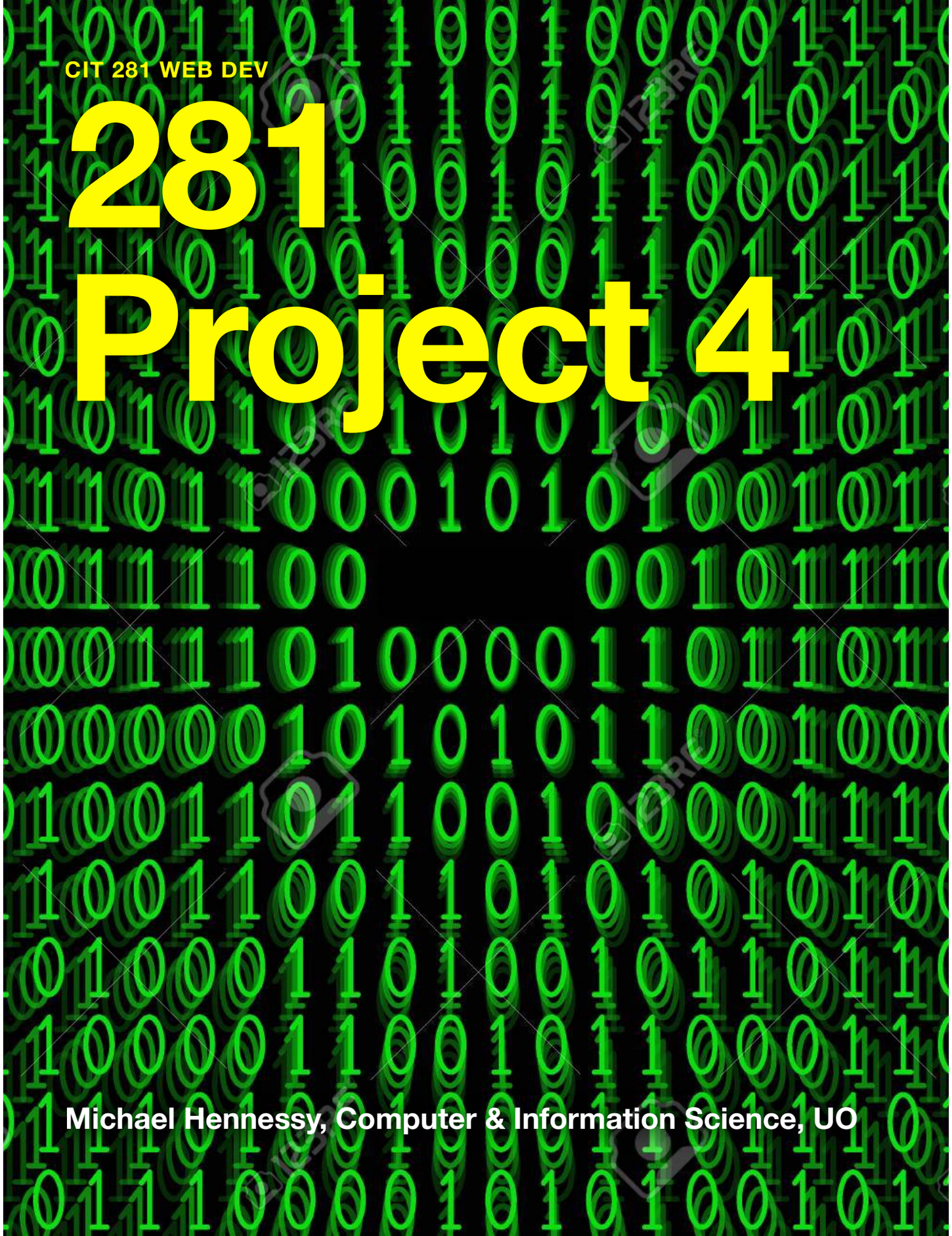
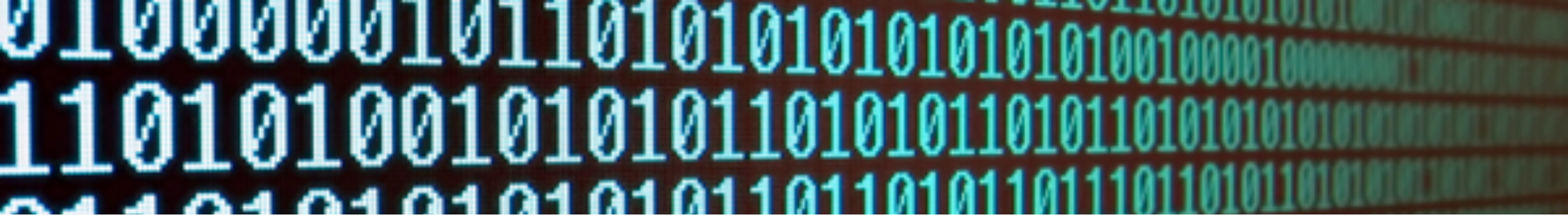

CIT 281 WEB DEV
281
Project 4
Michael Hennessy, Computer & Information Science, UO

CIT 281 Web Dev

Project 4

Learning Objectives

- Build a To-Do app that uses the Web Storage API
- The Web Storage API lets browsers store key/value pairs locally, to create a persistent data store
- ES6 Class Types: The Basics
- ES6 Class Types: Inheritance



Due Dates:

- **23:59 Fri 5/22 (XC)**
- **23:59 Mon 5/25**

GIT WORKFLOW FOR 281

Commits are usually made when you've completed something you are working on-- it's done, and you won't be changing it.

A good time to do a commit is when you are finished with a project requirement.

The Basic Git Workflow

```
$ git status
$ git add .
$ git commit -m ". . ."
$ git push origin master
$ git status
$ git log
```

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

Github Tip: never edit files directly on Github. Always make your changes in your local repo, and then push them to Github. This leaves your local and remote repos in the same state.

CREATE YOUR PROJECT-4 GITHUB REPO

- Sign in to Github.

- Paste this Project-4 repo invitation link into Chrome:
  https://classroom.github.com/a/ODDvhGv-

- Github will create a remote private repo for you, and add it to your other repos on Github.

Important: Notify me right away if your project-4 repo is not available on Github-- I can manually create a repo for you.

- Use the **`git clone`** command to download the remote repo to your computer

- Use Atom's `File > Add Project folder..` command to open your `project-4` repo folder

PROJECT REQUIREMENTS

1. **[20 pts] Add the Project 4 Learning Objectives to your README.md File.**

   a) In Atom open README.md. Note that it currently describes the Start-State of your repo. Replace that content with the Project 4 Learning Objectives.

   b) Open your README.md file from project 3. It currently has the learning objectives for P3, P2, and P1.

   c) Copy all the project learning objectives from your project 3 README.md. and add it to the end of your your project 4 README.md. When done, your P4 README will have the learning objectives for projects P4, P3, P2, and P1, in descending order.

   d) When you are done, use the *Basic Git Workflow* commands to *stage, commit*, and *push* your changes to the *master* branch on your remote repo.

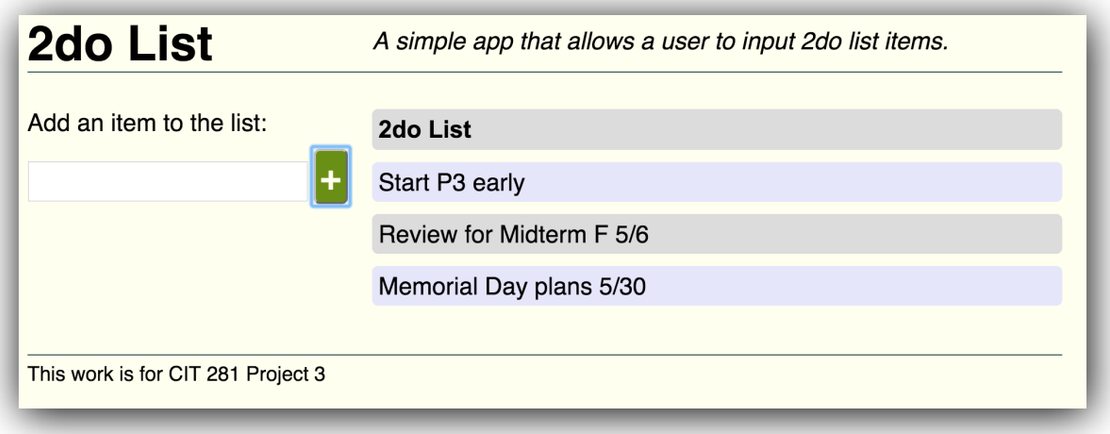
2. **[40 pts] Creating a To-Do App**. The Comments App is covered in ch. 4 of our textbook. See *Our First Interactive App*, pp. 99-110.

   For this requirement, you will modify the comments app to create a To-Do app.

   The To-Do app will use the Web Storage API to store list items.

   a) Read about the Web Storage API.

   b) Modify the .html file so that the app looks like this:



c) Modify the .js file to use the Web Storage API.

- The comments app adds new comments by creating a new paragraph element and appending it as a new child of a section element. The list of comments is maintained as a list of paragraph elements. That means comments do not persist--when you reload the page, the page returns to its start state, and you lose all the comments you added.

- Using the Web Storage API as a Data Store: For the To-Do app, save the list of paragraphs elements in local Storage.

- The contents of Local Storage persist even when the browser is closed and reopened.

- Note that the list of paragraph elements can be accessed as the *innerHTML* of the *section* element that contains them.

- jQuery's *$.html()* method can be used to both get and set the *innerHTML* of the *section* element.

- Writing the list to localStorage can be done with one statement:

```
localStorage.setItem("toDoList", $(".comments").html());
```

- When the To-Do app first loads in your browser, the web page should be populated with the list in localStorage, which again can be done with a single statement:

```
$(".comments").html(localStorage.getItem("toDoList"));
```

- This statement will have to be inside an *if* statement. If there is no *toDoList* key in localStorage, *getItem* returns *null*

d) When you are done, use the *Basic Git Workflow* commands to *stage, commit*, and *push* your changes to the *master* branch on your remote repo.

3. **[40 pts] JavaScript Class Types: The Animal Class.**

   We've learned about how to create objects in JavaScript using *object literal syntax*, and *JSON*.

   In ES6, includes a “class” construct, that introduces new

features which are useful for object-oriented programming. Learn about class types by reading the following:

`Class Types: Basic syntax`
https://javascript.info/class

`Class Types: Inheritance`
https://javascript.info/class-inheritance

a) In Atom, open *Animal.js*. Fill in the details for the class *Dog*, such that, when you run the test driver, the following output is produced:

```
> test_driver()
  White Rabbit runs with speed 5.
  White Rabbit hides!
  White Rabbit stopped.
  White Rabbit hides!
  White Rabbit
  10
  Fang runs with speed 20.
  Fang is a Airedale dog weighing 50 lbs. that can run 20 mph.
  Look, a cat! Fang barks: Woof Woof!
  Fang stopped.
  Woof Woof!
  Zero runs with speed 10.
  Zero is a Beagle dog weighing 22 lbs. that can run 10 mph.
  Look, a cat! Zero barks: Yip Yip!
  Zero stopped.
  Yip Yip!
```

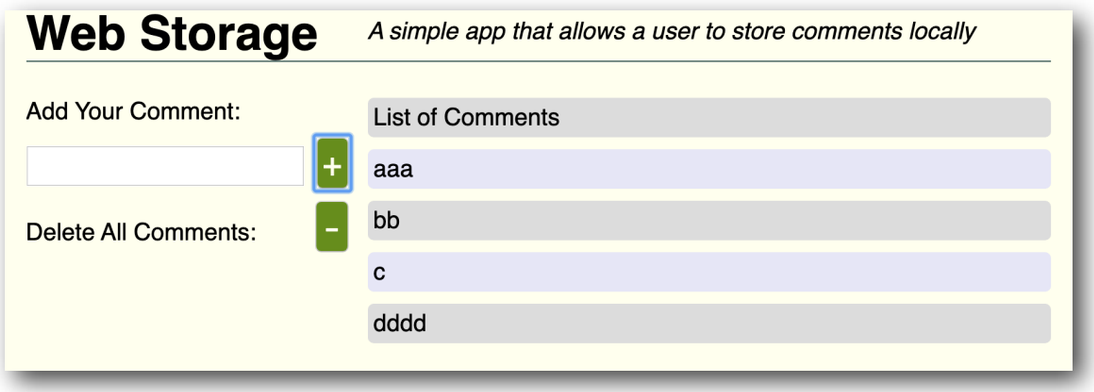
b) When you are done, use the *Basic Git Workflow* commands to *stage, commit*, and *push* your changes to the *master* branch on your remote repo.

4. **[+5pts XC] Optional Extra Credit**.

   a) Save `to-do.html` as `to-do-XC.html`. Save `to-do.js` as `to-do-XC.js`.

   b) Add a `delete` button to the `.html` file, as shown below. When the button is clicked, the to-do items are removed from `localStorage`.

   c) When you are done, use the *Basic Git Workflow* commands to *stage, commit*, and *push* your changes to the *master* branch on your remote 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*