1. According to Jeremy Keith's Resilient Web Design book - what is the "Core Purpose of the web"?
   1. The web is resilient by design and it is the job of people who create content for the web to ensure that resilience is preserved.
2. What is supposed to happen if a browser encounters a document with HTML and CSS errors?
   1. The browser will do what it can to present the data in a form that is readable without displaying any errors or blocking the page from the user.
3. JavaScript errors should \_\_\_\_\_\_\_\_\_\_\_\_\_\_
   1. Not prohibit the end user from accessing the content and completing their task.
4. List the 5 "Known unknowns" of the web
   1. Where the user is
   2. What their internet connection is like
   3. What devices they are using
   4. What user agents they are using
   5. What capabilities and limitations the user has at the moment
5. If a web site is resilient, how should disabling JavaScript and the default stylesheet of that web page effect it?
   1. The features that rely on them shouldn’t inhibit the use of the page without their presence. The user should still have access to all of the content.
6. What are the 4 HTML design principles as specified by the W3C? Briefly summarize each.
   1. Compatibility: making sure the web works
   2. Utility: making sure the web is useful
   3. Interoperability: making sure the web works consistently
   4. Universal Access: Making sure the web can be used by everyone.
7. HTML and CSS are declarative languages - what does that mean?
   1. They describe what the program should do instead of saying how it should accomplish the task.
8. JavaScript is an imperative language - what does that mean?
   1. They tell the program how the program should operate step by step.
9. Which kind of language is more resilient and more fault tolerant?
   1. Declarative Language

2—-----

1. What are the three core materials of the web?
   1. Content with html. Style with CSS. Enhanced with JS
2. Priority of constituents. Re-order the following constituents to their order of priority, highest to lowest:
   1. users
   2. Authors
   3. Implementors
   4. Specifiers
   5. theoretical purity
3. As the tools we use to write imperative code "distance" us from the code that the user sees, what is "introduced"?
   1. Fragility
4. Which "web stack" layer is the "presentation" layer?
   1. CSS
5. What is the difference between a "robust" website and a "resilient" one? Which is preferred?
   1. Robust websites are supposed to deliver the same experience to everyone all the time no matter what. Resilient websites on the other hand are built to deliver the same goal despite a difference in experiences.
6. Do web sites need to be "robust" meaning that they "look the same everywhere"?
   1. No
7. What do users "care about"?
   1. They care about the site working and that it will help them accomplish their goal on that website.

3—-----------

1. Give 2 examples of ways to make sure that web impermanence does not result in information loss
   1. Archiving all published content so that it won't be deleted
   2. Adding and extending eddinting features so that visitors can see old content
2. Give the 3 items from "Morten's Accessibility Checklist"
   1. Make it accessible.
   2. Make it fancy.
   3. Make sure the fancy doesn’t break accessibility.
3. Give one advantage to the "One-way Binding" that HTML links use
   1. Anyone can link to anything and you don’t need permission.
4. Resiliency means the \_\_\_\_\_\_\_\_\_\_\_ is met for all users in all circumstances
   1. purpose
5. Large media files mean \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
   1. Poor resiliency
6. List 3 best practices for resilient images
   1. Responsive images markup
   2. Modern formats with fallbacks
   3. Alt text and image captions
7. List 3 best practices for resilient audio and video
   1. Closed captioning
   2. Verbose transcripts
   3. Translations where possible

4—------

1. What does @supports do?
   1. Tests the browser for support of a feature and only runs the code if it is supported by the browser.
2. What is a JavaScript polyfill?
   1. An extra function that will instruct the browser on how to use functions that may not be supported.
3. Babel, webpack, et cetera, all automatically \_\_\_\_\_\_\_\_ modern code for older browsers enabling us to write modern code for all browsers knowing it'll work everywhere.
   1. Shim and pollyfill
4. What does a JavaScript service worker do?
   1. It can send and receive requests, cache resources, retrieve cached resources, and deliver push messages.
5. What are web components?
   1. A native platform method for building custom html elements with custom features and properties.
6. If you serve the browser with a custom element and the browser is unable to interpret it as a web component, it will interpret it as a \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
   1. Default <div>

5—------

1. What does CSR stand for?
   1. Client-Side Rendering
2. What does SSR stand for?
   1. Server-Side Rendering