> *You might not think that programmers are artists, but programming is an extremely creative profession. It's logic-based creativity.*
> —John Romero

Citaten toevoegen

1 Formulier op de pagina zelf

In het eerste filmpje maken we een eenvoudig formulier dat een gebruiker kan invullen om een nieuw citaat toe te voegen. We besteden aandacht aan:

- HTML: `form`, `label` (accessibility), `input`;
- JS: `value` opvragen van een input field;
- JS: `voegToe()` functie met `push` en `dispatchEvent`
- aandacht voor usability (o.a. melding);
- JS: `preventDefault()`;
- CSS: breedte van input fields en button.

2 Formuliervalidatie

Via het formulier kan de gebruiker interageren met de pagina. In dit geval: een nieuw citaat toevoegen. Maar is alles wel ingevuld? Is het citaat niet te lang? In een correcte taal?

Er kan vanalles fout lopen als een gebruiker dingen doet op een pagina. Daarom gaan we die input van de gebruiker nakijken (*valideren*)

Om de serverbelasting te verminderen kijken we al zoveel mogelijk dingen na aan de clientzijde, dus in de browser. Dat kan zowel met HTML (beperkt) als met JS.

Als een gebruiker een fout maakt, is het een goed idee om die fout zo snel en duidelijk mogelijk te signaleren met een duidelijke melding zodat de gebruiker niet te lang moet wachten op feedback. Qua feedback denken we trouwens niet alleen aan fouten, maar ook aan meldingen dat iets gelukt is.

Formuliervalidatie kan zowel in HTML als in JavaScript worden uitgevoerd, afhankelijk van de specifieke behoeften van je toepassing.

HTML-validatie is eenvoudig en kan worden bereikt door bvb. het gebruik van het `required` attribuut en andere attributen zoals `min`, `max`, en `pattern` voor meer specifieke validaties. Dit biedt een snelle en eenvoudige manier om te controleren of de invoer van de gebruiker voldoet aan bepaalde criteria *voordat* het formulier wordt verzonden.

JavaScript-validatie biedt meer flexibiliteit en controle. Je kan complexere controles uitvoeren, zoals het controleren van de geldigheid van een e-mailadres,

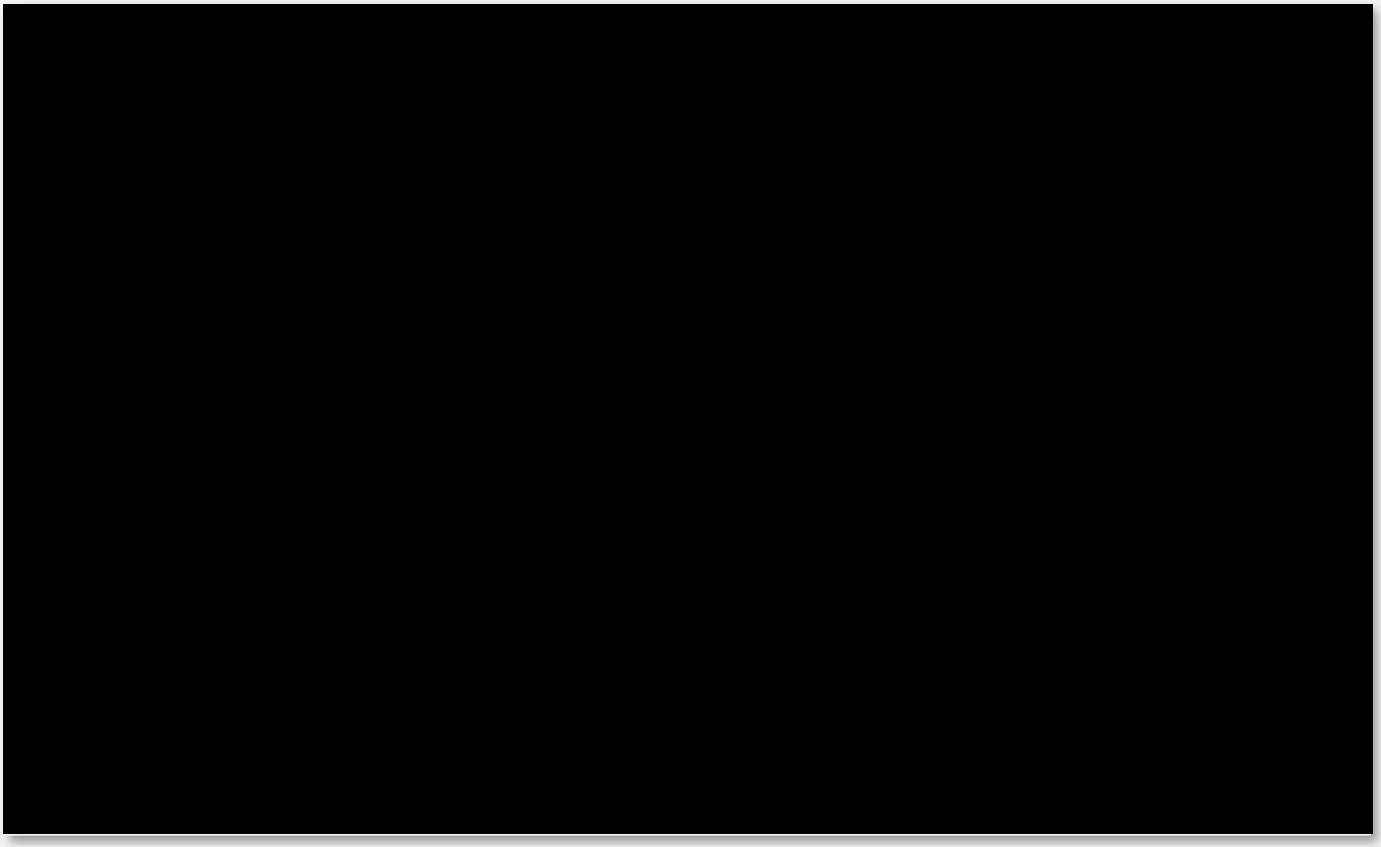
het vergelijken van twee wachtwoorden voor overeenstemming, of zelfs asynchrone validaties zoals het controleren van de beschikbaarheid van een gebruikersnaam. Dit vereist meer code en is iets complexer, maar het biedt een robuustere oplossing.

In de praktijk wordt vaak een *combinatie van beide* gebruikt: HTML-validatie voor snelle, client-side checks en JavaScript-validatie (java, PHP, .NET, ...) voor meer complexe scenario's. Het is ook belangrijk om server-side validatie te hebben als back-up, omdat client-side validatie kan worden omzeild.

Onderstaand filmpje zorgt ervoor dat geen enkel verplicht veld leeg mag blijven. Dat kan eenvoudig in HTML, maar we laten het liever ook zien hoe je het in JS doet.

- HTML required attribuut;
- falsy / truthy;
- correcte melding weergeven: fout / succes;
- oefening: controleer of de taal één van de beschikbare talen is.

We maakten de afspraak dat elk citaat een unieke titel heeft. Bij relationele databanken (SQL) zouden we dat een ‘primaire sleutel’ noemen. Het laatste filmpje voegt enkele regels JS toe die nakijken of de titel die de gebruiker invoegt nog niet bestaat.



↑ Naar top