**Dagger** is a fully static, compile-time **dependency injection framework** for both Java and Android. [It simplifies managing dependencies in your code by automatically generating the necessary boilerplate code and improving performance compared to reflection-based solutions1](https://developer.android.com/training/dependency-injection/dagger-basics).

Here are **five free resources** where you can learn more about Dagger:

1. [**Dagger Basics on Android Developers**](https://developer.android.com/training/dependency-injection/dagger-basics): This official Android guide covers the fundamentals of Dagger, including benefits, usage, and components[1](https://developer.android.com/training/dependency-injection/dagger-basics).
2. [**Dagger Tutorial**](https://dagger.dev/tutorial/): An interactive tutorial that walks you through building a sample application using Dagger. [It provides code snippets and explanations to help you understand how Dagger works](https://developer.android.com/training/dependency-injection/dagger-basics)[2](https://dagger.dev/tutorial/).
3. [**Introduction to Dagger 2 on Baeldung**](https://www.baeldung.com/dagger-2): A comprehensive tutorial introducing Dagger 2, its principles, and how to use it effectively for dependency injection in Java and Android applications[3](https://www.baeldung.com/dagger-2).
4. [**Dagger Documentation**](https://docs.dagger.io/): The official Dagger documentation provides detailed information about using Dagger, its features, and how to integrate it into your projects[4](https://docs.dagger.io/).
5. [**Dagger 2 on Stack Overflow**](https://stackoverflow.com/questions/41820133/what-is-dagger-and-why-we-use-it): This Stack Overflow thread discusses what Dagger is and why it’s considered an efficient dependency injection framework[5](https://stackoverflow.com/questions/41820133/what-is-dagger-and-why-we-use-it).

Feel free to explore these resources to enhance your understanding of Dagger! 🚀