## What is a Message Queue?

A message queue (MQ) allows data to flow between various services and applications within a system. Producers create messages, or data, while consumers receive data. [Workable MQs](https://blog.iron.io/top-10-uses-for-message-queue/) need to be highly scalable so that they use the right resources for the number of messages sent and received.



## The Benefits of a Message Queue

A message queue handles a variety of tasks asynchronously. The end-user doesn’t have to wait for one task to complete to go onto the next one. Think about placing an order online. You click the “buy now” button, and data gets sent off to shipping and distribution systems, but you don’t need to wait for that to happen. Instantly, you’re told that your order has gone through, and usually, you have an email confirmation and an order reference. A message queue will be shuttling these different bits of information where they need to go. It distributes them between systems and services, providing a better user experience and a smoother running system.



## What is RabbitMQ?

A RabbitMQ is one of the popular message brokers used to transmit messages from end to end. Messages are the way to transmit information. A message can contain any kind of information, and it could have information about a process or task. A message broker works as the post office; when we want some message, we put the information in the letter and keep it in the post box, so we can be sure that the letter carrier will eventually deliver them to the recipient.

Get Started with RabbitMQ and Python

<https://www.cloudiqtech.com/rabbitmq-an-open-source-message-broker/>

https://www.javatpoint.com/get-started-with-rabbitmq-and-python

GitHub code example

https://github.com/topics/rabbitmq-python

https://github.com/rabbitmq/rabbitmq-tutorials/blob/main/python/receive.py

Line bot development

https://engineering.linecorp.com/zh-hant/blog/line-device-10