CSE 154: Web Programming Autumn 2018

Practice Midterm Exam 2 | **Key**

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).

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 id, 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 and CSS		
The DOM		
JS/Animations		
Short Answer		

1. (HTML/CSS) A Special Spec for a Special Doggy

```
<!-- HTML Solution -->
<body>
   <h1>Abby's Style Guide</h1>
   <p>Abby's Top 3 tips on how to maximize puppy cuteness:</p>
   <div>
      <img src="sleepy-abby.png" />
      <p>1. Get at least 12 hours of beauty rest every day.</p>
   </div>
   <div>
      <img src="abby-with-toy.png" />
      <p>2. Always have someone to snuggle with.</p>
   </div>
   <div>
      <img src="seahawks-abby.png" />
      <p>3. Live half your life as a bean. With sports swag.</p>
   </div>
</body>
```

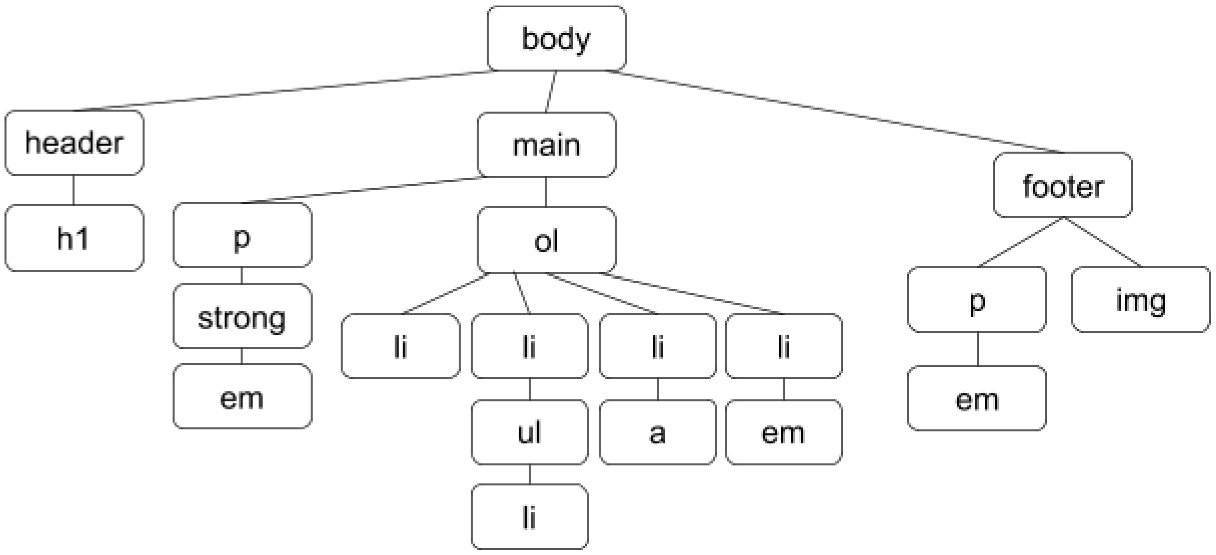
```
/* CSS Solution */
body {
     font-family: Verdana, sans-serif;
     text-align: center;
     width: 50%;
     margin: auto auto;
     border-left: 2px dashed #e91e63;
     border-right: 2px dashed #e91e63;
}
div {
     width: 350px;
     font-size: 12pt;
     font-weight: bold;
     margin: 20px auto;
}
div img {
     display: block;
     margin: 2px auto;
     width: 350px;
}
h1 {
     text-decoration: underline;
}
body > p {
     font-style: italic;
     font-size: 12pt;
}
```

2. Drawing the DOM

Consider the following HTML body:

```
<body>
   <header>
      <h1>Things to do after my CSE 154 midterm</h1>
   </header>
   <main>
      <p>
         Things I <strong><em>really</em></strong> look forward to:
      </p>
      <ol>
         <li>Find a comfy place to lay down and take a nap</li>
         <li>
            Finish any homework due today...
            <ul>
               <li>CP3 isn't due until next week! Yay!</li>
            </ul>
         </li>
         <li>Find <a href="https://imgur.com/r/puppies">puppy photos</a> on the int webz</li>
         <li>Get a <em>good</em> night's sleep</li>
      </ol>
   </main>
   <footer>
      <p>
         It's <em>almost</em> winter! :D
      </p>
      <img src="puppy-in-the-snow.jpg" alt="a happy puppy in the snow" />
   </footer>
</body>
```

Solution:



3. Turbo Turtles

Solution:

```
(function () {
   // Provided functions:
   function getRandomValue(min, max) { /* code not displayed here */ }
   function didFinish(turtle) { /* code not displayed here */ }
   function displayResults(pick, winner) { /* code not displayed here */ }

   // Solution begins here:
   let greenTimer, blueTimer = null;
   let myVote;
   window.addEventListener("load", function () {
      id("g-ftw").addEventListener("click", startRace);
      id("b-ftw").addEventListener("click", startRace);
   });

   function startRace() {
      if (this.id == "g-ftw") {
         myVote = id("g-turtle");
      } else {
         myVote = id("b-turtle");
      }
      id("g-ftw").disabled = true;
      id("b-ftw").disabled = true;
      let greenSpeed = getRandomValue(1, 250);
      let blueSpeed = getRandomValue(1, 250);

      /* version with one refactored move function */
      greenTimer = setInterval(function () {
         moveTurtle("g-turtle");
      }, greenSpeed);
      blueTimer = setInterval(function () {
         moveTurtle("b-turtle");
      }, blueSpeed);
   }

   function moveTurtle(turtleid) { // Solution could pass in DOM element or ID
      let turtle = id(turtleid);
      // ok if not done with window.getComputedStyle
      let leftPos = parseInt(window.getComputedStyle(turtle).marginLeft);
      leftPos += 4;
      turtle.style.marginLeft = leftPos + "px";
      if (didFinish(turtle)) {
         displayResults(myVote, turtle);
         clearInterval(greenTimer);
         clearInterval(blueTimer);
      }
   }
})();
```

4. Short Answers

1. Why is it important to avoid inline styles in HTML? (e.g. `<p style="color: green">...</p>`)

Possible solutions: To clearly separate content from appearance; to reduce redundancy and unused or overridden styles, to allow easily changing a webpage theme with a linked CSS file without needing to change the HTML.

2. For at least two of the following individuals write one thing you can do to make your website more accessible for the specified user:

a. blind user who uses a screen reader.

Possible Solutions: Use HTML5 semantic tags (screen readers use these to read content in an organized way)

b. low-vision user who uses magnification.

Possible Solutions: Use bold font for emphasis, clearly distinguish font sizes between headings/paragraphs; use easy-to-read font families, use em-based font sizes rather than px-based.

c. color-blind user.

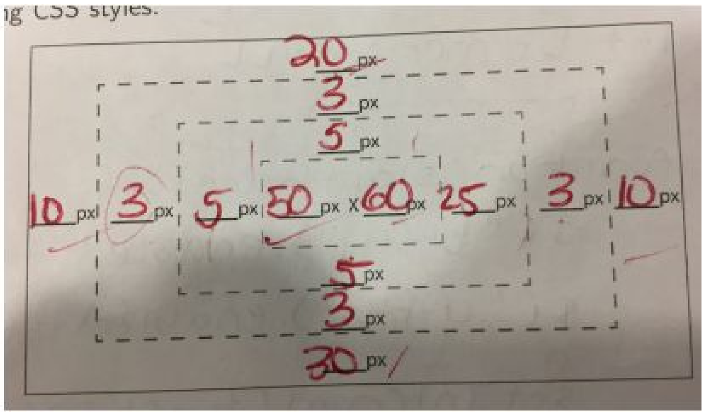
Possible Solutions: Use color palettes in CSS that are readable for different types of color-blindness (various software is available to see filters for different types of color-blind users)

d. mobility-limited who can't control a mouse.

Possible Solutions: Ensuring all interactions can be handled with keyboard interactions (tabs) - voice controls

3. In the Box Model diagram to the right, label the following CSS styles for a #content element:

```
#content {
  width: 50px;
  height: 60px;
  margin: 10px;
  margin-top: 20px;
  margin-bottom: 30px;
  border: 3px solid;
  padding: 5px;
  padding-right: 25px;
}
```



4. Provide and support one reason why the module pattern is important to use 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 to our js file (ideally within functions to localize scope as much as possible).

5. Give an example where using === and== would return different results when comparing the same two values in JavaScript.

Possible solution: `5 == "5"` is true but `5 === "5"` is false

6. Consider the following JSON object:

```
let miniJSON = {
  "waffle" : [“PANCAKE”],
  "pancake" : “waffle”,
  "POPTARTS" : {
    “frosted” : true,
    “flavors” : [“cherry”, “strawberry”, “jolly rancher”]
  }
};
```

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, specify this as your answer.

```
a. miniJSON.pancake : “waffle”
```

```
b. miniJSON["FOO"] : undefined
```

```
c. miniJSON[“POPTARTS”].flavors[1] : “strawberry”
```

```
d. miniJSON[miniJSON["pancake"]].length : 1
```

7. For the following JS program, label the _______ following each console.log statement with 1, 2, 3, or 4, corresponding to the relative order in which that statement will print (where 1 indicates the first statement printed).

```
console.log("Foo");                         // 1
(function() {

  console.log("Bar");                       // 2
  window.addEventListener(“load”, pageLoad);
  foo();

  function pageLoad() {
    console.log("Baz");                     // 4
  }

  function foo() {
    console.log("Mumble");                  // 3
  }
})();
```