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.
- 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.

<table>
<thead>
<tr>
<th>Question</th>
<th>Score</th>
<th>Possible</th>
</tr>
</thead>
<tbody>
<tr>
<td>HTML</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>CSS</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>JS/DOM</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>JS/Animations</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>Short Answer</td>
<td></td>
<td>10</td>
</tr>
</tbody>
</table>
1. What's Wrong with my HTML?

<!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-eze</li>
    <li>Malamint Magic</li>
    <li>Meow Meows</li>
  </ul>
</body>
</!DOCTYPE html>

Solution (any 5 of the following received full-credit):
1. The link tag needs the href attribute, not src
2. No content tags should be in <head> - <h1> should be moved into <body>
3. All text in the body should be in a content tag - “For Doggies;” could be in a <p> tag to fix this
4. There's no such thing as a closing </!DOCTYPE html> tag
5. The "Malamint Magic" should be followed by </li>, not <li>
6. Missing <html> and </html>
2. You Selected the Right Class.

Consider the following HTML:

```html
<html>
  <heading>
    <title>CSE 154 Course Web Page</title>
  </heading>
  <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>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>
```

**Solution:**

1. `p`  
   `#subtitle-1`
2. `ol li`  
   `#hw-1, #hw-2, #hw-3, #hw-4, #hw-5`
3. `li em`  
   `#em-2`
4. `ul > li`  
   `#topic-1, #topic-2, #topic-3, #topic-4`
5. `li li`  
   `#hw1, #hw-2, #hw-3, #hw-4, #hw-5`
3. Gotta Make That

Solution:

(function() {
    "use strict";
    let highscore = 0;

    window.addEventListener("load", function() {
        $('#paperclip-it').addEventListener("click", makePaperclip);
        $('#lucky-button').addEventListener("click", feelingLucky);
    });

    function makePaperclip() {
        let count = getPaperclipCount();
        if (count > highscore) {
            highscore = count;
        }
        if (count >= 50) {
            $('#secret-div').classList.remove("hidden");
        }
        setPaperclipCount(getPaperclipCount() + 1);
        updateHighscore();
    }

    // unlocked at 50 ppclips
    function feelingLucky() {
        let chance = Math.random();
        if (chance < 0.25) {
            setPaperclipCount(getPaperclipCount() * 2);
            updateHighscore();
        } else {
            setPaperclipCount(0);
        }
    }

    function getPaperclipCount() {
        return parseInt($('#count').innerText);
    }

    function setPaperclipCount(n) {
        $('#count').innerText = n;
    }

    function updateHighscore() {
        let paperclipCount = getPaperclipCount();
        if (paperclipCount > highscore) {
            highscore = paperclipCount;
            $('#highscore').innerText = highscore;
        }
    }
})();
4. The Little Traveler (JS)

Solution:

(function() {
    "use strict";

    window.addEventListener("load", function() {
        addBox();
        timer = setInterval(updateBox, 200);
    });

    function updateBox() {
        let box = qsa(".little-box")[0];

        let sides = []; // top, right, down, left
        let topSide = parseInt(window.getComputedStyle(box).top);
        let leftSide = parseInt(window.getComputedStyle(box).left);
        if (topSide >= 20) {
            sides.push("top");
        }
        if (leftSide >= 20) {
            sides.push("left");
        }
        if (topSide <= 480) {
            sides.push("bottom");
        }
        if (leftSide <= 480) {
            sides.push("right");
        }
        let randomSideIndex = Math.floor(Math.random() * sides.length);
        let randomSide = sides[randomSideIndex];
        if (randomSide == "top") {
            box.style.top = topSide - 20 + "px";
        } else if (randomSide == "bottom") {
            box.style.top = topSide + 20 + "px";
        } else if (randomSide == "right") {
            box.style.left = leftSide + 20 + "px";
        } else if (randomSide == "left") { // left
            box.style.left = leftSide - 20 + "px";
        }
    }

    function addBox() {
        let littleBoxCount = qsa(".little-box").length;
        let littleBox = document.createElement("div");
        littleBox.classList.add("little-box");
        $("box").appendChild(littleBox);
    }
})();
5. Short Answers

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

**Solution:** Inline elements (e.g. `<a>`, `<span>`, etc.) do not start a new line and have a default width of their content. Block elements (e.g. `<h1>`, `<section>`, `<p>`, etc.) do start a new line, and span 100% width of their parent element.

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

**Possible Solutions:**

- Users who cannot see the image due to vision impairment can have a textual description of the image (which can be spoken aloud by a screenreader)
- If the image fails to load (connection, broken path, etc.), the alt text is displayed instead
- SEO (Search Engine Optimization) benefits for page ranking

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

**Solution:**

![Diagram of Margin, Border, Padding, Content]

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;`)

**Solution:** To specify fallback fonts in case the primary font is not available on the system, with a system default font sharing the same font type as the preferred (earlier) fonts (e.g. serif, sans-serif, monospace, or cursive).

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

**Possible Solutions:**

- Wraps code in an anonymous function that is declared and immediately called so that there are 0 global symbols
- So variables don’t pollute the global namespace
- Localizing our variables within our JS file (ideally localized as much as possible within functions).

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

**Solution:** `setInterval` specifies a function to be repeated every given ms, while `setTimeout` specifies a function to be executed exactly once after a delay of the given ms.
7. Consider the following JSON object:

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

**Solutions:**

a. `miniJSON.foo` : `["b", 1, 2]`

b. `miniJSON["FOO"]` : "Foo?"

c. `miniJSON["FOO"][1]` : "o"

d. `miniJSON[foo]` : `error`

e. `miniJSON["foo"].length` : `3`