# JAVA - THE PROPERTIES CLASS

http://www.tutorialspoint.com/java/java_properties_class.htm

Properties is a subclass of Hashtable. It is used to maintain lists of values in which the key is a String and the value is also a String.

The Properties class is used by many other Java classes. For example, it is the type of object returned by System.getProperties( ) when obtaining environmental values.

Properties define the following instance variable. This variable holds a default property list associated with a Properties object.

```
Properties defaults;
```

The Properties define two constructors. The first version creates a Properties object that has no default values:

```
Properties( )
```

The second creates an object that uses propDefault for its default values. In both cases, the property list is empty:

```
Properties(Properties propDefault)
```

Apart from the methods defined by Hashtable, Properties define the following methods:

| SN | Methods with Description |
|---|---|
| 1 | **String getProperty(String key)**<br>Returns the value associated with key. A null object is returned if key is neither in the list nor in the default property list. |
| 2 | **String getProperty(String key, String defaultProperty)**<br>Returns the value associated with key. defaultProperty is returned if key is neither in the list nor in the default property list. |
| 3 | **void list(PrintStream streamOut)**<br>Sends the property list to the output stream linked to streamOut. |
| 4 | **void list(PrintWriter streamOut)**<br>Sends the property list to the output stream linked to streamOut. |
| 5 | **void load(InputStream streamIn) throws IOException**<br>Inputs a property list from the input stream linked to streamIn. |
| 6 | **Enumeration propertyNames( )**<br>Returns an enumeration of the keys. This includes those keys found in the default property list, too. |
| 7 | **Object setProperty(String key, String value)** |

Associates value with key. Returns the previous value associated with key, or returns null if no such association exists

8 **void store(OutputStream streamOut, String description)**

After writing the string specified by description, the property list is written to the output stream linked to streamOut

## Example:

The following program illustrates several of the methods supported by this data structure:

```
import java.util.*;

public class PropDemo {

   public static void main(String args[]) {
      Properties capitals = new Properties();
      Set states;
      String str;

      capitals.put("Illinois", "Springfield");
      capitals.put("Missouri", "Jefferson City");
      capitals.put("Washington", "Olympia");
      capitals.put("California", "Sacramento");
      capitals.put("Indiana", "Indianapolis");

      // Show all states and capitals in hashtable.
      states = capitals.keySet(); // get set-view of keys
      Iterator itr = states.iterator();
      while(itr.hasNext()) {
         str = (String) itr.next();
         System.out.println("The capital of " +
            str + " is " + capitals.getProperty(str) + ".");
      }
      System.out.println();

      // look for state not in list -- specify default
      str = capitals.getProperty("Florida", "Not Found");
      System.out.println("The capital of Florida is "
          + str + ".");
   }
}
```

This would produce the following result:

```
The capital of Missouri is Jefferson City.
The capital of Illinois is Springfield.
The capital of Indiana is Indianapolis.
The capital of California is Sacramento.
The capital of Washington is Olympia.

The capital of Florida is Not Found.
```