Redis

This blog post was translated by Mistral

Open Redis official website, the first sentence states that Redis is an open-source, in-memory data structure store, commonly used for databases and caching. `Redis` is very common. Install Redis

You can install Redis from the official website. Just like SQLite, how do we use Redis in Python after installation?

```
pip install redis
``` Import redis module.\
>>> import redis\
Create a Redis object with given parameters.\
>>> r = redis.Redis(host='localhost', port=6379, db=0)\
Set key 'foo' to value 'bar'.\
>>> r.set('foo', 'bar')\
Return True if successful.\
True\
Get value of key 'foo'.\
>>> r.get('foo')\
Return value as bytes.\
b'bar' Python documentation provided some examples. Here comes something like `pip`. `pip` is a package

`pip` usually comes with the installation of `python`. If the computer has multiple versions of `Python

```shell
alias python=/usr/local/Cellar/python@3.9/3.9.1_6/bin/python3
alias pip=/usr/local/Cellar/python@3.9/3.9.1_6/bin/pip3
``` The meaning is to specify a certain version of `python` and `pip`. One way is to install it using `

```bash
make
make test
make install
```

Translation:

Specify a certain version of Python and pip. One way is to install it using Homebrew. Another way is to

install it from the source code.

Build process:

make make test make install Redis is starting Redis version 6.2.1 64 bit Warning: no config file specified, using the default config Increased maximum number of open files to 10032 (it was originally set to 4864) monotonic clock: POSIX clock_gettime ⋯Server initialized 87684 M 10 Mar 2021 14:46:06.058 * Ready to accept connections

Version: 6.2.1 Open another terminal window for testing:

$ redis-cli 127.0.0.1:6379> set a 2 running the following code:

```
import redis


r = redis.Redis(host='localhost', port=6379, db=0)
result = r.get('a')
print(result)
```

The output of the code will be:

```
2
```

The Chinese text “127.0.0.1:6379> get a” translates to English as “running the following Redis command: get value for key ‘a’ ” . The number “2” is the value associated with key ‘a’ in the Redis database. `r.set('foo', 'bar')` # set value ‘bar’ for key ‘foo’ in Redis

`print(r.get('foo'))` # print the value associated with key ‘foo’ in Redis

Output:

`shell $ python fib_redis.py b'bar'` I. Redis Cache Example for Fibonacci Sequence To implement Fibonacci sequence using Redis. import redis

redis_instance = redis.Redis(host= ‘localhost’ , port=6379, db=0)

def function(n): stored_number = redis_instance.get(n) if stored_number is not None: return int(stored_number) “‘ if n < 2: # if statement condition is not met res_n = n

else: # if statement condition is met res_n = f(n-1) + f(n-2)

r.set(n, res_n) # set value of ‘r’ at index ‘n’ to ‘res_n’

return res_n

print(f(10))

Function definition and call

This function calculates the Fibonacci sequence recursively

def f(n): if n < 2: return n

else: return f(n-1) + f(n-2) This way it' s done.