circle(double d) { diameter = d; }  
}
```


Circle Class Revisted

- The Javadoc tags `@param` and `@return` describe the parameters and return type of the Java methods

```
/**
    Make the circle bigger

    @param factor Centimetres to be added to the diameter
    @return updated diameter
 */
public double grow(double factor) {
    diameter = diameter + factor;
    return diameter;
}
```

Some Javadoc Tags

- `@param` – Parameters
(More than one `@param` can be used for the same method)
- `@return` – Object returned by the method
- `@exception` and `throws` – Exceptions that can be generated by the method
- `@author` – Author of the class
- `@version` – Version of the class
- `@deprecated` – Indicates deprecated methods

Using Javadoc in Command Line

```
[clare@dimsum-cart tutorials]$ javadoc Circle.java
Loading source file Circle.java...
Constructing Javadoc information...
Standard Doclet version 1.4.2-02
Generating constant-values.html...
Building tree for all the packages and classes...
Building index for all the packages and classes...
Generating overview-tree.html...
Generating index-all.html...
Generating deprecated-list.html...
Building index for all classes...
Generating allclasses-frame.html...
Generating allclasses-noframe.html...
Generating index.html...
Generating packages.html...
Generating Circle.html...
Generating package-list...
Generating help-doc.html...
Generating stylesheet.css...
[clare@dimsum-cart tutorials]$
```