

메시지 보낼 때 자동으로 스크롤 내려가게 하기

스크롤 내릴 자리를 잡아주기

```
<div style={{
  width: '100%',
  height: '450px',
  border: '.2rem solid #ecec',
  borderRadius: '4px',
  padding: '1rem',
  marginBottom: '2rem',
  overflowY: 'auto'
}}>
  {searchTerm
    ? this.displayMessages(searchResults)
    : this.displayMessages(messages)}
  {this.renderTypingUsers(typingUsers)}
  <div />
</div>
```

잡은 자리를  
계속 참조할 수 있게 해 주기

Refs 를 생성  
React.createRef()

생성한 Refs를  
ref attribute을 통해 넣어주기

### 🔗 Creating Refs

Refs are created using `React.createRef()` and attached to React elements via the `ref` attribute. Refs are commonly assigned to an instance property when a component is constructed so they can be referenced throughout the component.

```
class MyComponent extends React.Component {
```

```

class MyComponent extends React.Component {
  constructor(props) {
    super(props);
    this.myRef = React.createRef();
  }
  render() {
    return <div ref={this.myRef} />;
  }
}

```

생성된 refs로  
참조할 곳에 넣어주기

```

{searchTerm
  ? this.displayMessages(searchResults)
  : this.displayMessages(messages)}
{this.renderTypingUsers(typingUsers)}
<div ref={node => (this.messagesEnd = node)} />

```

```
<div ref={node => console.log('node', node)} />
```

node    <div></div>

메시지를 보낼 때 마다  
참조하고 있는 곳으로  
스크롤 내려주기

```

componentDidUpdate() {
  if (this.messagesEnd) {
    this.messagesEnd.scrollToView({ behavior: "smooth" });
  }
}

```

## 메시지 로딩 중 스켈레톤 처리

하나의 스켈레톤을 만들기

```
function Skeleton() {  
  return (  
    <div className="skeleton">  
      <div className="skeleton-avatar" />  
      <div className="skeleton-author" />  
      <div className="skeleton-description" />  
    </div>  
  )  
}
```

하나의 스켈레톤을  
여러 개 나올 수 있게 해주기  
10번!

똑같은 걸 10번 나오게 하는 방  
법은 ?

같은 컴포넌트를 10번 넣기

Array Constructor를 이용!

[https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global\\_Objects/Array/Array](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/Array)

```
renderMessageSkeleton = (loading) =>  
  loading && (  
    <>  
      {[...Array(10)].map((undefine, i) => (  
        <Skeleton key={i} />  
      ))}  
    </>  
  )
```

엔터 키로 메시지 보내기

```
const handleKeyDown = event => {  
  console.log('event.keyCode', event.keyCode)  
  console.log('event.ctrlKey', event.ctrlKey)  
  
  if (event.ctrlKey && event.keyCode === 13) {  
    handleSubmit();  
  }  
}
```