Images/Assets in React JS

- Inside public Folder
- Inside src Folder

Inside public Folder

If you put a file into the public folder, it will not be processed by Webpack. Instead it will be copied into the build folder untouched.

To reference assets in the public folder, you need to use a special variable called PUBLIC_URL. Only files inside the public folder will be accessible by %PUBLIC_URL% prefix.

Normally we recommend importing stylesheets, images, and fonts from JavaScript.

<link rel="shortcut icon" href="%PUBLIC_URL%/favicon.ico">

- None of the files in public folder get post-processed or minified.
- Missing files will not be called at compilation time, and will cause 404 errors for your users.
- Result filenames won't include content hashes so you'll need to add query arguments or rename them every time they change.

Inside public Folder

When use Public Folder

- You need a file with a specific name in the build output, such as manifest.webmanifest.
- You have thousands of images and need to dynamically reference their paths.
- You want to include a small script like pace.js outside of the bundled code.
- Some library may be incompatible with Webpack and you have no other option but to include it as a <script> tags

Inside public Folder

How to use

Public Folder -> index.html

App.js

Inside src Folder

With Webpack, using static assets like images and fonts works similarly to CSS.

You can import a file right in a JavaScript module. This tells Webpack to include that file in the bundle. Unlike CSS imports, importing a file gives you a string value. This value is the final path you can reference in your code, e.g. as the src attribute of an image or the href of a link to a PDF.

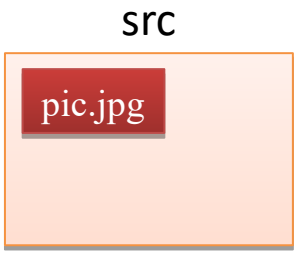
- Scripts and stylesheets get minified and bundled together to avoid extra network requests.
- Missing files cause compilation errors instead of 404 errors for your users.
- Result filenames include content hashes so you don’t need to worry about browsers caching their old versions.

Inside src Folder

How to Use

App.js

```
import pic from './pic.jpg';
<img src={pic} alt="mypic" />
```



This ensures that when the project is built, Webpack will correctly move the images into the build folder, and provide us with correct paths.