Q1: if i have code below in index.html:

```
<base href="/" />
```

and in React component:

```
<img className="product-rating-stars" src={`images/ratings/rating-$
{product.rating.stars * 10}.png`} />
```

then images/ratings/rating... is treated as /images/ratings/rating... right ?

A1: ✅Correct. With `<base href="/">`, all relative URLs (like `images/ratings/rating...`) are resolved from the root of the domain, i.e., `/images/ratings/rating...`.

Q2: is it general to set <base href="/" /> to other folder, not '/' ?

A2: Yes. `<base href>` can be set to any path, not just `/`. For example, `<base href="/myapp/">` makes relative URLs resolve relative to `/myapp/`.

Q3: I don't understand why it also depends on where web app is deployed. The '/' should be the project folder, so wherever I placed my project folder into whatever other project folder, the '/' always means starting from the project folder right? If so, why the location, or the folder where I placed the project folder into, matters?

A3: `/` always refers to the **domain root**, not the project folder. If you deploy your project in a subfolder, `/` still points to the domain root. So relative paths may break unless you adjust `<base>` or your build config.

Q4: so '/' will always refer to the domain root, not where the react project is uploaded.

A4: ✅Exactly. `/` and `<base href="/">` refer to the **root of the domain**, not the project folder.

Q5: if i am on the route /tracking, and inside tracking.jsx i have

```
<img src="image/logo.png" />
```

does this mean https://example.com/myapp/tracking/image/logo.png or https://example.com/myapp/image/logo.png?

A5: It depends on `<base>`: - Without `<base>`: relative paths resolve relative to the current URL. So `/myapp/tracking/` + `image/logo.png` → `https://example.com/myapp/tracking/image/logo.png` ❌usually wrong. - With `<base href="/myapp/">`: relative paths resolve from `/myapp/`, so it becomes `https://example.com/myapp/image/logo.png` ✅correct.

Q6: the default <base> which means I don't set <base> means that by default, the <img src="image/logo.png" /> becomes https://example.com/myapp/tracking/image/logo.png

A6: ✅Correct. Without `<base>`, relative paths are resolved relative to the current URL (including the route), so `/myapp/tracking/image/logo.png`.

Q7: does https://example.com/myapp/tracking/ mean i have a folder named tracking? Or search under https://example.com/myapp/tracking/ means search the image image/logo.png in the same folder level as tracking?

A7: Not necessarily. In React SPA, `tracking` is usually just a **route**, not a physical folder. The browser resolves relative paths as if it were a folder path, but the actual file may not exist there. So `<img src="image/logo.png">` → URL `https://example.com/myapp/tracking/image/logo.png` regardless of physical folder.

Q8: if i set <base href="/tracking/" /> this means the browser will look for <image> as https://example.com/myapp/tracking/log.png, but where is the physical location?

A8: `<base>` only affects **URL resolution**, not physical folders. Browser requests `https://example.com/tracking/log.png`. The server must serve that URL from some folder (physical folder name doesn't have to be `tracking`). In React, `public/` is mapped to `/`, so the server/build tooling handles the mapping.

Q9: you said "Server must have log.png in the folder that maps to /tracking/" but the folder name does not have to be tracking, like in react project, the public, src are mapped to "/"

A9: Correct. React maps `public/` to `/` in the URL. The physical folder name (`public`) doesn't appear in the URL. Browser requests `/tracking/images/logo.png`; build tooling or server must map this to the actual file in `public/`.

Q10: export default defineConfig({ base: '/tracking/', }) this is equivalent to <base href="/tracking/" /> right?

A10: ✅Yes. In Vite, `base` prepends the URL path for all assets, effectively acting like `<base href="/tracking/">` in `index.html`.

Q11: <base href="/tracking/" /> is somehow equal to say https://example.com/myapp/tracking/

A11: Not exactly. `<base href="/tracking/">` sets the **URL path**, browser automatically prepends the domain. To get `https://example.com/myapp/tracking/`, you must include the full path to your deployed app: `<base href="/myapp/tracking/">`.

Q12: So if your app is actually at https://example.com/myapp/ and you write <base href="/tracking/">, this is not a good way right, basically no one will use it

A12: ✅Correct. Setting `<base href="/tracking/">` when the app lives in `/myapp/` will break all relative paths. You should set `<base href="/myapp/">` to match your deployment folder.

Q13: basically why setting <base> in react is almost always a must

A13: Because in SPA, you have one `index.html` and routes can be anything (`/tracking`, `/settings`). Without `<base>`, relative paths break depending on the current route. Setting `<base>` ensures **all relative paths are resolved from the app root** regardless of route.

Q14: and i can always make sure manually that /myapp/ not only a domain, but also a physical folder, which fixes all the problems

A14: This can work, but only if you **mirror all folder structures for every route**, which is cumbersome. Using `<base href="/myapp/">` is easier and more reliable.

Q15: why do you want to set <base href="/tracking/">

A15: You would only set `<base href="/tracking/">` if you intend to deploy your app under `/tracking/` folder. Otherwise, for normal React apps deployed at `/myapp/` or root `/`, you set `<base href="/myapp/">` or `<base href="/">`.

Q16: the default <base> which means I don't set <base>, relative paths resolve relative to current URL, which may break on /tracking routes

A16: Correct. Without `<base>`, relative paths are resolved against the **current URL**, so `/tracking/image/logo.png` may 404 if file doesn’t exist there.

Q17: summary rule of thumb

A17: `<base href>` should match **the root URL where your app is deployed**. Relative paths then resolve consistently across all routes in a React SPA. Without it, assets may break when the route changes.

Q18: Original JSX example included for context

```
<div className="product-rating-container">
  <img
    className="product-rating-stars"
    src={`images/ratings/rating-${product.rating.stars * 10}.png`}
  />
</div>
```

This shows how the relative path `images/ratings/rating-...png` will be resolved according to `<base>` and current route, illustrating why `<base>` is important in SPAs.