INFO2180 - Lab 6

Due Date: Tuesday, October 31, 2017

For this lab you will be required to work through an AJAX example.

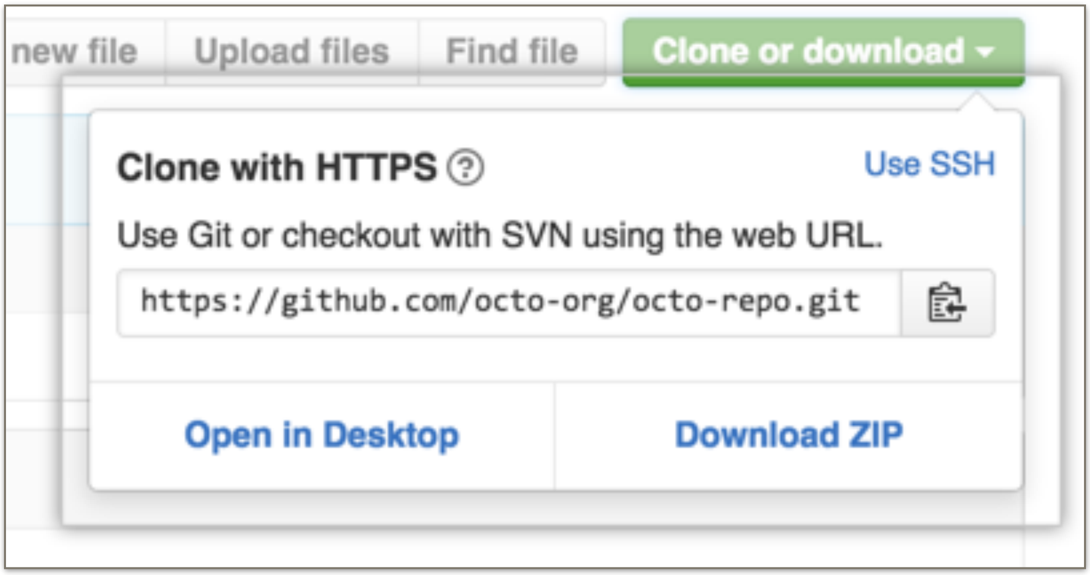
You will be required to submit the URL of both your Github account and your Cloud9 workspace by the deadline specified above.

Step 1 Get Started

To get started visit the following link and grab the starter code: https://www.github.com/uwi-info2180/info2180-lab6

Step 2 - Create a Github Repository

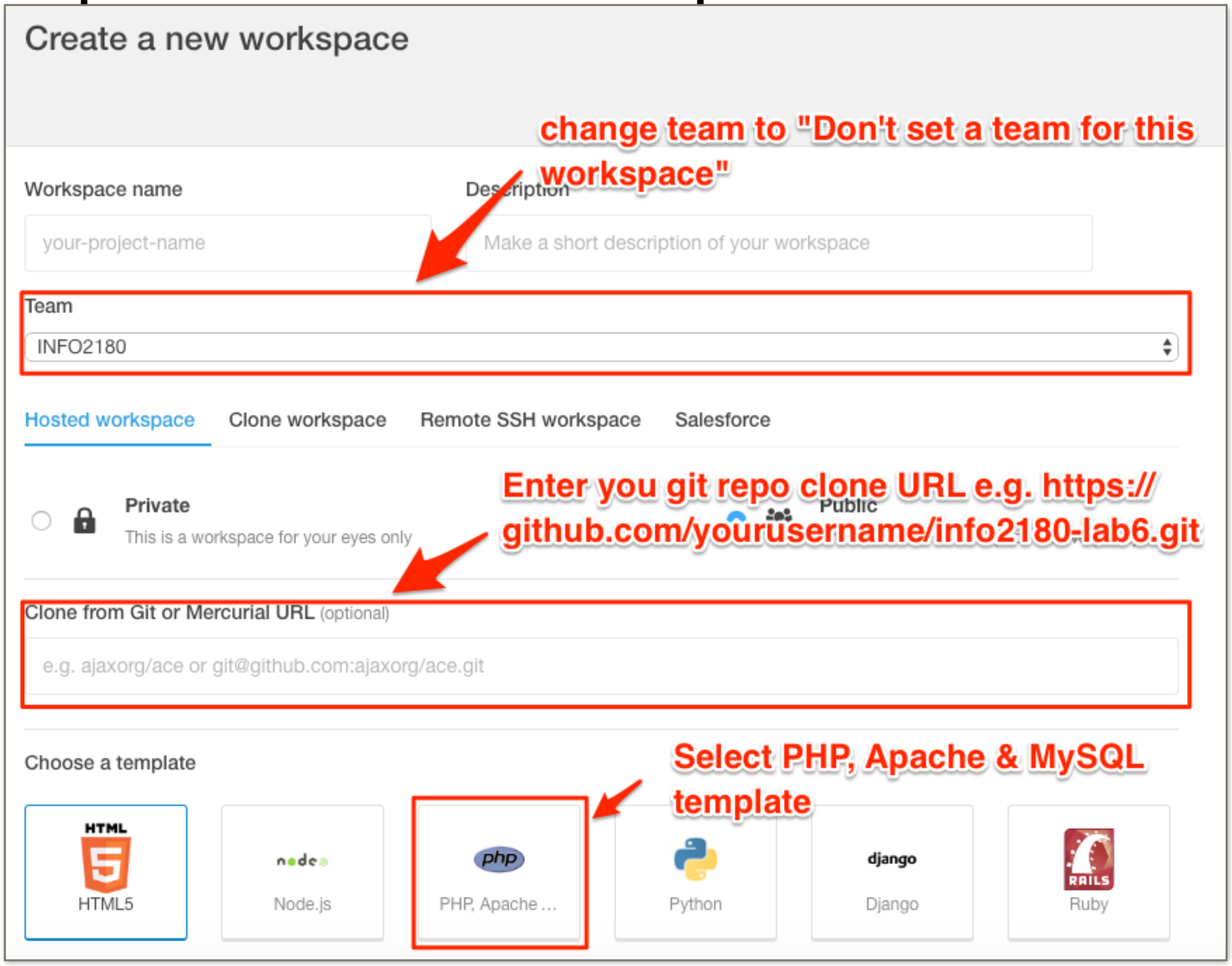
Create a Github repository called **`'info2180-lab6'`** and use the repository's '**clone**' url in the next step to clone the repository to your Cloud9 workspace.



Use Git or checkout with SVN using the web URL.

```
https://github.com/octo-org/octo-repo.git
```

Step 3 - Create a Cloud9 workspace


Create a new workspace

change team to "Don't set a team for this workspace"

Workspace name: your-project-name

Description: Make a short description of your workspace

Team: INFO2180

Enter you git repo clone URL e.g. https://github.com/yourusername/info2180-lab6.git

Private
This is a workspace for your eyes only

Public

Clone from Git or Mercurial URL (optional): e.g. ajaxorg/ace or git@github.com:ajaxorg/ace.git

Choose a template

Select PHP, Apache & MySQL template

Note: *You should have received an email at your UWI MyMona Email account with an invitation to sign up for Cloud9 without needing to enter any Credit Card details. Please check your student email account for the invitation that was sent to you. If you haven't received it, please let me know.*

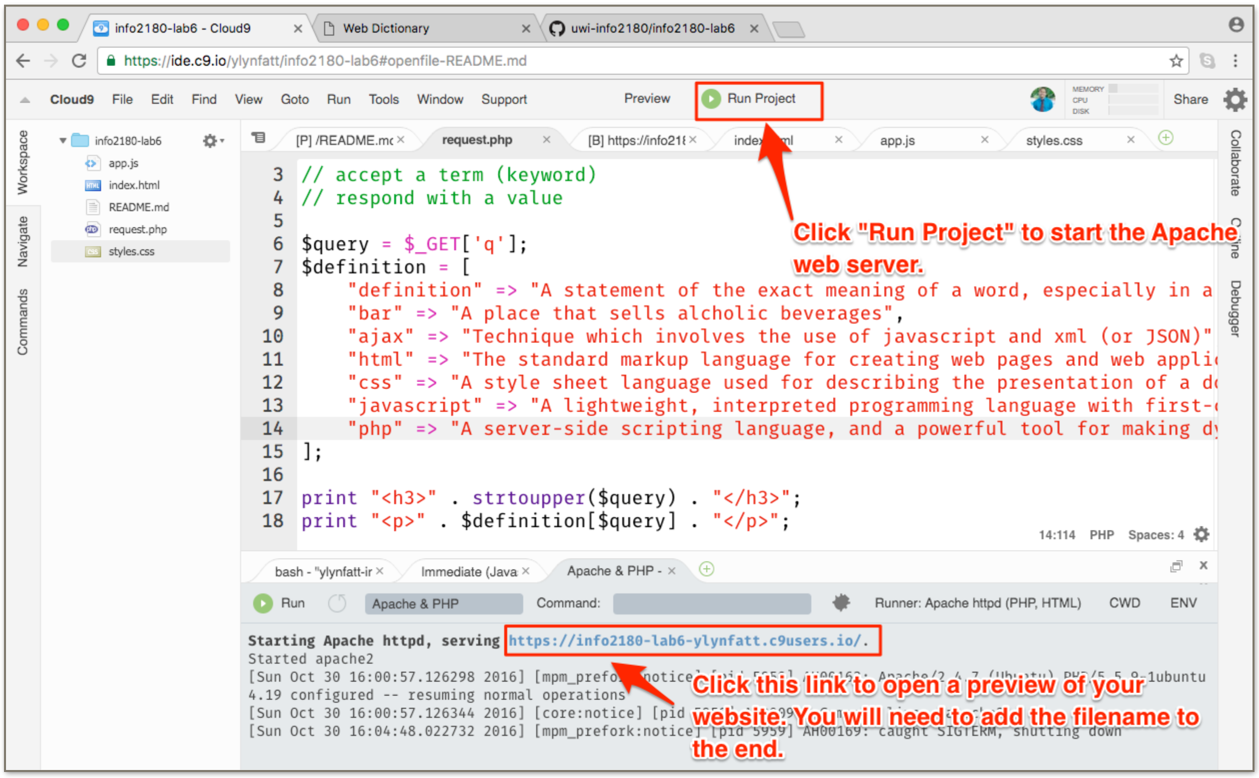
Create a workspace in the Cloud9 IDE https://c9.io/new.

Note: When creating the workspace ensure that you change the **"Team"** option to **"Don't set a team for this workspace"**, also ensure that you put the Github repository clone URL (from Step 2) into the **"Clone from Git or Mecurial URL field"** and also select the **PHP, Apache & MySQL** template.

Exercise 1 - Creating a PHP based AJAX Endpoint

Get the request.php file from the following repository https://github.com/uwi-info2180/info2180-lab6 .

Once you have the file saved in Cloud9, click the ‘**Run Project**’ button at the top of the IDE. This will start the Apache web server. In the Cloud9 console you should see the server start and a provide a link that you can click to browse to your website.



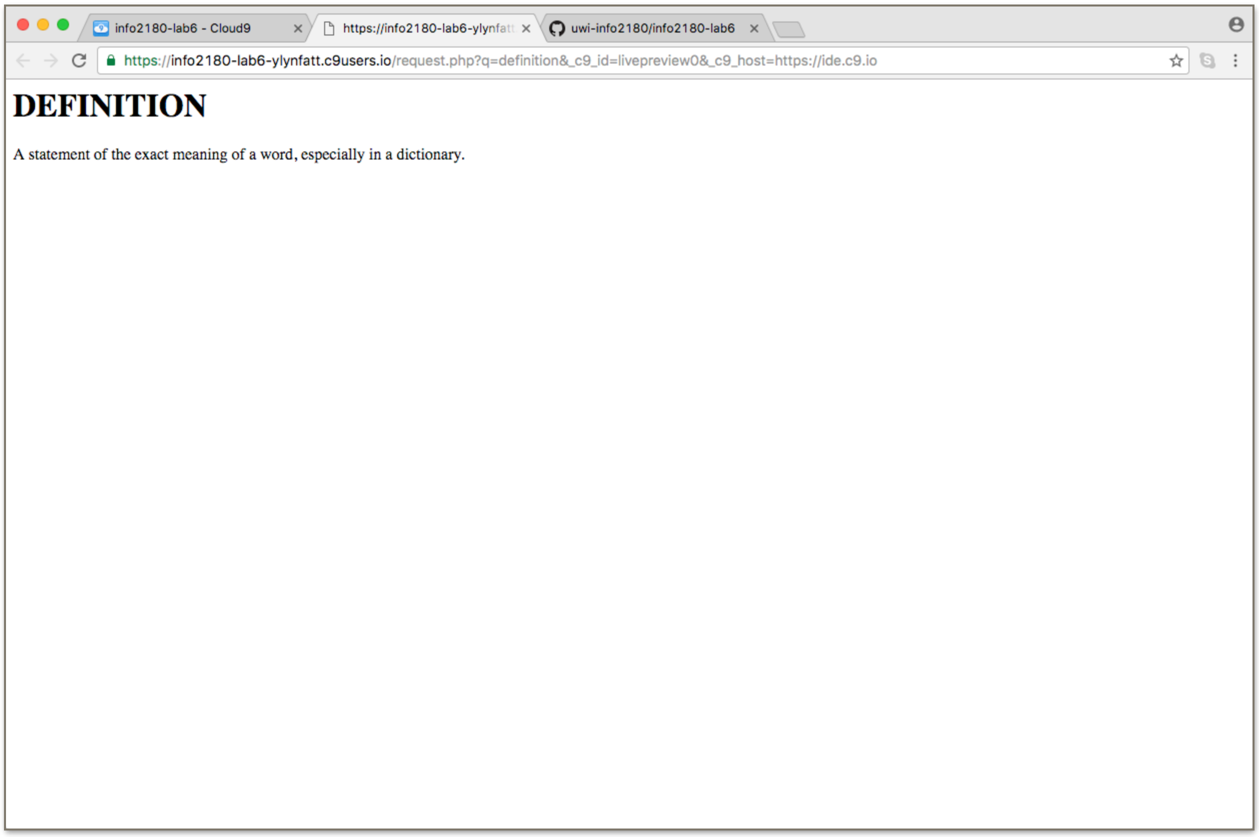
```
// accept a term (keyword)
// respond with a value

$query = $_GET['q'];
$definition = [
    "definition" => "A statement of the exact meaning of a word, especially in a
    "bar" => "A place that sells alcholic beverages",
    "ajax" => "Technique which involves the use of javascript and xml (or JSON)"
    "html" => "The standard markup language for creating web pages and web applic
    "css" => "A style sheet language used for describing the presentation of a do
    "javascript" => "A lightweight, interpreted programming language with first-c
    "php" => "A server-side scripting language, and a powerful tool for making dy
];

print "<h3>" . strtoupper($query) . "</h3>";
print "<p>" . $definition[$query] . "</p>";
```

```
Starting Apache httpd, serving https://info2180-lab6-ylynfatt.c9users.io/.
Started apache2
[Sun Oct 30 16:00:57.126298 2016] [mpm_prefork:notice] [pid 5959] AH00163: Apache/2.4.7 (Ubuntu) PHP/5.5.9-1ubuntu
4.19 configured -- resuming normal operations
[Sun Oct 30 16:00:57.126344 2016] [core:notice] [pid 5959] AH00094: Command line: 'apache2
[Sun Oct 30 16:04:48.022732 2016] [mpm_prefork:notice] [pid 5959] AH00169: caught SIGTERM, shutting down
```

Click the link and then add **‘request.php’** to the end. To the end of the url add **‘request.php?q=definition’** and you should see a page that looks like the following:


info2180-lab6 - Cloud9
https://info2180-lab6-ylynfatt
uwi-info2180/info2180-lab6
https://info2180-lab6-ylynfatt.c9users.io/request.php?q=definition&_c9_id=livepreview0&_c9_host=https://ide.c9.io

DEFINITION

A statement of the exact meaning of a word, especially in a dictionary.

Exercise 2 - Look up the word definition

Create an HTML page **`index.html`** and write some JavaScript code so that when a user clicks "**Search**", the definition of "**definition**" appears as a JavaScript **`alert`**.

Hint: Listen for a "**click**" event on a button. Fetch the data by opening an AJAX request which returns the result of **`request.php`**.

NOTE: You **MUST** make a commit after doing this exercise.

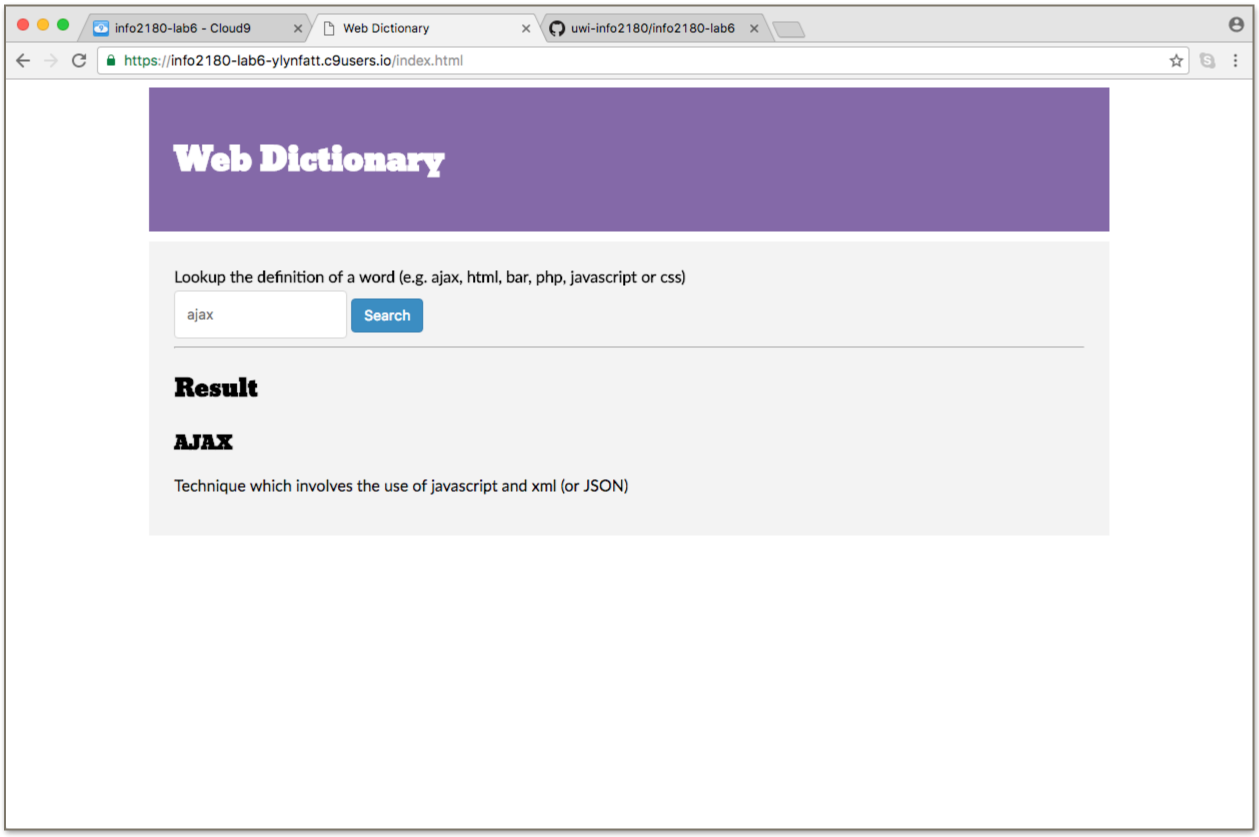
Exercise 3 - Look up any word (ajax, definition, bar, html, css, javascript or php)

Modify your code so that the definition of the word typed into a text field (not always "definition") is displayed.

- Some other words to test are: **`ajax, definition, bar, html, css, javscript or php`**.
- Show the result inside a **`div`** on your page with id of "**result**", rather than an alert.
- Ensure you are still using the **request.php** that was created earlier.

Hint: You can use the JavaScript innerHTML property (plain JavaScript) or the jQuery html() method if you are using jQuery.

The interface will look something like this (obviously you can call it whatever you would like and style it however you would like), but it MUST have the form field, button and display the output.


info2180-lab6 - Cloud9
Web Dictionary
uwi-info2180/info2180-lab6
https://info2180-lab6-ylynfatt.c9users.io/index.html
Web Dictionary
Lookup the definition of a word (e.g. ajax, html, bar, php, javascript or css)
ajax
Search
Result
AJAX
Technique which involves the use of javascript and xml (or JSON)

NOTE: You **MUST** make a commit after doing this exercise.

Exercise 4 - XML Data

Our Web Dictionary service can also send data in XML format. The XML data can return multiple definitions, and each definition comes with the author's name. To get the XML data, you will need to add another button **"Get All Definitions"** and pass our service a parameter via the URL with property **`all`**, set to a value of **`true`**.

Note:
You will need to alter **`request.php`** before this will work.

1. It will need to be able to return ALL values from your data when the **"Get All Definitions"** button is clicked.
2. It will need to be able to return XML see example https://github.com/uwi-info2180/info2180-ajax/blob/master/xmldata.php

The following is an example of the URL call for your AJAX request:

`request.php?q=&all=true`

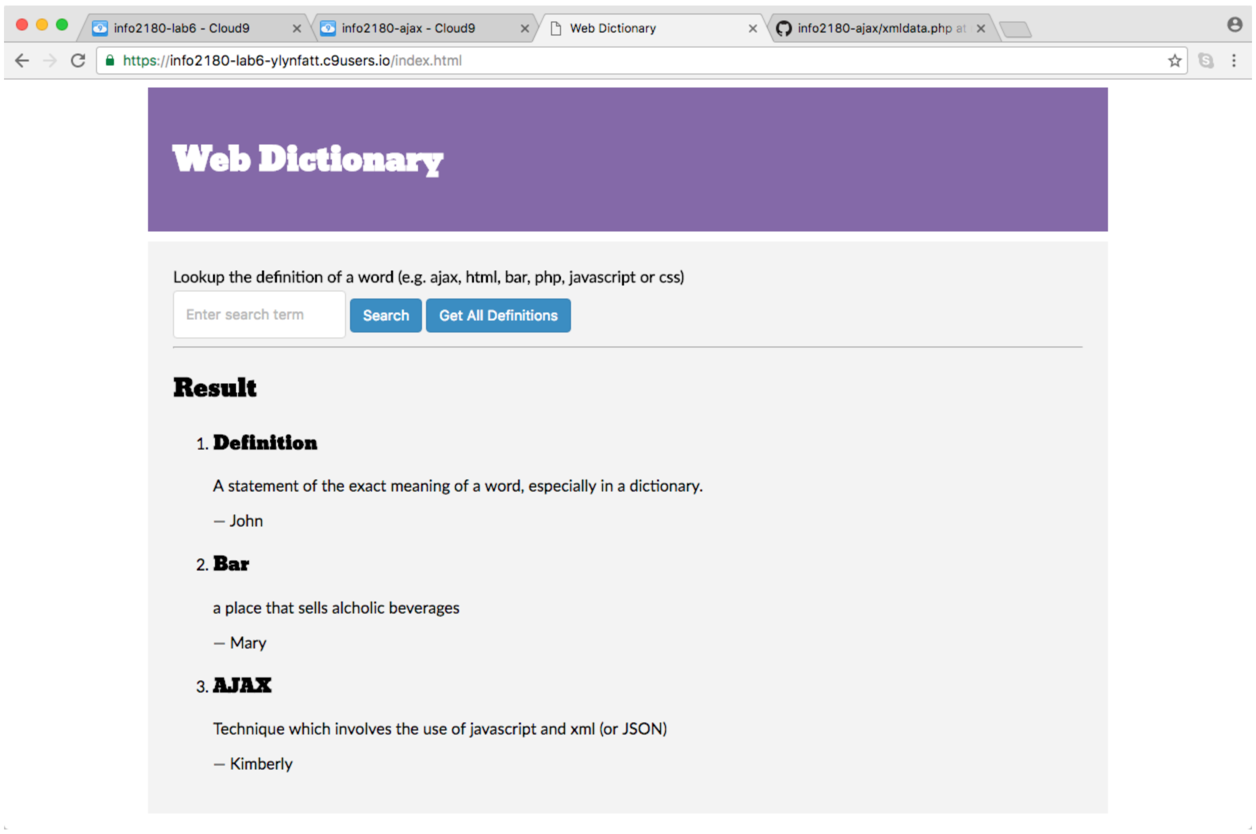
and the returned XML should look something like this:

```
<?xml version="1.0" encoding="UTF-8"?>
<entries>
    <definition name="definition" author="John">
        a statement of the exact meaning of a word, especially in a
dictionary.
    </definition>
    <definition name="bar" author="mary">
        a place that sells alcholic beverages
    </definition>
    <definition name="ajax" author="Kimberly">
        technique which involves the use of javascript and xml
    </definition>
</entries>
```

Possible steps:

Modify your code to display each of the definitions of the word:

- Create an ordered list (`ol`) of definitions to insert into the result div, using the DOM (`document.createElement`).
- For each definition in the XML, place a new list item (`li`) inside your ordered list.
- Insert a heading and two new paragraphs into each definition's `li`:
  - The name of the definition as a heading (e.g. `h3`)
  - The definition text in a paragraph
  - The author who submitted the definition in a paragraph (preceded by a dash)
- Use the Ajax request's `responseXML` property if you are using plain JavaScript or the `dataType` parameter set to `xml` if using jQuery’s `ajax()` method.



Web Dictionary

Lookup the definition of a word (e.g. ajax, html, bar, php, javascript or css)

Result

1. **Definition**

   A statement of the exact meaning of a word, especially in a dictionary.

   — John

2. **Bar**

   a place that sells alcholic beverages

   — Mary

3. **AJAX**

   Technique which involves the use of javascript and xml (or JSON)

   — Kimberly

NOTE: You **MUST** make a commit after doing this exercise.

Submission

Submit your information via the **"Lab 6 Submission"** link on OurVLE, with your Github repository URL (e.g. https://github.com/yourusername/info2180-lab6) and your Cloud9 workspace URL (eg. https://ide.c9.io/yourusername/info2180-lab6).

Note: You are not expected to create a Github page for this lab as Github is not able to interpret PHP.