

**How to iterate HashMap using JSTL forEach loop**

//Java

Map<String, String> countryCapitalList = new HashMap<String, String>();

countryCapitalList.put("United States", "Washington DC");

countryCapitalList.put("India", "Delhi");

countryCapitalList.put("Germany", "Berlin");

countryCapitalList.put("France", "Paris");

countryCapitalList.put("Italy", "Rome");

request.setAttribute("capitalList", countryCapitalList);

//JSP

<c:forEach var="country" items="${capitalList}">

    Country: ${country.key}  - Capital: ${country.value}

</c:forEach>

**request.getParameter() vs request.getAttribute()**

A "parameter" is a name/value pair sent from the client to the server - typically, from an HTML form. Parameters can only have [String](http://www.coderanch.com/t/410859/java/java/String-StringBuffer-StringBuilder-Performance) values. Sometimes (e.g. using a GET request) you will see these encoded directly into the URL (after the ?, each in the form name=value, and each pair separated by an &). Other times, they are included in the body of the request, when using methods such as POST.  
  
An "attribute" is a server-local storage mechanism - nothing stored in scoped attribues is ever transmitted outside the server unless you explicitly make that happen. Attributes have String names, but store Object values. Note that attributes are specific to Java (they store Java Objects), while parameters are platform-independent (they are only formatted strings composed of generic bytes).  
  
There are four scopes of attributes in total: "page" (for JSPs and tag files only), "request" (limited to the current client's request, destroyed after request is completed), "session" (stored in the client's session, invalidated after the session is terminated), "application" (exist for all components to access during the entire deployed lifetime of your application).  
  
The bottom line is: use parameters when obtaining data from the client, use scoped attributes when storing objects on the server for use internally by your application only.