You've most likely encountered forms

when registering an account on a website,

or when filling out your address

for something you've bought online.

As a developer, you'll use

HTML forms to capture user input.

Capturing input is one thing,

but the form also needs to ensure the data is usable.

How can you ensure the user input is valid?

For example, let's say you are

building an online food delivery website.

Say a user accidentally makes an error when

entering the delivery address and the form accepts it,

although the location does not exist.

It will cause a very bad user experience

when the food never arrives.

This is where form validation comes in.

Form validation is a process

that ensures that the data entered by

the user in a form is valid

and conforms to rules defined by the developer.

There are two methods of form validation,

client-side validation and server-side validation.

Client-side validation checks for

errors as soon as they are typed into the form.

This is done by the web browser or by JavaScript code,

and provide the user with immediate feedback.

The client-side validation process starts

by checking if a form has been filled out correctly.

If there are no errors,

the form will be submitted to the server for processing.

However, if there are errors,

an error message will alert the user of

the invalid data and how to

change it for successful submission.

For example, you decide to use

the input element with its type attributes set to email.

If the user does not enter a valid email address,

the web browser will display an error message

informing them that the data they

entered is not a valid email address.

On the other hand, server-side validation checks for

errors after the data has

been submitted to the web server.

Server-side validation is more

secure because it prevents malicious users

from tampering with your website's code

and submitting invalid data to your server.

When the form data is received by the web server,

the backend will validate the data before processing it.

This validation can run more complex checks,

such as checking the data against a database,

or validating the data against business requirements.

Most websites use both client-side and

server-side validation to provide

immediate feedback to users,

but also to protect against

malicious data submission and to ensure data integrity.

Let's examine how HTML allows

you to do simple client-side validation.

HTML has several input types

that are validated by the web browser.

As demonstrated in the example,

email is used for email addresses.

Others include "tel" for telephone numbers,

"url" for URLs such as www.data.com,

"date" for date values,

"time" for time values,

"number" for numeric values,

"range" for numeric values which

have a minimum and maximum value,

and "color" for color selection.

Say for instance, a user makes mistakes on

a user account form when entering

a telephone number and URL,

the browser will validate the entered data against

the requirements of the input type

and provide user feedback.

Another example is the required attribute,

which indicates that the user must

supply a value to an input field.

The web browser will alert a user if

a required value is outstanding.

That's the basics of client and

server-side validation of forms.

You'll learn more about HTML validation capabilities

in the next reading.

As a developer, you can add client-side validation to a form by using specific input types.



Yes



No

orrect. Client-side validation checks for errors as soon as they are typed into the form. The web browser does this by validating the submitted data against the specified input type. The web browser will show an error message if the wrong type of input has been submitted