

Angular Dependency Injection Decorators Cheat Sheet

✓ @Optional()

- Angular will inject `null` if the dependency is not found.
- Useful for optional services (e.g., LoggerService, Analytics).
- Example: constructor(@Optional() private logger?: LoggerService)

✓ @Self()

- Restricts resolution to current injector only.
- Throws error if dependency is not found in same component/directive.
- Example: constructor(@Self() private control: NgControl)

✓ @SkipSelf()

- Skips current injector and looks in parent injector.
- Useful when overriding a service locally but still want parent instance.
- Example: constructor(@SkipSelf() private parent: FormGroupDirective)

✓ @Host()

- Resolves dependency from the host component/directive only.
- Common in directives to access service from the component they're applied to.
- Example: constructor(@Host() private logger: LoggerService)

■ Tips:

- Combine with @Optional() to avoid injection errors.
- Use @Self() to ensure strict encapsulation of service usage.
- Use @SkipSelf() to chain service logic from parent components.
- Use @Host() for component-directive integration and scoped resolution.