The **Actor Model** is a design pattern where computation is organized around **independent actors** that communicate by sending messages to each other. [Each actor processes messages sequentially, ensuring encapsulation and avoiding locks1](https://mattferderer.com/what-is-the-actor-model-and-when-should-you-use-it/). Here are five free reference links to learn more about the Actor Model:

1. [**What is the Actor Model & When Should You Use it?**](https://mattferderer.com/what-is-the-actor-model-and-when-should-you-use-it/): This article provides an overview of the Actor Model and its historical context[1](https://mattferderer.com/what-is-the-actor-model-and-when-should-you-use-it/).
2. [**Down and Dirty: Understanding the Actor Model**](https://www.developer.com/design/down-and-dirty-understanding-the-actor-model/): A concise explanation of the Actor Model and its asynchronous communication approach[2](https://www.developer.com/design/down-and-dirty-understanding-the-actor-model/).
3. [**How the Actor Model works by example**](https://www.theserverside.com/blog/Coffee-Talk-Java-News-Stories-and-Opinions/How-the-Actor-Model-works-by-example): Learn about actors as cooperative entities and how they handle messages[3](https://www.theserverside.com/blog/Coffee-Talk-Java-News-Stories-and-Opinions/How-the-Actor-Model-works-by-example).
4. [**Akka Documentation: How the Actor Model Meets the Needs of Modern, Distributed Systems**](https://doc.akka.io/docs/akka/current/typed/guide/actors-intro.html): Detailed insights into using Akka, a popular Actor Model framework[4](https://doc.akka.io/docs/akka/current/typed/guide/actors-intro.html).
5. [**The actor model in 10 minutes**](https://www.brianstorti.com/the-actor-model/): A quick introduction to the Actor Model and its conceptual approach[5](https://www.reactivedesignpatterns.com/lectures/actors.html).

Feel free to explore these resources to deepen your understanding of this powerful design pattern! 🌟