Util.java

```java
// Copyright 2013 Bill Campbell, Swami Iyer and Bahar Akbal-Delibas

package jminusminus;

/**
 * This class defines helper functions.
 */

class Util {

    /**
     * Escape the special XML characters in the specified string and return the
     * escaped string.
     * 
     * @param s
     *            string to escape.
     * @return the escaped string.
     */

    public static String escapeSpecialXMLChars(String s) {
        String escapedString = s.replaceAll("&", "&amp;");
        escapedString = escapedString.replaceAll("<", "&lt;");
        escapedString = escapedString.replaceAll(">", "&gt;");
        escapedString = escapedString.replaceAll("\"", "&quot;");
        escapedString = escapedString.replaceAll("'", "&#39;");
        return escapedString;
    }

    /**
     * Unescape the escaped characters in the specified string and return the
     * unescaped string.
     * 
     * @param s
     *            string to unescape.
     * @return the unescaped string.
     */

    public static String unescape(String s) {
        StringBuffer b = new StringBuffer();
        for (int i = 0; i < s.length(); i++) {
            char c = s.charAt(i);
            if (c == '\\') {
                i++;
                if (i >= s.length()) {
                    break;
                }
                c = s.charAt(i);
                switch (c) {
                case 'b':
                    b.append('\b');
                    break;
                case 't':
                    b.append('\t');
                    break;
                case 'n':
                    b.append('\n');
                    break;
                case 'f':
                    b.append('\f');
                    break;
                case 'r':
                    b.append('\r');
                    break;
                case '"':
                    b.append('"');
                    break;
                case '\'':
                    b.append('\'');
                    break;
                case '\\':
                    b.append('\\');
                    break;
                }
            } else {
                b.append(c);
            }
        }
        return b.toString();
    }

}

/**
 * A utility class that allows pretty (indented) printing to STDOUT.
 */

class PrettyPrinter {

    /** Width of an indentation. */
    private int indentWidth;

    /** Current indentation (number of blank spaces). */
    private int indent;

    /**
     * Construct a PrettyPrinter with an indentation width of 2.
     */

    public PrettyPrinter() {
        this(2);
    }

    /**
     * Construct a PrettyPrinter given the indentation width.
     * 
     * @param indentWidth
     *            number of blank spaces for an indent.
     */

    public PrettyPrinter(int indentWidth) {
        this.indentWidth = indentWidth;
        indent = 0;
    }

    /**
     * Indent right.
     */

    public void indentRight() {
        indent += indentWidth;
    }

    /**
     * Indent left.
     */

    public void indentLeft() {
        if (indent > 0) {
            indent -= indentWidth;
        }
    }

    /**
     * Print an empty line to STDOUT.
     */

    public void println() {
        doIndent();
        System.out.println();
    }

    /**
     * Print the specified string (followed by a newline) to STDOUT.
     *
     * @param s
     *            string to print.
     */

    public void println(String s) {
        doIndent();
        System.out.println(s);
    }

    /**
     * Print the specified string to STDOUT.
     *
     * @param s
     *            string to print.
     */

    public void print(String s) {
        doIndent();
        System.out.print(s);
    }

    /**
     * Print args to STDOUT according to the pecified format.
     *
     * @param format
     *            format specifier.
     * @param args
     *            values to print.
     */

    public void printf(String format, Object... args) {
        doIndent();
        System.out.printf(format, args);
    }

    /**
     * Indent by printing spaces to STDOUT.
     */

    private void doIndent() {
        for (int i = 0; i < indent; i++) {
            System.out.print(" ");
        }
    }

}

```