

Basis HTML elementen

Web Frontend Basics

1	BASIS HTML ELEMENTEN	3
1.1	b element	3
1.2	strong element	3
1.2.1	b of strong?	4
1.3	i element	5
1.4	em element	5
1.5	u element	5
1.6	mark element	7
1.7	cite element	7
1.8	small element	8
1.9	code element	8
1.10	samp element	8
1.11	kbd element	8
1.12	pre element	9
1.13	sup en sub element	9
1.14	blockquote element	10
1.15	q element	10

3 BASIS HTML ELEMENTEN

In dit deel van de cursus nemen we een aantal basis HTML-elementen onder de loep en bespreken we hun doel en werking.

3.1 B ELEMENT

Type: inline element

Het **b** element (*Bring Attention To element*) wordt gebruikt om de aandacht van de lezer te trekken, zonder extra betekenis toe te voegen.

De *user-agent* zal in de meeste gevallen de opmaak van de inhoud van het **b** element in het vet weergeven. Gebruik dit element echter niet om snel een stukje tekst van stijl te voorzien. Hiervoor ga je aan de slag met **CSS** en de *font-weight* property.

Het **b** element wordt hoofdzakelijk gebruikt om sleutelwoorden in een tekst of productnamen in een recensie te omsluiten. Het wordt ook vaak toegepast om de eerste zin van een artikel in te plaatsen om er de nodige aandacht van de lezer mee te trekken.

```
<p>In dit hoofdstuk zullen wat meer leren over <b>HTML</b> elementen.</p>
```

Voorbeeld gebruik van het **b** element



Enkele richtlijnen en tips voor het gebruik:

- Verwar het **b** element niet met de elementen **strong**, **em** of **mark**.
 - Het **strong** element geeft tekst weer met een bepaald belang
 - Het **em** element legt enige nadruk op de tekst
 - Het **mark** element geeft tekst aan met een zekere relevantie.

Het **b** element geeft geen betekenis mee, gebruik het daarom alleen als de andere elementen niet passen. Zie verder voor de verduidelijking van bovenvermelde elementen.
- Gebruik het **b** element niet om koppen aan te geven. Hiervoor zijn de **h1** tot en met **h6** elementen ter beschikking.

Bron: [Mozilla Developer Network - b element](#)

3.2 STRONG ELEMENT

Type: inline element

Het **strong** element is bedoeld voor inhoud die van groot belang is, zaken van grote ernst of urgentie (*zoals waarschuwingen*). Dit kan zelfs een zin zijn die van groot belang is voor de hele pagina, of je kan erop wijzen dat sommige woorden van groter belang zijn dan inhoud in de buurt.

```
<p>
  Voor je verder gaat, <strong>zorg ervoor dat je je veiligheidsbril aanhebt</strong>.
</p>
```

Voorbeeld gebruik van het **strong** element

De *user-agent* zal in de meeste gevallen de opmaak van de inhoud van het **strong** element in het vet weergeven. Gebruik dit element echter niet om snel een stukje tekst van stijl te voorzien. Hiervoor ga je aan de slag met **CSS** en de *font-weight* property.

Een ander geaccepteerd gebruik van het **strong** element is om de labels van alinea's aan te duiden die opmerkingen of waarschuwingen in de tekst van een pagina vertegenwoordigen.

```
<p><strong>Belangrijk:</strong> Wees voorzichtig tijdens het rijden in de mist.</p>
```

Voorbeeld gebruik van het strong element

3.2.1 B OF STRONG?

Het kan vaak verwarrend zijn als je van start gaat met HTML waarom er zoveel verschillende manieren zijn om iets op het scherm te plaatsen wat er voor de gebruiker uiteindelijk precies hetzelfde uitziet. Het **b** en **strong** element zijn hier beiden misschien een van de meest voorkomende bronnen van verwarring. Waardoor al vaak de vraag komt: "*Moet ik nu **b** of **strong** gebruiken? Doen ze niet allebei hetzelfde?*"

Wel, niet precies. Het **strong** element is voor inhoud die van groter belang is, terwijl het **b** element wordt gebruikt om de aandacht op tekst te vestigen zonder aan te geven dat het belangrijker is.

Het kan helpen om te beseffen dat beide geldige en semantische elementen zijn en dat het toeval is dat ze in de meeste browsers dezelfde standaardstijl (*vetgedrukt*) hebben. Elk element heeft echter zijn plaats om gebruikt te worden gebruikt in bepaalde scenario's. En het is net het doel, de betekenis wat bepalen zal welk element je gebruikt en niet het visuele resultaat in de browser.

3.3 I ELEMENT

Type: inline element

Het **i** element wordt gebruikt om inhoud te omsluiten die om de een of andere reden afwijkt van de normale tekst. Voorbeelden hiervan zijn technische termen, uitdrukkingen in een vreemde taal of gedachten van fictieve personages. We zien dan ook vaak het **lang** attribuut op dit element opduiken om verdere verduidelijking te geven. Zorg ervoor dat de tekst in kwestie niet beter geschikt is voor een ander element.

De *user-agent* zal in de meeste gevallen de opmaak van de inhoud van het **i** element cursief weergeven. Gebruik dit element echter niet om snel een stukje tekst van stijl te voorzien. Hiervoor ga je aan de slag met **CSS** en de *font-style* property.

```
<p>In het Frans noemen ze dit <i lang="fr">à l'improviste</i>.</p>
<p><i lang="la">Taraxacum officinale</i> is
door ons beter gekend als paardenbloem.</p>
```

Voorbeeld gebruik van het **i** element

3.4 EM ELEMENT

Type: inline element

Het **em** element (*Emphasis*) legt op de inhoud een zekere nadruk. **em** elementen kunnen genest worden in elkaar waarbij elk niveau van nesten een grotere mate van nadruk aangeeft. Dit element is bedoeld om een nadruk op woorden te leggen ten opzichte van de omringende tekst. Vaak beperkt de inhoud van dit element zich tot een woord of woorden van een zin en heeft invloed op de betekenis van de zin zelf.

De *user-agent* zal in de meeste gevallen de opmaak van de inhoud van het **em** element cursief weergeven. Gebruik dit element echter niet om snel een stukje tekst van stijl te voorzien. Hiervoor ga je aan de slag met **CSS** en de *font-style* property.

```
<div>
  <h2>EM element</h2>
  <p>Laten dit <em>nu</em> doen!</p>
  <p>We moeten hier <em>dringend</em> iets aan doen.</p>
</div>
```

Voorbeeld gebruik van het **em** element

3.5 U ELEMENT

Type: inline element

Het **u** element (*Unarticulated Annotation element*) gebruik je voor om tekst te markeren als een vorm van niet-tekstuele annotatie. Wat hiermee concreet bedoeld wordt, is dat dit een verklarende visuele 'opmerking' toevoegt ter verduidelijking van de tekst die omsloten werd. Geldige scenario's waarin je dit element kan gebruiken zijn onder meer het annoteren van spelfouten, het toepassen van een **proper name mark** om

eigen namen in Chinese tekst aan te duiden (zal je hoogstwaarschijnlijk nooit tegenkomen, maar even meegeven om volledig te zijn) en andere vormen van annotatie.

De *user-agent* zal in de meeste gevallen de opmaak van de inhoud van het `u` element onderlijnen. Gebruik dit element echter niet om snel een stukje tekst van stijl te voorzien. Hiervoor ga je aan de slag met **CSS** en passende properties.

In vele gevallen zal je ook nog de nodige aanpassingen maken via CSS om de gebruiker een visueel herkenbare onderlijning weer te geven. Op deze manier onderscheid je ook meteen de interpretatie van een gebruiker dat iets wat onderlijnd is veelal een hyperlink voorstelt.

In onderstaand voorbeeld wordt hiervoor gebruik gemaakt van een class attribuut om de nodige bijkomende opmaak te voorzien op het element. Het class attribuut en hoe je hiermee omgaat in CSS wordt in het volgende hoofdstuk besproken.

```
<p>Deze alinea bevat een woord dat <u class="spelling">verkreed</u> gespeld werd.</p>
```

Voorbeeld gebruik van het `u` element samen met een class attribuut

```
.spelling {
  text-decoration: red wavy underline;
}
```

Voorbeeld van de CSS selector op class attribuut

Deze alinea bevat een woord dat verkreed gespeld werd.

Voorbeeld output in een browser

Indien je enkel maar op zoek bent naar een manier om te onderlijnen, maar het `u` element niet relevant is om te gebruiken dan kan je op soortgelijke manier aan de slag gaan. Maak dan gebruik van een niet semantisch element omheen de inhoud van de tekst die je wil onderlijnen. We bekijken even een voorbeeld.

```
<span class="underline">Dit willen we onderlijnen:</span>
<br />
Vervolgens komt dit op een nieuwe lijn.
```

Voorbeeld gebruik van het `span` element samen met een class attribuut om stijl te voorzien

```
.underline {
  text-decoration: underline;
}
```

Voorbeeld van de CSS selector op class attribuut

Dit willen we onderlijnen:
Vervolgens komt dit op een nieuwe lijn.

Voorbeeld output in een browser

3.6 MARK ELEMENT

Type: inline element

Het **mark** element vertegenwoordigt tekst die is gemarkeerd voor referentie- of notatiedoeleinden. Dit kan zijn vanwege de relevantie, of het belang van de gemarkeerde passage in de omsluitende context.

Als output zal het **mark** element de inhoud in het fluo aanduiden. Dit element wordt voornamelijk gebruikt in een dynamische context waar men bijvoorbeeld - aan de hand van javascript - woorden uit een zoekopdracht zal markeren in een stuk tekst.

```
<p>In deze tekst kunnen we gaan markeren dat het <mark>correct gebruik van HTML Elementen </mark> zeker en vast van belang is!</p>
```

Voorbeeld gebruik van het mark element

In deze tekst kunnen we gaan markeren dat het correct gebruik van HTML Elementen zeker en vast van belang is!

Voorbeeld output in een browser

3.7 CITE ELEMENT

Type: inline element

Het **cite** element (*Citation*) element wordt gebruikt om een verwijzing naar een geciteerd creatief werk te beschrijven en moet de titel van dat werk bevatten.

In de context van het **cite** element zou een creatief werk bijvoorbeeld een van de volgende kunnen zijn:

- Een boek
- Een onderzoekspaper
- Een verhandeling
- Een gedicht
- Een muzikale score
- Een lied
- Een toneelstuk of filmscript
- Een film
- Een televisieprogramma
- Een spel
- Een sculptuur
- Een schilderij
- Een theatrale productie
- Een toneelstuk
- Een opera
- Een musical
- Een tentoonstelling
- Een juridisch dossier
- Een computerprogramma
- Een website
- Een webpagina
- Een blogbericht of reactie
- Een forumbericht of reactie
- Een tweet
- Een Facebook-bericht
- Een schriftelijke of mondelinge verklaring
- ...

De *user-agent* zal in de meeste gevallen de opmaak van de inhoud van het **cite** element cursief weergeven. Wens je dit liever niet, dan kan je uiteraard altijd aan de slag met **CSS** en de *font-style* property om dit op te heffen of je eigen opmaak te voorzien.

```
<p>Een aan te raden boek voor alle toekomstige developers is <cite>Clean Code</cite> geschreven door Uncle Bob.</p>
<p>Meer informatie kan je terugvinden in <cite>[ISO 20000-1]</cite>.</p>
```

Voorbeeld gebruik van het cite element

3.8 SMALL ELEMENT

Type: inline element

Het **small** element vertegenwoordigt rand commentaar en ‘de kleine lettertjes’, zoals copyright, auteursrecht en juridische tekst(en). Standaard maakt het tekst binnenin één lettergrootte kleiner (bv. van big naar medium of van small naar x-small).

```
<small>Een of meerdere regels copyright en/of disclaimers</small>
```

Voorbeeld gebruik van het small element

3.9 CODE ELEMENT

Type: inline element

Het **code** element heeft tot doel om een kort fragment van computercode te omsluiten. De inhoud wordt meestal weergegeven met het standaard monospaced lettertype van de *user agent* (zoals Courier of Lucida Console).

```
<p>Roep vervolgens de functie <code>updateTextElements()</code> aan via javascript.</p>
```

Voorbeeld gebruik van het code element



Aandachtspunt

Als je meerdere coderegels wenst weer te geven is het de afspraak om het **code** element in een **pre** element (zie verder) te plaatsen. Het **code** element vertegenwoordigt op zichzelf slechts een fragment of een enkele coderegel.

3.10 SAMP ELEMENT

Type: inline element

Het **samp** element (*Sample*) wordt gebruikt om tekst in te sluiten die een voorbeeld-output (of *citaat*) van een computerprogramma weergeeft. De inhoud wordt meestal weergegeven met het standaard monospaced lettertype van de *user agent* (zoals Courier of Lucida Console).

```
<p>Na uitvoeren van de applicatie krijg je in de console <samp>Update Successful</samp> te zien.</p>
<p><samp>Keyboard not found<br /> F1 to continue</samp></p>
```

Voorbeeld gebruik van het samp element

3.11 KBD ELEMENT

Type: inline element

Het **kbd** element (*Keyboard Input element*) geeft een gebruikersinvoer aan die gerelateerd kan zijn aan een toetsenbord, spraakinvoer of een ander tekstinvoerapparaat. De inhoud wordt meestal weergegeven met het standaard monospaced lettertype van de *user agent* (zoals Courier of Lucida Console).

```
<p>Om iets te kopiëren, dien je <kbd>CTRL</kbd> + <kbd>C</kbd> in te drukken.</p>
```

Voorbeeld gebruik van het kbd element

3.12 PRE ELEMENT

Type: block element

Het **pre** element (*Preformatted*) staat voor voorgeformatteerde tekst. Met andere woorden tekst die exact moet worden gepresenteerd zoals geschreven in het HTML-bestand. De inhoud wordt meestal weergegeven met het standaard monospaced lettertype van de *user agent* (zoals Courier of Lucida Console). Witruimte binnen dit element wordt, in tegenstelling tot wat er normaal gebeurt bij het renderen, wel weergegeven.

```
<pre>
body {
  color:red;
}
</pre>
```

Voorbeeld gebruik van het pre element

```
body {
  color:red;
}
```

Voorbeeld output in een browser



Aandachtspunt

Misbruik dit element niet om alles een plaats (bv.: aliniëren, spatiëren, e.d.) te geven op je pagina. Gebruik dit enkel voor kleine fragmenten waar dit nodig/passend is. Bijvoorbeeld in combinatie met het **code** element.

3.13 SUP EN SUB ELEMENT

Type: inline elementen

Het **sup** element (*Superscript*) en het **sub** element (*Subscript*) geeft inline-tekst aan die uitsluitend om typografische redenen een andere weergave vereist. Superscript wordt meestal weergegeven met een verhoogde basislijn met kleinere tekst en subscript wordt meestal weergegeven met een verlaagde basislijn met kleinere tekst. Veelal gebruikt voor wiskundige notaties.

```
<p>x<sup>2</sup> = y en x<sub>1</sub> ... x<sub>n</sub></p>
```

Voorbeeld gebruik van het sup en sub element

$$x^2 = y \text{ en } x_1 \dots x_n$$

Voorbeeld output in een browser

3.14 BLOCKQUOTE ELEMENT

Type: block element

Het **blockquote** element (*Block Quotation element*) geeft aan dat de omsloten tekst een uitgebreid citaat is. Meestal wordt dit visueel ook weergegeven door het inspringen van de betreffende tekst. Een URL voor de bron van het citaat kan worden gegeven met behulp van het **cite attribuut** (*niet verwarren met het cite element*), terwijl een tekstweergave van de bron dan weer kan worden meegegeven met behulp van het **cite element**.

```
<blockquote cite="https://cleancoders.com/">
  <p>
```

```
You should name a variable using the same care with which you name a first-
  born child.
```

```
  </p>
  <cite>Clean Coders</cite>
</blockquote>
```

Voorbeeld gebruik van het blockquote element, samen met het cite attribuut en het cite element

3.15 Q ELEMENT

Type: inline element

Het **q** element (*Quotation*) geeft aan dat de ingesloten tekst een kort citaat is. De meeste moderne browsers implementeren dit door de tekst tussen aanhalingstekens te plaatsen. Dit element is dus bedoeld voor korte citaten waarvoor geen alinea-eindes nodig zijn. Gebruik voor lange(re) citaten steeds het **blockquote** element wat we hier net voor zagen.

```
<p>
```

```
<q cite="http://www.caesar.com" lang="la">Veni, vidi, vici</q> is een gekende
uitspraak
  van Caesar.
</p>
```

Voorbeeld gebruik van het q element

"Veni, vidi, vici" is een gekende uitspraak van Caesar

Voorbeeld output in een browser

abbr element

Type: inline element

Het **abbr** element (*Abbreviation element*) laat ons toe om een afkorting aan te duiden en geeft ons bijkomend de mogelijkheid om de volledige omschrijving ter verduidelijking toe te voegen. Indien het **title** attribuut wordt benut in het element, hoort deze de volledige omschrijving te bevatten en niets anders.

```
<p>Maak gebruik van <abbr title="Cascading Style Sheets">CSS</abbr> om je <abbr title="HyperText Markup Language">HTML</abbr> stijl te geven.</p>
```

Voorbeeld gebruik van het **abbr** element



Oefening

Aansluitend kan je met **oefening 01 c** uit **oefenreeks 01** uitwerken om zelf even aan de slag te gaan met alle elementen die we tot hiertoe besproken hebben.



[Oefening 01 c via Github](#)

