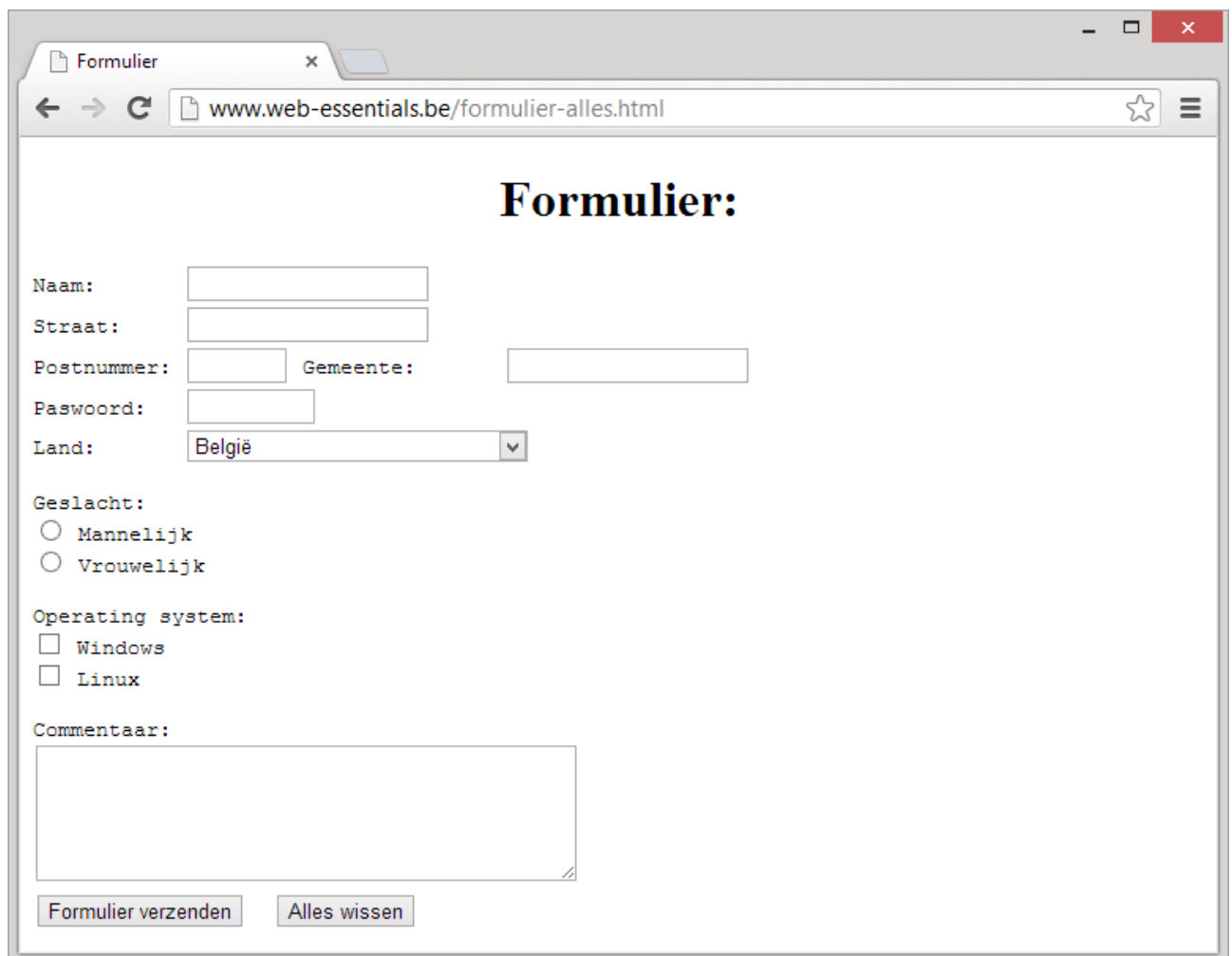


1 Formulieren

1.1 Omschrijving

Formulieren verhogen de interactiviteit van de website en zijn te vergelijken met het dialoogvenster van een programma. Formulieren worden niet enkel gebruikt voor het opgeven van contactgegevens en het versturen van bevestigingen, maar zijn zeker op zeer interactieve pagina's onmisbaar. Bij het klassieke gebruik van een formulier stuurt de websurfer gegevens naar een webserver die daar verwerkt worden door een webapplicatie via een achterliggend script (cgi, php, perl, ...). Die verwerkte gegevens kunnen in een databank bewaard worden en/of desgewenst onder de vorm van een HTML-pagina teruggestuurd worden naar de verzender als respons. De ingevulde gegevens kunnen ook via e-mail verzonden worden. Op interactieve pagina's die bijvoorbeeld simulaties toelaten, worden de gegevens verwerkt door een scripttaal (JavaScript, VBScript, php, ...).

Om ten volle van de meerwaarde van een formulier te genieten moet je dit doordacht opbouwen. Bedenk dus op voorhand goed waarvoor je het formulier nodig hebt en beslis wat er moet gebeuren met de verzamelde gegevens. Het aantal formulieren dat kan opgenomen worden in een HTML-pagina is onbeperkt, maar formulieren kunnen niet genest worden. Indien alle gegevens met één klik van de muis moeten verzonden worden is het wel nodig om alles binnen één formulier te houden.



The screenshot shows a web browser window with a single tab titled 'Formulier'. The address bar displays 'www.web-essentials.be/formulier-alles.html'. The page content features a heading 'Formulier:' followed by a form with the following elements:

- Naam:** A single-line text input field.
- Straat:** A single-line text input field.
- Postnummer:** A single-line text input field.
- Gemeente:** A single-line text input field.
- Paswoord:** A single-line text input field.
- Land:** A dropdown menu with 'België' selected.
- Geslacht:** Two radio buttons labeled 'Mannelijk' and 'Vrouwelijk'.
- Operating system:** Two checkboxes labeled 'Windows' and 'Linux'.
- Commentaar:** A multi-line text area.
- Buttons:** Two buttons at the bottom: 'Formulier verzenden' and 'Alles wissen'.

1.2 Elementen en attributen

Formulieren kunnen zonder problemen in een gewone HTML-pagina opgenomen worden. Ze zijn prima combineerbaar met tekst, afbeeldingen, tabellen, opmaakcodes, enzovoort.

Met formulierspecifieke elementen kan je knoppen, keuzerondjes, keuzelijsten, tekstvakken, paswoorden, enzovoort toevoegen. Aan de gebruiker wordt zo een snelle en efficiënte manier geboden om een website interactief te bezoeken en te verkennen.

❖ Het <form>-element

De container <form>...</form> wordt geplaatst op de plaats in het HTML-document waar je het formulier wil opnemen. De minimale structuur van een formulier:

```
<form>
    Inhoud van het formulier
</form>
```

Om de webserver te vertellen hoe een formulier moet worden verwerkt, gebruik je de attributen `method` en `action`. Het encryptietype geef je aan in `enctype`. Indien een leesbaar ASCII-tekst moet verstuurd worden staat dit op "text/plain".

HTML	waarde
<code>accept-charset</code>	Specificeert de karakterset voor de server.
<code>action="URL"</code>	Hiermee wordt aangegeven wat de server moet doen met de gegevens die je hebt gepost ('post') of die de server heeft opgehaald ('get'). 'URL' geeft het adres aan van het script dat deze gegevens moet verwerken. Door gebruik te maken van een script kan de bezoeker onmiddellijk over de resultaten beschikken.
<code>autocomplete</code>	Schakelt het automatisch aanvullen aan of uit.
<code>enctype</code>	Geeft aan hoe het formulier verzonden wordt naar de server. Met <code>enctype="text/plain"</code> wordt gekozen voor ASCII-tekst (enkel bij <code>method="post"</code>).
<code>method="post"</code> of <code>method="get"</code>	Met 'method' geef je aan hoe de gegevens gecodeerd zullen worden. Met 'post' worden de gegevens niet getoond in de URL, met 'get' worden ze wel getoond.
<code>name</code>	Naam voor het formulier.
<code>novalidate</code>	Geeft aan of veldvalidatie moet gebeuren alvorens te verzenden.
<code>target</code>	Het doel om het resultaat in te tonen.

```
<form action="http://www.domein.com/cgi-bin/..." method="post"
enctype="text/plain">
```

Gebruik een `mailto`-koppeling als alternatief voor browsers (zeer klein percentage) die geen formulieren ondersteunen. Hier moet `method="post"`, wil je ook data in de mail versturen (=body).

```
<form action="mailto:iemand@domein.com" method="post">
```

Binnen de mailto-koppeling kan ook een onderwerp, een carbon copy en een blind carbon copy opgegeven worden. Deze dienen gescheiden te worden van het mailto-adres door een vraagteken en onderling door een ampersand.

```
<form  
action="mailto:iemand@domein.com?subject=onderwerp&cc=iemands.anders@d  
omein2.com&bcc=nogiemand.anders@domein3.com" method=post">
```

Er moet een mail cliënt geïnstalleerd zijn, opdat dit werkt: de gebruiker moet immers de mail zelf versturen.

1.3 Invoerelementen

De verschillende invoervelden vormen de basisbestanddelen van een formulier. Hiervoor gebruik je het `<input />`-element, waarin het type van invoerveld wordt gedefinieerd met `type=""`. Al vanaf de vroegere HTML-versies kan je met het input-element tekstinvoervelden (text, password, hidden en file), keuze-elementen (checkbox en radio), knoppen (submit, reset, image en button) maken. In HTML5 zijn er vele types toegevoegd aan het input-element (email, url, number, tel, local, range, date, month, week, time, datetime, datetime-local, search en color). Wanneer de nieuwe input-elementen niet ondersteund worden door de browser, werken ze als een invoerveld van het type tekst. Daarnaast zijn er de elementen `datalist` en `output` bijgekomen.

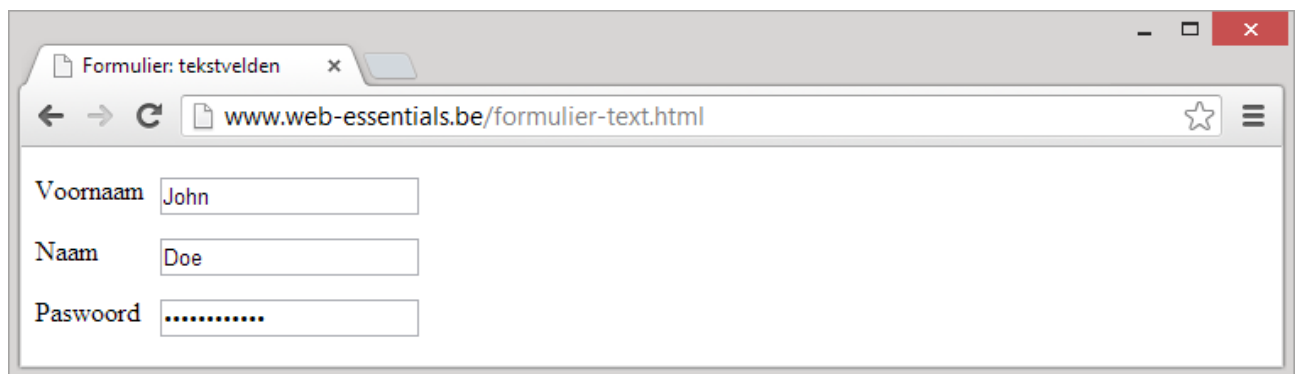
Sommige attributen zijn op bijna alle `<input>`-formulierelementen toepasbaar.

HTML	waarde
autocomplete	Automatisch aanvullen.
autofocus	Automatisch focus krijgen.
disabled	Voorkomt dat de gegevens doorgestuurd worden naar de server.
formnovalidate	Zorgt ervoor dat een element niet gevalideerd wordt.
list	Refereert naar een datalist.
maxlength	Geeft het maximaal aantal tekens aan.
name	Zonder dit attribuut stuurt de browser de invoervelden niet door naar de webserver.
pattern	Reguliere expressie.
placeholder	Toont een hint.
readonly	Voorkomt dat de waarde gewijzigd kan worden.
required	Maakt het invoerveld verplicht.
size	Duidt de beginbreedte van het element aan in aantal tekens.
type	Invoertype: button, checkbox, color, date, datetime, datetime-local, email, file, hidden, image, month, number, password, radio, range, reset, search, submit, tel, text, time, url, week.
value	Specificeert de beginwaarde.

❖ Tekstvelden: text

Tekstvelden zijn het meest geschikt voor het verzamelen van algemene informatie. Het is het eenvoudigste veld om als lezer tekstuele en numerieke informatie door te geven. Het invoerveld moet een naam krijgen via het name-attribuut. Zonder dit attribuut stuurt de browser de invoervelden niet door naar de webserver.

```
<input type="text" name="..." />
```



De labels kunnen gewoon opgegeven worden, maar kunnen ook door middel van het `label`-element aangegeven worden. Er is geen verschil in weergave, maar er kan wel een functioneel verschil zijn (vb. text-to-speech).

```
<label for="naam">Naam</label><input type="text" id="naam" name="naam" />
```

Met HTML5 kunnen er mogelijkheden voor een invoerveld gedefinieerd worden met een `datalist`-container. In deze container plaats je `option`-elementen. Als het element goed ondersteund wordt toont de browser de mogelijkheden. Je kan aan een `label` ook een `value` koppelen. Als de gebruiker dan de `label`-waarde ingeeft, dan komt de `value`-waarde in het invoerveld. De gebruiker blijft altijd de mogelijkheid hebben om toch een vrije waarde in te geven. Het werkt ook als een vorm van autocompletion. Als je enkel `value` voorziet en je begint te typen wordt de keuze mogelijkheid beperkt tot wat past bij wat je typt.

```
Gemeente: <input list="gemeentelijst" name="gemeente" />
<datalist id="gemeentelijst">
  <option label="3500" value="Hasselt">
  <option label="3600" value="Genk">
  <option label="3700" value="Sint-Truiden">
</datalist>
```

❖ Wachtwoordvelden

Wachtwoordvelden werken analoog aan het tekstveld. Het enige verschil bestaat erin dat er in het wachtwoordveld enkel bolletjes verschijnen bij het intikken. Het is enkel een bescherming bij het invoeren van gegevens in aanwezigheid van derden. Alles wat de bezoeker in het wachtwoordveld intikt, wordt echter ongecodeerd over het internet verzonden en kan onderschept worden.

```
<input type="password" name="..." />
```

❖ Hidden velden

Met hidden velden is het ook mogelijk om velden aan te maken die de lezer niet ziet. Deze kunnen dus niet ingevuld worden, maar zijn soms handig om binnen een script te gebruiken.

```
<input type="hidden" name="verborgen veld" />
```

❖ E-mailvelden

Het e-mailveld is bedoeld om het e-mailadres op te vragen. Het is de bedoeling om dan automatisch te controleren of de vormgeving geldig is, met name de aanwezigheid van punten en het @-teken. Dit wordt nog niet goed ondersteund, er wordt wel op het @-teken gecontroleerd, maar nog niet op punten.

```
<input type="email" name="e-mailadres" />
```

❖ Nummervelden

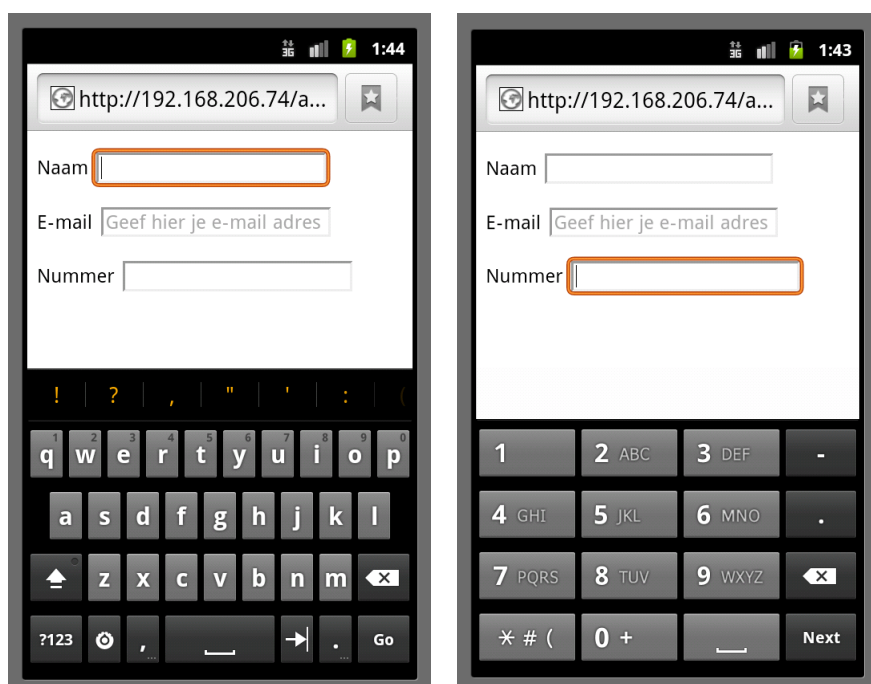
De nummervelden laten toe om een getal in te geven. Met het min- en max-attribuut kan je de range bepalen.

```
<input type="number" name="nummer" min="0" max="10" />
```

Bij mobiele apparaten zonder toetsenbord wordt vaak automatisch een numeriek toetsenbord gekozen als de focus op dit element komt. Voor andere invoerelementen wordt dan een andere toetsenbordlay-out gekozen. Bij iPhone en iPad werkt dit al voor nummers, e-mailadressen en url-adressen.



De mobiele browsers voor Android-toestellen maken dit onderscheid ook. Ze kiezen een geschikt toetsenbord, afhankelijk van het invoerveld dat de focus heeft.



❖ Tekstinvoer: url

Het `url`-type is een tekstveld dat de vormgeving van een webadres kan valideren en een foutmelding genereert als een incorrecte url wordt ingegeven. Dit is ook enkel HTML5.

```
<input type="url" name="..." />
```

❖ Datumveld: date, month, week, time, datetime, datetime-local

Het invoegen van een vast gedefinieerde datum wordt nog onvoldoende ondersteund, maar zal in de toekomst zeker een meerwaarde zijn als de datum nodig is om berekeningen op uit te voeren om in een databank op te nemen. Er bestaan meerdere datumvelden: `date`, `month`, `week`, `time`, `datetime`, `datetime-local`.

```
<input type="date" name="..." />
```

In Chrome werkt dit reeds: er wordt getoond hoe de datum moet ingegeven worden en als je in het veld klikt, krijg je een kalender.

❖ Zoekveld: search

Het `search`-type werkt hetzelfde als het gewone tekstveld, maar kan een iets andere opmaak meekrijgen (toevoeging van wis-kruis). De bedoeling is om een aparte betekenis te geven aan zoekvelden.

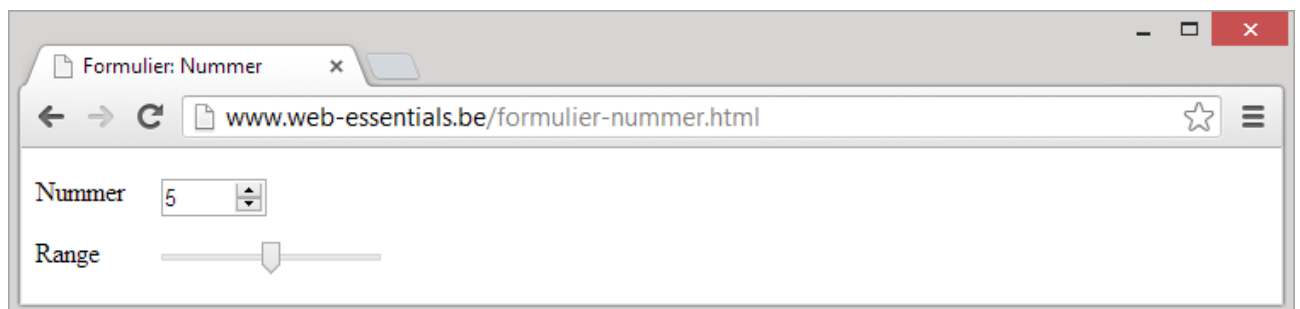
```
<input type="search" name="..." />
```

❖ Range velden

Het range veld kan ook met een minimum en maximum uitgerust worden, eventueel aangevuld met stapsgewijze sprongen.

```
<input type="range" name="range" min="0" max="100" step="10" />
```

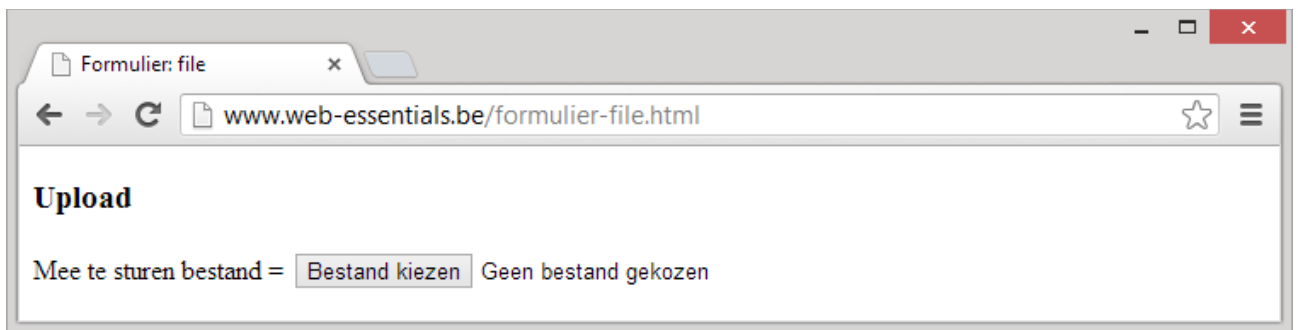
HTML	waarde
<code>max</code>	Maximum waarde
<code>min</code>	Minimum waarde
<code>step</code>	Stapsgewijze wijziging



❖ File velden

Wanneer bestanden dienen verzonden te worden naar de webserver kan je gebruik maken van het file-type. Natuurlijk moet de webapplicatie op de server verwachten dat er bestanden (mee)gestuurd worden. De weergave van een file-veld verschilt nogal van browser tot browser, "Bestand kiezen" of "Bladeren"-knop wordt automatisch toegevoegd.

```
<input type="file" name="bijvoegsel" />
```



❖ Selectievakken: checkbox

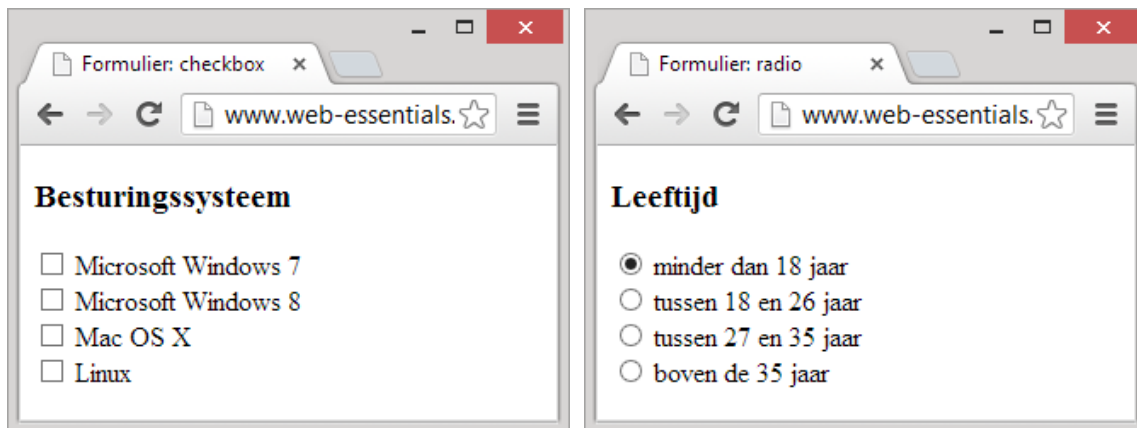
Wanneer de bezoeker een keuze dient te maken uit meerdere mogelijkheden maak je gebruik van selectievakjes. Deze kunnen geactiveerd (voorzien van een vinkje) of gedeactiveerd zijn. Het attribuut 'checked' wordt gebruikt om een vakje standaard aan te vinken.

```
<input type="checkbox" name="..." value="..." checked="checked" />
```

❖ Keuzerondjes: radiobuttons

Analoog aan de selectievakjes met dit verschil dat de bezoeker maar één van de mogelijkheden kan selecteren. Dit bereik je door alle keuzerondjes in de lijst dezelfde name toe te kennen. Om toch te weten te komen welk keuzerondje uiteindelijk is geselecteerd, geef je ze allemaal een andere waarde mee (value="..."). Deze waarde wordt doorgezonden aan de webserver.

```
<input type="radio" id="leeftijd" name="..." value="..." checked="checked" />  
<label for="leeftijd">minder dan 18 jaar</label>
```



❖ Kleurenveld: color

Het color-type is nieuw in HTML5 en nog slecht ondersteund. Het laat toe om een kleur te kiezen (soms zelfs met een colorpicker) of een hexadecimale kleurwaarde in te geven.

```
<input type="color" name="..." />
```

❖ Uitvoerveld: output

Nieuw in HTML5 is de logische toevoeging van uitvoervelden bovenop de invoervelden. Het output-element is nutteloos zonder een event handler met achterliggend JavaScript (zie Scripting). De meeste browsers ondersteunen dit nieuw element.

```
<form>
  <input type="number" name="getal1" value="0" /> x <input
type="number" name="getal2" value="0" /> = <output id="product"
for="getal1 getal2" ></output>
</form>
```

1.4 Keuzelijsten

❖ Dropdownmenu: select

Qua structuur zijn keuzelijsten te vergelijken met genummerde of ongenummerde lijsten. Ze stellen de gebruiker in staat om één of meer items te selecteren uit een menu. Qua functie lijken keuzelijsten op keuzerondjes en selectievakjes. De schermweergave verschilt echter.

Een keuzelijst wordt gedefinieerd binnen de container-tag `<select>...</select>`. De afzonderlijke opties binnen de keuzelijst worden aangeduid door de tag `<option>...</option>`.

De `<select>`-tag kan de attributen `autofocus`, `disabled`, `form`, `name`, `size` en `multiple="multiple"` bevatten.

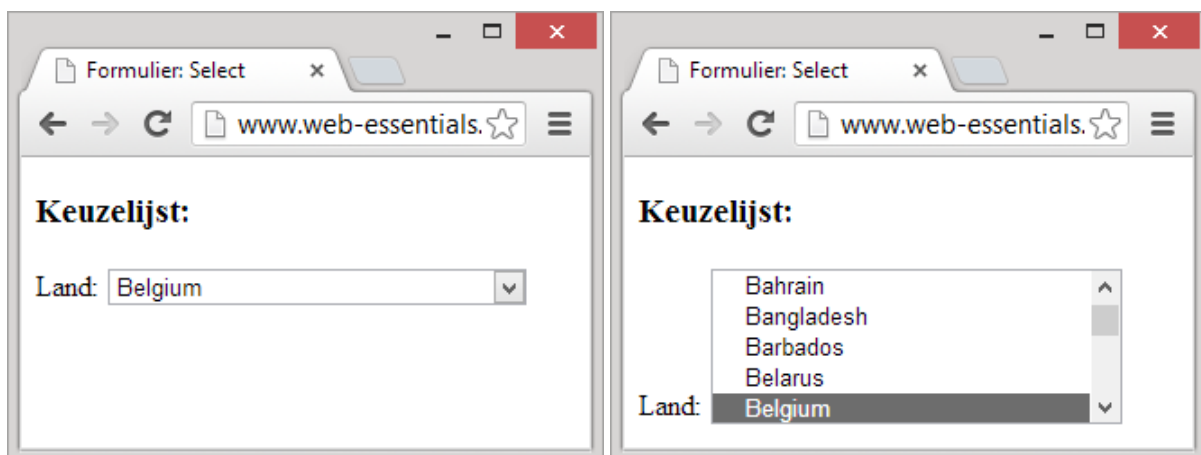
HTML	waarde
<code>autofocus</code>	Zet de focus op de keuzelijst bij het laden van de pagina.
<code>disabled</code>	Zet de keuzelijst op disabled.
<code>form</code>	Definieert de form waar de keuzelijst toe behoort.
<code>multiple="multiple"</code>	Zorgt ervoor dat de gebruiker meerdere items tegelijkertijd kan selecteren door middel van de Ctrl- of Shift-toets in Windows of de Commandtoets bij Mac.

name	Zonder dit attribuut stuurt de browser de invoervelden niet door naar de webserver.
size	Staat voor het aantal items dat de browser effectief toont.

De <option>-tag kan de attributen selected="selected" en value="..." bevatten.

HTML	waarde
selected="selected"	Maakt het mogelijk om een standaardwaarde te selecteren.
value	Specificeert de doorgestuurde waarde.

```
<select name="..." size="..." multiple="multiple">
  <option>...</option>
  <option selected="selected">...</option>
  <option>...</option>
</select>
```



1.5 Tekstvakken

❖ Tekstveld met meerdere regels: textarea

Voor het versturen van langere commentaren maak je best gebruik van de containertag <textarea>...</textarea>. Met de attributen rows="..." en cols="..." wordt de breedte en hoogte bepaald, uitgedrukt in het aantal karakters. Als je al tekst in het tekstveld wil, kan je die in de container plaatsen.

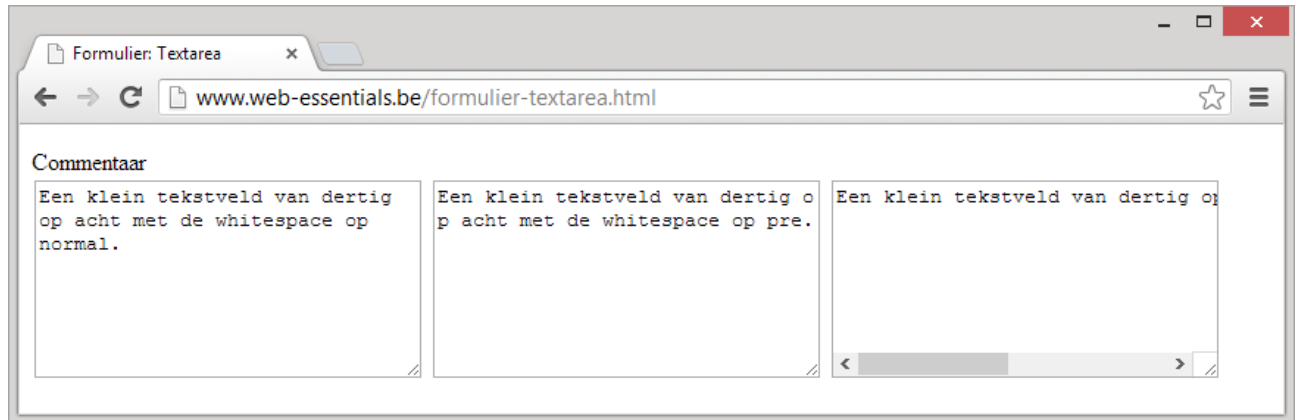
Het al dan niet toevoegen van lijneinden (wrapping) bij het verzenden kan gebeuren met het wrap-attribuut dat op soft of hard gezet kan worden. Bij soft wrapping wordt er geen nieuwe regel meegezonden als die er is in het tekstveld, bij hard wrapping wordt wel een combinatie van linefeed en carriage return meegezonden.

Met de CSS-stijl *white-space* kan het uitzicht binnen het tekstveld bepaald worden. De werking verschilt echter per browser.

- *white-space: normal* is de standaardinstelling, de tekst breekt vanzelf af en begeeft zich op de volgende regel als het eind van het tekstveld bereikt wordt.
- *white-space: pre* schrijft de tekst tot de grens van het venster en gaat dan naar een volgende regel, echter zonder af te breken.

- *white-space:nowrap* breekt helemaal niet af en geeft een schuifbalk als de gebruiker blijft doortypen zonder enters in te geven.

```
<label for="commentaar">Commentaar</label>
<textarea cols="30" rows="8" id="commentaar" wrap="hard"> ...
</textarea>
```



1.6 Knoppen

❖ Zend- en wisknop: submit, reset

De waarden die de bezoeker heeft ingevuld, moeten naar de server verstuurd worden. Hiervoor maak je gebruik van een submit-knop. Indien je een eigen tekst op de knop wil plaatsen kan dit met het attribuut `value="..."`. Als je geen value meegeeft, komt de standaardtekst op de knop te staan en deze verschilt van browser tot browser. Het indrukken van de submitknop triggert de actie `action="..."` en verstuurt de informatie naar de server. Elk formulier kan maar één submitknop bevatten.

```
<input type="submit" value="Verzenden" />
```

Om al de invoervelden in een formulier te wissen gebruik je het reset-type:

```
<input type="reset" value="Wissen" />
```

❖ Knop: button

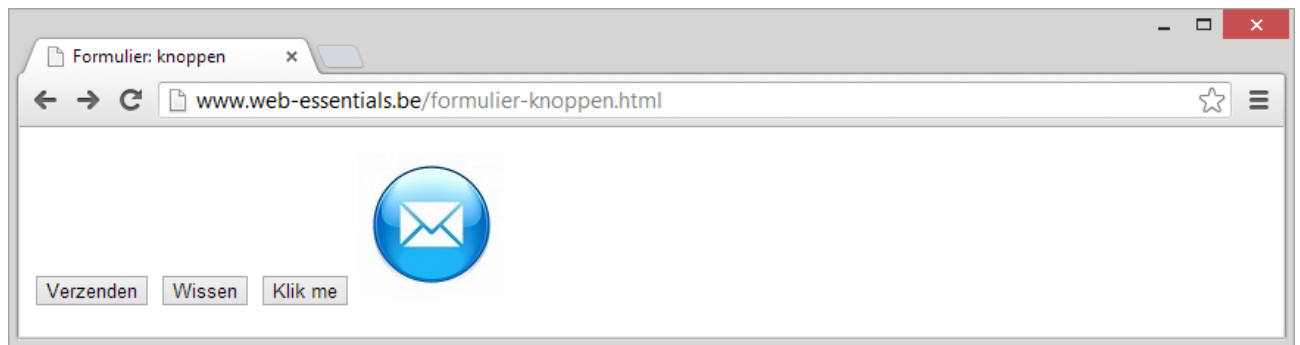
Voor een knop zonder voorgedefinieerde werking en zonder voorgedefinieerd label gebruik je het button-type. Hieraan moet je een actie koppelen met een event-handler in JavaScript.

```
<input type="button" value="Klik me" />
```

❖ Figuurknop: image

Als je een figuur als submit-knop wil gebruiken, kan dat met het image-type.

```
<input type="image" src="e-mail.jpg" alt="e-mail me" />
```



Voor de submit- en image-knop bestaan er attributen om af te wijken van de in de form gedefinieerde action, enctype, method en target.

HTML	waarde
formation	overschrijft de standaardactie van de submit of imageknop.
formenctype	overschrijft de standaardencryptie van de submit of imageknop.
formmethod	overschrijft de standaardmethode van de submit of imageknop.
formtarget	overschrijft de standaardtarget van de submit of imageknop.

```
<form action="script.asp" method="get">
  Voornaam: <input type="text" name="voornaam" /><br />
  Achternaam: <input type="text" name="achternaam" /><br />
  <input type="submit" value="Verzenden" />
  <input type="submit" formmethod="post" formaction="postsript.asp"
  value="Verzenden met POST" />
</form>
```

1.7 Structuur in het formulier brengen

❖ Fieldset en legend

Indien bepaalde invoerelementen duidelijk samenhangen kan je best wat structuur aan je formulier toevoegen. Hiervoor is het fieldset-element ontwikkeld. Elke fieldset kan je een titel geven met het legend-element.

```
<fieldset>
  <legend>Persoonlijke gegevens</legend>
  <label for="naam">Naam</label>
  <input type="text" id="naam" name="naam" />
  <label for="voornaam">voornaam</label>
  <input type="text" id="voornaam" name="voornaam" />
</fieldset>
```

Formulier

www.web-essentials.be/formulier-fieldset.html

Persoonlijke gegevens

Naam Voornaam

Straat

Postnummer Gemeente

Commentaar

Verstuur formulier

Formulier verzenden Alles wissen

In combinatie met CSS geeft dit een goed gestructureerd formulier.

Formulier

www.web-essentials.be/formulier-fieldset-css.html

Persoonlijke gegevens

Naam Voornaam

Straat

Postnummer Gemeente

Commentaar

Verstuur formulier

Formulier verzenden Alles wissen

1.8 Toegankelijk formulier

Ook mensen met een beperking surfen op het internet. Zij gebruiken hiervoor hulpmiddelen, zoals een screenreader (= schermuitleesprogramma) voor blinden en slechtzienden. Wat er nog allemaal kan, kan je vinden op de website <http://anysurfer.be/nl>.

Op bovenstaande site vind je ook een checklist om die toegankelijkheid te testen. Maar het is beter om de aanbevelingen die hierin staan al onmiddellijk bij het maken van de website toe te passen.

Als eerste moet je zorgen dat je met de tabtoets logisch door het formulier kan navigeren. Indien je hier niets wijzigt is dat default van links naar rechts en van boven naar beneden. Maar let op met horizontaal floaten van elementen, in dat geval kan het nodig zijn dat je zelf de tabvolgorde manipuleert. Voordat je dit gaat gebruiken, best even deze pagina lezen: <http://anysurfer.be/nl/in-de-praktijk/websites/tabvolgorde>.

Als we nu gaan kijken naar de checklist om een formulier toegankelijk te maken, is dat niet zo moeilijk om rekening te houden met toegankelijkheid.

Probeer ook er voor te zorgen dat een reeks opeenvolgende velden op dezelfde manier kunnen ingevuld worden: met muis of met toetsenbord. Dus niet veld 1 met muis, veld 2 met toetsenbord, veld 3 weer met muis Probeer ze te combineren, zodat mensen met een motorische beperking niet telkens moeten wisselen.

❖ **Labels en invulvelden (input) zijn met elkaar verbonden:**

- Voeg aan ieder input veld een id-attribuut toe.
- Plaats een label in een label-element.
- Voeg een for-attribuut toe in het label, met de waarde van het id-attribuut van het bijbehorende inputveld.

❖ **Input zonder label**

In dit geval kan je geen koppeling maken. Gebruik hier dan een tooltip: deze voeg je toe met het title-attribuut.

Een tooltip wordt getoond als je met de muis beweegt over een invulveld en wordt ook voorgelezen en in braille weergegeven.

❖ **Label bij datumvelden**

Zorg ervoor dat je in het label meegeeft hoe de datum moet ingevuld worden. Bijvoorbeeld: geboortedatum (dd/mm/jjjj). Indien je een datumpicker zou toevoegen, zorg er dan voor dat dit niet de enige manier is om de datum in te geven, maar dat je hem ook nog gewoon kan typen.

Indien je type=date gebruikt, zul je merken dat je of zelf kan typen, of gebruik kan maken van een datumpicker. (=Chrome)

Omdat het type date nog niet door elke browser ondersteund wordt, kan je ook zelf het datumveld opbouwen, waarbij je voor dag, maand en jaar telkens een aparte selectielijst maakt.

❖ **Samenhorende formulervelden zijn ingesloten door een fieldset.**

Om duidelijk aan te geven dat bepaalde invoervelden (zoals checkboxes, radiobuttons ...) samen horen, is een label niet altijd voldoende. Beter gebruik je hier ook telkens een fieldset, in het legend-element zet je dan de omschrijving.

❖ **Verplichte velden in label duidelijk maken**

Maak duidelijk welke velden verplicht ingevuld moeten worden door in het label (verplicht) of (*) erbij te zetten.

❖ Een duidelijke verzendknop

Maak geen keuzelijsten met het onchange-event; kan problemen geven voor mensen, die met behulp van de pijltjestoetsen door de keuzelijst navigeren. Als je dan bij aanklikken van een keuze (pijltje) al naar een andere html pagina zou gaan

Formulieren kunnen soms ook verstuurd worden met de entertoets. Voorzie in dat geval toch altijd een verzendknop, voor mensen die het toetsenbord niet kunnen gebruiken.

Als je een afbeelding als knop gebruikt, vergeet dan zeker het alt-attribuut niet, waarin je duidelijk de actie van de knop omschrijft.

❖ Bij validatie je foutboodschap aanpassen

Het is aanbevolen om altijd een boodschap te geven, zowel bij fouten als bij een goed versturen van het formulier.

Dit moet met duidelijke boodschappen gebeuren en niet enkel met kleuraanpassingen.

Omdat je hiervoor met een script moet werken is dit hier niet uitgebreid beschreven.

Als je met javascript kan werken, kan je bij anysurfer op zoek gaan hoe je dat best aanpakt.

❖ Voorbeelden toegankelijkheid:

Van alle hier aangehaalde punten zijn voorbeelden terug te vinden op de pagina's van de checklist bij <http://anysurfer.be/nl>

Ga ook voor de andere onderdelen van je website eens kijken in die checklist.