noraesae / Front-end-Developer-Interview-Questions-And-Answers
forked from h5bp/Front-end-Developer-Interview-Questions

Branch: master ▾ Front-end-Developer-Interview-Questions-And-Answers / answers / general-questions.md

Find file | Copy path

noraesae Answer to the general questions dc3ef5e on Jan 10, 2016

1 contributor

225 lines (167 sloc) 8.5 KB

General Questions

What did you learn yesterday/this week?

I studied about a Node.js ORM called Bookshelf.js and Express middleware implementation. I also solved some problems in Project Euler using Haskell, in which I am currently interested

What excites or interests you about coding?

I'm generally interested in quite everything about coding. Nowadays I'm especially fascinated by Haskell, one of the most beautiful languages I've ever experienced.

Also I like to create tools and services I and many others can be helped by and enjoy using.

What is a recent technical challenge you experienced and how did you solve it?

Recently I worked on building a web service and we needed to write test cases for APIs of it. When writing a test case, it needs several database entities which will be used in the test case, and it is really a pain to write all the insertion code for the database and cleanup at the end.

To resolve the problem, I implemented concepts called 'state' and 'environment'. 'State' refers to a database instance which a session contains. 'Environment' helps the 'state' manage its own entities. With the concepts, entities can be created and managed easily in each test session and also cleaned all together with the `cleanup` method implemented in a state instance.

What UI, Security, Performance, SEO, Maintainability or Technology considerations do you make while building a web application or site?

- **UI**: I like minimal UI which contains only what it should. I believe it results in the better user experience, as a user knows what to do intuitively.
- **Security**: I always try to make both frontend and backend secure, concerning CSRF, XSS, etc.
- **Performance**: I consider space and time complexity for the algorithms and logics I use and write.
- **SEO**: Set meta tags for search engines and consider and consider server-side rendering for SPA.
- **Maintainability**: Try to keep the source code consistent and make objects immutable. Use statically typed languages such as TypeScript. Use CI with tests and lints.
- **Technology**: I like to learn new technologies, but if the project is in production, I would consider using technologies which is well-documented and widely used.

Talk about your preferred development environment.

I like to work in terminal shell environments. For the time being, my favourite dev environment are like below:

- OS X
- iTerm2
- Tmux

- Vim

Nevertheless, I don't want to say they are the silver bullets for everything. I always try to find the best environment for each language and requirement.

Which version control systems are you familiar with?

Git. Not really familiar with others, but at least have experience of Subversion and Mercurial too.

Can you describe your workflow when you create a web page?

I usually use Node.js to build a web page, so will describe the workflow with it.

1. Decide a CSS preprocessor. I may consider using SCSS, but Less and Stylus are also viable options.
2. Decide a HTML template engine. I may go with Pug(formerly Jade).
3. Decide a JavaScript preprocessor or other languages being compiled to it. I may go with TypeScript or ES6 with Babel.
4. Decide a task manager. I recently like to just use NPM scripts instead of using huge task managers like Gulp or Grunt.
5. Write tests and make them fail.
6. Write app code and check the tests succeed.
7. Set CI.
8. Publish the code and check a task in CI succeed.

If you have 5 different stylesheets, how would you best integrate them into the site?

Use a CSS preprocessor to nest them with `@import` statements in class names for each stylesheet, and merge them into a built file. In production, minify the built file with a CSS minifier.

Can you describe the difference between progressive enhancement and graceful degradation?

Progressive enhancement is a way to implement a web page where basic features, which are supported by most environments, are implemented first and then progressively enhance them for advanced environments.

On the other hand, **graceful degradation** is an opposite. The advanced features are freely implemented at any time, and additional works are done to support the environment where the features don't work well.

How would you optimize a website's assets/resources?

Minimise CSS and JavaScript using minifier(or uglifier), archive them using gzip, use separated file servers, use CDN, etc.

How many resources will a browser download from a given domain at a time?

It depends on browser implementations. Usually 6 to 8 in the modern browsers, and less in the old browsers.

What are the exceptions?

When we use several subdomains pointing the same domain, we can increase the concurrency level of the download.

Name 3 ways to decrease page load (perceived or actual load time).

- Use minifier and gzip to decrease the page size - *actual*
- Show spinner or progress bar - *perceived*
- Preload the page before actually loading it using libraries like InstantClick - *both actual and perceived*

If you jumped on a project and they used tabs and you used spaces, what would you do?

- I would use tabs because it is the convention used for the project.
- Introduce a linter or other scripts to ensure indentations are consistent
- Use a tool like EditorConfig to configure editors team members are using automatically

Describe how you would create a simple slideshow page.

```html
<div class='slide-page'>...</div>
<div class='slide-page'>...</div>
<div class='slide-page'>...</div>
```

```css
html, body, .slide-page {
  width: 100%;
  height: 100%;
}

.slide-page {
  position: fixed;
  top: 0;
  left: 0;
  display: none;
}

.slide-page:first-child {
  display: block;
}
```

```js
window.addEventListener('click', () => {
  document.querySelector('.slide-page').remove();
});
```

If you could master one technology this year, what would it be?

Haskell. Haskell really helps developers a lot to perform much better, even though they don't really use Haskell directly because through learning Haskell, we can have better understand for other languages as well. It has been, and will be a well-designed language and many other languages have been adopting functional concepts from it and I would say it is fascinating and valuable language for every developer.

Explain the importance of standards and standards bodies.

Standards describe how a thing does and should work. It is extremely important especially in software, because the *thing* can be used by many people for different perposes. For example, there are several engines for JavaScript including V8, JavaScriptCore, Rhino, etc, and if there is no standard for the language, developers and users cannot feel ensured when doing something with it.

Standards bodies, in the same manner, do a key role to form a standards and are essential in everywhere including both software and hardware.

What is Flash of Unstyled Content? How do you avoid FOUC?

It is caused when content is loaded before styles are applied to the content. It happens when style tags are placed after other content, or applied asynchronously, for example, by scripts.

To avoid FOUC, the styles should be placed in order that they can be loaded and applied in the same rendering process as HTML elements do. The easiest way is to place them in the `head`, and avoid applying styles by scripts at the first load.

Explain what ARIA and screenreaders are, and how to make a website accessible.

They are for accessibility. To make a website accessible, we should try to follow the usage of HTML elements, for example, `h1` for headers and `section` for sections. Also it's good to take care of using visual contents, such as not forgetting to add an `alt` attribute to `img` tags.

Explain some of the pros and cons for CSS animations versus JavaScript animations.

CSS animations

- **pros**: They use GPU, so they are CPU-efficient. Don't consume JavaScript event loops.

- **cons**: Hard to handle, as CSS doesn't contain logics. Not supported in old browsers.

JavaScript animations

Opposite to CSS animations

What does CORS stand for and what issue does it address?

CORS stands for cross-origin resource sharing. There could be situation where some resources should be allowed from sources having different origin. CORS is a standard to enable cross-site HTTP requests for:

- AJAX API call
- Web Fonts
- WebGL textures
- Image/video frames drawn to a canvas using drawImage
- Stylesheets
- Scripts