1. **● - What is `NPM`?**

Npm is package manager that manages packages and dependencies on behalf of an application in Node.js. It provides a lightweight way of installing, upgrading, configuring,

1. **● - What is `Parcel/Webpack`? Why do we need it?**

Parcel and webpack both are bundlers which help us convert our raw code into production ready code , parcel is off zero configuration where as webpack requires consfiguration setup before using , parcel is for small to medium projects where as webpack is for high level projects.

Features of Parcel are

1. Dev server
2. Hot reloading
3. Diagnostics
4. Reliable caching
5. Lazy mode dev--

In development, it can be frustrating to wait for your entire app to build before the dev server starts up. This is especially true when working on large apps with many pages. If you’re only working on one feature, you shouldn’t need to wait for all of the others to build unless you navigate to them.

You can use the --lazy CLI flag to tell Parcel to defer building files until they are requested in the browser, which can significantly reduce development build times. The server starts quickly, and when you navigate to a page for the first time, Parcel builds only the files necessary for that page. When you navigate to another page, that page will be built on demand. If you navigate back to a page that was previously built, it loads instantly.

1. Tree shaking
2. Minification
3. Image optimization
4. Compression
5. Code splitiing
6. Content hashing
7. **● - What is `.parcel-cache`**

It’s a dedicated directory where Parcel stores information about your project’s assets, dependencies, and transformations.

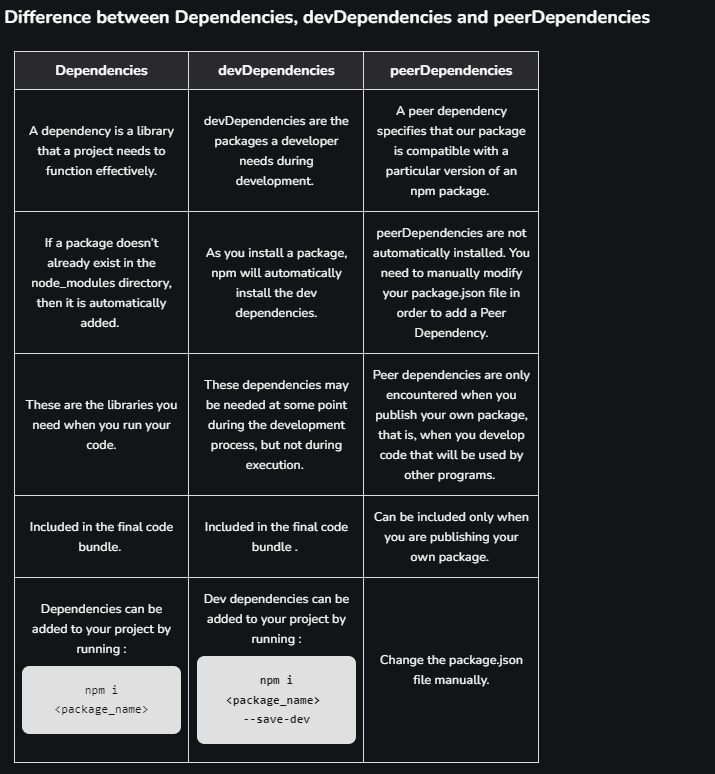
It acts as a comprehensive map, guiding Parcel to recreate your project’s bundle intelligently and efficiently, avoiding unnecessary work.

* It typically includes:
* Cached versions of parsed files
* Information about dependency relationships
* Metadata about transformations applied to assets

1. **● - What is `npx` ?**

The npx stands for node package executer , if we install npm it automatically install npx . with the help of npx we can execute packages even without installing them locally .

1. **● - What is difference between `dependencies` vs `devDependencies`**

****

1. **● - What is Tree Shaking?**

**Tree shaking is a term commonly used within a JavaScript context to describe the removal of dead code.**

**It relies on the**[**import**](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/import)**and**[**export**](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/export)**statements to detect if code modules are exported and imported for use between JavaScript files.**

1. **● - What is Hot Module Replacement?**
   1. **Hot Module Replacement (HMR) is a feature that allows developers to update modules in a running application without reloading the page. This can help speed up development by saving time and retaining application state**
2. **● - List down your favourite 5 superpowers of Parcel and describe any 3 of them in your**

**own words.**

**Treeshaking**

**Hot module replacement**

**Minfication**

**Dev lazy mode**

**Diagnostics**

**Reliabile caching**

**Treeshaking is process of removing dead code by analysing imports and exports in amoung javascript files.**

**Dev lazy mode . when we start application we don’t have to load all the modules ,instead we can specify the page we want to run , based on the navigation in the browser , when we select the next page it will build the next page and go on. This is really a good feature.**

**Image optimization in parcel has built in support for resizing, converting, and optimizing images. Images can be referenced from HTML, CSS, JavaScript, or any other file type.**

1. **● - What is `.gitignore`? What should we add and not add into it?**

We will specify the files inn gitignore which can be regenerated when ever we want, for example node modules etc , if we don’t specify nodemodules ,we have to capture them in git**.**

1. **What is the difference between `package.json` and `package-lock.json`**

The package.json file focuses on project metadata and specifying the desired versions of dependencies, while the package-lock.json file ensures deterministic installations by locking the exact versions of dependencies and their dependencies. Both files are essential for managing dependencies, but they serve different purposes in the Node.js development workflow**.**

1. **● - Why should I not modify `package-lock.json`?**

Npm automatically modifies the dependencies on the package-lock.json ,if we modify it may cause to inconsistencies.

1. **● - What is `node\_modules` ? Is it a good idea to push that on git?**

When we install any packages with npm ,all the external packages relateded files and dependensies will be stored in git , so it has huge no of files which can be regenerated ,so its is not preferred to push into it.

1. **● - What is the `dist` folder?**

**It contains the production ready code of our project.**

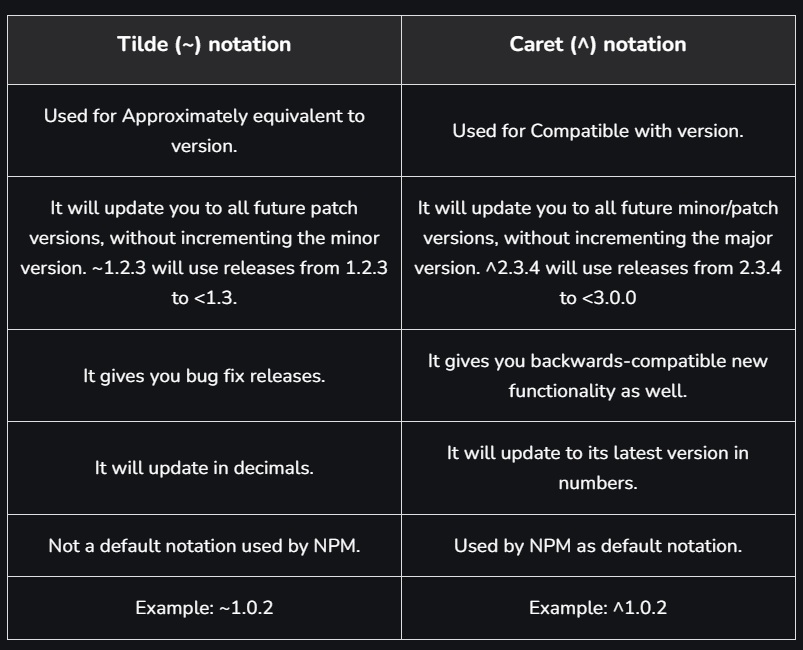
1. **● - What is `browserlists`**

Browserslist is a single configuration that can be used by all front-end tools to determine which browsers are supported. It can be used to:

1. **Read about dif bundlers: vite, webpack, parcel**

**https://www.linkedin.com/pulse/comprehensive-comparison-webpack-parcel-vite-which/**

1. **● Read about: ^ - caret and ~ - tilda**

****

1. **● Read about Script types in html (MDN Docs)**