CSE 154: Web Programming   Autumn 2018

Practice Midterm Exam 3

Note: We strongly recommend printing out practice exams and working through them with only your cheatsheet (provided on the course website) - it's important to be comfortable taking a spec and writing code on paper without the convenience of autocomplete/debugging tools!

Also note that provided exams adapt problems from previous quarter exams, but the number/format of problems may be different (see other provided practice exams for other example problems). This exam in particular is a bit longer than we'd expect for 50 minutes.

Name:

UWNet ID: @uw.edu

TA (or section):

Rules:

- You have 60 minutes to complete this exam.
- You will receive a deduction if you keep working after the instructor calls for papers.
- This is a closed-note exam, but you may use the provided cheatsheet for reference. As noted on the cheatsheet, you may assume $, qs, and qsa are provided in JS as shorthand for document.getElementById, document.querySelector, and document.querySelectorAll, respectively.
- You may not use any electronic or computing devices, including calculators, cell phones, smartwatches, and music players.
- Unless otherwise indicated, your code will be graded on proper behavior/output, not on style.
- Do not abbreviate code, such as writing ditto marks ("") or dot-dot-dot marks (…). You may not use JavaScript frameworks such as jQuery or Prototype when solving problems.
- If you enter the room, you must turn in an exam and will not be permitted to leave without doing so.
- You must show your Student ID to a TA or instructor for your submitted exam to be accepted.

Question	Score	Possible
HTML Validation		
CSS and the DOM		
JS/DOM/UI		
JS/Animations		
Short Answer		

1. What's wrong with my HMTL?

Consider the following HTML:

```html
<!DOCTYPE html>
<html lang="en">
    <head>
        <script src="index.js"></script>
        <link href="styles.css" rel="stylesheet"/>
    </head>
    <body>
        <h1 "Best page ever!" />
        <div>
            <a>Check out this cool page:
              <href>http://www.pointerpointer.com</href>
            </span>
        </div>
    <body>
</html>
```

This HTML document won't validate, and would generate errors and warnings in the W3C Validator. However, it is possible to make 5 modifications to the HTML to make it pass validation. Each modification might result in multiple text "changes" to the HTML document, but is considered one modification because it is addressing the same root problem.

Indicate the 5 modifications we need in order to make it pass validation. Write directly on the HTML. Below, briefly describe the changes that you made, and why you had to make each change in order to validate. If it is unclear what changes go with which explanations, feel free to number your changes on the HTML. No need for an essay — 10 words or less should be plenty.
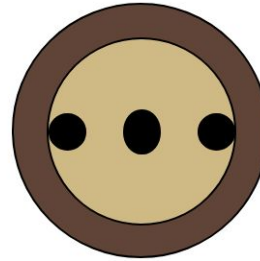
1. _____

2. _____

3. _____

4. _____

5. _____

2. Cute, Slow, and Sleepy.

In this problem, you will **A**. finish drawing a DOM tree of a provided HTML body and **B**. write CSS with the provided HTML to produce the expected page output below (appearance details are given to supplement the expected output image where needed).

```
<body>
    <main>
        <header>
          <h1>Baby Sloth!</h1>
        </header>
        <div id="s">
            <div id="l">
                <span id="o"></span>
                <span id="t"></span>
                <span id="h"></span>
            </div>
        </div>
    </main>
</body>
```

Baby Sloth!

Part A (Drawing the DOM Tree): Finish the DOM tree for the body, using only nodes for HTML elements **(ignore ids and any text content).** We have started the tree with the body and section element for you - the final tree should have exactly as many nodes as there are HTML elements.

```
body
  |
main
```
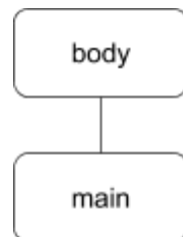
Write the CSS for **Part B** on the following page.

Part B (Writing CSS): Appearance Details:

- The `main` element is 40% of the page's width. The text and sloth image are centered on the page.
- The `h1` text has a font family of Helvetica, falling back to Arial if Helvetica is not available on the system, falling back to the system's sans-serif default font if Arial is also not available.
- The background color of the outermost circle is `sienna` and the background color of the inner circle containing the "eyes" and "nose" (which each have a black background) of the sloth is `peru`.
- The two div circles have a 1px-width solid black border and a border radius of 50%.
- The outermost circle has a width and height of 150px. Its inner circle has a width and height of 110px.
- The `#t` span is positioned in the center of the face and has a height of 25px - the `#o` and `#h` spans both have a height of 20px and touch the very left and right borders of the parent `#l` div, respectively. All span elements have a width of 20px and a border radius of 50%.

Write your CSS solution below:

3. Define-It!

In this problem, you will write JavaScript to add functionality to a personal dictionary webpage that allows a user to manage an entry list of terms and definitions.

Define-It!

Term: [_____]

Definition:

[_____]

[Add Entry!]

Empty dictionary view (initial/after removing all added dictionary entries)

Define-It!

Term: [zetetic____]

Definition:

[proceeding by inquiry or investigation_____]

[Add Entry!]

Current dictionary entries:

- oikofugic: Relating to or characterized by the desire to travel, migrate, or wander
- shallop: a light sailing boat used chiefly for coastal fishing
- uroboros: a circular symbol depicting a snake (or a dragon) swallowing its tail, intended as an emblem of wholeness or infinity

Entering an Entry for "Zectetic"

Define-It!

Term: [_____]

Definition:

[_____]

[Add Entry!]

Current dictionary entries:

- oikofugic: Relating to or characterized by the desire to travel, migrate, or wander
- shallop: a light sailing boat used chiefly for coastal fishing
- uroboros: a circular symbol depicting a snake (or a dragon) swallowing its tail, intended as an emblem of wholeness or infinity
- zetetic: proceeding by inquiry or investigation

Clicking "Add Entry" given term/definition of left screenshot

Requirement Details

- When the page loads, there are no dictionary entries on the page.
- There is a text input with id #term for a user to type a term and a text area input with id #definition for a user to enter the corresponding definition. If the users clicks the "Add Entry!" button, the current word and definition should be added to the list of current dictionary entries if both the term and definition input areas are non-empty. Whenever a new entry is added to the entry list, both term and definition input areas should be cleared. Nothing should happen if either the term or definition fields are empty.
- Each entry should be added as a single item to the #entries list in the format TERM: DEFINITION, replacing TERM with the value typed in the term input box and DEFINITION with the value typed in the definition text area.
- Whenever there are dictionary entries in the #entries list, #current-entries should be visible, otherwise it should be hidden. The linked CSS has a hidden class which sets display: none for an element with that class. You should use this class when hiding/showing an element.
- Whenever a user double-clicks on an entry in the #entries list, that entry should be removed from the list. Any other entries should maintain their order in the list whenever an entry is removed.

```html
<!DOCTYPE html> <!-- Provided HTML for Question 3 -->
<html lang="en">
    <head>
        <meta charset="UTF-8">
        <title>Define-It!</title>
        <link href="dictionary.css" rel="stylesheet" />
        <script src="dictionary.js"></script>
    </head>
    <body>
        <section>
            <header>
                <h1>Define-It!</h1>
            </header>
            <fieldset id="entry-input">
                Term: <input id="term" type="text" /><br/>
                Definition:<br/>
                <textarea rows="5" cols="40" id="definition"></textarea>
                <button id="add-entry">Add Entry!</button>
            </fieldset>
            <article id="current-entries" class="hidden">
                <header>
                    <h2>Current dictionary entries:</h2>
                </header>
                <ul id="entries"></ul>
            </article>
        </section>
    </body>
</html>
```

Write your JavaScript solution below (continue on next page as needed):

```javascript
(function() {
  "use strict";
```

```javascript
// Continue your JavaScript solution to Problem 3 below as needed:
```

```javascript
})(); // end module pattern
```

4. JavaScript DOM and Animations

Given the provided HTML and CSS, implement graph.js. After the window loads, a new point div should be added **within** the #graph element at a **random position** every 1 second.
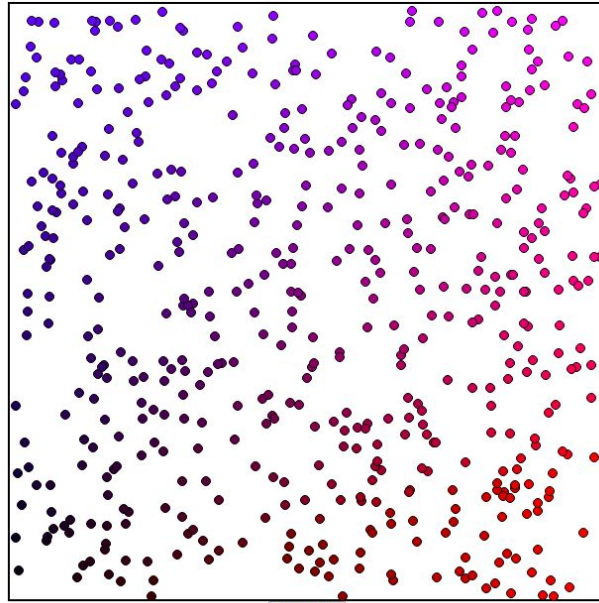
Output after a bunch of points have been added

Additionally, when you create a point, you are to set the background color of the point based on the top and left coordinates that you randomly picked for the location:

- The green color component of all the points is 0.
- The red component of the color is determined by the horizontal position
- The blue component is determined by the vertical position of the point.

The red and blue components start at a value of 0 (in points at the left and bottom edges, respectively), and have a max value of 255 (in points at the right and top edges, respectively), and scale linearly in between.

Requirements:

Each point is to appear at a random location in the #graph - both the top and left coordinates must be integers - the circle representing each point must fit entirely within the #graph div - you may not change the #graph div in any way, besides appending point elements inside it - every possible (integer pair) location must have the same chance of a point appearing there - the color of a point is based off of the coordinates of it's top/left location - when any point is double-clicked, it should be removed from the page - you must correctly encapsulate your JavaScript in a module

Hints:

- This problem is harder if you try to calculate/work with the center-points of the circles. It is much easier if you only think about a point as its top/left coordinates - though, remember that the points in the upper right corner of the #graph are the most blue and the most red
- The CSS rgb() color value doesn't work with floating point numbers -- it needs integers - to figure out if a circle fits inside the #graph, determine if a square that circumscribes the circle fits in the #graph

Use the following HTML and CSS as reference for Problem 4.

```html
<html>
    <head>
        <script src="graph.js"></script>
        <link rel="stylesheet" href="graph.css"/>
    </head>
    <body>
        <div id="graph"></div>
    </body>
</html>
```

```css
#graph {
  position: relative;
  width: 600px;
  height: 600px;
  margin-left: auto;
  margin-right: auto;
  border: 2px solid black;
}

.point {
  position: absolute;
  width: 8px;
  height: 8px;
  border-radius: 5px;
  border: 1px solid black;
}
```

Write your JS solution on the next page.

Start your JS here (remember you may assume you have $, qs, and qsa defined):
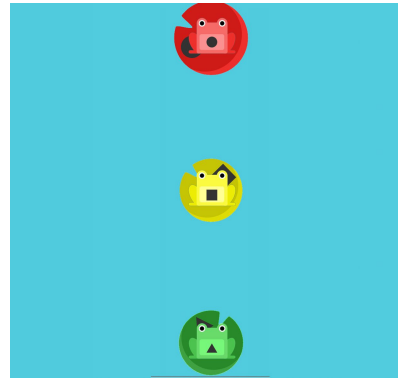
```
(function() {
  "use strict";
```

```
})(); // end of module pattern
```

```
(function() {
  "use strict";
```

5. Short Answers

1. If you were playing Flexbox Froggy and you needed to get the Froggies from their starting position (shown on the left) to the ending position on the lilypads (shown on the right), what flex rules would you add to the pond CSS selector? (Note: Because this printed in greyscale, you must match the symbols on the frogs with the symbols on the pads. The lilypad symbols may appear rotated from what the frog symbols are).
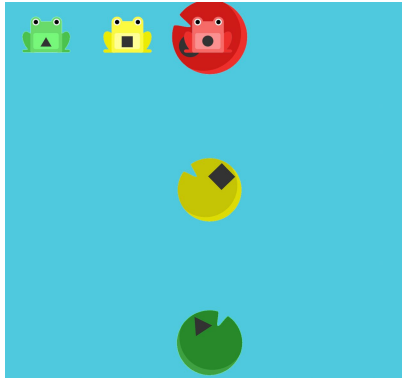
Flexbox Froggy starting position   Flexbox Froggy ending position

Add the rules to the ruleset below:

```
#pond {
  display: flex;



}
```

2. Why is it important to limit the use of module-global variables in our JavaScript programs?

3. What is the role of the id of a timer returned by `setTimeout` or `setInterval`? In particular, when do we need it?

4. Consider the following HTML body:

```html
<body>
  <button id="my-btn">Click Me!</button>
  <p id="result"></p>

  <script>
    document.getElementById("my-btn").addEventListener("click", function() {
      document.getElementById("result").innerHTML = "you clicked the button!";
    });
  </script>
</body>
```

Because the <script> tag is at the bottom of the body, this actually shows "You clicked the button!" on the page when the button is clicked. But based on what we've learned about HTML/CSS/JS so far, what are 2 issues *related to the script tag and JavaScript code* which demonstrate poor code quality? Briefly justify both of your answers (1 sentence is sufficient), and provide a better alternative to each (ensuring the behavior and appearance of the page is still the same). Do not use commenting/indentation/spacing as your answers.

Issue A and justification: Better alternative:
Issue B and justification: Better alternative:

5. Consider the following JSON declaration:

```javascript
let mystery = {
  "i" : ["j", 0, 1],
  "ii" : "ii",
  "I" : "i"
};
```

Write the JavaScript value that would be returned for each of the following statements. Include "" around any string values. Hint: at least one value is `undefined`.

Statement	Value
mystery["i"]	
mystery[0]	
mystery.ii.length	
mystery["i"][0].length	