CSC 337, Fall 2013
Assignment 2
Due: Wednesday, September 25 at 22:00:00

Assignment one used the Bad Idea of mapping 300 points in the write-up to 50 of the semester's 600 assignment points. I apologize! On this assignment the point values of problems correspond directly to assignment points. There are six problems with a total of 27 points that in total represent 2.7% of your final grade in this course.

Remember that late assignments are not accepted and that there are no late days; but if circumstances beyond your control interfere with your work on this assignment, there may be grounds for an extension. See the syllabus for details.

About restrictions...

If you've ever done any strength training for sports, you know that particular exercises like presses, curls, and squats develop different muscle groups. When you're on the field or the floor, it doesn't matter how much you can bench press; but the hope is that doing all those presses, squats, and curls make you more likely to win and less likely to get injured.

Just like strength training exercises focus on certain muscle groups, problems on my assignments often focus on certain skills by using restrictions. On assignment one I wouldn't let you use CSS. Perhaps that made you learn a little more about HTML. On this assignment you'll be using CSS, but you will find some restrictions to focus you on particular aspects of CSS in various problems. For example, in `paragraphs` you can only add CSS rules and not touch the `body` element. In `useclasses` you can change only the value of the `class` attribute in elements. In `construct` you'll have to figure out a tree structure for `body` that will match a complex selector.

In the workplace you'll typically have far more freedom, but my hope is that these restrictions will help you be better rounded. Maybe you had a soccer coach who made you learn to shoot with both feet or a basketball coach who made you learn to dribble, pass and shoot with both hands. It's the same idea here.

Because it's often the restrictions that produce the challenge in a problem, a solution that doesn't honor the restrictions will earn a greatly reduced score, typically a zero. The TAs and I are always happy to take a look at a solution in progress to be sure it doesn't violate any restrictions, but be sure to give us plenty of time for that. Do that by mailing solutions to be inspected to all three of us. NEVER EVER post pieces of solutions on Piazza, even with private posts—it's all too easy to make a mistake and "blow" a problem. (Review the syllabus to remind yourself of the typical penalty for "blowing" a problem.)

Be careful with "*Why doesn't X work?*" questions on Piazza

On assignment one a number of students posted Piazza notes of the form, "I'm trying to use X to do Y but it doesn't work. Any ideas?". It was often the case that X was the correct solution, but they didn't realize it. That counts as blowing a problem even though the poster might not realize it at the time.

To play it safe, use mail for questions about why something doesn't work. Remember that the first thing we usually want to do is to reproduce the problem. On assignment one we saw plenty of students write hundreds of words describing a problem and all the things they've tried and all the things they might try but not send us enough code to easily reproduce the problem ourselves. Typically a sentence or two and **all** your code is all we need to help you.

Stupid questions

You've probably heard more than one teacher say, "There are no stupid questions." That is incorrect. There are lots of stupid questions. In hopes of heading off stupid questions about this assignment, here is an example of a stupid question that I fear someone will ask:

> "I see that on `sequence` we are only allowed to change the `style` element, but I need to change something in the `body` element in order for my CSS rules to work. Is that ok?"

My response would be something like,

> "That's not ok! As the problem write-up says, you may change only the `style` element. Think some more about how you might do it using only CSS. If you're stumped, send us your solution. We'll take a look at it."

I would not add "p.s. What a stupid question that was!" but I might be unable to stop myself from thinking that.

Sample files and such

In this write-up the notation `$a2/x.txt` is used to indicate that a file can be found both at the URL `http://cs.arizona.edu/classes/cs337/fall13/a2/x.txt` and on the CS machines using the path `/cs/www/classes/cs337/fall13/a2/x.txt`.

A zip of the entire `a2` directory can be found at `$a2/all.zip`.

This convention will be used on all assignments.

Suggestion: Set up an "`a2`" search engine as shown on slide 28. Use this as the target: http://cs.arizona.edu/classes/cs337/fall13/a2/%s To download the zip just type (on Windows) Control-L a2 all.zip ENTER. On Macs, Cmd-L a2 sequence.html will bring up the starting point for the first problem.

Problem 1. (5 points) `sequence`

There have been several demos in lectures where CSS rules were added to a `style` element one at a time, with discussion of the effect of each.

In this problem you are to create a series of a CSS rules such that each rule has the effect, **and only the effect**, described in the corresponding step in `$a2/sequence.html`.

For each step you may only add rules; you may **never** delete or comment-out a rule. For example, one of the steps is `/* make all text black */`. You must do that by adding rules, not removing or commenting earlier rules. Think of the embedded stylesheet as append-only! (Recall that a `<style>` element is known as an *embedded stylesheet*—see CSS slide 12.)

The task is to produce the specified changes of appearance for only the `body` in `sequence.html`. You are **not** being asked to create styles that will work for an arbitrary file.

You'll see that the `style` element of `sequence.html` looks like this:

```
<style>
     /*
      * Underline all text.
      */
     CHANGEABLE

     /*
      * Make all paragraph text be bold.
      */
     CHANGEABLE

     /*
      * Make all the underlines go away.
      */
     CHANGEABLE

</style>
```

Here's how that `style` element should look in your solution:

```
<style>
   /*
    * Underline all text.
    */
   body { text-decoration: underline }

   /*
    * Make all paragraph text be bold.
    */
   p { font-weight: bold }

   /*
    * Make all the underlines go away.
    */
   body { text-decoration: none }

</style>
```

Hint: Take special note of that last rule: `body { text-decoration: none }`. That's being used to undo the effect of an earlier rule. You may need to do something similar in various situations on this assignment.

Note that the appearance after each step matters. Develop your solution by writing a rule for the first step and testing to be sure it's right. Then write a rule for the second step and test it, etc. Leave the comments in place and put your rules where the text `CHANGEABLE` appears in `$a2/sequence.html`.

I'll make available a series of images showing renderings after each step, but I don't yet know what form that will take. Watch Piazza for more on this.

Problem 2. (5 points) `paragraphs.html`

Change the contents of the `style` element in `$a2/paragraphs.html` so that it renders as shown in `$a2/paragraphs.png`. You may change no text other than that in the `style` element.

Problem 3. (5 points) `bq.html`

Change the contents of the `style` element in `$a2/bq.html` so that it renders as shown in `$a2/bq.png`. You may change no text other than that in the `style` element.

Problem 4. (2 points) `useclassses.html`

Change `$a2/useclassses.html` so that it renders as shown in `$a2/useclasses.png` For this problem the only text you may change is the value of the `class` attributes on the `span` elements in `body`.

Problem 5. (4 points) `construct.html`

In this problem you are to devise a `body` element for `$a2/construct.html` such that the one CSS rule already specified in the embedded stylesheet causes the word `Hello` to appear in red. You'll see that the challenge is to devise a structure that is matched by the selector of the lone rule in the stylesheet.

I'll get you started:

```
<body>Hello</body>
```

`$a2/construct.png` shows how my solution renders. The only thing that's important about the appearance is that `Hello` is in red.

If you've been reading in HFHC you might be inclined to simply add `style="color:red"` to the opening `body` tag. Don't do that! That inline style would be what causes the text to be red, not the rule in the embedded stylesheet.

Analogy: I'm giving you a set of street directions. You've got to build the streets, and put the word `Hello` at the end.

`contstruct.html` must validate as HTML5 at http://validator.w3.org/#validate_by_input

Problem 6. (6 points) `diagrams`

Using the style shown on HTML slide 20, create diagrams for `sequence.html`, `bq.html`, and `useclasses.html`. Your final result should be three image files, one for each of the HTML files. Files may be in GIF, JPEG, or PNG format only. For text nodes, just show the first couple of words and then "...".

Keep in mind that you can use things like Chrome DevTools to explore the structure.

If you take pictures of diagrams drawn on paper, we'd appreciate it if you'd reduce the image resolution to correspondingly reduce the file size. Something like 800x600 should be plenty of resolution, but be sure the images are legible. Please submit a separate image for each diagram, not an all-in-one.

You may use drawing tools but you are on your honor to not use any automated tools that directly produce diagrams from markup. If you've got a favorite tool for creating diagrams, tell everybody about it with a post to the resources folder on Piazza.

Turning in your work

Use the D2L Dropbox named `a2` to submit a zip file named `a2.zip` that contains all your work. If you submit more than one `a2.zip`, we'll grade your final submission.

Here's the full list of deliverables:

`sequence.html`
`paragraph.html`
`bq.html`
`useclasses.html`
`construct.html` (must validate!)
`{sequence,bq,useclasses}.{gif,jpg,png}` (three files in all)

Note that all characters in the file names are lowercase. It's ok if your zip includes other files, too.

If you find yourself wanting to use aspects of CSS we haven't yet covered, you're probably overlooking something and/or making a problem more complicated than I intend. We will be talking in more detail about fonts, colors, and borders later, but I believe the examples of rules, properties, and property values that we've seen in class should sufficiently equip you for this assignment. If you find yourself wanting to use dimensions, like `1px`, you're probably making something a little too hard.

Miscellaneous

My estimate is that it will take a typical student between two and eight hours to complete this assignment. If you're a CS senior, you'll probably do it in less. If you've taken only a single programming class, you might be on the high end, or beyond.

Other than `construct.html` validating as HTML5 your solutions do not need to validate with either the HTML5 validator or the "Jigsaw" CSS validator, but you may find the CSS validator to be helpful in diagnosing mysterious problems with CSS. (The typical mysterious CSS problem is that a declaration is ineffective, like those on slide CSS slide 16.)

I'll put up a Piazza poll that asks you to report how many hours you spent on this assignment. You don't need to participate in that, but if you want to, make a little effort to track your time.

Keep in mind the point value of each problem; don't invest an inordinate amount of time in a problem or become incredibly frustrated before you ask for a hint or help. Remember that the purpose of the assignments is to build understanding of the course material by applying it to solve problems. If you reach the six-hour mark, regardless of whether you have specific questions, it's probably time to touch base with us. Give us a chance to speed you up! Our goal is that everybody gets 100% on this assignment and gets it done in an amount of time that is reasonable for them.

Remember that I award Bug Bounty points; if you find problems, there's a potential reward for reporting them. I prefer that bugs be reported by mail to me. I'll put up an "Assignment 2 FAQs and Corrections" post on Piazza and update it as things come in.

Use the `a2` folder for any Piazza posts about the assignment.