**Documentation**

Note: The Redis Documentation is also available in raw (computer friendly) format in the [redis-doc github repository](http://github.com/antirez/redis-doc). The Redis Documentation is released under the [Creative Commons Attribution-ShareAlike 4.0 International license](https://creativecommons.org/licenses/by-sa/4.0/).

Programming with Redis

* [The full list of commands](https://redis.io/commands) implemented by Redis, along with thorough documentation for each of them.
* [Pipelining](https://redis.io/topics/pipelining): Learn how to send multiple commands at once, saving on round trip time.
* [Redis Pub/Sub](https://redis.io/topics/pubsub): Redis is a fast and stable Publish/Subscribe messaging system! Check it out.
* [Redis Lua scripting](https://redis.io/commands/eval): Redis Lua scripting feature documentation.
* [Debugging Lua scripts](https://redis.io/topics/ldb): Redis 3.2 introduces a native Lua debugger for Redis scripts.
* [Memory optimization](https://redis.io/topics/memory-optimization): Understand how Redis uses RAM and learn some tricks to use less of it.
* [Expires](https://redis.io/commands/expire): Redis allows to set a time to live different for every key so that the key will be automatically removed from the server when it expires.
* [Redis as an LRU cache](https://redis.io/topics/lru-cache): How to configure and use Redis as a cache with a fixed amount of memory and auto eviction of keys.
* [Redis transactions](https://redis.io/topics/transactions): It is possible to group commands together so that they are executed as a single transaction.
* [Mass insertion of data](https://redis.io/topics/mass-insert): How to add a big amount of pre existing or generated data to a Redis instance in a short time.
* [Partitioning](https://redis.io/topics/partitioning): How to distribute your data among multiple Redis instances.
* [Distributed locks](https://redis.io/topics/distlock): Implementing a distributed lock manager with Redis.
* [Redis keyspace notifications](https://redis.io/topics/notifications): Get notifications of keyspace events via Pub/Sub (Redis 2.8 or greater).
* [Creating secondary indexes with Redis](https://redis.io/topics/indexes): Use Redis data structures to create secondary indexes, composed indexes and traverse graphs.

Redis modules API

* [Introduction to Redis modules](https://redis.io/topics/modules-intro). A good place to start learing about Redis 4.0 modules programming.
* [Implementing native data types](https://redis.io/topics/modules-native-types). Modules scan implement new data types (data structures and more) that look like built-in data types. This documentation covers the API to do so.
* [Blocking operations](https://redis.io/topics/modules-blocking-ops) with modules. This is still an experimental API, but a very powerful one to write commands that can block the client (without blocking Redis) and can execute tasks in other threads.
* [Redis modules API reference](https://redis.io/topics/modules-api-ref). Directly generated from the top comments in the source code inside src/module.c. Contains many low level details about API usage.

Tutorials & FAQ

* [Introduction to Redis data types](https://redis.io/topics/data-types-intro): This is a good starting point to learn the Redis API and data model.
* [Writing a simple Twitter clone with PHP and Redis](https://redis.io/topics/twitter-clone)
* [Auto complete with Redis](http://autocomplete.redis.io/)
* [Data types short summary](https://redis.io/topics/data-types): A short summary of the different types of values that Redis supports, not as updated and info rich as the first tutorial listed in this section.
* [FAQ](https://redis.io/topics/faq): Some common questions about Redis.

Administration

* [Redis-cli](https://redis.io/topics/rediscli): Learn how to master the Redis command line interface, something you'll be using a lot in order to administer, troubleshoot and experiment with Redis.
* [Configuration](https://redis.io/topics/config): How to configure redis.
* [Replication](https://redis.io/topics/replication): What you need to know in order to set up master-slave replication.
* [Persistence](https://redis.io/topics/persistence): Know your options when configuring Redis' durability.
* [Redis Administration](https://redis.io/topics/admin): Selected administration topics.
* [Security](https://redis.io/topics/security): An overview of Redis security.
* [Encryption](https://redis.io/topics/encryption): How to encrypt Redis client-server communication.
* [Signals Handling](https://redis.io/topics/signals): How Redis handles signals.
* [Connections Handling](https://redis.io/topics/clients): How Redis handles clients connections.
* [High Availability](https://redis.io/topics/sentinel): Redis Sentinel is the official high availability solution for Redis.
* [Latency monitoring](https://redis.io/topics/latency-monitor): Redis integrated latency monitoring and reporting capabilities are helpful to tune Redis instances for low latency workloads.
* [Benchmarks](https://redis.io/topics/benchmarks): See how fast Redis is in different platforms.
* [Redis Releases](https://redis.io/topics/releases): Redis development cycle and version numbering.

Embedded and IoT

* [Redis on ARM and Raspberry Pi](https://redis.io/topics/ARM): Starting with Redis 4.0 ARM and the Raspberry Pi are officially supported platforms. This page contains general information and benchmarks.

Troubleshooting

* [Redis problems?](https://redis.io/topics/problems): Bugs? High latency? Other issues? Use [our problems troubleshooting page](https://redis.io/topics/problems) as a starting point to find more information.

Redis Cluster

* [Redis Cluster tutorial](https://redis.io/topics/cluster-tutorial): a gentle introduction and setup guide to Redis Cluster.
* [Redis Cluster specification](https://redis.io/topics/cluster-spec): the more formal description of the behavior and algorithms used in Redis Cluster.

Other distributed systems based on Redis

* [Roshi](https://github.com/soundcloud/roshi) is a large-scale CRDT set implementation for timestamped events based on Redis and implemented in Go. It was initially developed for [the SoundCloud stream](http://developers.soundcloud.com/blog/roshi-a-crdt-system-for-timestamped-events).

Specifications

* [Redis Design Drafts](https://redis.io/topics/rdd): Design drafts of new proposals.
* [Redis Protocol specification](https://redis.io/topics/protocol): if you're implementing a client, or out of curiosity, learn how to communicate with Redis at a low level.
* [Redis RDB format](https://github.com/sripathikrishnan/redis-rdb-tools/wiki/Redis-RDB-Dump-File-Format) specification, and [RDB version history](https://github.com/sripathikrishnan/redis-rdb-tools/blob/master/docs/RDB_Version_History.textile).
* [Internals](https://redis.io/topics/internals): Learn details about how Redis is implemented under the hood.

Resources

* [Redis Cheat Sheet](http://www.cheatography.com/tasjaevan/cheat-sheets/redis/): Online or printable function reference for Redis.

Use cases

* [Who is using Redis](https://redis.io/topics/whos-using-redis)

Books

The following is a list of books covering Redis that are already published. Books are ordered by release date (newer books first).

* [Mastering Redis (Packt, 2016)](https://www.packtpub.com/big-data-and-business-intelligence/mastering-redis) by [Jeremy Nelson](https://www.packtpub.com/books/info/authors/jeremy-nelson).
* [Redis Essentials (Packt, 2015)](http://www.amazon.com/Redis-Essentials-Maxwell-Dayvson-Silva-ebook/dp/B00ZXFCFLO) by [Maxwell Da Silva](http://twitter.com/dayvson) and [Hugo Tavares](https://twitter.com/hltbra)
* [Redis in Action (Manning, 2013)](http://www.manning.com/carlson/) by [Josiah L. Carlson](http://twitter.com/dr_josiah) (early access edition).
* [Instant Redis Optimization How-to (Packt, 2013)](http://www.packtpub.com/redis-optimization-how-to/book) by [Arun Chinnachamy](http://twitter.com/ArunChinnachamy).
* [Instant Redis Persistence (Packt, 2013)](http://www.packtpub.com/redis-persistence/book) by Matt Palmer.
* [The Little Redis Book (Free Book, 2012)](http://openmymind.net/2012/1/23/The-Little-Redis-Book/) by [Karl Seguin](http://twitter.com/karlseguin) is a great *free* and concise book that will get you started with Redis.
* [Redis Cookbook (O'Reilly Media, 2011)](http://shop.oreilly.com/product/0636920020127.do) by [Tiago Macedo](http://twitter.com/tmacedo) and [Fred Oliveira](http://twitter.com/f).

The following books have Redis related content but are not specifically about Redis:

* [Seven databases in seven weeks (The Pragmatic Bookshelf, 2012)](http://pragprog.com/book/rwdata/seven-databases-in-seven-weeks).
* [Mining the Social Web (O'Reilly Media, 2011)](http://shop.oreilly.com/product/0636920010203.do)
* [Professional NoSQL (Wrox, 2011)](http://www.wrox.com/WileyCDA/WroxTitle/Professional-NoSQL.productCd-047094224X.html)

Credits

Redis is maintained and developed by [Salvatore Sanfilippo](http://twitter.com/antirez). In the past [Pieter Noordhuis](http://twitter.com/pnoordhuis) and [Matt Stancliff](https://matt.sh/) provided a very significant amount of code and ideas to both the Redis core and client libraries.

The full list of Redis contributors can be found in the [Redis contributors page at Github](https://github.com/antirez/redis/graphs/contributors). However there are other forms of contributions such as ideas, testing, and bug reporting. When it is possible, contributions are acknowledged in commit messages. The [mailing list archives](http://groups.google.com/group/redis-db) and the [Github issues page](https://github.com/antirez/redis/issues) are good sources to find people active in the Redis community providing ideas and helping other users.

Sponsors

The work [Salvatore Sanfilippo](http://antirez.com/) does in order to develop Redis is sponsored by [Redis Labs](http://redislabs.com/). Other sponsors and past sponsors of the Redis project are listed in the [Sponsors page](https://redis.io/topics/sponsors).

License, Trademark and Logo

* Redis is released under the three clause BSD license. You can find [additional information in our license page](https://redis.io/topics/license).
* The Redis trademark and logos are owned by Salvatore Sanfilippo, please read the [Redis trademark guidelines](https://redis.io/topics/trademark) for our policy about the use of the Redis trademarks and logo.