ICT-4570 Homework 4B

Purpose

This exercise allows you to include a JavaScript file for data, and use it to populate your form. You will also get a taste for timers.

What to Hand In

Canvas submission instructions:
Please combine multiple files into a single "zip" archive, and save it in a location that you will remember. When you are ready to submit the assignment solution, open the assignment in Canvas, and click on the Submit for Evaluation button at the top, attach the file, and click Submit for Evaluation at the bottom.

Problems

I. Politics and poker part 2
   1. Incorporate the provided data source, and use it to populate your election results.
   2. There are several sets of data for the candidates in the data set, each at a different index in the array.
   3. After applying each update, set a timeout after which the next update will be applied.

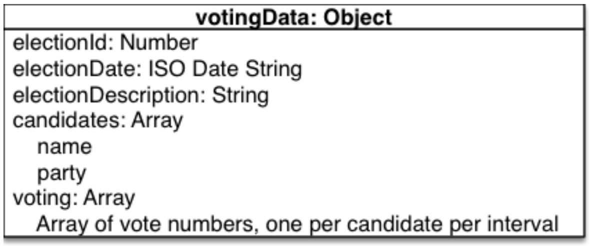
Notes

- This data object, which is in the supplied `voting41.js`, is incorporated by adding a script tag with the data as its src
- The structure of the JavaScript object is depicted in the picture.
- Since the candidates will only be set once, do this at initialization time.
- You may find it convenient to change the form: say, changing some input fields which are now read-only to span or output elements
- To organize this to work smoothly, you will want a function which takes an array and just updates the vote totals, votes, and percentages.

```
function updateVoteCount(array) {
  "use strict";
  // The code to update the votes
  // Call your existing function that
  // computes the vote total and percentages
}
```

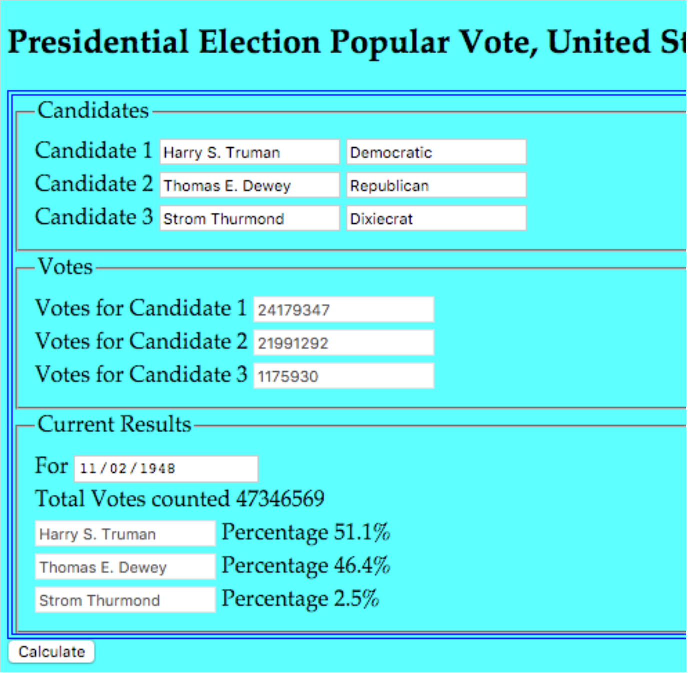
- You will also want a function that remembers manages sending the next data via timeout. It will rely on the `voting` part of the JavaScript object. Here is one way to write it:

```
var timeDelayMs = 5000; // 5-second delay
function voteUpdates(voting,myIndex) {
  "use strict";
  // The if test prevents the voteUpdates
  // function from recursing forever
  if ( voting.length > myIndex ) {
    updateVoteCount(voting[myIndex]);
    // Recursively call voteUpdates function after a delay, passing the voting array and the next ind
    setTimeout(voteUpdates, timeDelayMs, voting, myIndex+1);
  }
}
```



votingData: Object
electionId: Number
electionDate: ISO Date String
electionDescription: String
candidates: Array
  name
  party
voting: Array
  Array of vote numbers, one per candidate per interval

Description of voting41.js data structure



Presidential Election Popular Vote, United St

Candidates
Candidate 1 Harry S. Truman Democratic
Candidate 2 Thomas E. Dewey Republican
Candidate 3 Strom Thurmond Dixiecrat

Votes
Votes for Candidate 1 24179347
Votes for Candidate 2 21991292
Votes for Candidate 3 1175930

Current Results
For 11/02/1948
Total Votes counted 47346569
Harry S. Truman Percentage 51.1%
Thomas E. Dewey Percentage 46.4%
Strom Thurmond Percentage 2.5%

Sample Screen

- Your initialization function will need to change to ensure the candidates get set (and their parties if you wish). Things that might be updated once include:
  - Update the title and heading with the electionDescription and electionId
  - Set the date from the electionDate
  - Set up the candidate names (and parties)

  The last thing it should do is call voteUpdates with the voting array and index 0.
- Following is a copy of the data from `voting41.js`

```
var votingData = {
  electionId: 41,
  electionDate: "1948-11-02",
  electionDescription: "Presidential Election Popular Vote, United States",
  candidates: [{
      name: "Harry S. Truman",
      party: "Democratic"
    },
    {
      name: "Thomas E. Dewey",
      party: "Republican"
    },
    {
      name: "Strom Thurmond",
      party: "Dixiecrat"
    }],
  voting: [
    [
      336122, 612446, 711326
    ],
    [
      4663122, 7001244, 822506
    ],
    [
      10437916, 12443910, 930971
    ],
    [
      17623004, 18680779, 1060304
    ],
    [
      24179347, 21991292, 1175930
    ]
  ]
};
```

Evaluation

Criteria	Weight
HTML and CSS updated to properly display assignment data	20
Function to apply vote array to form	20
Application of candidate names to inputs	20
Set up timeout to apply the next update at the desired interval	20
Screenshots of results.	20