**Answer Key**

for Spring 2021 Lesson 17

**Organizing files:**

1. F: filenames \*are\* case-sensitive
2. Mostly F: it works, but it looks janky with %20 in place of every space (and easier to mess it up)
3. F: you can refer to other folders within the root
4. – advantage: easier to move the project, e.g. from your computer to an online server.
– risk: links could break silently, especially if you change folder nesting

EXT: <a href="">, <img src="">, <link href="">

**HTML basics**

1. Metadata
2. <!-- html comments -->
3. F: trick question. <img />, <br />, and <hr /> are self-closers, with no element content outside the tag itself.
4. True! In HTML, whitespace is whitespace is whitespace.

EXT: Every <head> should include <meta charset='UTF-8'/> and a <title>... and most likely a <link rel="stylesheet">

**Image basics**

1. T: img doesn't function without a src
2. Mostly F: width and height are often better set in css, because they’ll often affect layout. (But you should probably make sure to set at least one of them there). An exception might be if images are rare, widely varied in size, and/or not part of the layout (e.g. just thrown in for the text to wrap around)
	* If you only set one, the other is automatically calculated so as to preserve the aspect ratio
3. Mostly T: alt is required. If you have a purely ornamental image (e.g. a decorative background), you can write alt="" (i.e. empty, but not missing)

EXT: jpgs are lossy, but with photos it's hard to tell. With line diagrams, jpg leads to compression artifacts, and png offers transparency.

**CSS basics**

1. Curly brackets assign rules to selectors / define a ruleset; semicolons distinguish rules *within* a ruleset
2. Connect multiple selectors with a comma before a shared set of {}
3. /\* css comments \*/
4. – external stylesheet

– page-specific <style> in the <head>
– inline style in the attributes of an html element

1. Inline styles override the stylesheet, can't make changes in one fell swoop (or for accessibility purposes)

EXT: Use external stylesheets most of the time, for max flexibility (e.g. consistent theme); use page-specific style for one-time modifications to a site’s primary style (e.g. for a special landing page); use inline style sparingly, when you have to override a system where you can post html but not css (e.g. on Canvas or someone else’s WordPress site)