

281 Project 2

Michael Hennessy, Computer & Information Science, UO

Project 2

Due: 23:59 Fri 4/24

Learning Objectives

In the 21st century, every business— from fashion to finance—is a tech company. Every company has a web presence, and many have smart phone apps. Coding and web dev, therefore, are high-demand skills for every job, and differentiate you from most people in the job market. All employers rank STEM skills very highly. By completing this project you will practice the following skills:

- A web API or *Application Programming Interface* is an API accessed using the HTTP protocol.
- Web APIs often deliver data formatted as JSON.
- A web API can be accessed programmatically using JavaScript.
- The API documentation must first be consulted in order to understand how the API works and what it returns.
- Some web APIs are public, and others require an access key.

Flickr offers several public APIs.

Due: 23:59 Fri 4/24

Github. This leaves your local and remote repos in the same state.

SET UP

1. **Required:** Sign in to our Canvas course, read *How to Set your Canvas Notification Preferences*, and set your notifications so that you don't miss important announcements.
2. Accept the Project-2 invitation link:
<https://classroom.github.com/a/YZfXqfnj>

Github will create your private repo, which you can access by [signing in to Github](#).

GIT & GITHUB TIPS

1. **Github Tip: never edit files directly on Github.** Always make your changes in your local repo, and then push them to

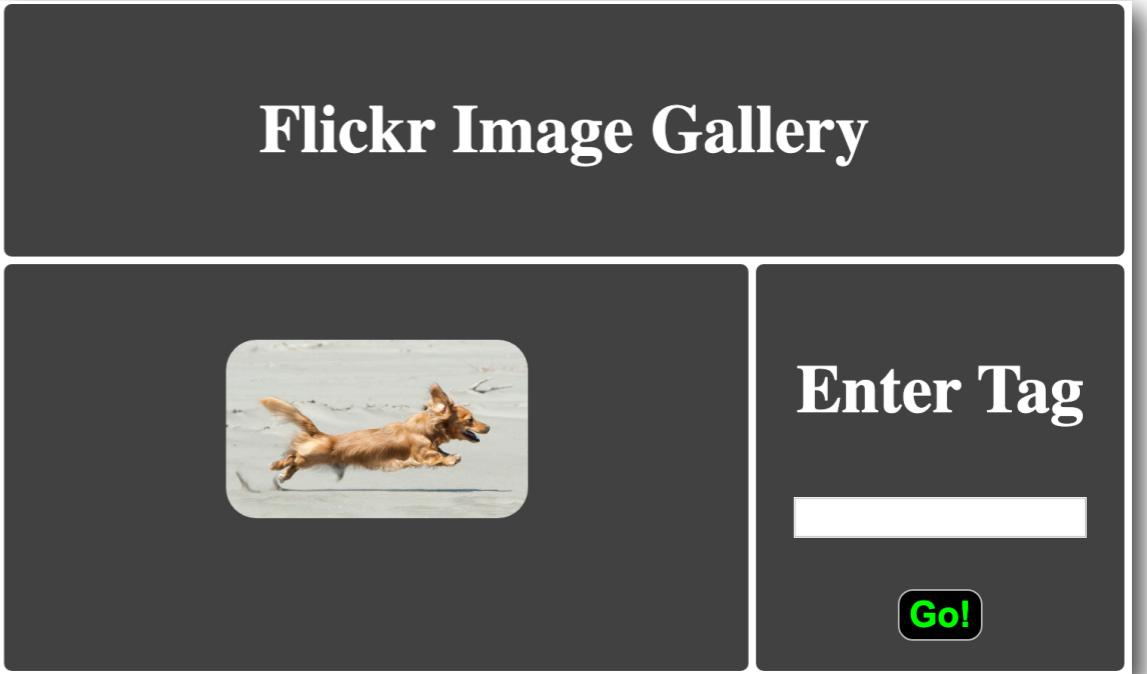
PROJECT LEARNING OBJECTIVES

- A web API or *Application Programming Interface* is an API accessed using the HTTP protocol.
- Web APIs often deliver data formatted as JSON.
- A web API can be accessed programmatically using JavaScript.
- The API documentation must first be consulted in order to understand how the API works and what it returns.
- Some web APIs are public, and others require an access key. Flickr offers several public APIs.
- In this project, you will use Flickr's public API.

PROJECT REQUIREMENTS

1. [20 pts] Flickr Image Gallery.

- a) Use Atom's File > Add Project Folder.. command to add your project-2-yourGitHubUserName folder.
- b) In Atom, open *gallery.html*, *gallery.css*, and *gallery.js*.
- c) Rename *main* as *controller*.
- d) Modify the .html file as follows: a) Add a text box; b) add a style rule to the .css file, with a rule for the input element that sets the font size to 18px; c) Modify the headers as shown; d) Add an *images* folder to the *client* folder, containing one image of your choice; e) When the page loads, the image is displayed, fitting on the page, as shown. Here's an example:

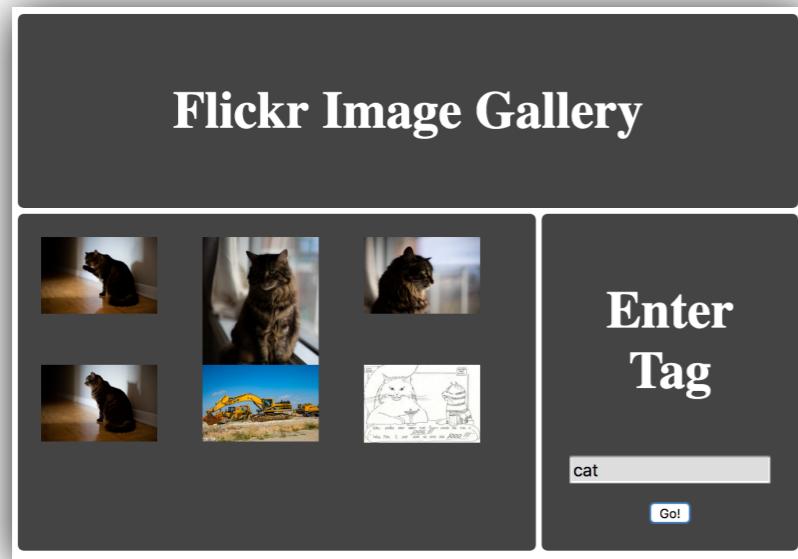


- d) Modify the web app so that the user can enter a tag value into a textbox on the web page (cat or duck, for example), and then click the button to retrieve photos related to the tag.

This is how to register the *onclick* handler:

```
//without using jQuery
window.addEventListener("load", function() {
    //select the button and register the handler
    document.querySelector("button").addEventListener("click",
        controller);
});
```

- e) Test the web app. Enter *cat* in the textbox and click *Go!* Six images of cats should be displayed:



When complete, stage, commit, and push your changes to the *master* branch your Github repo.

2. [40 pts] Lab Week 3: The Articles API.

Complete all parts of the of the week 3 lab.

3. [30 pts] Coding Practice.

In Atom, create a new file named *functions.js* in your *javascripts* directory.

a) Write a function named *revWords1* that accepts a sentence or phrase (a string) and returns a string with the words in s reversed. Words are delimited by a single space. Use a *for* loop, and *arrow* syntax. Hint: *String.split* method.

Examples:

```
reverseAllWords("Web App Dev") -> "beW ppA veD"  
reverseAllWords("Delta Echo") -> "atleD ohcE"
```

b) Save *revWords1* as *revWords2*. Modify the function to use the *Array.forEach* method.

c) Save *revWords 2*as *revWords3*. Use a *for..of* loop, and *arrow* syntax.

Stage, commit, and push your changes to the *master* branch your Github repo.

4. [10 pts] Edit your README.md file to add the learning objectives for this project.

In Atom, edit *README.md* by adding the following markdown elements:

a) An h2 header with this content: *281 Project 2 Learning Outcomes*. Following the header, add the learning objectives for Project 2.

b) Add a second h2 header with this content: *281 Project 1 Learning Outcomes*. Following the header, copy and paste the learning objectives from project 1.

f) Save, stage, and commit your changes. Then push your changes to the *master* branch on your Github repo.

In Chrome, reload your Github repo to verify the push succeeded.

5. [+5pts XC] Optional Extra Credit.

In *functions.js*, write a function named *containsDuplicates* that accepts a string *s*, and returns true if *s* contains duplicate characters. To receive credit, you must use an array, the *charCodeAt* method, a *for..in* loop, and arrow syntax.

Start by creating an array of size 26, with all elements initialized

to zero: `let charCount = new Array(26).fill(0);`

Examples:

```
containsDuplicates("Web App Dev") -> true  
containsDuplicates("Od or Ev") -> true  
containsDuplicates("Nth") -> false
```

When done, *stage*, *commit*, and *push* your changes to the master branch on your Github repo.

Meeting the Deadline

How to Handle the Deadline

- Start working on your project early. Do not delay.
- Turn in what you have by the deadline-- partial credit is better than none.



[Instructors] are a Superstitious Sect, Great Keepers of Set Times and Places.

-- from *Poor Richard's Almanac*

