

CS110 Review Questions and *Solutions*

This is *not* a “practice test.” It is much longer than the test will be and covers more material. Some of the questions may be harder or more time-consuming than the test questions. Furthermore, this is not exhaustive: there may be kinds of questions on the test that are not exhibited here. Therefore, you should not study *only* by doing these questions. These questions are here for you to think about and to trigger your own thoughtful inquiry into the material.

Review your own notes, the lecture and lab notes, and the textbook reading in an inquisitive way: not only “do I understand this,” but “can I explain it,” and “can I use it,” and even “what little bits of syntax are tricky here.” Consider compiling your own summary of reference material, concepts and so forth.

Work through problems on your own without reference to the course material to test whether you actually understand it. While the test is open book and open notes, there will not be time to learn a technique or look up things on every problem and finish in 70 minutes.

Vocabulary

1. The GIF image format tends to be good for drawn graphics and animations, but the JPEG format is better for photos.
2. A piece of text that the code can remember inside a variable or concatenate with other pieces of text is called a(n) string.
3. A hyperlink like `` is an example of a(n) relative URL.
4. A hyperlink like `` is an example of a(n) absolute URL.
5. The information put in front of a curly brace in a CSS rule is called a(n) selector.

True/False

1. true JavaScript can use either single or double quotes around its strings.
2. true A GIF with 33 colors requires a bit-depth of 6.
3. false It is *not* possible to show images on a web page without the `` tag. False because you can use the CSS background-image property

Short Answer

For questions with numerical answers, show how you arrived at your answer and mark the answer clearly. For other questions, write 2–3 clear, concise, complete English sentences. Your answer will be graded 50% for correctness and 50% for quality of explanation.

1. How many bytes does it take to represent a 100x100 image with 3 colors, assuming you represent each pixel with some bits that refer to a color palette? Also calculate how many bytes are in the color palette. (Ignore the bytes that would be necessary to specify the width and height of the image at the beginning of the file. Assume 24-bit color.)

answer:

There are $100 \times 100 = 10,000$ pixels. Each pixel can be represented by 2 bits if we let 00 be the first color, 01 be the second, and 10 be the third, so that's 20,000 bits. Divide by 8 to get the number of bytes for the image: 2,500. For our palette, we need 3 bytes (24 bits) to represent each color in the palette, one for each R, G, and B value, which makes $3 \times 3 = 9$ extra bytes. (We don't need to write the codes in the palette, because we know that 00 is the first one, 01 is the second, and so on.) So the total number of bytes is 2,509.

2. What decimal number does binary number 11111111 represent?

answer:

$128 + 64 + 32 + 16 + 8 + 4 + 2 + 1 = 255$. We could also take the shortcut of noticing that this is 8 bits, which is the same number of bits we get for an R, G, or B value, and remember that 255 is the biggest possible value for a byte. It's also $2^8 - 1 = 256 - 1 = 255$.

3. What decimal number does hexadecimal number AB represent?

answer: The digit "A" stands for 10, and "B" stands for 11, so this number is a 10 in the 16s place and 11 in the one's place, for a total value of $10 \times 16 + 11 = 171$.

4. Convert 75 to binary and to hexadecimal.

answer: In binary, we can see what places we need: the place values are 64, 32, 16, 8, 4, 2, and 1. We'll need 64, leaving us with $75 - 64 = 11$ to represent. So we don't need 32 or 16, but we do need 8, leaving us with $11 - 8 = 3$ to represent. To get 3, we need the 2 and 1. Final answer is 1001011.

In hexadecimal, we just have to ask ourselves how many 16s there are in 75. Divide 16 into 75 and we get a quotient of 4 and a remainder of 11. We can represent 11 with the digit "B." Therefore, the answer is 4B.

5. Consider 11_{10} and 11_2 (both decimal and binary). Which of the two is larger?

answer: If we convert them both to decimal, we can compare. Convert 11_2 to decimal and we get 3, and clearly $11 > 3$, so the first is larger.

6. Each row of the following table shows how the number for that row can be written in different bases. The first row is an example. Your task is to fill in the missing entries for the other rows. Show your work.

Decimal (base10)	Binary (base 2)	Hexadecimal (base 16)
110	01101110	6E
255	11111111	FF
129	10000001	81
46	00101110	2E

7. How would "Is it \$99?" be represented in ASCII? Don't represent the quotation marks. Here's a slightly modified form of the ASCII table you saw in class and in the reading:

	0	1	2	3	4	5	6	7
32	SPC	!	"	#	\$	%/	&	'
40	()	*	+	,	-	.	/
48	0	1	2	3	4	5	6	7
56	8	9	:	;	<	=	>	?
64	@	A	B	C	D	E	F	G
72	H	I	J	K	L	M	N	O
80	P	Q	R	S	T	U	V	W
88	X	Y	Z	[\]	^	_
96	`	a	b	c	d	e	f	g
104	h	i	j	k	l	m	n	o
112	p	q	r	s	t	u	v	w
120	x	y	z	{		}	~	DEL

answer:

"I" is on the row starting with 72 and in column 1, so its ASCII value is $72 + 1 = 73$. Working similarly for the other characters, we get:

I	s		i	t		\$	9	9	?
73	115	32	105	116	32	36	57	57	63

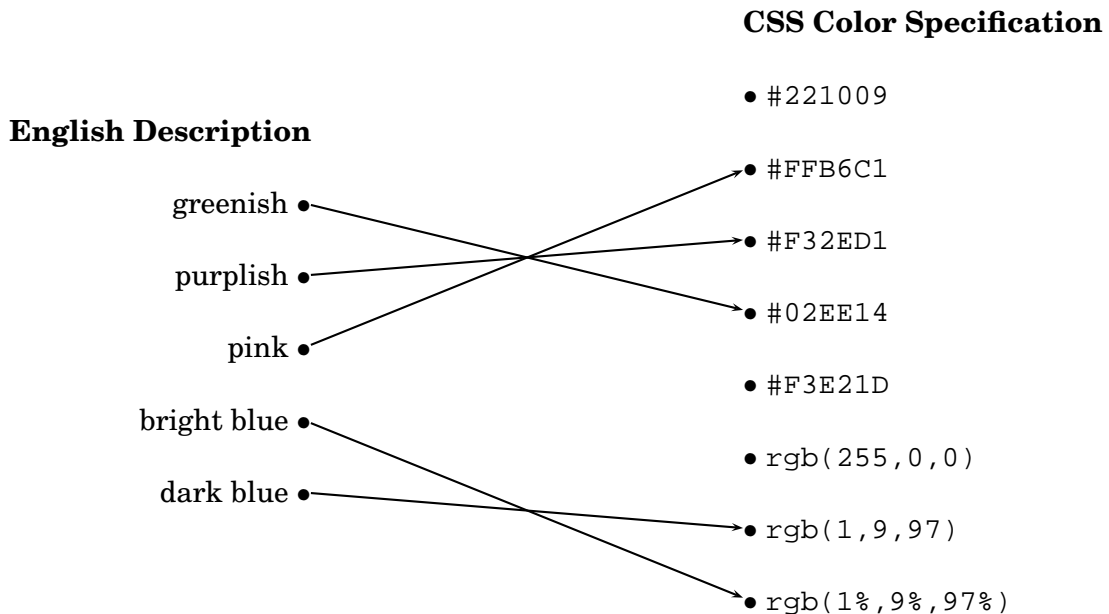
8. What part of the computer performs arithmetic operations and decides which instructions to execute next?

answer: The Central Processing Unit, or CPU.

9. Where are the bits that represent programs stored on the computer when the programs are not running?

answer: The hard drive.

10. For each English description in the left column, draw a line to the CSS color specification in the right column that best matches the English description. Since there are more color specifications than English descriptions, some color specifications will be unused.



answer: In the RGB color representation:

- A greenish color has a large green (first) component and small red (first) and blue (third) components. The only choice meeting these criteria is #02EE14.
- A purplish color has large red (first) and blue (third) components and small green (second) component. The only choice meeting these criteria is #F32ED1.
- Pink is like white with a little less green and blue; the only color like this is #FFB6C1.
- A bright blue color has a very large blue (third) component and small green (second) and red (first) components. The only choice meeting these criteria is `rgb(1%, 9%, 97%)`.
- A dark blue color has a small red (first) component, small green (second), and medium blue (third) component. The only choice meeting these criteria is `rgb(1, 9, 97)`.

11. You're picking colors for a web site. Your client likes dark violet (#9400D3) and plum (#DDA0DD), and asks you to find a color that is "right in between." What color would you offer? Show your computations.

answer: You can do the computations in hex or decimal. Here's how they would go:

- Red: halfway between 94 and DD isn't easy to do in hex, so let's convert to decimal. $94_{16} = 9 \times 16 + 4 = 148_{10}$ and $DD_{16} = 13 \times 16 + 13 = 221_{10}$. Halfway between 148 and 221 is $(221 + 148)/2 = 369/2 = 185$. Converting $185_{10} = 11 \times 16 + 9 = B9_{16}$.
- Green: halfway between 0 and $A0_{16} = 160_{10}$ is clearly 80. Converting $80_{10} = 5 \times 16 + 0 = 50_{16}$. You can also do this directly in hex and observe that 5 is half of A, so 50 is half of A0.
- Blue: halfway between D3 and DD is D8; there's no need to convert to decimal.

So, the intermediate color is #B950D8

12. Which of these two colors is *lighter*: aliceblue (#F0F8FF) or cornflowerblue (#6495ED)?

answer: AliceBlue is lighter, since all three of its components are larger numbers, so it's closer to white (#FFFFFF).

Relative URLs

This section tests your understanding of the syntax of relative URLs, which are incredibly useful when designing all but the smallest of web sites.

You're creating a music web site with the following folder and file structure:

```
music/
  index.html
  format/
    cd-used.html
    cd-new.html
  genre/
    rock/
      classic/
        doors.html
        stones.html
      punk/
        ramones.html
        sex-pistols.html
    rap/
      icecube.html
      DMX.html
```

Give the relative URL for each of the following connections:

- From `cd-used.html` to `stones.html`
`../genre/rock/classic/stones.html`
- From `icecube.html` to `ramones.html`
`../rock/punk/ramones.html`
- From `doors.html` to `stones.html`
`stones.html`
- From `index.html` to `DMX.html`
`genre/rap/DMX.html`

HTML

Write the complete HTML and CSS to produce this web page:



The image of this star comes from the following URL:

`http://cs.wellesley.edu/~cs110/lectures/M02-fireworks/star1.gif`

The image of the question mark comes from a local file, named `questionMark.png`.

Issues to consider:

- the right URLs for the star and the question mark image
- how to center headers, images, and lines of paragraphs
- how to control the size of the text in the header and to bump it up for a few words
- how to get the horizontal line across the page, above the word “--traditional”
- how to get the word “--traditional” to be positioned where it is.

answer: Note that your code need not match our code exactly. We would be shocked if it did. The following solution is just one of many ways to solve the problem. Capturing the ideas is what’s important.

```
<!DOCTYPE html>
<!-- saved from url=(0055)http://cs.wellesley.edu/~cs110/midterm-review/star.html -->
<html><head><meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
  <title>Little Star</title>
  <meta charset="utf-8">
  <link href="star.css" rel="stylesheet">
</head>

<body>
  <h1>Twinkle, Twinkle <span>Little Star</span></h1>

  <figure>
  
  <figcaption>How I wonder what you are</figcaption>
  </figure>

  <figure>
  
  <figcaption>Up above the world so high,<br>
```

```

    like a diamond in the sky...</figcaption>
  </figure>

  <footer>
    <p>--Traditional</p>
  </footer>

</body></html>

```

And here's the external CSS file:

```

/* Filename: star.css */

/* The body is centered with fixed width. */
body {
  width: 600px;
  margin: 0px auto;
  text-align: center;
  font-family: Verdana, Arial, sans-serif;
}

/* make the text Little Star inside h1 bigger and italics*/
h1 span {
  font-size: larger;
  font-style: italic;
}

/* apply some padding to captions*/
figcaption {
  padding: 25px 10px;
}

/* show text in footer on the right side and with space from top border */
footer {
  border-top: thick solid black;
  padding-top: 10px;
  text-align: right;
}

```

CSS

Write the code that would go into your CSS file to solve each of the following problems. You need not set values for attributes that are not immediately relevant to the problem. When classes and ids are used, write out the tag name too.

1. You told two images to float left, but instead of going all the way to the left side of the screen, the lower one seems to be bumping against the lower right corner of the upper image. (Assume the “float” and display type is set elsewhere.)

answer:

```

img {
    clear: left;
}

```

2. You'd like text within paragraphs of class “answer” to be black against black, so that people can only see the text when they highlight it. (Black on black is sufficient for this effect.)

answer:

```
p.answer {
    color: black;
    background: black;
}
```

3. You'd like the background color of a section containing a floating image to extend beyond the containing text so that the image doesn't stick out.

answer:

```
section {
    overflow: hidden;
}
```

Here are some slightly more lengthy CSS questions.

1. Suppose that you're writing a web site for an online store. In addition to its thousands of regular offerings, it occasionally has sale items, often dozens at a time. To make the sale items stand out, your boss says that they should be written in a bold font in lime green. How do you solve this problem? Briefly explain your design decisions.

answer: Your first decision is whether sale items are separate paragraphs (implying the use of P or DIV) or in-line text (implying the use probably of SPAN). Suppose you decide to use DIV for each item. There are clearly two classes of items, the regular items and the sale items. So, you should probably define a class of DIV. What will you name it? You should probably name it based on its *purpose*, not its look, so you might call it "sale." Thus, in the external style sheet, you code:

```
DIV.sale {
    font-weight: bold;
    color: lime;
}
```

In the web pages, you code, for example:

```
<div class="sale">
all-wool bikinis, now just \$5.99!
</div>
```

Note that this also makes it easy to change the look of all the sale items, when your boss decides that this looks ugly.

There are several key ideas here. One is how to code it at all (using DIV, P or SPAN) and another is the use of classes and the naming of them.

2. For a website for a newspaper, you decide to mimic the look of a newspaper and have the content of the site in columns, but just three columns each 2 inches wide. How would you do this?

answer: One approach, and probably the simplest, is to have three SECTIONS (or DIVs), each of which is floating left, and is exactly two inches wide. Because they all float, they won't overlap each other.


```
#left, #middle, #right {
  float: left;
  width: 2in;
  padding: 0.1in;
}
```

And, in the page, do

```
<section id="left">
<p>Here is the contents of the left column, etc.
</section>
<section id="middle">
<p>This is the contents of the middle column, etc.
</section>
<section id="right">
<p>And here is the contents of the right column, etc.
</section>
```

3. It turns out that HTML has a special code to produce a bullet, namely `•`. So, one could do a bullet list as follows:

```
<p>Here is a list of items:
<p style="margin-left: 3em">&bull; apples
<p style="margin-left: 3em">&bull; bananas
<p style="margin-left: 3em">&bull; dates
```

The result can be indistinguishable from conventional bullet lists. Indeed, many GUI programs for creating web sites do exactly this. What advantages or disadvantages does this have compared to conventional bullet lists?

answer: The given code concentrates on what the result *looks like* rather than its *structure*. This means that programs that rely on structure, such as screen readers for the blind, will not be able to convey that structure accurately. It will read this list as if it were an ordinary sequence of paragraphs, which may be less useful to the visitor. Secondly, each paragraph gets its own style, so if we decide to indent all bullet lists a little more or a little less, each bullet item must be individually updated. These are two big disadvantages. About the only advantage is that you don't have to think about abstractions like lists.

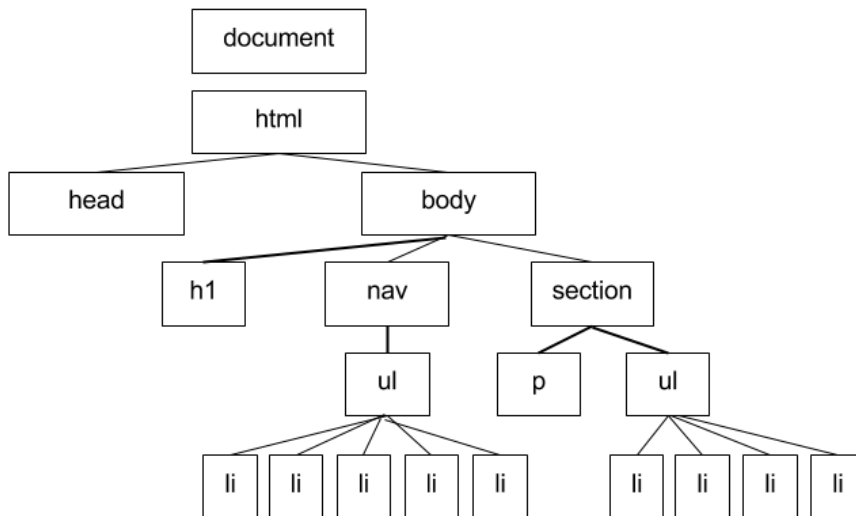
4. You are given a screenshot of how the document is rendered by the browser (see picture below) as well as the DOM tree of representing its HTML (next picture). Write the CSS rules that were used to style the page. The exact values in pixels (or the colors) are not important. A correct solution will have a rule for every styled element using the proper selector. Every rule contains the needed properties that will receive some reasonable value. Explain with words the role that each rule plays in the page appearance. Note: there are no ID or CLASS attributes in the HTML document

Welcome!

Home
About Us
Members
Events
Contact

Our values:

- Selfless
- Peaceful
- Honest
- Brave



answer:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <style>
      nav { float: left;
      }
      section {
        margin-left: 180px;
      }
      nav ul {
        list-style-type: none;
        margin-top: 0px;
        margin-left: 0px;
        padding: 0px;
      }
    </style>
  </head>
  <body>
    <h1>Welcome!</h1>
    <nav>
      <ul>
        <li>Home</li>
        <li>About Us</li>
        <li>Members</li>
        <li>Events</li>
        <li>Contact</li>
      </ul>
    </nav>
    <section>
      <p>Our values:</p>
      <ul>
        <li>• Selfless</li>
        <li>• Peaceful</li>
        <li>• Honest</li>
        <li>• Brave</li>
      </ul>
    </section>
  </body>
</html>
```

```

    }
    nav li {
        padding: 5px 20px;
        width: 100px;
    }
    nav li:nth-of-type(odd){
        background-color: #EBCAEB;
    }
    nav li:nth-of-type(even){
        background-color: #CFABCF;
    }
</style>
</head>
<body>
<h1>Welcome!</h1>
<nav>
    <ul>
        <li>Home
        <li>About Us
        <li>Members
        <li>Events
        <li>Contact
    </ul>
</nav>
<section id="about">
    <p>Our values:
    <ul>
        <li>Selfless
        <li>Peaceful
        <li>Honest
        <li>Brave
    </ul>
</section>
</body>

</html>

```

JavaScript

1. What is the result of the following code? Correct any errors.

```

<script type="text/JavaScript">
    var x = 7;
    var y = 5;
    var z = 14;
    x = y;
    z = z % x;
    alert(y + z);
</script>

```

answer:

The fourth line sets x to be 5, so the “remainder” operator on line 5 computes $(14\%/5)$, which is 4. So, the alert is of $(5 + 4)$, which is 9.

2. What is the result of the following code? Correct any errors.

```
<script type="text/JavaScript">
  var a = 1;
  a = a + 1;
  var b = "a is " + a;
  a = 5;
  alert(b);
</script>
```

answer: The third line makes b have the value "a is 2". This is unaffected by the fourth line, so the alert says "a is 2".

3. What is the result of the following code? Correct any errors.

```
<script type="text/JavaScript">
  var a = prompt("enter a number", "31");
  var b = a + 5;
  var c = "b is " + b;
  alert(c);
</script>
```

answer: The second line is actually concatenation, not addition so the result is "b is 315" Here's the correct code:

```
<script type="text/JavaScript">
  var a = parseInt(prompt("enter a number", "31"));
  var b = a + 5;
  var c = "b is " + b;
  alert(c);
</script>
```

4. What is the result of the following code? Correct any errors.

```
<script type="text/JavaScript">
  var name = prompt("enter your name", "Harry Potter");
  if( name = "Hermione Granger" ) {
    alert("A special hello to you, Hermione!");
  } else {
    alert("Hello, "+name);
  }
</script>
```

answer:

The second line is an assignment, not comparison, so the name will become "Hermione Granger" and the special greeting will happen.

```

<script type="text/JavaScript">
  var name = prompt("enter your name","Harry Potter");
  if( name == "Hermione Granger" )
    alert("A special hello to you, Hermione!");
  else
    alert("Hello, "+name);
</script>

```

5. What is the result of the following code? Correct any errors.

```

<script type="text/JavaScript">
  var name = prompt("enter your name",
    "Albus Percival Wolfric Brian Dumbledore");
  if( name > 20 ) {
    alert("Wow, that's quite a mouthful!");
  }
</script>

```

answer: The condition compares a string with a number; not what we want. Find the length of the string:

```

<script type="text/JavaScript">
  var name = prompt("enter your name",
    "Albus Percival Wolfric Brian Dumbledore");
  if( name.length > 20 )
    alert("Wow, that's quite a mouthful!");
</script>

```

6. What is the result of the following code? Correct any errors.

```

<script type="text/JavaScript">
var sales = parseInt(prompt("What were your sales this quarter?",
    "10000"));
var msg;
if( sales > 24000 ) {
  msg = "you deserver a raise.";
}
if( sales > 10000 ) {
  msg = "well done; keep it up.";
}
if( sales > 5000 ) {
  msg = "umm, step it up or else ...";
}
alert(msg);
</script>

```

answer: These should be mutually exclusive, so we have to use `else if`. There should also be a catchall for someone who doesn't even make 5000 in sales.

```
<script type="text/JavaScript">
var sales = parseInt(prompt("What were your sales this quarter?",
                             "10000"));

var msg;
if( sales > 24000 )
    msg = "you deserver a raise.";
else if ( sales > 10000 )
    msg = "well done; keep it up.";
else if ( sales > 5000 )
    msg = "umm, step it up or else ...";
else
    msg = "you're fired.";

alert(msg);
</script>
```