

9 captures (/web/*/http://web.stanford.edu:80/class/cs142/project3.html)

one column for each day of the week.

- Once you have created the JavaScript class, we have provided a file `datepicker.html` containing two empty `div` elements, plus a bit of JavaScript code that invokes the `DatePicker` class to display a date picker in each of the `div`s. One of the date pickers initially displays the current month and the other displays the month of January 2009. It should be possible to change the month of each date picker *independently* using the controls on that date picker.

Problem 2: Simple Table Template Processing (15 points)


Requirements for the `fillIn` method:

-

For example if you find a `td` element in the matching column (assuming `columnName` is specified) that has the text string "The date is the {{day}} day of month {{month}} of the year {{year}}" and the dictionary object argument is `{month: "January", day: "30", year: "2016"}`, you should update the text of the `td` to be "The date is the 30 day of month January of the year 2016".

9 captures (<http://web.archive.org/web/20161016224300/http://web.stanford.edu:80/class/cs142/project3.html>)

Calling `tableTemplate.fillIn("table", dict, 'Part Number')`, where "table" is the table on the left and dict is a dictionary of strings, should generate the table on the right:

original table		filled-in table																				
<table><tr><th>{{PartNumber}}</th><th>{{Length}}</th></tr><tr><td>{{n14926}}</td><td>{{n47}}</td></tr><tr><td>{{n773}}</td><td>{{n3_5}}</td></tr><tr><td>{{n9318}}</td><td>{{n10}}</td></tr><tr><td>{{n3045}}</td><td>{{n4}}</td></tr></table>	{{PartNumber}}	{{Length}}	{{n14926}}	{{n47}}	{{n773}}	{{n3_5}}	{{n9318}}	{{n10}}	{{n3045}}	{{n4}}		<table><tr><th>Part Number</th><th>Length</th></tr><tr><td>14926</td><td>{{n47}}</td></tr><tr><td>773</td><td>{{n3_5}}</td></tr><tr><td>9318</td><td>{{n10}}</td></tr><tr><td>3045</td><td>{{n4}}</td></tr></table>	Part Number	Length	14926	{{n47}}	773	{{n3_5}}	9318	{{n10}}	3045	{{n4}}
{{PartNumber}}	{{Length}}																					
{{n14926}}	{{n47}}																					
{{n773}}	{{n3_5}}																					
{{n9318}}	{{n10}}																					
{{n3045}}	{{n4}}																					
Part Number	Length																					
14926	{{n47}}																					
773	{{n3_5}}																					
9318	{{n10}}																					
3045	{{n4}}																					

If the template specifies a property that is not defined in the dictionary object the template should be replaced with an empty string. Your system need only handle properly formatted templates. Its behavior can be left undefined in the case of nested templates (e.g., `{{foo {{bar}}}}`) or unbalanced `{{ (`). You do not need to handle nested tables or complex table cells.

You should use your `cs142-template-processor.js` solution from project 2 to help you implement the `fillIn` method. See the script tag ordering in the html file in the `project3` directory to see how your `cs142-template-processor.js` code would be loaded.

Beware that browsers insert a `<tbody>` element around all of the `<tr>` elements in a table, if the table doesn't already contain a `<tbody>`.

Once your function has processed the entire table you should examine the `visibility` of the table and if it is `hidden` update it to be `visible`.

Once you have created the JavaScript class, open the file `cs142-test-table.html` in your browser. This file represent an HTML page containing a sample template table that that will run your code and shows what your output should look like.

Additional Requirements, Hints, etc.

- You may not use jQuery or any other JavaScript library package in this project (or anywhere else in CS 142). If you were developing a real website then you should definitely take advantage of existing JavaScript libraries, but for this class we want you to learn about the low-level foundational JavaScript/DOM features.
- You may, however, use any JavaScript built-in functions. For example, you may find the following built-in functions useful in this project:
 - Methods of the Date class.
 - `parseInt`, `parseFloat`, `isNaN`.
- You may find some of the following DOM element properties useful in this project:
 - `cells`
 - `firstChild`
 - `nextSibling`
 - `rows`

- tagName
- textContent
- innerHTML



(https://archive.org/account/login.php)

http://web.stanford.edu:80/class/cs142/project3.html

Go

OCT (HTTP://WEB.ARCHIVE.ORG/WEB/201710162243

The JavaScript function `console.log` is very useful for debugging. It takes a string argument, which it prints to the JavaScript log. You can display this log with Tools -> JavaScript console in the Chrome control menu. If you are having trouble figuring out what is happening in your JavaScript, sprinkle

`console.log` statements around your code so you can see which code is being executed.

- You may also find the `debugger` statement useful in debugging. If the text `"debugger;"` is executed in a JavaScript program, it causes the program to drop into the JavaScript debugger. This is analogous to a breakpoint, except that your code can control when it triggers.
- If nothing seems to be happening in your JavaScript code, it's possible that your code contains an error that is causing it to be aborted. To find out if this is happening, open the JavaScript console to see if there are any errors.
- The JavaScript console and Chrome DevTools will be your best friends for the remaining projects. We encourage you to spend some time this project learning how these tools work. Feel free to stop by any of the office hours for a hands-on demonstration.

Style Points (10 points)

Your JavaScript must be clean (appropriate use of classes, no global variables other than constructor functions, etc.), readable, and JSHint warning-free.

Deliverables

Use the standard class submission mechanism (`submit.html`) to submit the entire `project3` directory.

Designed by Raymond Luong for CS142 at Stanford University

Powered by Bootstrap (<http://web.archive.org/web/20171227021106/http://getbootstrap.com/>) and Jekyll

(<http://web.archive.org/web/20171227021106/https://jekyllrb.com/>) – [learn more](#) ([website.html](#))