

# Tạo Javadoc

Ngôn ngữ Java hỗ trợ 3 kiểu comment sau:

Comment	Miêu tả
<code>/* text */</code>	Trình biên dịch bỏ qua mọi thứ từ <code>/*</code> tới <code>*/</code> .
<code>// text</code>	Trình biên dịch bỏ qua mọi thứ từ <code>//</code> tới cuối dòng.
<code>/** documentation */</code>	Đây là một Documentation Comment và nói chung nó được gọi là <b>doc comment</b> . Công cụ <b>JDK javadoc</b> sử dụng <i>doc comment</i> khi tự động chuẩn bị documentation đã tạo.

Chương hướng dẫn này giải thích về Javadoc. Chúng ta sẽ thấy cách sử dụng Javadoc để tạo Documentation hữu ích cho Java code của bạn.

## Javadoc là gì?

Javadoc là một công cụ đi với JDK và nó được sử dụng để tạo Java code documentation trong định dạng HTML từ Java source code mà đã yêu cầu Documentation trong một định dạng đã định trước.

Trong ví dụ đơn giản sau, phần màu đỏ của code biểu diễn Java comment:

```
/**  
 * The HelloWorld program implements an application that  
 * simply displays "Hello World!" to the standard output.  
 *  
 * @author   Zara Ali  
 * @version  1.0  
 * @since    2014-03-31  
 */  
  
public class HelloWorld {  
    public static void main(String[] args) {  
        /* Prints Hello, World! on standard output.  
        */  
    }  
}
```

```
        System.out.println("Hello World!");
    }
}
```

Bạn có thể bao gồm các thẻ HTML được yêu cầu bên trong phần mô tả, ví dụ sau sử dụng `<h1>....</h1>` cho heading và `<p>` được sử dụng để tạo các đoạn văn:

```
/**
 * <h1>Hello, World!</h1>
 * The HelloWorld program implements an application that
 * simply displays "Hello World!" to the standard output.
 * <p>
 * Giving proper comments in your program makes it more
 * user friendly and it is assumed as a high quality code.
 *
 *
 * @author  Zara Ali
 * @version 1.0
 * @since   2014-03-31
 */
public class HelloWorld {
    public static void main(String[] args) {
        /* Prints Hello, World! on standard output.
        System.out.println("Hello World!");
        */
    }
}
```

## Các thẻ javadoc trong Java

Công cụ javadoc thừa nhận các thẻ sau:

Thẻ	Miêu tả	Cú pháp
@author	Thêm author của một lớp	@author name-text

<code>{@code}</code>	Hiển thị text trong code font mà không thông dịch text như là các thẻ HTML markup hoặc javadoc được lồng.	<code>{@code text}</code>
<code>{@docRoot}</code>	Biểu diễn relative path tới thư mục root của tài liệu từ các trang được tạo	<code>{@docRoot}</code>
<code>@deprecated</code>	Thêm một comment chỉ dẫn rằng API này không còn được sử dụng nữa	<code>@deprecated deprecated-text</code>
<code>@exception</code>	Thêm một <b>Throws</b> subheading tới documentation được tạo, với tên lớp và text mô tả	<code>@exception class-name description</code>
<code>{@inheritDoc}</code>	Kế thừa một comment từ lớp có thể kế thừa <b>agăn nhất</b> hoặc interface có thể triển khai	Kế thừa một comment từ một lớp cha trung gian.
<code>{@link}</code>	Chèn một in-line link với nhãn text có thể nhìn thấy mà chỉ tới Documentation cho package, lớp hoặc tên member đã xác định của một lớp tham chiếu	<code>{@link package.class#member label}</code>
<code>{@linkplain}</code>	Giống <code>{@link}</code> , ngoại trừ label của link được hiển thị ở dạng plain text thay vì code font.	<code>{@linkplain package.class#member label}</code>
<code>@param</code>	Thêm một tham số với parameter-name đã xác định được theo sau bởi description đã xác định tới khu vực "Parameters".	<code>@param parameter-name description</code>
<code>@return</code>	Thêm một khu vực "Returns" với description	<code>@return description</code>
<code>@see</code>	Thêm một "See Also" heading với một link hoặc text mà chỉ tới reference	<code>@see reference</code>

@serial	Được sử dụng trong doc comment cho một trường có thể xếp thứ tự.	@serial field-description   include   exclude
@serialData	Thu thập thông tin dữ liệu được viết bởi phương thức writeObject( ) hoặc writeExternal( )	@serialData data-description
@serialField	Thu thập thông tin một thành phần ObjectOutputStreamField	@serialField field-name field-type field-description
@since	Thêm một "Since" heading với since-text đã xác định tới Documentation đã tạo	@since release
@throws	Thẻ @throws và @exception là như nhau	@throws class-name description
{@value}	Khi {@value} được sử dụng trong doc comment của một trường static, nó hiển thị giá trị của hằng số đó	{@value package.class#field}
@version	Thêm một "Version" subheading với version-text đã xác định tới Documentation đã tạo khi tùy chọn -version được sử dụng	@version version-text

## Ví dụ:

Chương trình sau sử dụng một số thẻ quan trọng có sẵn cho Documentation comment. Bạn có thể sử dụng các thẻ khác tùy theo yêu cầu của bạn.

Documentation về lớp AddNum sẽ được tạo trong HTML file là AddNum.html nhưng cùng thời điểm đó một master file với tên là index.html cũng sẽ được tạo.

```
import java.io.*;

/**
 * <h1>Add Two Numbers!</h1>
```

```
* The AddNum program implements an application that
* simply adds two given integer numbers and Prints
* the output on the screen.
* <p>
* <b>Note:</b> Giving proper comments in your program makes it more
* user friendly and it is assumed as a high quality code.
*
* @author  Zara Ali
* @version 1.0
* @since   2014-03-31
*/
public class AddNum {
    /**
     * This method is used to add two integers. This is
     * a the simplest form of a class method, just to
     * show the usage of various javadoc Tags.
     * @param numA This is the first paramter to addNum method
     * @param numB This is the second parameter to addNum method
     * @return int This returns sum of numA and numB.
     */
    public int addNum(int numA, int numB) {
        return numA + numB;
    }

    /**
     * This is the main method which makes use of addNum method.
     * @param args Unused.
     * @return Nothing.
     * @exception IOException On input error.
     * @see IOException
     */
    public static void main(String args[]) throws IOException
    {
```

```
AddNum obj = new AddNum();  
int sum = obj.addNum(10, 20);  
  
System.out.println("Sum of 10 and 20 is :" + sum);  
}  
}
```

Bây giờ, xử lý AddNum.java file bởi sử dụng tiện ích javadoc như sau:

```
$ javadoc AddNum.java  
Loading source file AddNum.java...  
Constructing Javadoc information...  
Standard Doclet version 1.7.0_51  
Building tree for all the packages and classes...  
Generating /AddNum.html...  
AddNum.java:36: warning - @return tag cannot be used in method with void return type.  
Generating /package-frame.html...  
Generating /package-summary.html...  
Generating /package-tree.html...  
Generating /constant-values.html...  
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 /help-doc.html...  
1 warning  
$
```

Bạn có thể kiểm tra tất cả về Documentation đã tạo ở đây: [AddNum](#). Nếu bạn đang sử dụng JDK 1.7 thì javadoc không tạo một **stylesheet.css** tốt, vì thế tôi đề nghị bạn tải và sử dụng Stylesheet từ <http://docs.oracle.com/javase/7/docs/api/stylesheet.css>