**Q. What is Webpack?**
Webpack is a static module (css, js, images, fronts) bundler for modern javascript applications.
Webpack also offers multiple functions, like:

- Merging modules/module bundling,
- Code minimizations ( minimization by removing white spaces, remarks, junk code, and code reduction)
- SASS or typescript compilation,
- Npm integration.

**Q. What problem Webpack actually solves?**
In javascript, we have had dependency hierarchy issues many times. That means we must always maintain the order of dependencies when we write code in javascript.
Example,

- When we work with Ajax, ajax needs to be loaded first then we should import our custom code otherwise it fails to load up Ajax. So, we have to maintain a dependencies hierarchy when we want to use Ajax.

Webpack solves the issue of dependencies hierarchy allowing us to make our code modular as much as we want or would like without bothering with dependencies order, import syntax (modern or old), and finally push all the code into a single file and this file will do everything.

**Q. How does Webpack work?**
When Webpack processes our application, it internally builds a dependency graph from one or more entry points and then combines every module of our application into one or more bundles which are static assets to service our contents.

**Q. Core concepts?**
To get started, we need to familiarize ourselves with some **Core Concepts**:

- **Entry**: we need to define what should be the entry point of our project or from which module Webpack will start building an internal dependency graph.
  Default entry,
  `./src/index.js`

- **Output:** The output property tells the Webpack where to emit the bundles it creates and how to name those files.
  It default to,
  `./dist/main.js` (for main output file) and
  `./dist/` (other generated files)

- **Loaders:**
  The job of the loader is to simply work on the things when things are getting loaded. That means before generating one single output (main.js or output.js) all the configurations that require to be done are part of the Loader.
  - css loader,

    - svg loader

- **Plugins:**
  Configurations that need to be done after the compilation or one single output or bundle (bundle.js/ main.js) is being created are part of the plugin.
    - **HtmlWebpackPlugin**:
      i) create `index.html` file inside `dist` directory,
      ii) inject main bundle/output (main.js / bundle.js) into `index.html`

- **Mode:**
  **i) development:**
    - More forgiving / not strict,
    - Aim at debugging or error tracing,
    - Does not aim at minifying or compression.

  **ii) production:**
    - More strict,
    - Doesn't aim at debugging or Error tracing,
    - Enable bundle compression as much as possible.

**References:**
i) https://webpack.js.org/concepts/
ii) https://www.youtube.com/watch?v=yiwSVeHYosQ
iii) https://www.youtube.com/watch?v=9c3dBhvtt6o