

← Previous Next →

P Redis streams consumer groups

## Running the code

Run the Program

Type the following commands into a terminal window:

cd src

python consumer\_group.py

You should see the project start up and generate output similar to the following:

\$ python consumer\_group.py

BOB-1: 2 is a prime number

BOB-2: 3 is a prime number

BOB-0: 5 is a prime number

BOB-2: 7 is a prime number

•••

BOB-0: 79 is a prime number

BOB-2: 89 is a prime number

**CHAOS: Restarted BOB-1** 

BOB-1: Recovering pending messages...

BOB-0: 97 is a prime number

BOB-1: 101 is a prime number

What You Will See

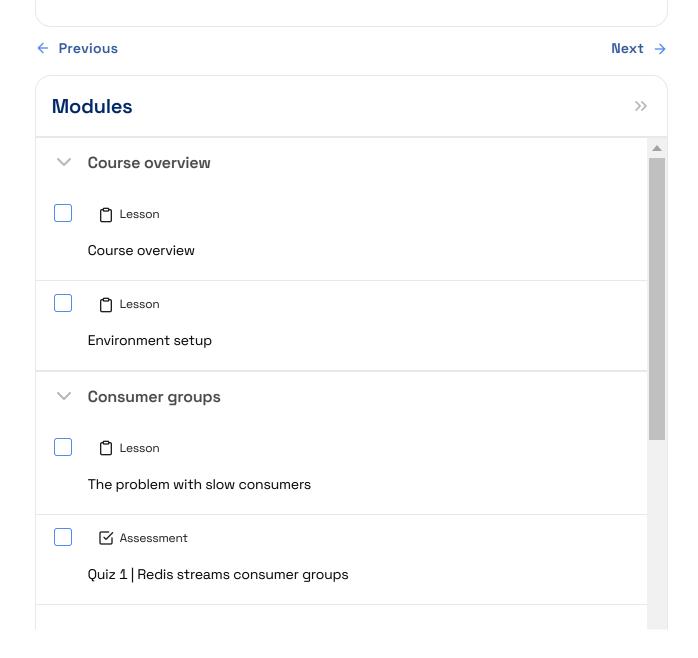
From the output, we can see three consumers (BOB-O, BOB-1, BOB-2) working together as a consumer group to process a stream of numbers and determine if each number is prime or not. Each consumer processes messages one-by-one, but as the message processing takes a variable amount of time, it's not possible to determine which consumer will process which message.

The "Chaos" Function

Periodically, a chaos function (**CHAOS**) restarts one of the consumers, which then has to read its pending entry list so that it can resume processing where it left off. The program also starts a producer process, which doesn't output anything but will keep pushing increasing natural numbers to the stream at semi-random intervals.

## STOPPING THE PROGRAM

To stop the program, click in the terminal window that it is running in, and press **Ctrl + C**.



Lesson	
Consumer groups	
Assessment	
Quiz 2   Redis streams consumer groups	
Lesson	
Adding consumers to a group	
☐ Assessment	
Ouiz Zl Padia atrooma concumor groups	

## Redis

Cloud

Software

Pricing

Support

University Feedback

University Help

Contact us

f

Legal notices



in

0

 $\mathbb{X}$