

# JSF - H:OUTPUTFORMAT

[http://www.tutorialspoint.com/jsf/jsf\\_outputformat\\_tag.htm](http://www.tutorialspoint.com/jsf/jsf_outputformat_tag.htm)

Copyright © tutorialspoint.com

The h:outputFormat tag renders an HTML text but can accept parameterised inputs.

## JSF Tag

```
<h:outputFormat value="parameter 1 : {0}, parameter 2 : {1}" >
  <f:param value="Item 1" />
  <f:param value="Item 2" />
</h:outputFormat>
```

## Rendered Output

parameter 1 : Item 1, parameter 2 : Item 2

## Tag Attributes

| S.N. | Attribute & Description                                                                                                                     |
|------|---------------------------------------------------------------------------------------------------------------------------------------------|
| 1    | <b>id</b><br>Identifier for a component                                                                                                     |
| 2    | <b>binding</b><br>Reference to the component that can be used in a backing bean                                                             |
| 3    | <b>rendered</b><br>A boolean; false suppresses rendering                                                                                    |
| 4    | <b>styleClass</b><br>Cascading stylesheet CSS class name                                                                                    |
| 5    | <b>value</b><br>A component's value, typically a value binding                                                                              |
| 6    | <b>converter</b><br>Converter class name                                                                                                    |
| 7    | <b>style</b><br>Inline style information                                                                                                    |
| 8    | <b>title</b><br>A title, used for accessibility, that describes an element. Visual browsers typically create tooltips for the title's value |

## Example Application

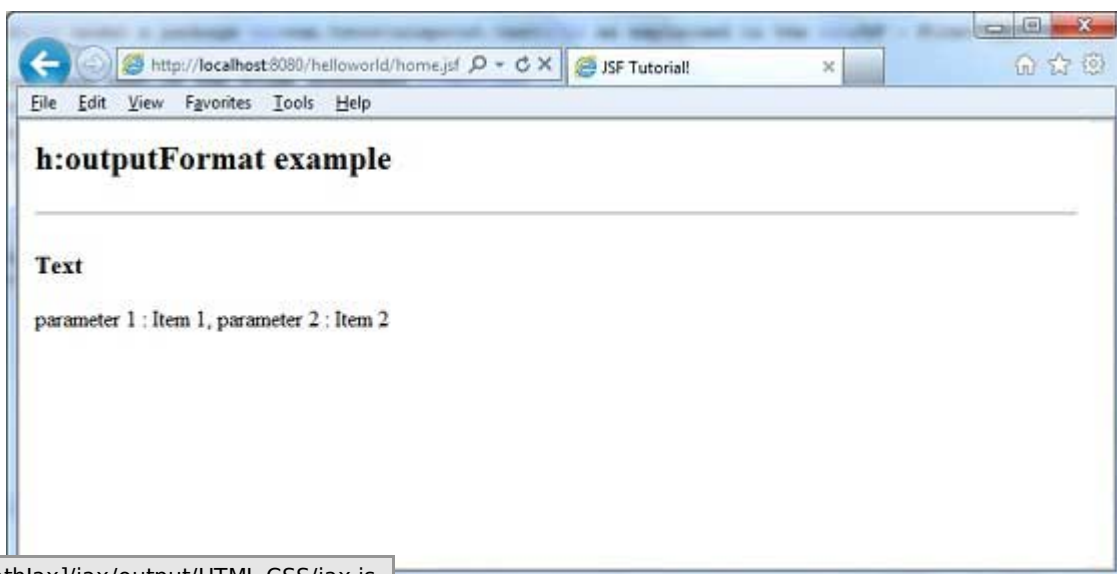
Let us create a test JSF application to test the above tag.

| Step | Description                                                                                                                                               |
|------|-----------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1    | Create a project with a name <i>helloworld</i> under a package <i>com.tutorialspoint.test</i> as explained in the <i>JSF - First Application</i> chapter. |
| 2    | Modify <i>home.xhtml</i> as explained below. Keep rest of the files unchanged.                                                                            |
| 3    | Compile and run the application to make sure business logic is working as per the requirements.                                                           |
| 4    | Finally, build the application in the form of war file and deploy it in Apache Tomcat Webserver.                                                          |
| 5    | Launch your web application using appropriate URL as explained below in the last step.                                                                    |

## home.xhtml

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title>JSF Tutorial!</title>
</head>
<body>
  <h2>h:outputFormat example</h2>
  <hr />
  <h:form>
    <h3>Text</h3>
    <h:outputFormat value="parameter 1 : {0}, parameter 2 : {1}" >
      <f:param value="Item 1" />
      <f:param value="Item 2" />
    </h:outputFormat>
  </h:form>
</body>
</html>
```

Once you are ready with all the changes done, let us compile and run the application as we did in JSF - First Application chapter. If everything is fine with your application, this will produce following result:



Loading [Mathjax]/jax/output/HTML-CSS/jax.js