1. **CDN:** CDN is a content delivery network its key function is serve the files from the edge locations depending on the request it is distributed network with caching included which enhances the websites response time.
2. **React Name to reactjs**: it's called React because it reacts. It was developed by Facebook (a site that CONSTANTLY updates their data) to improve the user interface development and more effectively change (REACT to) what the user sees when they're doing things like mouse clicking, submitting and typing.
3. **Script cross origin**: Anonymous it tells that there is no requirement of credentials, use-credentials it tells the user credentials are required for every request.
4. **React vs ReactDOM**: reactDOM contains dom related api’s like render and react contains state and provides dom modification related api’s

Like react.children,react.createElement etc.

1. **Difference between react.developement.js, react.production.js:** React fine tunes more in production environment which makes it more performance improvement compared to react.developement mode.
2. **Async vs defer:** The **defer** attribute tells the browser not to wait for the script. Instead, the browser will continue to process the HTML, build DOM. The script loads “in the background”, and then runs when the DOM is fully built. **async** scripts load in the background and run when ready. The DOM and other scripts don’t wait for them, and they don’t wait for anything.
3. **Npm:** npm not stands for node package manager officially but it’s manages the packages of node js and maintains package up to date by auto versioning by maintaining the package.json
4. **Parcel/ webpack:** Both are bundlers which bundles the code do miracle things like code splitting, tree shaking, hot reloading, build development server, caching etc using algo’s we can use them to build starting boiler plate code like create-react-app
5. **What is. parcel.cache**: parcel use smart caching to improve the speed for every refresh it will cache the build files and it will not rebuild those files this is the reason for fast build
6. **What is npx:** npx is executing the npm package It allows developers to execute any JavaScript Package available on the NPM registry without even installing it.
7. **Tree Shaking:** It is the process of remove dead code or unused code In modern JavaScript applications, we use module bundlers (e.g., webpack or Rollup) to automatically remove dead code when bundling multiple JavaScript files into single files. This is important for preparing code that is production ready, for example with clean structures and minimal file size.
8. **Hot Module Replacement:** HMR is one of the main features of modern webpack/bundler. Developing a JavaScript application involves reloading the browser each time you save code changes in order to refresh the user interface. Developer tools like Webpack can even run in watch mode to monitor your project files for changes. As soon as Webpack detected a change, it will rebuild the application and the browser is reloaded automatically. But soon developers started to think, is there a way to actually save and reflect changes to the browser without reloading? That’s where the hot module replacement shines his is made possible in Webpack by creating an HMR server inside Webpack Development Server (webpack-dev-server) that communicates with HMR runtime in the browser through a WebSocket.
9. **What is gitignore:** git ignore is a file where we list out the contents to ignore to push to GitHub client files like. env,node\_modules we don’t require such files it is unnecessary to push those to GitHub it makes the client bulky and security vulnerable. And we should not include files like package.json , app.py which are necessary to run the application.
10. **Package.json vs Package-lock.json:** package.json holds the main normal ,dev dependencies and scripts which are necessary to run the application package.lock.json contains the all entire tree of dependencies with exact version etc it tracks/lock the current version downloaded so that if the another coder clones your project the version’s should match
11. **Why should I not modify package.lock.json:** Because it contains the exact version if you alter then it can lead to version mismatch for the application
12. **What is node modules is it a good idea to push to git:** The node modules folder contains the library code which is required for the application to run basically it contains the code written by other developer who build the library and its dependent node modules. It is the bad idea to push it to GitHub because it makes it bulky.
13. **What is the dist folder:** It includes the compiled, minified code which is used to build the production build of the application.
14. **What is the Browser List:** Browser list enables the sharing of target browser and Node. js version configurations between various frontend tools. It's used in Autoprefixer, Babel, postcss-preset-env, eslint-plugin-compat, stylelint-no-unsupported-browser-features, postcss-normalize, obsolete-webpack-plugin. It helps in code transpilation browserlist indicates what kind of browsers we are targeting then bundler uses it and transpile the code accordingly.
15. **Different versioning symbols in package.json(^,~):** The package.json includes the ^/~ symbols on packages we install it indicates the auto upgrading the package ^ indicates the upgrade to only minor updates where ~ indicates to only upgrade batch updates majors there are kinds of upgrade like major, minor and batch ex: ‘react’: “^17.8.9” this package only get’s upgrade to minor updates such as 17.9.0
16. **Script type attribute:** The script tag has the type attribute whose value may be module, import type the default attribute is treated as classic JavaScript code.

Module: This value causes the code to be treated as a JavaScript module. The processing of the script contents is deferred. The charset and defer attributes have no effect.

Import type: This value indicates that the body of the element contains an import map. The import map is a JSON object that developers can use to control how the browser resolves module specifiers when importing JavaScript modules