CLIENT-SIDE VALIDATION

REMEMBER: Client-side JS is NOT secure.

- Fully visible to the user
- Fully alterable by the user

Client-side JS provides **convenience**, not **security**

"Validation" is one such convenience.

WHAT IS VALIDATION?

- Preventing the user from submitting invalid information
- Informing the user of what changes need to happen

There are MANY approaches

Does **not replace** server-side validation

But may be the friendly version

STANDARDS-BASED VALIDATION

There are some standards to allow HTML to do the validation automatically

These standards are pretty minimal

Support for them is even more minimal

Liking the UI is even less than that

Most validation is JS-based (AND server-based)

SIMPLE EXAMPLE

The chat application allows you to enter an empty message

We can disable the submit button until they have text

No messaging needed -- **discoverability**

CREATE SOME CLIENT-SIDE JS

Add to our HTML

```
  <script src="/chat.js"></script>
```

Create a chat.js file **in public/** (static asset)

```
console.log("Hello world");
```

REMEMBER client-side JS is just text to the server, it runs on the browser, not the server.

ATTEMPT A SMALL CHANGE

```
const sendButton = document.querySelector(".send button");
const toSend = document.querySelector(".to-send");
sendButton.disabled = true;
```

WHY DID THAT FAIL?

Remember to check your console.

The JS executes on the login - no such elements

OKAY, ONLY WHEN ELEMENTS PRESENT

```
const sendButton = document.querySelector(".send button");
const toSend = document.querySelector(".to-send");
if( sendButton && toSend ) {
  sendButton.disabled = true;
}
```

NOW WHAT?

You can debug via Sources tab.

Set a breakpoint.

See what happens.

AFTER BODY

You can set all your JS to run after the page an event says the page is done loading.

But that still pauses the page to get the JS.

Better to not even load it until you have a page.

Place the `<script>` tag just before the closing `</body>` tag

YAY! EXCEPT...

You are polluting the global scope

Put your code in an IIFE:

```
(function IIFE() {
  // Your code here
})();
```

ADD SOME COMPLEXITY

```
( function IIFE() {
  const sendButton = document.querySelector(".send button");
  const toSend = document.querySelector(".to-send");
  if(toSend && sendButton) {
    sendButton.disabled = !toSend.value;
    toSend.addEventListener('input', (e) => {
      sendButton.disabled = !e.target.value;
    });
  }
})();
```

ARE YOU REQUIRING JS?

You should always consider if you're **requiring** client-side JS.

Arguments exist on whether this is a reasonable requirement.

You should consider the cost/benefits.

Progressive Enhancement - make it work without JS, then add JS.

NOW EXTEND

- Don't allow login without a username
  - Does your server-side blow up? client-side JS won't fix that!