CSE 154: Web Programming

Practice Midterm Exam 1

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.
- 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.
- This is a closed-note exam, but you may use the provided cheatsheet for reference. As noted on the cheatsheet, you may assume id, qs, and qsa are provided in JS as shorthand for document.getElementById, document.querySelector, and document.querySelectorAll, respectively.
- 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		5
CSS		5
JS/DOM		15
JS/Animations		15
Short Answer		10

1. What’s Wrong with my H⊥ML?

```
<!DOCTYPE html>
<head>
    <h1>Mowgli's Magical Muffins</h1>
    <link src="mypage.css" rel="stylesheet" />
</head>
<body>
    <p>For Doggies' Best Friends:</p>
    <ul>
        <li>Multi-grain Melody</li>
        <li>Merry-Mint-Chip</li>
    </ul>
    For Doggies:
    <ul>
        <li>The Malt-ese</li>
        <li>Malamint Magic<li>
        <li>Meow Meows</li>
    </ul>
</body>
</!DOCTYPE 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. You Selected the Right Class.

Consider the following HTML:

```html
<html>
    <head>
        <title>CSE 154 Course Web Page</title>
    </head>
    <body>
        <header id="title-1">
            <h1 id="title-2"><em id="em-1">All the CSE 154 Course Stuffff Ever</em></h1>
        </header>
        <p id="subtitle-1">Topics:</p>
        <ul id="list-1">
            <li id="topic-1">What is the Internet</li>
            <li id="topic-2">How to do the Internet</li>
            <li id="topic-3">How to make the Internet</li>
            <li id="topic-4">
                Make cool projects:
                <ol id="list-2">
                    <li id="hw-1">Make Pies</li>
                    <li id="hw-2">Watch Lion King</li>
                    <li id="hw-3">Read <em id="em-2">rly rly rly</em> fast</li>
                    <li id="hw-4">Push squares around</li>
                    <li id="hw-5">Catch 'em all!</li>
                </ol>
            </li>
        </ul>
        <div id="div-1">
            <img id="img-1" src="mowgli.jpg">Our course mascot!</img>
        </div>
    </body>
</html>
```

Write the ID’s of the elements selected by each of the given selectors:

1. `p`

____

2. `ol li`

____

3. `li em`

____

4. `ul > li`

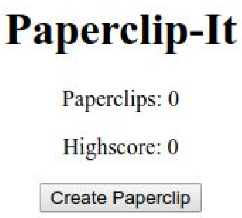
____

5. `li li`

____

3. Gotta Make That 📎📎📎

Write a JavaScript program to add to the given HTML and CSS (next page). This program lets a user run their own paperclip factory at the click of a button! Who knew your new web development skills could come in handy?



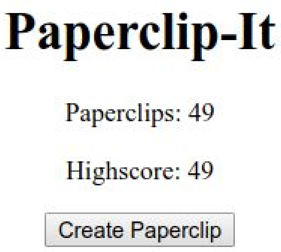
Paperclip-It

Paperclips: 0

Highscore: 0

Your job is to simulate creating a single paperclip whenever the “Create Paperclip” (`#paperclip-it`) button is clicked. When the page is initially loaded, the user has 0 paperclips. When the button is clicked, the current paperclip count (in `#count`) should be incremented by 1.

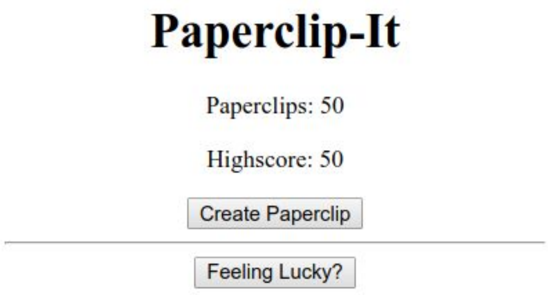
Feeling Lucky: The first time the user generates 50 paperclips, they unlock the ”Feeling Lucky” option, which should display the `#secret-div` (this div is already hidden with a .hidden class when the page is loaded). When visible, the user can test their luck at the paperclip jackpot by clicking on the “Feeling Lucky” button. When a user clicks “Feeling Lucky” then there should be a 25% chance that the user will double their paperclips, and a 75% chance that they instead lose all their paperclips (the “Feeling Lucky” button still remains visible).



Paperclip-It

Paperclips: 49

Highscore: 49



Paperclip-It

Paperclips: 50

Highscore: 50

Before user unlocks “Feeling Lucky” button

After user unlocks “Feeling Lucky” button

Highscores: Below the paperclip count is the user’s current highscore (the highest number of paperclips they have ever had). You should update the high score (in #highscore) whenever the current paperclip count increases and is higher than the current high score. For example, if a user’s highscore is 12 paperclips and they go from having 12 to 13 paperclips, their highscore should become 13 as well. If their highscore was instead 57 and they went from 12 to 13 paperclips, the highscore would remain 57.

Write your solution (JavaScript) on the next page, under the provided HTML and CSS.

```html
<!DOCTYPE HTML> <!-- HTML for Problem 3 -->
<html>
    <head>
        <link href="paperclips.css" rel="stylesheet" />
        <script src="paperclips.js"></script>
    </head>
    <body>
        <h1>Paperclip-It</h1>
        <p>Paperclips: <span id="count">0</span></p>
        <p>Highscore: <span id="highscore">0</span></p>
        <button id="paperclip-it">Create Paperclip</button>
        <div class="hidden" id="secret-div">
            <hr />
            <button id="lucky-button">Feeling Lucky?</button>
        </div>
    </body>
</html>
```

```css
/* CSS for Problem 3 */
body {
  margin: auto auto;
  text-align: center;
  width: 30%;
}
.hidden {
  display: none;
}
```

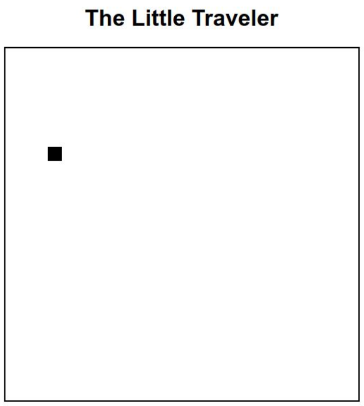
Write your solution (JavaScript) on the next page.

Continue your solution (JavaScript) for Problem 3 here if needed.

4. The Little Traveler (JS)

Given the HTML and CSS on the following page, write the JavaScript code that adds a `.little-box` div to the top left corner of the `#box` div, and moves the little box inside of the `#box` 20px up, down, left, or right randomly every 100ms. The little box may only move to a position that is inside of the boundaries of the parent `#box`. Note that the `#box` parent has a width and height of 500px, and the `.little-box` div has a width and height of 20px (without any border):

To the right is a screenshot of the little box during an animation. Write your JavaScript solution on the next page following the provided HTML and CSS:



```html
<!DOCTYPE html> <!-- HTML for Problem 4 -->
<html lang="en">
    <head>
        <link href="traveler.css" rel="stylesheet" />
        <script src="traveler.js"></script>
    </head>
    <body>
        <h1>The Little Traveler</h1>
        <div id="box"></div>
    </body>
</html>
```

```css
/* CSS for Problem 4 */
h1 {
  font-family: Helvetica, Arial, sans-serif;
  text-align: center;
}
.little-box {
  background-color: #000;
  height: 20px;
  position: absolute;
  width: 20px;
}
#box {
  border: 2px solid black;
  height: 500px;
  margin: auto auto;
  position: relative;
  width: 500px;
}
```

5. Short Answers

1. What is the difference between inline elements and block elements?

2. Why do we always want to include an alt attribute on img tags?

3. What's the difference between margin, borders, and padding? (You may provide a labeled diagram)

4. Why is it important to specify multiple font styles for the same element in your CSS? (e.g., `font-family: Helvetica, Arial, sans-serif;`)

5. Why is it important to use the module pattern in JavaScript?

6. What is the difference between `setInterval` and `setTimeout`?

7. Consider the following JSON object:

```
let miniJSON = {
  "foo" : ["b", 1, 2],
  "bar" : 0,
  "FOO" : "Foo?"
};
```

For each of the following statements, write the value that would be returned (include ”” around any string values; if any expression would result in an error, write error. If any expression would return the value undefined or null, specify this as your answer.

```
a. miniJSON.foo : ________

b. miniJSON["FOO"] : ________

c. miniJSON["FOO"][1] : ________

d. miniJSON[foo] : _______

e. miniJSON["foo"].length : _______
```