1. Add custom element polyfill to support custom elements (npm install --save @ungap/custom-elements)
2. Include @angular/elements library as it provides the Angular function that uses the polyfill to render Angular components
3. Replace app-root with the component tag name
4. Adjust build process
   1. Remove app.component files
   2. Update app.module to remove app component.
   3. Add component tag as entry component
   4. Use the app module constructor to inject our component and use the registerElement polyfill to define this component as a custom element in app.module.ts as below:

*const el = createCustomElement(DemoBoxComponent, { injector });*

*customElements.define('demo-box', el);*

1. Update ng build command in package.json as below
   1. ng build --prod && cat dist/angular-web-component/runtime\*.js dist/angular-web-component/polyfills\*.js dist/angular-web-component/main\*.js > example/demo-box.js && cp dist/angular-web-component/styles.\*.css example/demo-box.css
2. Generate Single JS & CSS component using below command
   1. npm run-script build

Once all above steps are followed, we get JS and CSS files for the component in exported version.

To use this component in plain HTML, we can use a syntax like below.

e.g. If the exported component’s name is **component-name**

**<component-name></component-name>**