Angular InjectionToken Cheat Sheet

✔ What is InjectionToken?

- A wrapper object used to inject non-class-based dependencies like strings, config, booleans.

✔ Basic Syntax:

```
export const API_URL = new InjectionToken<string>('API_URL');
```

✔ Providing Token Value:

```
@NgModule({
  providers: [
    { provide: API_URL, useValue: 'https://api.example.com' }
  ]
})
```

✔ Injecting Token:

```
constructor(@Inject(API_URL) private apiUrl: string) {}
```

✔ Tree-Shaking Behavior:

- If no class injects the token, it will be excluded from bundle.
- Prefer 'useValue' or 'useFactory' for lightweight tree-shakable injection.

✔ Use Cases:

- Runtime configuration (URLs, feature flags)
- Locale/i18n settings (LOCALE_ID)
- Environment-specific setups
- Dynamic injection in lazy modules

✔ Best Practices:

- Always define tokens with types: InjectionToken<T>
- Avoid string literals as tokens (fragile, not tree-shakable)
- Use factory providers when value depends on other DI tokens
- Store tokens centrally in a `tokens.ts` file