Homework Assignment 7: Baby Names
EXTRA FEATURES

The following are four **extra features** you can optionally complete. Every student **must complete one** of these four features to get full credit on the assignment.

If you choose to do additional extra features beyond that, each additional extra feature you complete is worth **+1 extra late day** to be applied to this or future assignments. In your JavaScript file's comment header, indicate which feature(s) you have completed.

Regardless of how many additions you write, the page behavior and appearance **should still work as specified**. You may not modify the XHTML/CSS files; each feature must be done entirely through JavaScript. If you have a different idea for an addition to the program, please ask us and we may approve it.

Extra Feature #1. Better loading feedback

The page as currently described does not give the user very good feedback while it is loading its data, both when initially loading the list of names, and later when a name is chosen and that name's ranking and meaning data must be fetched.

For this feature, add code to do **all** of the following:

- Display a **"Loading" message** with a *throbber* image of some sort to indicate that the page is fetching data. You can find good Ajax "Loading" throbbers at http://ajaxload.info/ and other sites found by searching. Upload your throbber image to your Webster space and link to it using an absolute URL.

- **Disable the select box**, and modify the rest of the page visually to **indicate certain things are disabled** by "graying out" items or putting placeholder text into them.

- Change the user's mouse cursor to an **hourglass/watch cursor** using the CSS `cursor` property. http://www.w3schools.com/css/pr_class_cursor.asp

You should make the above changes as soon as you begin fetching data, and they should persist until **all** Ajax requests have been completed (i.e., not until just the first request comes back). After the last Ajax request comes back, your page should revert to its normal idle state so the user can select a new name—**even if any requests were unsuccessful**.

To help you test, the **babynames.php** service can accept an optional query parameter `delay`, whose value should be an integer for a number of seconds of delay that the server should wait before sending its response. For example, requesting the following URL would retrieve the rankings for the name "Stefanie" after a delay of 10 seconds:

https://webster.cs.washington.edu/cse190m/babynames.php?type=rank&name=stefanie&delay=10

Extra Feature #2. Ability to compare multiple names

By default the page displays data about one name at a time. For this feature, add code so that if a second name is chosen, the previous data remains, with the new data superimposed in a different color. This provides a way to compare their relative popularity. Add some kind of "Clear" button/feature so the user can erase previous names.

You can decide details such as what to show in the Origin/Meaning box, and how to show both sets of bars. There is no exact specification of the appearance, so long as both sets of bars can be seen. You can decide what should happen if the user chooses a third name: Does the first go away? Do all 3 show? Etc.

Extra Feature #3. Animation effects

Once the user chooses a baby name, use **Scriptaculous** (found in book Ch. 12, and in extra lecture slides not covered this quarter) to make the bars animate into existence rather than simply appearing immediately on the page. Add at least one other animation effect to the page, such as fading in ranking numbers or shrinking/growing an element on the page.

Extra Feature #4. Auto-complete babies' names

Use the DOM to remove the `select` box from the page and replace it with an `input` text box where the user can type a name. As the user types, use **Scriptaculous** (found in book Ch. 12, and in extra lecture slides not covered this quarter) to display any names in the data that start with or contain the partial text entered. The program should fetch rank/meaning data for the name when the user selects a from the auto-complete list and/or clicks a Search button that you inject into the page.

To help you write this feature, there are three additional optional parameters that our **babynames.php** service can accept when you are performing a list query. First is `format`, which, if set to `html`, will output the list of names as a `ul` list. The other two are `prefix` and `substring`, indicating an optional beginning or substring to filter on (case insensitive). For example, the following query would output a list of names in HTML that begin with "mar":

https://webster.cs.washington.edu/cse190m/babynames.php?type=list&format=html&prefix=mar

```
<ul>
    <li>Mara</li>
    <li>Maranda</li>
    <li>Marc</li>
    <li>Marcel</li>
    <li>Marcelina</li>
    ...
</ul>
```