Exercises: Basic Custom Tags

1. Make a random number tag that inserts a random number (double) between 0 and 1 into the page. For example, you might use it like this: `<sample:randomNum/>`. Use the Java-based approach where you extend `SimpleTagSupport`. Recall that `Math.random()` returns a double between 0 and 1.

Hint: start by grabbing some of my code from the basicTags project.

- Go to src/coreservlets.tags, copy SimplePrimeTag.java, and drop it into your project. (You can drop it into any package: Eclipse will automatically fix the package statement for you.) Then, erase the part that refers to prime numbers, and replace it with your logic. But the main class declaration and the signature of the `doTag` method will stay the same.
- Similarly, go to my project, go to WebContent/WEB-INF/tlds, and grab my TLD file. Drop it in your project in WebContent/WEB-INF or some folder inside WebContent/WEB-INF. Erase all the `tag` declarations from my TLD file, but keep the top-level `taglib` declaration. Insert your `tag` declarations inside.

2. Redo the previous tag using tag files.

3. Make a random int tag that inserts a random integer between 1 and some optional limit (default 10). For instance: `<sample:randomInt limit="782"/>`. If you multiple the result of `Math.random()` by the appropriate value, cast the result to an int, and add 1, you can easily generate random integers from 1 to some limit.

4. Redo the previous tag using tag files.

5. Make a tag that results in whatever text it encloses being displayed in a large, bold, red, blinking format. For example:
`<sample:annoying>This is a test</sample:annoying>`.

If you want to see the text actually blinking, test on Netscape 4, Netscape 7, or Firefox; Internet Explorer and Netscape 6 wisely ignore the `BLINK` tag. (Firefox actually lets you set whether or not `BLINK` is ignored, but it is displayed by default.)

6. Redo the previous tag using tag files.