Ordering Newest Messages First
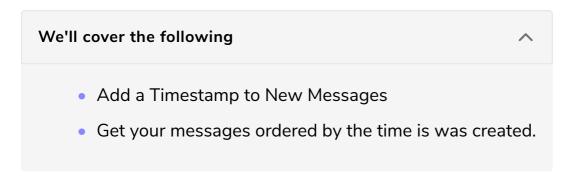
Now we want to show our messages in ascending order. Without this step, they will end up displaying randomly.

We'll cover the following ⌃

- Add a Timestamp to New Messages
- Get your messages ordered by the time is was created.

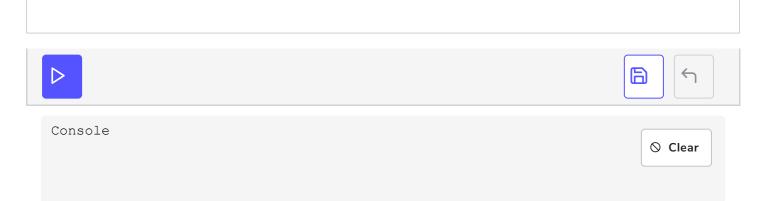# Add a Timestamp to New Messages #

With the JavaScript Date constructor we can generate the current time in milliseconds. We then pass that into Firebase's timestamp method as we save it to the database. Modify your message adding event Firebase query like this:

```
document.querySelector('#message-form').addEventListener('submit', e => {
    e.preventDefault()

    let message = document.querySelector('#message-input').value
    db.collection('messages')
    .add({
        name: name,
        message: message,
        date: firebase.firestore.Timestamp.fromMillis(Date.now())
        })
    .then(docRef => {
        console.log(`Document written with ID: ${docRef.id}`)
        document.querySelector('#message-form').reset()
    })
    .catch(error => {
        console.log(`Error adding document: ${error}`)
    })
})
```

Get your messages ordered by the time is was created.

Now we can make the query we use to get our data more powerful using the `.orderBy()` method. Modify the query like this to show messages in ascending order:

```
db.collection('messages')
.orderBy('date', 'asc')
.onSnapshot(snapshot => {

        document.querySelector('#messages').innerHTML = ''
        snapshot.forEach(doc => {
                let message = document.createElement('div')
                message.innerHTML = `
                <p class="name">${doc.data().name}</p>
                <p>${doc.data().message}</p>
                `
                document.querySelector('#messages').prepend(message)
        })
})
```

This code requires the following API keys to execute: ∧

apiKey	Not Specified...
authDomain	Not Specified...
databaseURL	Not Specified...
projectId	Not Specified...
storageBucket	Not Specified...
messagingSenderId	Not Specified...
appId	Not Specified...

HTML   CSS   JavaScript

Click Run for output...   output

▷   💾   ↩

Console
                                                              ⊘ Clear

In the next lesson, I will show you how to simulate a two-way conversation. See you there.