

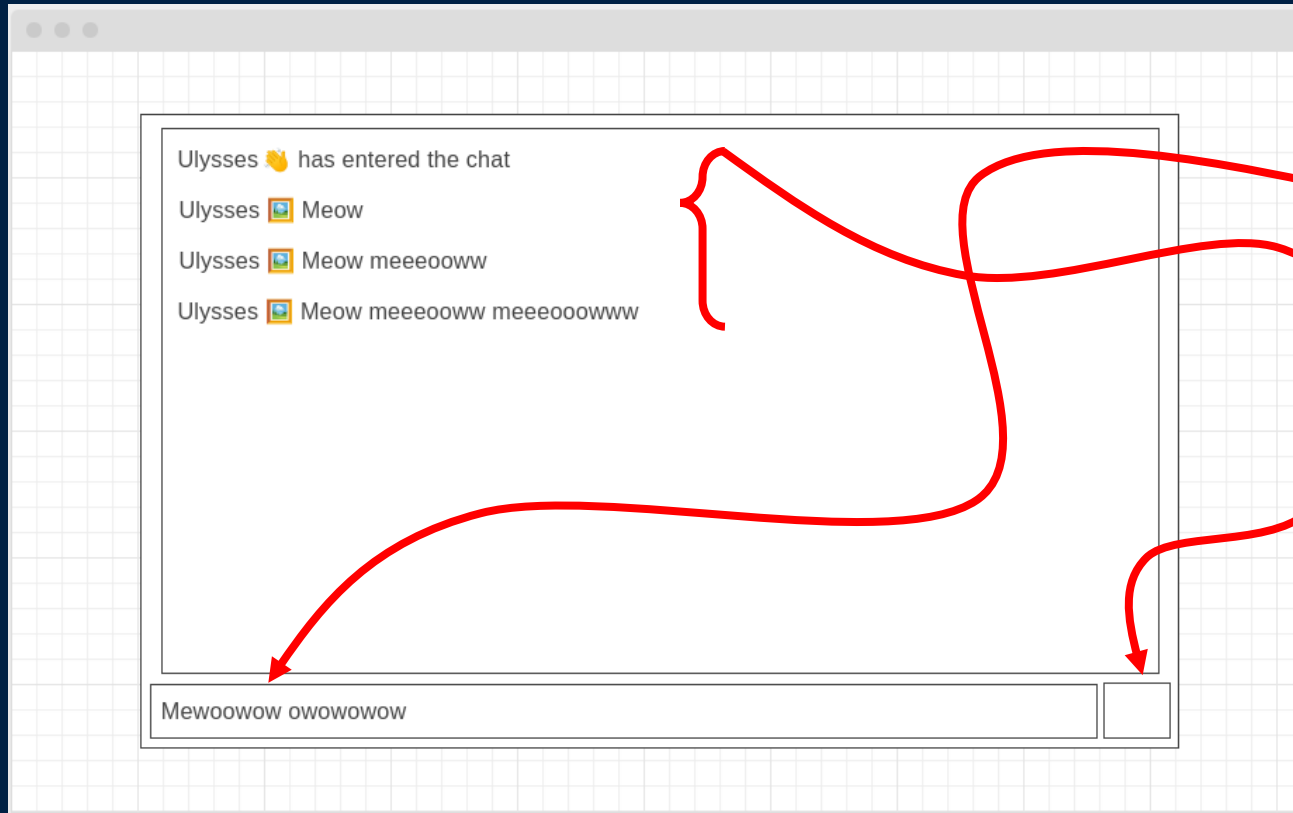


CMPSC 302

WEB DEVELOPMENT



Clicky click 2: Chatroom Bugalú



This interface is driven by the following actions:

- ✦ Text entry in the <textarea>
- ✦ Entry submission via button click or press of <Enter> key
- ✦ Messages from Heroku nodeJS application appearing as <p> in the designed chat window



How to: Bugalú

```
chat.window = document.getElementById("chat-window");
```

Ulysses 🐱 has entered the chat

Ulysses 🐱 Meow

Ulysses 🐱 Meow meeeooww

Ulysses 🐱 Meow meeeooww meeeoowww

Mewoowow owowowow

We need to script this. But how?

Wait! It's part of an HTML <form>!

Back to our HTML!

Handled by HTML <form>



<form>ing opinions

We can script a “submit” event from the form itself, and save some work!

```
<form id = "chatForm" onsubmit="return chat.send(this, 'normal')">
```

```
  <label for="sendMsg" hidden>Message Entry</label>
```

```
  <textarea id = "sendMsg"></textarea>
```

```
  <button id = "sendBtn">
```

```
    &raquo;
```

```
  </button>
```

```
</form>
```

HTML “entity” for “right-angled quote” (raquo): »



<form>ing opinions

```
<form id = "chatForm" onsubmit="return chat.send(this, 'normal')">  
  <label for="sendMsg" hidden>Message Entry</label>  
  <textarea id = "sendMsg"></textarea>  
  <button id = "sendBtn">  
    &raquo;  
  </button>  
</form>
```



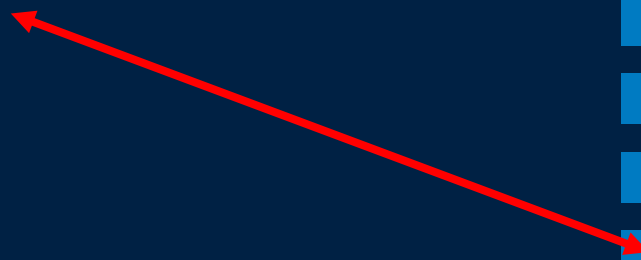
For the grader's sake, I need everyone to name the button and form the same as what I have here!



HTML Savings Time (-10 minutes)

```
sendMsg.addEventListener("keydown", (evt) => {  
  if(evt.key == "Enter"){  
    sendBtn.click();  
    evt.preventDefault();  
  }  
});
```

```
send: (message, type) => {  
  let msg = {  
    user: chat.name,  
    text: sendMsg.value || message,  
    type: type  
  }  
  if(typeof(msg.text) !== "string") return false;  
  chat.socket.send(JSON.stringify(msg));  
  if(type !== "ping") sendMsg.value = "";  
  return false;  
}
```





In any event...



`<form> submit`



`chat.send`



Some extras

Either the value of our `<textarea>` or a populated message variable

```
let msg = {  
  user: chat.name,  
  text: sendMsg.value || message,  
  type: type  
}  
  
if(typeof(msg.text) !== "string") return  
false;
```

A blank `<textarea>` can send a type of `object` here; this prevents that.



post-al service

```
post: (message) => {
```

```
  let msg = document.createElement("p");
```

```
  let text = document.createElement("span")
```

```
  text.className = "chat-msg";
```

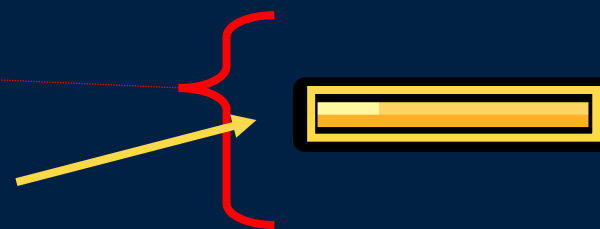
```
  text.innerText = `${message}`;
```

```
  msg.appendChild(text);
```

```
  chat.window.appendChild(msg);
```

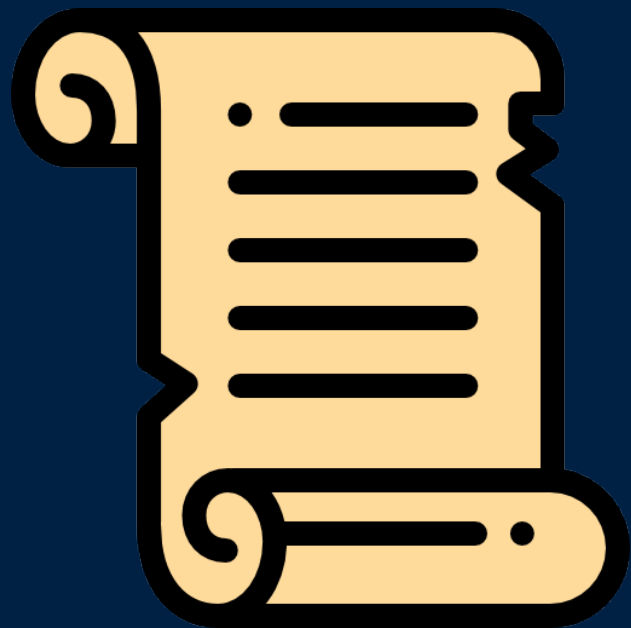
```
  return false;
```

```
}
```

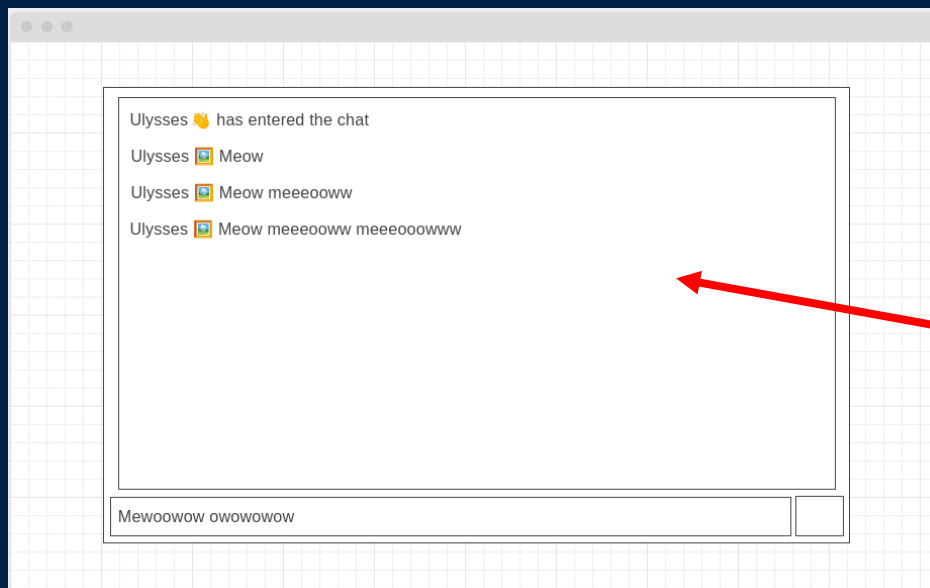




Scroll out



As new messages get added, how do we allow the window to scroll?

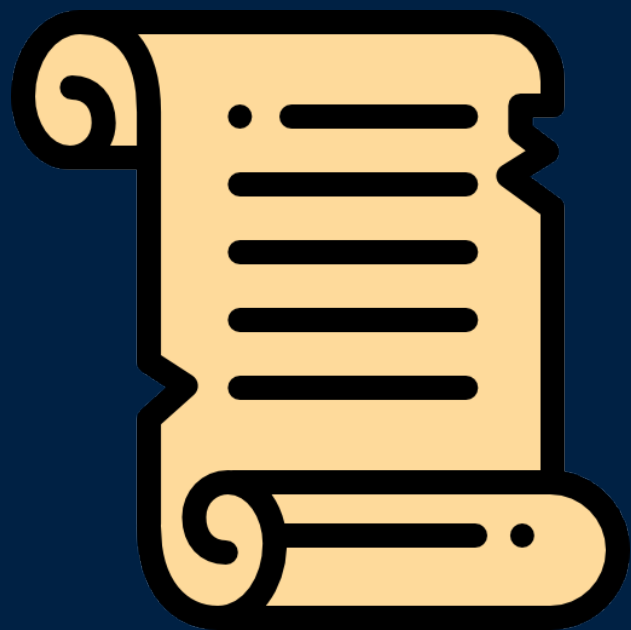


CSS:

`overflow-y: scroll;`



Scroll out

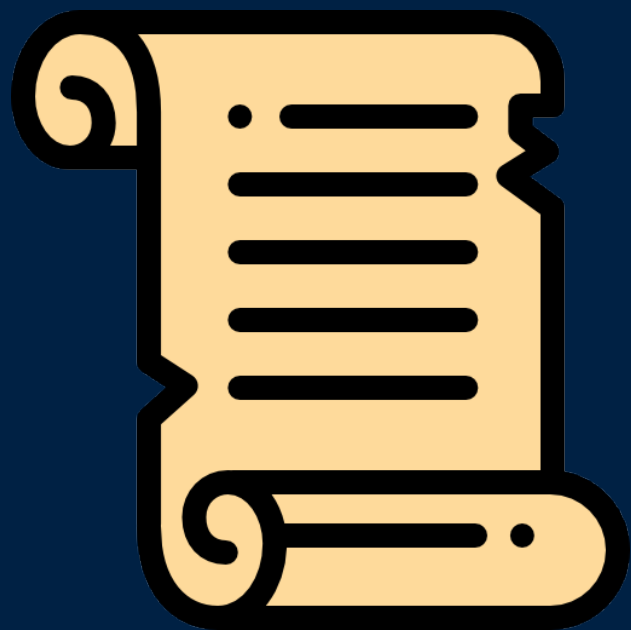


Javascript:

```
scroll: () => {  
  let msgs = Array.from(  
    document.getElementsByClassName("chat-msg")  
  );  
  let pos = msgs[msgs.length - 1].offsetTop;  
  document.getElementById("chat-window").scrollTo({  
    top: pos,  
    behavior: "smooth"  
  });  
  return false;  
}
```



Scroll out



Javascript:

```
post: (message) => {  
  let msg = document.createElement("p");  
  let text = document.createElement("span")  
  text.className = "chat-msg";  
  text.innerText = `${message}`;  
  msg.appendChild(text);  
  chat.window.appendChild(msg);  
  return false;  
  chat.scroll();  
},
```