

**CT 310, Web Development**  
Things Covered in the First Third of CT 310  
Ross Beveridge  
Last Update February 22, 2017

Here is a list derived from a review of the material covered prior to the first Midterm. Please keep two things in mind when using this list. First, it is intended to be relatively complete and a useful study guide. Second, it is **not** a contract. **Material covered in lecture or recitation not explicitly listed below may of course appear on the midterm.**

1. **ALL CODE USED AS PART OF CLASS EXAMPLES**
2. You can explain the term DNS and tell the difference between a domain name and an IP address. You also understand the concept of DNS and reverse DNS lookup.
3. If someone mentions Tim Berners--Lee to you, you can explain who he is and why it is important.
4. If someone tells you Tim Berners---Lee invented the idea of hyperlinks you should be able to correct them.
5. You can now discuss intelligently the difference between a LAN and a WAN.
6. You can now take apart and explain the meaning of the various constituent parts of a URL. You also know what the acronym URL stands for.
7. You can name and explain the use of the UNIX utility that lets you discover how TCP/IP packets are routed between a Web server and client.
8. You can explain the manner in which there is a clean separation between the specification of substance and specification of appearance inherent in modern webpage construction.
9. You could walk someone unfamiliar with the power of CSS to alter the presentation of a page through an interactive demonstration.
10. If asked to name ten common HTML tags, you could generate them quickly and efficiently.
11. There are three distinct ways/places at which to introduce CSS into an HTML document. You can name each and recognize examples.
12. You know how to center text in a paragraph and how to center a division on a page, and further you understand the difference.
13. You understand the difference between specifying a margin and specifying padding.
14. You understand the use of borders including some of the commonly used different styles including, for example, solid, double, and grooved.
15. You understand how to specify colors for common aspects of a webpage, including the color of text, the color of backgrounds, and the changeable colors of links.
16. You understand how the Bootstrap Grid functionality allows you to build pages that are responsive to display width and automatically reconfigure page layout between cell phones and laptops.

17. You understand the size of things as specified in CSS, and in particular the difference between 'em' and 'px'.
18. You understand the value of relative, i.e. percent change to the default, size specifications for text.
19. You understand the difference between and 'id' and a 'class' designator in CSS.
20. You understand how to float an image or div, and more importantly and more generally how to work with floating divisions within a webpage.
21. You understand the importance of a markup validation service, and how to explain to others what such a service does and where on the Internet to find one.
22. You understand how to create a table in HTML and you also know better than to mistake a table as an appropriate means of doing webpage layout.
23. You understand and can explain at least three reasons why a website constructed using only HTML, as opposed to some form of server-side language, will ultimately fail to scale up and become unmanageable.
24. You understand and can explain the difference between a development and a production Web server.
25. You understand the syntactic trick associated with creating a variable in PHP and how to therefore instantly recognize a variable.
26. You appreciate and can explain the power of the PHP include command as a mechanism for code reuse and enforcement of consistency across a website.
27. You are not confused between the appearance of the HTML source you produce as opposed to the appearance of the final page. Further, you strive to make the HTML obvious and readable.
28. You can now explain the DOM as it relates to the JavaScript Namespace.
29. Early on we said the distinction between CSS class and CSS id was important to JavaScript, you now understand how and why.
30. You can correct mistakes in JavaScript programs, such as a 'hello world' program, or programs that dynamically populate lists and tables.
31. You can now debug and explain the operation of simple JavaScript event handling such as encountered with 'onChange' or 'onmouseover'.
32. You know how to create a JavaScript function.
33. You are comfortable defining and using JavaScript Objects and JavaScript array of Objects.
34. You know the syntax for specifying a JavaScript function by name when establishing linkages to event callback functions. You also know how to create an anonymous JavaScript function.
35. You know name of the root of a document in JavaScript.
36. The keyword 'innerHTML' now has immediate and practical meaning for you.
37. The practical value of 'onload' in JavaScript is now clear to you.
38. Someone new to web programming may be surprised that developer tools show different page content when showing source versus showing the DOM. You now

understand why and can provide and or interpret examples.

39. You have a basic understanding of how to debug at run-time JavaScript running in a browser, including how to set break points and inspect variables.
40. Events in JavaScript need not come from I/O devices.
41. You know enough about PHP to set the default time zone before using the date function.
42. You know how to write a page footer that will display the current date and time.
43. You know enough to be able to list by memory at least two of the nine super global variables used by PHP.
44. Over a lunchtime interview, you understand immediately how to respond to the question: "Show me the syntax required to declare an integer variable in PHP."
45. You understand and can explain the kind of information resident in the `$_SERVER` super global PHP variable.
46. You know better than to get the post, or post the get, and more generally how to convey user data entered through a form back to the Web server.
47. You understand the most common types of input in a form including text, textarea, hidden and submit.
48. You understand how to test for the presence of data arriving from a form submitted on a client.
49. You understand the risks of using an independent page as the recipient of form data as specified through the action field. Consequently, you understand the value of writing a page that handles its own form submission data.
50. You understand how to conditionally send plain html to the client using control constructs in PHP combined with starts and stops of the PHP interpreter.
51. You can instantly think of an extremely common case where the openness of get variables as embedded in the URL is useful.
52. You can instantly think of an extremely common case where the openness of get variables as embedded in the URL is disastrous if utilized.
53. You understand how to provide a submit button with a textual label of your own choosing.
54. You understand string manipulation and string output in PHP, including in particular when it is best to concatenate literals and when it is best to escape special characters in regular strings.
55. You understand standard iteration in PHP with explicit numeric indices.
56. You know better than to ever use a for loop with a numeric index when your intention is to iterate over all the elements of an array, and moreover you know and understand the syntax of the appropriate construct.
57. You understand the power and the danger of the 'mail' command in PHP. Just as importantly, you understand how to use it.
58. You understand how to open files, read from files, write to files, and close files all from within PHP. You also understand the power and the danger of the connection between

PHP and the file system on the server. You can explain at least one mistake you can imagine making relative to security and file I/O.

59. A pernicious bug when constructing elements of a page such as a footer is an image that hangs down below the bottom of the footer. You can now readily spot this bug and explain others how to avoid it.
60. You know better than to mistake a PHP array for a mere contiguous sequence of data indexed exclusively by integers. Moreover, you know how to explain to others why such a view of the PHP array is naïve.
61. You can explain why a common mistake in a powerful language such as PHP is to use iteration when a mapping construct is better. Moreover, you can offer your explanation in concrete terms using actual examples that employ the `array_reduce` function.
62. If asked to explain the statement “not all PHP functions you write need be, nor should be, named,” you could do so handily and cite at least concrete motivating example.
63. Since PHP is an object oriented programming language, you could easily demonstrate how objects are specified by writing down an example of a simple class. For example, a class in which each instance represents a state and conveys both the name of the state in the name of the state capital.
64. Objects allow for a special `toString()` function in PHP. You can recognize this functionality as well as explain how it may be useful.
65. Hashing of passwords using older approaches such as MD5 is now something you well understand. If asked at lunch about 'rainbow-tables' you can provide a succinct explanation.
66. Consider carefully your answer to this question: “Under what circumstances should a website record and store a user’s password?”
67. **Be able to answer all the quiz questions up to this point.**