**Q: What is Emmet?**

Emmet is a powerful tool for web developers that accelerates HTML and CSS writing. By using concise abbreviations, developers can rapidly expand code snippets into full-fledged structures.

**Q: What is a CDN? Why do we use it?**

A Content Delivery Network (CDN) is a distributed group of servers that work together to deliver content quickly. CDNs enhance website performance and availability by distributing content across multiple geographic locations.

**Q: Why is React called React?**

React gets its name from its core functionality: reacting to data changes. Created by Facebook to handle frequent updates, React efficiently updates the user interface in response to user interactions like clicks, submissions, and typing.

**Q: What is the crossorigin attribute in the script tag?**

The crossorigin attribute enables cross-origin resource sharing (CORS) in scripts. It specifies how the browser should handle requests to resources from different domains. This attribute is crucial for security and interoperability in modern web applications.

**Syntax:**

HTML

<script crossorigin="anonymous|use-credentials">

**Q: What is the difference between React and ReactDOM?**

React is a JavaScript library for building user interfaces, while ReactDOM is another library that bridges React and the browser's DOM. React handles component logic, while ReactDOM renders components to the DOM.

**Q: What is the difference between react.development.js and react.production.js files via CDN?**

The primary difference between development and production builds of React is performance. Development builds include additional code for debugging and development tools, making them slower. Production builds are optimized for speed and size, suitable for deployment to live environments.

**Q: What are the async and defer attributes in the <script> tag?**

* **async:** This attribute loads the script asynchronously without blocking page rendering. The script executes as soon as it's available.
* **defer:** Like async, it loads the script asynchronously. However, the script executes after the page has fully parsed.

**Syntax:**

HTML

<script async src="script.js"></script>

<script defer src="script.js"></script>

**Q: Difference between a Library and Framework?**

A **library** is a collection of reusable code for specific tasks. Developers incorporate libraries into their projects as needed. A **framework** provides a complete structure for building applications, dictating the overall application flow and how components interact.

React is an example of a library, while Angular is a framework.

**Q1: What is npm?** npm is the world's largest software registry, hosting over 800,000 code packages. It's the package manager for Node.js, enabling developers to share and reuse code.

**Q2: How to initialize npm?**

* npm init: Creates a package.json file with default settings.
* npm init -y: Creates a package.json file with default settings, skipping prompts.

**Q3: What are Parcel/Webpack? Why do we need them?**

Parcel and Webpack are bundlers that combine multiple JavaScript files into a single file for efficient delivery. They optimize code by minifying, removing unused code (tree shaking), and improving performance.

**Q4: Key benefits of Parcel/Webpack?**

Faster load times, smaller file sizes, improved code organization.

**Q5: How to install Parcel?**

Bash

npm install -D parcel

* -D flag installs Parcel as a development dependency.

**Q6: Parcel commands**

* npx parcel <entry\_point>: Starts a development server.
* npx parcel build <entry\_point>: Creates a production build.

**Q7: What is the .parcel-cache folder?**

The .parcel-cache folder stores information about your project to speed up subsequent builds. It improves build performance by caching intermediate results.

**Q8: What is npx?**

npx is a package runner that executes packages from the npm registry without installing them globally. It's useful for trying out tools or running scripts from packages.

**Q9: Difference between dependencies and devDependencies?**

* **dependencies:** Packages required by your application in production.
* **devDependencies:** Packages used for development and testing, not needed in production.

**Q10: What is Tree Shaking?**

Tree shaking is a process that removes unused code from your production build, resulting in smaller and faster applications.

**Q11: What is Hot Module Replacement (HMR)?**

HMR updates parts of an application without a full reload, improving development speed and experience.

**Q12: Parcel's Superpowers**

Parcel excels in:

* Development speed due to its fast build times and HMR.
* Code splitting for efficient loading of large applications.
* Image optimization for improved performance.
* Tree shaking for smaller bundle sizes.

**Q13: What is .gitignore?**

.gitignore is a file that tells Git which files or directories to ignore when committing changes. It helps prevent unnecessary files from being included in your repository.

**Q14: Difference between package.json and package-lock.json?**

* **package.json:** Describes your project, its dependencies, and scripts.
* **package-lock.json:** Stores exact versions of installed dependencies to ensure reproducibility.

**Q15: Why avoid modifying package-lock.json?**

Modifying package-lock.json manually can break dependency management. It's automatically generated to maintain dependency consistency.

**Q16: What is node\_modules?**

node\_modules is a folder containing installed packages and their dependencies. It's generally not committed to version control due to its size and the ability to recreate it from package.json.

**Q17: What is the dist folder?**

The dist folder contains the production-ready build of your application. It includes minified and optimized code.

**Q18: What is Browserslist?**

Browserslist defines the target browsers for your project, allowing build tools to optimize code accordingly.

**Additional Notes:**

* Consider using a linter to enforce code style and catch potential errors.
* Explore additional build tools like Webpack for more complex projects.
* Optimize images and other assets for performance.