Form validation

Used to make sure the user is entering valid data. Two ways Server-side client-side.

Client- Most validation is done here. Used to make sure input fields are filled, etc. JavaScript is used for this most of the time, however some elements have attributes that can do this for us such as:

<input type=”text” required>

Server-side- Used for sanitizing user input, especially when connecting to a DB.

Constraints

Limit the number of characters with min- or maxlength attribute. For number min and max for the number such as a scale between 1 and 10

Error messages

Can only be changed with javascript

var email = document.getElementById("mail");

email.addEventListener("input", function (event) {

if (email.validity.typeMismatch) {

email.setCustomValidity("I expect an e-mail, darling!");

} else {

email.setCustomValidity("");

}

});

Fetch

async function postData(url = '', data = {}) {

// Default options are marked with \*

const response = await fetch(url, {

method: 'POST', // \*GET, POST, PUT, DELETE, etc.

mode: 'cors', // no-cors, \*cors, same-origin

cache: 'no-cache', // \*default, no-cache, reload, force-cache, only-if-cached

credentials: 'same-origin', // include, \*same-origin, omit

headers: {

'Content-Type': 'application/json'

// 'Content-Type': 'application/x-www-form-urlencoded',

},

redirect: 'follow', // manual, \*follow, error

referrer: 'no-referrer', // no-referrer, \*client

body: JSON.stringify(data) // body data type must match "Content-Type" header

});

return await response.json(); // parses JSON response into native JavaScript objects

}