

University of Washington, CSE 154

Homework Assignment 8: Baby Names

EXTRA FEATURES / CSE Majors Features

CSE majors must complete **at least one (1)** of the following extra features to add new functionality to the program. It is fine for a CSE major to complete more than one extra feature, but since this is the last assignment, no late-day reward will be given for doing so.

Non-CSE majors are not required to complete any of these extra features. If they do, those features will be ignored for grading as long as they do not break existing core functionality.

Regardless of how many additions you write, the page behavior and appearance should still work as described in the main spec document. If you have a different idea for an extra feature, please ask us and we may approve it. If you choose to do any of these extra features, **indicate in your file's comment header** which one(s) you did.

Extra Feature #1. Use a JavaScript library

Incorporate a JavaScript library such as Prototype, Scriptaculous, and/or jQuery into your page. If you choose to do this feature, you should try to embrace the preferred syntax and idioms of the library you use. For example, if you use Prototype, use the `$` and `$$` functions rather than `document.getElementById` and `querySelectorAll`, and use Prototype's `Ajax.Request` rather than the built-in `XMLHttpRequest`. Or if you use jQuery, use its `$` function to query the DOM and use its `$.ajax` function to perform Ajax behavior.

Extra Feature #2. Animation effects

Once the user chooses a baby name, make the bars and other data 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. To achieve this, you may want to use **Scriptaculous** (found in book Ch. 10, and taught in class) or the jQuery UI library.

Extra Feature #3. 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 #4. Auto-complete babies' names

Change the `select` box on the page into an `input` text box where the user can type a name. As the user types, display any names in the data that contain or start with the partial text. Fetch rank/meaning data for the name once the user selects a name from the auto-complete list and clicks the Search button.

One way to implement this feature would be to use an HTML5 `datalist` element. See the following link:

- http://www.w3schools.com/html/html5_form_elements.asp

Another way to implement auto-completion is to use the **Scriptaculous** auto-completer as found in the slides.

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. The first two are `prefix` and `substring`, indicating an optional beginning or substring to filter on (case insensitive). Another is `format`, which, if set to `html`, will output the list of names as a `ul` list. For example, the following query would output a list of names in HTML that begin with "mar":

- <https://webster.cs.washington.edu/cse154/babynames.php?type=list&format=html&prefix=mar>