formulieren

## formulieren

### Voorbeeld

#### input

<p>Dit is een tekstveld</p>

<input type="text" />

<p>Dit is een emailveld</p>

<input type="email" />

<p>Dit is een wachtwoordveld</p>

<input type="password" />

<p>Dit is een checkbox</p>

<input type="checkbox" />

<p>Dit is een radiobutton</p>

<input type="radio" />

<p>Dit is een slider</p>

<input type="range" />

<p>Dit is een nummeriek veld</p>

<input type="number" />

<p>Dit is een datumveld</p>

<input type="date" />

<p>Dit is een bestandsupload</p>

<input type="file" />

<p>Dit is een kleurkiezer</p>

<input type="color" />

### Te gebruiken bij:

-

### Eigenschappen:

* Om een formulier te maken, gebruiken we de <form></form>-tag.
* versturen wanneer de gebruiker op "verzenden" drukt, zullen we echter JavaScript nodig

### Info

Op alle input-elementen kun je een aantal standaard attributen kwijt, die de werking beïnvloeden.

* Het placeholder-attribuut: deze geeft tekst of andere waardes weer zolang het veld leeg is;
* Het value-attribuut: deze stelt de waarde van het veld vooraf in.
* Het checked-attribuut: deze zet input's van type="checkbox" of type="radio" vooraf ingesteld "aan" of "uit";
* Het min en max attribuut: deze zet het minimale of maximale getal in een nummeriek invoerveld;
* Het minLength en maxLength-attribuut: deze zet het minimale of maximale toegestane aantal karakters in een tekstueel invoerveld.

### Buttons

#### buttons buiten een form

<button type="button">

Ik sta buiten een formulier!

</button>

#### buttons binnen een form

<form>

<button type="button">

Ik sta binnen een formulier!

</button>

</form>

We kunnen ook bepalen of een button klikbaar is of niet. Wanneer producten in een webshop uitverkocht zijn, of de gebruiker niet alle verplichte velden van het formulier heeft ingevuld,

<button type="button" disabled="true">

Deze knop is niet klikbaar

</button>

#### Types buttons

het type-attribuut op de button expliciet moeten meegeven:

type="button": Dit is het standaardtype voor een knop. Het gedraagt zich als een gewone knop waarop gebruikers kunnen klikken.

htmlCopy code

<button type="button">Klik hier</button>

type="submit": Dit type wordt vaak gebruikt binnen formulieren. Wanneer de knop wordt ingedrukt, wordt het bijbehorende formulier verzonden naar de server.

htmlCopy code

<button type="submit">Verzenden</button>

type="reset": Dit type wordt gebruikt om alle ingevulde waarden in een formulier te resetten naar de oorspronkelijke waarden.

htmlCopy code

<button type="reset">Reset</button>

type="checkbox": Dit type wordt gebruikt om een selectievakje (checkbox) te maken.

htmlCopy code

<input type="checkbox" name="voorkeur" value="ja">Ik ga akkoord met de voorwaarden

type="radio": Dit type wordt gebruikt voor een keuzerondje (radio button). Het wordt meestal gebruikt in combinatie met andere radio buttons waaruit slechts één optie kan worden geselecteerd.

htmlCopy code

<input type="radio" name="gender" value="male">Man <input type="radio" name="gender" value="female">Vrouw

Dit zijn slechts enkele voorbeelden van de typen attributen voor knoppen in HTML. Er zijn nog andere attributen en types beschikbaar, afhankelijk van de specifieke behoeften en functionaliteit van de knoppen in je webpagina.

#### Radiobuttons en checkboxes

De eigenschap van radiobuttons is dat de gebruiker er altijd maar één tegelijkertijd kan aanvinken. Denk aan het samenstellen van een burrito bij de Mexicaan om de hoek: wil je een meergranen tortilla, of een gewone witte? Bij checkboxes mag de gebruiker meerdere waardes aanvinken. Denk bijvoorbeeld aan de toppings die men kan aankruisen: wil je er kaas, salsa en avocado op of slechts één van de drie? Door deze functionaliteit zul je met twee dingen rekening moeten houden:

* De gebruiker begrijpt welke opties hij of zij aankruist, omdat we dit communiceren via het label-element. Het is echter niet mogelijk om label-informatie mee te sturen naar de server: alleen de gegevens van input-elementen worden verzameld. Om ervoor te zorgen dat de server toch begrijpt welke waarde er zojuist aangekruist is, hebben radio- en checkbox inputs altijd een value-attribuut nodig. De gebruiker ziet hier niets van.
* We kunnen de gebruiker één vraag stellen die één of meerdere antwoorden mag bevatten. Daarom moeten we duidelijk aangeven welke radiobuttons en welke checkboxes onderdeel zijn van dezelfde vraag.

<form onsubmit="hier-hebben-we-js-voor-nodig">

<p>Kies een tortilla:</p>

<label for="whole-weat-radio">

<input type="radio" id="whole-weat-radio" value="whole-weat-tortilla" name="type"/>

Meergranen

</label>

<label for="normal-radio">

<input type="radio" id="normal-radio" value="normal-tortilla" name="type" />

Normaal

</label>

<p>Kies toppings:</p>

<label for="cheese-checkbox">

<input type="checkbox" id="cheese-checkbox" value="cheese-topping" name="toppings"/>

Kaas

</label>

<label for="avocado-checkbox">

<input type="checkbox" id="avocado-checkbox" value="avocado-topping" name="toppings" />

Avocado

</label>

<label for="salsa-checkbox">

<input type="checkbox" id="salsa-checkbox" value="salsa-topping" name="toppings" />

Salsa

</label>

<button type="submit">Verzenden</button>

</form>

Nu doet ons formulier wat we verwachten, en krijgt de server de data netjes en duidelijk aangeleverd:

type: whole-weat-tortilla

toppings: cheese-topping, salsa-topping

### labels

<!-- label en input los van elkaar -->

<label for="user-email-field">Email:</label>

<input type="email" id="user-email-field"/>

<!-- label en input genest -->

<label for="user-newsletter-field">

<input type="checkbox" id="user-newsletter-field" />

Ik wil de nieuwsbrief ontvangen

</label>

#### for

Stel, we maken een registratieformulier waarin mensen hun naam, geboortedatum en woonplaats kunnen invullen omdat we een buurtonderzoek zijn gestart. De reden dat we gebruikers dit formulier laten invullen, is omdat we deze data ergens op willen slaan zodat iets mee kunnen doen.

Om dit te kunnen realiseren, zullen we onze data dus altijd eerst naar de backend moeten versturen. We hebben alvast het volgende formulier gebouwd:

<form onsubmit="hier-hebben-we-js-voor-nodig">

<label for="name-field">Naam</label>

<input type="text" id="name-field"/>

<label for="dob-field">Geboortedatum</label>

<input type="date" id="dob-field"/>

<label for="birthplace-field">Geboorteplaats</label>

<input type="text" id="birthplace-field"/>

<button type="submit">Verzenden</button>

</form>

Wanneer een gebruiker dit in zou vullen, versturen we wellicht het volgende pakketje aan data naar de server:

Klara Pietersen

05-09-1987

Muiden

### Input type

<input>-tag heeft *geen* closing tag: hij is **self closing**. Dit betekent dat de sluitende tag zich al in de beginnende tag begint: <input />. Het input-element verandert in *werking* en *uiterlijk* op basis van het type-attribuut. Dit type attribuut kan één van de volgende waardes hebben:

* type="text" geeft een tekstveld;
* type="email" geeft een tekstveld die en e-mailadres verwacht;
* type="password" geeft een tekstveld waarvan de ingetypte waardes als \*\*\*\* verschijnen;
* type="checkbox" geeft een checkbox die aan- en uitgevinkt kan worden;
* type="radio" geeft een radio button die aan- en uitgevinkt kan worden;
* type="range" geeft een slider;
* type="number" geeft veld om nummerieke waardes in te voeren;
* type="date" geeft veld om datums in te voeren (als *dd/mm/yyyy*);
* type="file" geeft de mogelijkheid tot het uploaden van een afbeelding of bestand;
* type="color" geeft een kleurkiezer om alle kleuren op het spectrum te kiezen.

### Input attributen

* Het placeholder-attribuut: deze geeft tekst of andere waardes weer zolang het veld *leeg* is;
* Het value-attribuut: deze stelt de waarde van het veld vooraf in.
* Het checked-attribuut: deze zet input's van type="checkbox" of type="radio" vooraf ingesteld "aan" of "uit";
* Het min en max attribuut: deze zet het minimale of maximale getal in een *nummeriek* invoerveld;
* Het minLength en maxLength-attribuut: deze zet het minimale of maximale toegestane aantal karakters in een tekstueel invoerveld.
* name: Hiermee wordt de naam van het input-veld opgegeven, wat wordt gebruikt om de waarde ervan te identificeren wanneer het formulier wordt ingediend.
* required: Als dit attribuut aanwezig is, wordt aangegeven dat het invoerveld verplicht is. Het formulier kan niet worden verzonden zonder een waarde in dit veld.
* disabled: Hiermee wordt het invoerveld gedeactiveerd, zodat gebruikers het niet kunnen bewerken of er gegevens in kunnen typen.
* readonly: Hiermee wordt het invoerveld alleen-lezen gemaakt, zodat gebruikers de waarde kunnen bekijken maar niet kunnen bewerken.

### lijsten

In HTML zijn er verschillende manieren om lijsten weer te geven. De twee meest voorkomende typen lijsten zijn ongeordende lijsten en geordende lijsten.

Ongeordende lijst (<ul>):

Gebruik het <ul>-element om een ongeordende lijst te maken.

Gebruik het <li>-element om elk item in de lijst te definiëren.

Plaats de gewenste inhoud tussen het openingstag <li> en het sluitingstag </li>.

Hier is een voorbeeld van een ongeordende lijst met drie items:

html

Copy code

<ul>

<li>Item 1</li>

<li>Item 2</li>

<li>Item 3</li>

</ul>

Geordende lijst (<ol>):

Gebruik het <ol>-element om een geordende lijst te maken.

Gebruik het <li>-element op dezelfde manier als bij een ongeordende lijst.

Hier is een voorbeeld van een geordende lijst met drie items:

html

<ol>

<li>Item 1</li>

<li>Item 2</li>

<li>Item 3</li>

</ol>

Daarnaast zijn er ook geneste lijsten waarbij je een lijst binnen een lijst kunt maken. Dit kan handig zijn om een hiërarchie of structuur aan te geven. Je kunt simpelweg een <ul> of <ol> element en <li> elementen binnen een ander <li> element plaatsen.

Hier is een voorbeeld van een geneste lijst:

html

<ul>

<li>Item 1</li>

<li>Item 2

<ul>

<li>Subitem 1</li>

<li>Subitem 2</li>

</ul>

</li>

<li>Item 3</li>

</ul>

Dit zou een ongeordende lijst creëren waarbij "Item 2" een geneste lijst bevat met twee subitems.

Tot slot kan je ook lijsten binen lijsten maken op deze manier

<ul>

<li>Item 1</li>

<li>Item 2

<ul>

<li>

<ul>

<li>Subitem 1</li>

<li>Subitem 2</li>

</ul>

</li>

<li>

<ul>

<li>Subitem 1</li>

<li>Subitem 2</li>

</ul>

</li>

</ul>

</li>

</ul>

Met deze verschillende lijstelementen kun je inhoud op een gestructureerde en georganiseerde manier weergeven op je webpagina's

### navigeren

Het <nav>-element in HTML wordt gebruikt om een sectie van de pagina te definiëren die navigatielinks bevat. Hiermee kun je een navigatiemenu maken voor je website. Het <nav>-element wordt meestal gebruikt in combinatie met andere elementen zoals <ul> (ongeordende lijst) en <li> (lijstpunt) om de navigatielinks te structureren.

Hier is een voorbeeld van het gebruik van het <nav>-element:

html

<nav>

<ul>

<li><a href="#home">Home</a></li>

<li><a href="#about">Over ons</a></li>

<li><a href="#services">Diensten</a></li>

<li><a href="#contact">Contact</a></li>

</ul>

</nav>

In dit voorbeeld wordt een navigatiemenu gemaakt met vier links: "Home", "Over ons", "Diensten" en "Contact". De links zijn opgenomen in een ongeordende lijst <ul>, en elk item in de lijst wordt weergegeven met het <li>-element. De daadwerkelijke linktekst wordt gedefinieerd met het <a>-element, waarbij de href-attribuut wordt gebruikt om de bestemming van elke link aan te geven.

Het <nav>-element helpt bij het aangeven van de sectie van de pagina die als navigatie dient, en dit kan nuttig zijn voor zoekmachines en toegankelijkheidsdoeleinden. Het is een semantisch element