

ICT-4570 Homework 3

Purpose

This exercise allows you to use control logic in JavaScript, and to experience how arithmetic is done.

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

1. Build a simple form containing: Inputs for at least two candidate names, candidate parties, an "as-of" date, and places for raw vote counts.
2. Also put on the form two output locations for total votes, and percentages for each candidate, which will be reported with 1 decimal place.
3. The form should also have a button entitled Calculate which will tie to a function to read the inputs and place the output results.
4. Be sure to add JavaScript logic to catch errors, or other, incorrect input and properly handle it: Unparsable dates and other errors should be reported or corrected upon pressing the calculate button. See notes for detailed checks.
5. Your JavaScript functions should accomplish the following:
 - A function to look up an HTML element by *id* and return the element
 - A function which takes an *Array* of numbers as a parameter, and returns the sum.
 - A function which takes an *Array* of numbers as a parameter, and returns an array of percentages (see notes for this computation).
 - A function which formats a number into a percentage (see notes for this computation).
 - A function to collect all the vote counts and return an *Array*, ordered by candidate, of their vote counts.
 - A function to (a) collect all the counts; (b) get the sum of all votes; (c) compute the percentages for each candidate; and (d) place that result on the form in the correct locations.
 - A function to initialize the form by setting the date to the current date, and attaching the click action to the click button.

Election Calculator

The screenshot shows a web form titled "Election Calculator". It is divided into three main sections: "Candidates", "Votes", and "Current Results".

- Candidates:** Contains three rows. Each row has a label (Candidate 1, 2, 3), a text input field for the name, and a dropdown menu for the party. The data entered is: Candidate 1: Pat, Democratic; Candidate 2: Robin, Independent; Candidate 3: Alex, Republican.
- Votes:** Contains three rows, each with a label (Votes for Candidate 1, 2, 3) and a text input field for the vote count. The data entered is: 12440, 13960, and 11990.
- Current Results:** Contains a date input field labeled "As of" with the value "12/20/2015", and three lines of text showing the results: "Total Votes counted 38390", "Pat Percentage 32.4%", "Robin Percentage 36.4%", and "Alex Percentage 31.2%".

At the bottom of the form is a "Calculate" button.

Figure 1: Sample form with data filled in and computation completed

Notes

- Please ensure the user sees error text in **red**
- Be sure to handle the following situations explicitly (comment if no code is required):
 - Input is blank
 - Number input is not a number
 - Number input is zero or negative
 - Date is not parseable (see note below)
- Include screenshots of the cases you're handling through JavaScript code.
- If you post errors, warnings, or log messages to the console, include the console in the screenshot.
- Some browsers enable the input type to be selected to prevent illegal values from being selected. You are welcome to use these, but please remember that browsers are NOT obligated to support these, and some will just treat them as input text fields (so you still need to code the checks).
- Remember that getting elements by *id*, *class*, or *tag name* is pretty straightforward.
- To compute a *percentage* from a *vote count*, simply divide the candidate's *vote count* by the *total count*
- To format a percentage, multiply the number by 100, convert the number with *toFixed(n)*, where *n* is replaced with the number of decimal places desired, and add the string "%".
Example of *toFixed*: (234.5).toFixed(2) will yield "234.50"
- *input* fields of type date need an ISO-formatted date. Unfortunately, JavaScript doesn't have a function to take care of

this (yet). The following function will return a date as an ISO string, suitable to assigning to the input field's value property.

It might be called like this:

```
today = new Date();
isodate = toISODate(today);
```

toISODate:

```
function toISODate(date) { // yyyy-mm-dd
  "use strict";
  var yyyy, mm, dd;
  // JavaScript provides no simple way to format a date-only
  yyyy = "" + date.getFullYear();
  mm = date.getMonth() + 1; // Months go from 0 .. 11
  dd = date.getDate();
  // Need leading zeroes to form the yyyy-mm-dd pattern.
  if (mm < 10) {
    mm = "0" + mm; // This converts it to a string
  }
  if (dd < 10) {
    dd = "0" + dd; // This converts it to a string
  }
  return "" + yyyy + "-" + mm + "-" + dd;
}
```

- Just for fun: If you add a button and attach a function that calls `print()`, you can print your result (or save them to, say, a PDF if your dialog allows). Interestingly, the `window.print` specification is in the HTML specification document!

Evaluation

Criteria	Weight
HTML and CSS, to create the form	40
JavaScript to read inputs and create output	40
Screenshots of results, including error cases.	20