

Angular Standalone Directives: Usage and Best Practices

Angular Standalone Directives: Usage and Best Practices

What We Solved

Problem:

- Standalone directive created.
- Error: "Directive is standalone and cannot be declared in NgModule."

Solution:

- Standalone Directives must be imported (not declared).
- Added the directive to imports array of AppModule.
- Console logs now fire correctly and directive is working.

Quick Rules

| Situation | What To Do |

|:-----|:-----|

| Directive is standalone: true | Import it into imports array |

| Directive is NOT standalone | Declare it inside declarations array |

Correct Example

In app.module.ts (when directive is standalone):

```
import { HighlightDirective } from './directives/highlight.directive';
```

```
@NgModule({  
  declarations: [  
    AppComponent,  
    // Do NOT declare HighlightDirective here  
  ],  
  imports: [  
    BrowserModule,
```

Angular Standalone Directives: Usage and Best Practices

```
HighlightDirective // Correct place to add  
  
],  
  
bootstrap: [AppComponent]  
  
})  
  
export class AppModule { }
```

If AppComponent was also standalone (optional approach):

```
@Component({  
  selector: 'app-root',  
  standalone: true,  
  templateUrl: './app.component.html',  
  imports: [HighlightDirective]  
})  
  
export class AppComponent { }
```

Best Practices for Standalone Directives

- Use standalone: true for all small reusable directives.
- Group multiple standalone directives into a single Shared Module only when needed.
- Prefer Renderer2 over direct DOM manipulation (nativeElement.style) for safety.
- Name directive selectors uniquely with prefixes like app, e.g., [appHighlight].
- Keep logic minimal inside directives (Single Responsibility Principle).
- Add @Input() parameters if customization needed.
- Document standalone usage clearly in code comments.
- Lazy-load directive with loadComponent if used in dynamic contexts.

Future Tip: Going Fully Standalone

- Modern Angular (15+) recommends even AppComponent, feature modules, pages be standalone.
- Smaller NgModules = Faster app bootstrapping and better lazy loading.
- Cleaner and more maintainable project structures.

Final Checklist

Angular Standalone Directives: Usage and Best Practices

| Item | Status |

|:-----|:-----|

| Console log working? | Yes |

| Directive imported correctly? | Yes |

| Best practices followed? | Yes |

Happy Angular coding! Always verify the directive's structure and imports.

(Generated by ChatGPT Advanced Angular Deep Dive Assistant)