

HTML

INFO

Lesinhoud

In deze les maken we kennis met de ontwikkeling van webpagina's en structuur van het HTTP-protocol. We zetten een ontwikkelomgeving op aan de hand van Webstorm, waarna we aan de slag gaan om onze eerste webpagina aan te maken. We maken kennis met de structuur en opbouw van een eenvoudige HTML5-pagina.

Doelstellingen

De student(e) kan:

- een webpagina opbouwen in HyperText Markup Language (HTML);
- de structuur van een webpagina opbouwen met (structuur)elementen en attributen;
- vanuit een ontwerp, HTML en CSS combineren tot een werkend functioneel geheel;
- de developer tools van browsers gebruiken bij het ontwikkelen van een website;

▼ Inhoud

- [Inleiding](#)
 - [Webtalen](#)
 - [HTTP-Protocol](#)
 - [URL's](#)
 - [Integrated Development Environment \(IDE\)](#)

- [\(Google Chrome\) DevTools](#)
- [Starten met HTML](#)
 - [Onze eerste HTML pagina](#)
 - [Elementen en tags](#)
 - [Nesten](#)
 - [Attributen](#)
 - [Commentaar](#)
 - [Speciale karakters - entiteitsnamen](#)
 - [Conventies en validatie](#)
- [Structuur](#)
 - [Blokelementen en inline-elementen](#)
 - [Structuurelementen](#)
- [Hyperlinks](#)
 - [Absolute koppeling](#)
 - [Relatieve koppeling](#)
 - [Interne koppeling](#)
 - [Koppeling naar een e-mailadres](#)
- [Lijsten](#)
 - [De geordende lijst \(\)](#)
 - [Ongeordende lijst \(\)](#)
 - [Definitie lijst \(<dl>\)](#)
 - [Geneste lijsten](#)
- [Media](#)
 - [Afbeeldingen toevoegen](#)
 - [Afbeeldingsformaten](#)
 - [Afbeeldingen groeperen](#)
 - [Afbeelding als koppeling](#)
 - [Andere media: iframe](#)
- [HTML in 5 minuten](#)

Inleiding

Webtalen

Het web zoals we het kennen, is gebouwd op drie fundamentele technologieën: HTML, CSS en JavaScript. Samen vormen ze de ruggengraat van elke website of webapplicatie.

- **HTML** (**H**yper**T**ext **M**arkup **L**anguage): bepaalt de structuur en inhoud van een webpagina.
- **CSS** (**C**ascading **S**tyl**S**heets): verzorgt de opmaak en de stijl van de HTML-elementen.
- **JavaScript**: een programmeertaal om webpagina's interactief en dynamisch te maken.

Door deze drie technologieën te combineren, kan je niet alleen informatie tonen, maar ook aantrekkelijke en gebruiksvriendelijke ervaringen creëren. In deze cursus starten we met HTML en CSS, de bouwstenen van elke webpagina.

HTTP-Protocol

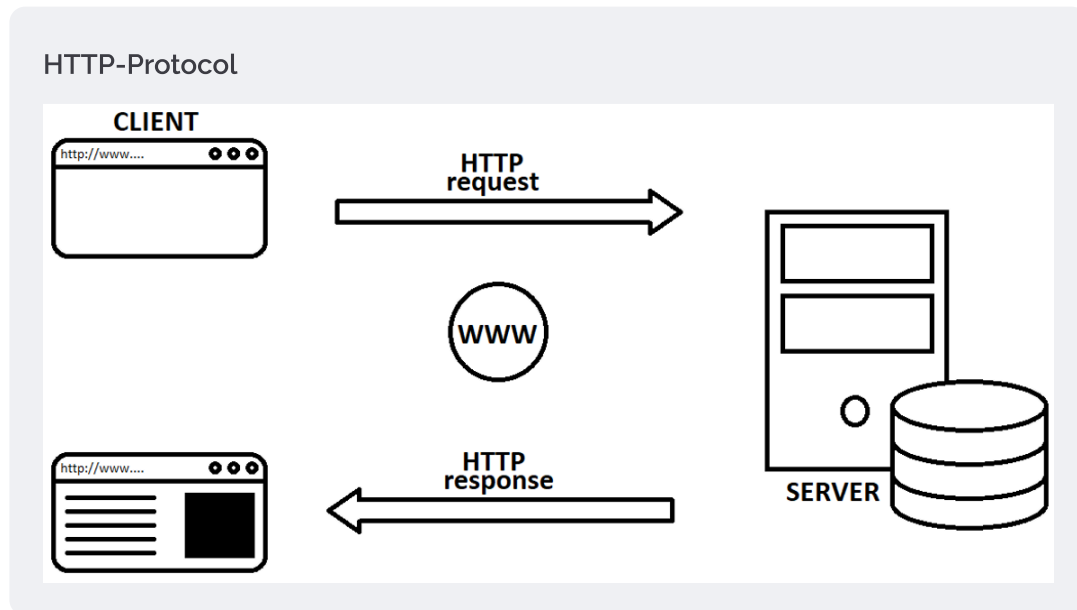
Het HTTP-protocol (**H**yper**T**ext **T**ransfer **P**rotocol) is de standaardtaal waarmee computers op het internet informatie uitwisselen. Het werkt volgens een **vraag-en antwoordmodel** (request/response) voor de **communicatie** tussen een **client** (meestal jouw browser) en een **server**.

Wanneer je bijvoorbeeld www.example.com bezoekt:

- De browser (client) stuurt een HTTP-request naar de server: "Mag ik de homepage?"
- De server antwoordt met een HTTP-response: "Hier is de HTML, en eventueel afbeeldingen en CSS-bestanden."

Je kan het dus zien als een gesprek: de client stelt een vraag, de server geeft een antwoord.

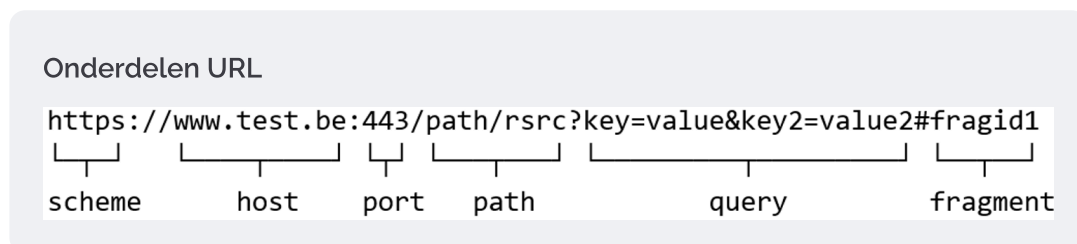
HTTP is **platformonafhankelijk** en vormt de basis van bijna alle communicatie op het web. Tegenwoordig gebruiken we meestal HTTPS, een beveiligde versie van HTTP waarbij alle gegevens versleuteld worden om privacy en veiligheid te garanderen.



URL's

- Uniform Resource Locator
- Geeft de unieke locatie van een document weer in de vorm van een adres of pad
- Bestaat uit verschillende onderdelen

Onderdelen:



- Scheme/Protocol: het soort van transport dat gebruikt wordt - http, https, ftp, file, ... - *Auto of trein*
- Host: de domeinnaam of het IP-adres - *De stad waarin je rijdt*
- Poortnummer: de poort die je wil gebruiken (TCP-poort die gebruikt wordt, optioneel, standaard 80 voor het *http*-protocol en 443 voor het *https*-protocol) - *Via waar je de stad binnenrijdt*
- Padnaam: welk deel je specifiek wil aanspreken - *De straat en huisnummer*
- Query: extra instructies die je wil meegeven voor het specifieke deel (niet-hiërarchische data, vraagteken gevolgd door *key-value*-paren gescheiden door ampersand) - *Toeter 3 maal als je voor het huis staat*
- Fragmentidentifier: een specifieke plaats over het deel dat wordt weergegeven (subdocument of specifiek onderdeel van het document) - *Een specifieke kamer in het huis*

▼ Oefening

Bekijk onderstaande URL's en benoem alle onderdelen.

- <http://127.0.0.1:8080/oefening1.html>
- https://mydb/?view=tbl_articles&id=23
- <http://shop.myserver.be/article/23/reviews>
- <ftp://192.168.97.24/examenopgave/java.zip>
- https://nl.wikipedia.org/wiki/Hogeschool_PXL#Geschiedenis
- <http://www.google.be/search?q=url>

Integrated Development Environment (IDE)

Een IDE (Integrated Development Environment) is een programma dat alle essentiële tools voor softwareontwikkeling samenbrengt in één omgeving. Denk aan een slimme teksteditor, foutopsporing (debugging) en hulpmiddelen voor projectbeheer.

In deze cursus wordt er meestal gebruikgemaakt van **JetBrains WebStorm**. Een veelgebruikt gratis en open-source alternatief is [Visual Studio Code](#).

JetBrains WebStorm

Je kan een gratis studentenlicentie voor WebStorm krijgen via <https://www.jetbrains.com/community/education/#students>

(Google Chrome) DevTools

DevTools (Developer Tools) is een ingebouwde **set van hulpmiddelen** in moderne browsers, zoals Google Chrome, waarmee je websites kan inspecteren, testen en optimaliseren. Met DevTools kan je:

- De HTML-structuur en CSS-styling bekijken en live aanpassen.
- JavaScript-fouten opsporen via de Console.
- Netwerkverkeer analyseren om te zien welke bestanden geladen worden en hoe snel.
- Prestaties en toegankelijkheid testen met Lighthouse.

Google Chrome DevTools

Je kan deze openen via rechtermuisklik **Inspecteren** of met een van de volgende shortcuts:

- **F12** (Windows/Mac)
- **Ctrl+Shift+I** (Windows)
- **Command + Option + I** (Mac)

Starten met HTML

Onze eerste HTML pagina

Elke webpagina begint met een basisstructuur in HTML. Deze structuur vertelt de browser hoe de inhoud van de pagina is opgebouwd. Hieronder zie je een voorbeeld van een lege webpagina met de minimale elementen die nodig zijn.

```
<!doctype html>
<html lang="nl">
  <head>
    <meta charset="utf-8" />
    <title>Dit is de titel van de webpagina</title>
  </head>
  <body></body>
</html>
```

html

<!doctype html> laat de browser weten dat we HTML5 gebruiken en is verplicht mee te geven:

```
<!doctype html>
<html lang="nl">
  <head>
    <meta charset="utf-8" />
    <title>Dit is de titel van de webpagina</title>
  </head>
  <body></body>
</html>
```

html

<html> -element representeert de *root* van een HTML-document:

```
<!doctype html>
<html lang="nl">
  <head>
    <meta charset="utf-8" />
```

html

```
<title>Dit is de titel van de webpagina</title>
</head>
<body></body>
</html>
```

Het `lang` -attribuut (*attributen komen verder in detail aan bod*) geeft aan in welke taal de inhoud van een element geschreven is. Dit helpt browsers, zoekmachines en hulpmiddelen zoals schermlezers om de tekst correct te interpreteren en uit te spreken.

Het `<head>` -element bevat alle informatie die betrekking heeft op het HTML-document maar niet in het documentvenster wordt weergegeven:

```
<!doctype html>
<html lang="nl">
  <head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width,
initial-scale=1.0" />
    <meta name="description" content="Dit is een
beschrijving van de webpagina" />
    <meta name="author" content="Jouw naam" />
    <link rel="stylesheet" href="style.css" />
    <title>Dit is de titel van de webpagina</title>
  </head>
  <body></body>
</html>
```

- Informatie voor browsers. Bijvoorbeeld:
 - Een `<meta>` -element met de charset (welke tekstkarakters er dienen gebruikt te worden)
 - Een `<meta name="viewport">` -element met de viewport (wordt behandeld in een andere les)
- Informatie voor zoekmachines. Bijvoorbeeld:

- Een `<meta name="description">` -element met de beschrijving van de webpagina
- Een `<meta name="author">` -element met de auteur van de webpagina
- Linken naar stylesheets (CSS-bestanden) en JavaScript-bestanden. Bijvoorbeeld:
 - Een `<link>` -element naar een CSS-bestand (wordt behandeld in een andere les)
- Een `<title>` -element, dit is de titel die in de tab van de browser en in zoekresultaten wordt weergegeven.

Het `<body>` -element bevat alle verdere teksten, afbeeldingen en HTML-elementen:

```
<!doctype html>
<html lang="nl">
  <head>
    <meta charset="utf-8" />
    <title>Dit is de titel van de webpagina</title>
  </head>
  <body>
  </body>
</html>
```

html

▼ Oefening

Maak een lege webpagina aan via kladblok:

- Noem het bestand `voorbeeld1.html`.
- Voorzie de basiscode voor een lege webpagina.
- Voorzie in het head element de titel 'Web Essentials'.
- Zoek online hoe je een hoofding of kop `Week 1` kan voorzien in het body element.
- Sla het bestand op.

- Open het bestand in je browser (door bijvoorbeeld te dubbelklikken op het bestand).

Elementen en tags

Een *element* is een onderdeel van een webpagina, zoals een paragraaf, een kop of een afbeelding.

Een element bestaat meestal uit drie delen:

- **Starttag**: geeft aan waar het element begint.
- **Inhoud**: de tekst of andere elementen binnenin.
- **Eindtag**: geeft aan waar het element eindigt.

Normale elementen (met inhoud) hebben een begin- en eindtag.

```
<p>dit is een paragraaf</p>
```

html

```
<!-- Starttag = <p> -->
```

```
<!-- Inhoud = Dit is een paragraaf -->
```

```
<!-- Eindtag = </p> -->
```

Sommige elementen hebben geen inhoud. Deze noemen we **lege** of **self-closing** elementen.

```
 <br />
```

html

TIP

Het is mogelijk om de slash `/` weg te laten, onderstaande elementen zijn dus identiek:

- `` of ``

- `
` of `
`

- Een **tag** is de code tussen `<` `>` die het begin of einde van een element aanduidt.
- Een **element** is het geheel van starttag, inhoud en eindtag (of alleen een self-closing tag bij lege elementen).

Nesten

In HTML kunnen elementen andere elementen bevatten, dit noemen we **nesten** (nesting).

Bijvoorbeeld: een paragraaf kan een vetgedrukt woord bevatten.

```
<p>dit is een <strong>belangrijk</strong> woord in een  
paragraaf</p>
```

html

- Het element `` zit binnen het element `<p>`.
- De volgorde van openen en sluiten moet correct zijn: eerst open je `<p>`, dan ``, en je sluit ze in omgekeerde volgorde.

Om de leesbaarheid te verhogen laten we geneste elementen inspringen via de tabtoets.

Voorbeeld gebruik tabs

Goed:

```
<!doctype html>  
<html lang="nl">  
  <head>  
    <meta charset="utf-8" />  
    <title>Dit is de titel van de webpagina</title>  
  </head>  
  <body></body>  
</html>
```

html

Fout:

```
<!DOCTYPE html>
<html lang="nl">
<head>
<meta charset="utf-8">
<title>Dit is de titel van de webpagina</title>
</head>
<body>
</body>
</html>
```

html

▼ Engelstalige video: Elementen en Nesten - OPTIONEEL

HTML - Elements - W3Schools.com



Attributen

Een **attribuut** geeft extra informatie over een HTML-element. Het staat altijd in de **starttag** van het element.

Soorten attributen:

- **Globale attributen** kunnen bij elk element gebruikt worden (Bijvoorbeeld `class` en `id`)

- **Specifieke attributen** gelden enkel voor bepaalde elementen
(Bijvoorbeeld `src` en `alt` bij ``, `href` bij `<a>`)

Voorbeelden:

```

<a href="page2.html">pagina 2</a>
<p class="main-article" id="par">paragraaf</p>
```

html

Opbouw van een attribuut

- Een `naam` en een `waarde` gescheiden door een `=`-teken
- Dubbele aanhalingstekens `" "` (of enkele aanhalingstekens `' '`) rond de waarde
- **Geen** spatie voor of na de `=`
- De waarde kan wél spaties bevatten

Let op

Let goed op voor het verschil in typografische `" "` en normale `" "` aanhalingstekens. De typografische (schuine) worden niet verwerkt door de browser en kunnen tot fouten leiden! Dit kan bijvoorbeeld gebeuren als je de code kopieert van een website of Word-document.

Best practice: inhoud en opmaak moet steeds gescheiden worden!

- Alle elementen en attributen die enkel opmaak beschrijven zijn verboden.
- Alle elementen en attributen moeten een inhoudelijke betekenis hebben.

Meer hierover in de les over CSS.

Niet doen

```
<p class="example" id="D2D" style="border:1px solid
```

html

```
red;">paragraaf</p>
```

`style` -attribuut niet gebruiken, stijl voegen we toe via aparte CSS-bestanden

▼ Oefening

1. Maak een link (`<a>`) naar <https://example.com> die opent in een nieuw tabblad. (Hint: gebruik het attribuut `target`.)
2. Voeg aan een afbeelding (``) een alternatieve tekst toe voor toegankelijkheid.

▼ Engelstalige video: Attributen - OPTIONEEL

HTML - Attributes - W3Schools.com



Commentaar

Commentaar is een manier om extra informatie toe te voegen aan je HTML-code zonder dat deze zichtbaar is voor de gebruiker in de browser. Het helpt om je code leesbaar en begrijpelijk te houden, vooral bij grotere projecten of wanneer meerdere mensen samenwerken.

Commentaar start met `<!--` en eindigt op `-->`.

html

```
<!-- Dit is een stukje commentaar -->
```

- Commentaar wordt genegeerd door de browser en verschijnt dus niet op de webpagina.
- Je kan commentaar gebruiken om:
 - Uitleg te geven over een stuk code.
 - Tijdelijk code uit te schakelen zonder deze te verwijderen.
 - Secties in je HTML te markeren voor overzicht.

▼ Oefening

Voeg een commentaar toe boven een `<h1>`-element waarin je uitlegt wat de titel van de pagina is.

▼ Engelstalige video: Commentaar - OPTIONEEL

HTML - Comments - W3Schools.com



Speciale karakters - entiteitsnamen

Doordat de browser bepaalde karakters een speciale betekenis geeft, moeten we een andere manier vinden om deze karakters te typen indien we deze toch willen tonen op de webpagina. Bijvoorbeeld `<` en `>`

worden gebruikt om het begin en einde van een HTML-element aan te geven. Om deze karakters toch te kunnen gebruiken, maken we gebruik van **entiteitsnamen**.

Teken	Entiteit	Betekenis
<	<code>&lt;</code>	kleiner dan
>	<code>&gt;</code>	groter dan
&	<code>&amp;</code>	ampersand
<i>vaste spatie</i>	<code>&nbsp;</code>	Non Breaking SPace
"	<code>&quot;</code>	dubbel aanhalingsteken
...		

Bekijk hier een volledige lijst van alle mogelijkheden:

<https://www.freeformatter.com/html-entities.html>

▼ Oefening

Schrijf een paragraaf waarin je `< & >` correct toont met entiteitsnamen.

Conventies en validatie

Om correcte en browser-onafhankelijke HTML te schrijven, volgen we deze regels:

- De `<!DOCTYPE html>` -declaratie is verplicht.
- De `<meta charset="utf-8">` is verplicht.
- We hanteren een correcte volgorde en nesting van elementen.
- Alle HTML-elementen en attributen worden in kleine letters geschreven.
- De waarden van de attributen staan altijd tussen dubbele aanhalingstekens.

- Elementen met een eindtag moeten altijd gesloten worden.
- Algemene teksten moeten altijd in een paragraaf geplaatst worden.

Code Valideren

Je moet steeds je code valideren via <https://validator.w3.org> om te komen tot [gestandaardiseerde en browser-onafhankelijke html-code](#)

Een oefening is pas af al deze gevalideerd is zonder errors!

▼ Oefening

- Kopieer de volgende code in een HTML-bestand:

```
html
<html>
<head>
  <TITLE>De schoenbekooievaar
</head>
<body>
<H1>De Schoenbekooievaar</H2>

<p>De schoenbekooievaar is een <strong><span>vogel</strong>
</span> met een grijs verenkleed. </p>

<p>
  Danzij zijn brede bek kan hij makkelijk vis vangen
  (scheppen) in het troebele water. Het dier verblijft in de
  natte gebieden en moerassen van de <A href=
  https://nl.wikipedia.org/wiki/Sudd>Suddregio</a>

  
</BODY>
</HTML>
```

- Valideer de code aan de hand van de online validator
- Probeer de fouten weg te werken

Structuur

Blokelementen en inline-elementen

In HTML zijn er twee belangrijke categorieën: **blokelementen** en **inline elementen**

- Blokelementen nemen altijd een nieuwe regel in en gebruiken standaard de volledige breedte van hun container.
- Inline elementen blijven in dezelfde regel en zijn slechts zo breed als hun inhoud.

Je kan het zien als *blokken* (block) die onder elkaar staan en *woorden* (inline) die naast elkaar in een zin staan.

Later leer je dat je dit gedrag met CSS kan aanpassen, maar voor nu focussen we op de standaardinstellingen.

Dit is een BLOCK element

Dit is een INLINE-BLOCK element

Dit is een **INLINE** element

Eigenschap	block	inline-block	inline
Komt op een nieuwe regel	✓ Ja	✗ Nee	✗ Nee
Breedte instelbaar	✓ Ja	✓ Ja	✗ Nee (past zich aan inhoud aan)
Hoogte instelbaar	✓ Ja	✓ Ja	✗ Nee
Naast elkaar mogelijk	✗ Nee	✓ Ja	✓ Ja

Eigenschap	block	inline-block	inline
Omvat alleen inhoud	✗ Nee (neemt volledige breedte)	✗ Nee (je bepaalt zelf de breedte)	✓ Ja
Voorbeeld-elementen	<code><div></code> , <code><p></code> , <code><section></code>	<code></code> , <code><button></code>	<code></code> , <code><a></code> , <code></code>

Blokelementen

Blokelementen vormen de bouwstenen van de structuur van een webpagina. Ze worden gebruikt om grotere secties of blokken inhoud te groeperen, zoals paragrafen, koppen, navigatiebalken of containers.

Kenmerken van blokelementen:

- Ze beginnen altijd op een nieuwe regel.
- Ze nemen standaard de volledige breedte van hun container in.
- Ze kunnen andere blokelementen of inline-elementen bevatten.
- Voorbeelden van blokelementen zijn onder andere koppen (`<h1>` tot `<h6>`), paragrafen (`<p>`), en containers zoals `<div>` en `<section>` .

Element	HTML Tag	Voorbeeld
Paragraaf	<code><p>...</p></code>	This is a paragraph.
Kop 1	<code><h1>...</h1></code>	Main Title
Kop 2	<code><h2>...</h2></code>	<hr/> Subtitle

Element	HTML Tag	Voorbeeld
Kop 3	<code><h3>...</h3></code>	Section Title
Kop 4	<code><h4>...</h4></code>	Subsection Title
Kop 5	<code><h5>...</h5></code>	Smaller Title
Kop 6	<code><h6>...</h6></code>	Tiniest Title
Divisie	<code><div>...</div></code>	This is a division.
Blockquote/citaat	<code><blockquote>...</blockquote></code>	<div>This is a quote.</div>
Preformatted (<i>tekst wordt exact weergegeven, inclusief spaties, tabs en regeleinden</i>)	<code><pre>...</pre></code>	Preformatted text
Ongeordende lijsten	<code></code>	<ul style="list-style-type: none"> • Item 1 • Item 2
Geordende lijsten	<code></code>	<ol style="list-style-type: none"> 1. First 2. Second

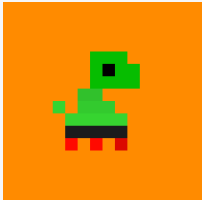
Inline-elementen

Inline-elementen worden gebruikt om kleinere stukken inhoud binnen een regel te markeren of te structureren, zoals een woord, een link of een icoon. Ze breken geen nieuwe regel en nemen enkel zoveel ruimte in als hun inhoud nodig heeft.

Kenmerken van inline-elementen:

- Ze blijven in dezelfde regel als de omringende tekst.
- Ze zijn zo breed als hun inhoud.
- Ze kunnen andere inline-elementen bevatten, maar geen blokelementen.
- Voorbeelden van inline-elementen zijn onder andere `<a>` (hyperlink), `` (belangrijke tekst), `` (nadruk) en `` (generieke inline-container).

Je kan inline-elementen zien als woorden in een zin: ze staan naast elkaar en vormen samen een groter geheel.

Element	HTML Tag	Voorbeeld
Code blok	<code><code>...</code></code>	<code>print("Hello")</code>
Quote	<code><q>...</q></code>	"This is a quote"
Hyperlink	<code><a>...</code>	Link
Tekst-selectie	<code>...</code>	This is selected text
Urgent	<code>...</code>	Important!
Superscript	<code><sup>...</sup></code>	x ²
Subscript	<code><sub>...</sub></code>	H ₂ O
Uitgelicht	<code>...</code>	Bold text
Andere stemming	<code><i>...</i></code>	<i>Italic text</i>
Afbeelding	<code></code>	
Inputveld	<code><input /></code>	
Selectieveld	<code><select></code>	Option 1

Element	HTML Tag	Voorbeeld
Line break	<code>
</code>	Line 1 Line 2
Word break	<code><wbr /></code>	Supercalifragilistic

▼ Oefening

Maak de volgende webpagina volledig na.

- Gebruik de voorbeeldelementen uit de voorgaande secties.
- Zoek via [W3Schools](https://www.w3schools.com/html/html_tags.asp) op hoe je een horizontale lijn kan toevoegen in HTML.

Dit is een kop op niveau 1

Dit is een kop op niveau 2

Dit is een kop op niveau 3

Dit is een kop op niveau 4

Dit is een kop op niveau 5

Dit is een kop op niveau 6

Dit is een paragraaf

Dit is een citaat. Een citaat is een letterlijke uitspraak van iemand, die door iemand anders aangehaald wordt.

Deze paragraaf gaan we over twee lijnen verdelen met een line break.

In deze paragraaf gebruiken we een ^{superscript} en een _{subscript}.

In deze paragraaf **lichten we iets uit** en zorgen we voor *een andere stemming*.

▼ Oplossing

```

<!doctype html>
<html lang="nl">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width,
initial-scale=1.0" />
    <title>Voorbeeldpagina</title>
  </head>
  <body>

```

html

```

<h1>Dit is een kop op niveau 1</h1>
<h2>Dit is een kop op niveau 2</h2>
<h3>Dit is een kop op niveau 3</h3>
<h4>Dit is een kop op niveau 4</h4>
<h5>Dit is een kop op niveau 5</h5>
<h6>Dit is een kop op niveau 6</h6>

<p>Dit is een paragraaf</p>

<blockquote>
  Dit is een citaat. Een citaat is een letterlijke
  uitspraak van iemand, die
  door iemand anders aangehaald wordt.
</blockquote>

<p>
  Deze paragraaf gaan we over <br />
  twee lijnen verdelen met een line break.
</p>

<p>
  In deze paragraaf gebruiken we een
  <sup>superscript</sup> en een
  <sub>subscript</sub>.
</p>

<!-- Horizontale lijn -->
<hr />

<p>
  In deze paragraaf <strong>lichten we iets
  uit</strong> en zorgen we voor
  <em>een andere stemming</em>.
</p>
</body>
</html>

```

▼ Oefening - EXTRA

Maak een webpagina over een huisdier naar keuze. Geef hierbij algemene informatie over het ras, de verzorging, de voeding, de huisvesting, ...

- Gebruik minstens 5 verschillende blokelementen.

- Gebruik minstens 6 verschillende inline elementen.

Je kan dit ook over je favoriete zanger, game of sport doen.

▼ Engelstalige video: Block en Inline elementen - OPTIONEEL

HTML - Block and Inline - W3Schools.com



Structuurelementen

Vroeger werd er vaak gebruikgemaakt van `<div>`-elementen om de structuur van een webpagina op te bouwen. Het probleem? `<div>` zegt niets over de betekenis van de inhoud.

Tegenwoordig gebruiken we **semantische structuurelementen**. Deze elementen beschrijven **wat** de inhoud is. Dit helpt bij:

- de toegankelijkheid: schermlezers begrijpen beter wat elk deel van de pagina doet
- SEO (**S**earch **E**ngine **O**ptimization, Nederlands: zoekmachineoptimalisatie): zoekmachines kunnen de inhoud beter indexeren

Online zal je vaak nog `div`-elementen tegenkomen, probeer dit alleen te gebruiken als er geen semantisch alternatief is.

Structuurelementen geven, zoals de naam het zegt, je website meer structuur en zijn **verplicht** te gebruiken.

Element	HTML Tag
Hoofding	<code><header>...</header></code>
Navigatieblok	<code><nav>...</nav></code>
Hoofdgedeelte	<code><main>...</main></code>
Secties	<code><section>...</section></code>
Artikels	<code><article>...</article></code>
Voetnoot	<code><footer>...</footer></code>
Randinformatie	<code><aside>...</aside></code>

Het is niet altijd eenvoudig om te bepalen wanneer je wat gebruikt.

Meer info

Je kan meer lezen over de verschillende structuurelementen op [W3Schools](https://www.w3schools.com/html/html5_structural_elements.asp).

Element	HTML Tag	Beschrijving	Voorbeelden
Artikel	<code><article>...</article></code>	De inhoud van dit structuurelement kan op zichzelf gelezen of gedeeld worden. De inhoud is onafhankelijk van de rest van de pagina.	Een krantenartikel of nieuwsitem. Een recept in een kookboek.

Element	HTML Tag	Beschrijving	Voorbeelden
Sectie	<pre><section>... </section></pre>	De inhoud van dit structuurelement is een onderdeel van een groter geheel en is op zichzelf onvolledig.	Een hoofdstuk uit een boek. De eerste pagina van een blogpost.
Randinformatie	<pre><aside>... </aside></pre>	De inhoud van dit structuurelement kan worden weggelaten. Het wordt gezien als iets 'extra', maar het is niet nodig om de inhoud te lezen of te begrijpen.	Een wist-je-datje o tip. Een reclamebanner.
Hoofding	<pre><header>... </header></pre>	De inhoud van dit structuurelement heeft een inleidend of verwijzend karakter.	De titel en subtitel van een blogpost.
Hoofdgedeelte	<pre><main>... </main></pre>	De inhoud van dit structuurelement is dominant en essentieel.	De eigenlijke tekst van een blogpost.
Voettekst	<pre><footer>... </footer></pre>	De inhoud van dit structuurelement	De auteur en aanmaakdatum onderaan een

Element	HTML Tag	Beschrijving	Voorbeelden
		omvat meta- of extra detailinformatie.	blogpost. De copyrightinformatie onderaan een webpagina.
Navigatieblok	<pre><nav>... </nav></pre>	De inhoud van dit structuurelement omvat navigatielinken.	De linken naar andere webpagina's binnen een website.

- Een volledig uitgewerkt voorbeeld kan er zo uitzien:

```

<!doctype html>
<html lang="nl">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width,
initial-scale=1.0" />
    <title>Document</title>
  </head>
  <body>

    <header>
      <h1>Paginatitel</h1>
      <p>Tekst</p>
    </header>

    <nav>
      <ul>
        <li><a href="link1.html">Link 1</a></li>
        <li><a href="link2.html">Link 2</a></li>
        <li><a href="link3.html">Link 3</a></li>
      </ul>
    </nav>

```

```
<section>
  <header>
    <h2>Titel</h2>
  </header>
  <p>Inhoud van de sectie</p>
  <footer>
    <p>sectievoetnoot</p>
  </footer>
</section>

<aside>
  <p>Bijkomende inhoud</p>
</aside>

<footer>
  <p>Paginavoetnoot</p>
</footer>

</body>
</html>
```

Hyperlinks

Met hyperlinks - gemaakt met `<a> ... ` - kan je navigeren naar andere pagina's of bestanden in hetzelfde venster of in een nieuw venster.

Belangrijke attributen

- `href` -attribuut: bevat de URL of het pad naar de bestemming.
- `target` -attribuut geeft aan hoe de link geopend moet worden:
 - `target="_self"` : in het huidige venster (standaard)
 - `target="_blank"` : in een nieuw venster
 - andere opties zijn minder van belang

Enkele voorbeelden:

html

```
<a href="pagina1.html">Pagina 1</a>  
<a href="http://www.andere-website.be/pagina2.html">Pagina  
2</a>  
<a href="/txt/tekst3.txt" target="_blank">Open tekstbestand  
in map txt</a>
```

▼ Oefening

Maak een HTML-pagina met:

- Een link naar een interne pagina.
- Een link naar een externe website die opent in een nieuw tabblad.

Absolute koppeling

Een **absolute koppeling** verwijst naar een specifiek document op een webserver en bevat het volledige adres (URL), inclusief:

- Protocol (http:// of https://)
- Domeinnaam
- Eventueel het pad naar het bestand

html

```
<a href="http://www.andere-website.be/pagina2.html">Pagina  
2</a>
```

Wanneer gebruik je een absolute koppeling?

- Voor links naar externe websites.
- Voor bestanden die niet op je eigen server staan.

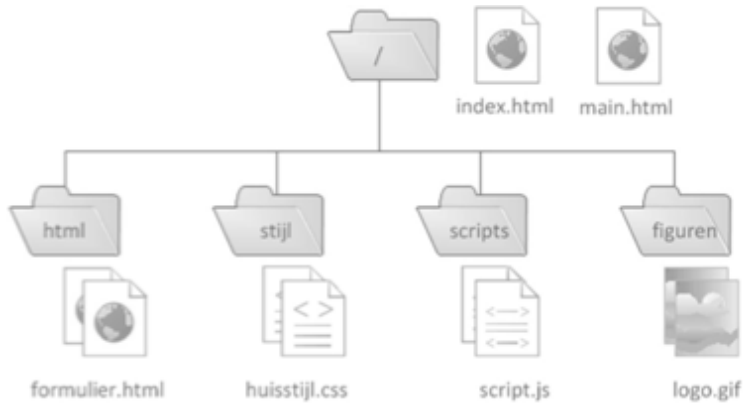
Relatieve koppeling

Een **relatieve koppeling** verwijst naar een ander bestand, maar doet dit relatief ten opzichte van de locatie van het huidige HTML-bestand. In

tegenstelling tot een absolute koppeling bevat een relatieve koppeling geen domeinnaam of protocol.

Voorbeeld:

Vooropgesteld hebben we volgende bestandstructuur:



1. Link naar een bestand in dezelfde map:

We vertrekken vanuit het `index.html` bestand en linken naar `main.html`:

```
<a href="./main.html">...</a>
```

html

We verwijzen 'vanuit de huidige map' (`./`) naar `main.html`.

2. Link naar een bestand één map dieper in de folderhiërarchie (dalen)

We vertrekken vanuit het `index.html` bestand en linken naar `formulier.html`:

```
<a href="./html/formulier.html">...</a>
```

html

Ga naar de huidige map (`./`) en dan naar de map `html` en vervolgens naar het bestand `formulier.html`.

3. Link naar een bestand één map hoger in de folderhiërarchie (stijgen)

We vertrekken vanuit het `formulier.html` bestand en linken naar `index.html` :

```
<a href="../index.html">...</a>
```

html

Er staat `../`, wat betekent `ga één map hoger` en dan naar `index.html`.

4. Link een stijlbestand

Om `huisstijl.css` te linken in zowel `index.html` als `formulier.html`, zou je het volgende doen:

`index.html` :

```
<link rel="stylesheet" href="../stijl/huisstijl.css" />
```

html

`formulier.html` :

```
<link rel="stylesheet" href="../../stijl/huisstijl.css" />
```

html

`../` is optioneel

De browser gaat standaard uit van de huidige map.

Met expliciete `../`.

```
<a href="/main.html">...</a>
```

html

Zonder expliciete `../`.

```
<a href="main.html">...</a>
```

html

▼ Oefening

Maak een HTML-fragment met:

- Een absolute link naar <https://www.example.com>.
- Een relatieve link naar `contact.html`.

▼ Engelstalige video: Relatieve vs Absolute koppeling - OPTIONEEL

Wil je wat meer informatie over relatieve en absolute koppelingen, dan kan je altijd onderstaande video bekijken.

HTML File Paths | Absolute vs Relative URLs | Tutorial for B...



Interne koppeling

Een interne koppeling wordt gebruikt om naar een specifieke positie te gaan binnen **dezelfde pagina** of naar een bepaald onderdeel op een **andere pagina**. Dit is handig voor lange pagina's of inhoudsopgaven.

Een interne koppeling bestaat altijd uit **twee delen**:

- Een anker (Engels: anchor) om een element te identificeren. Hiervoor voeg je een `id` -attribuut toe aan het element (bv. `id="someId"`).
- Een hyperlink die met een hashtag verwijst naar het anker waar je naartoe wilt gaan (bv. `href="#someId"`).

Koppeling binnen dezelfde pagina:

```
<h2>Inhoudsopgave</h2>
<ul>
  <li><a href="#deel1">deel 1</a></li>
  <li><a href="#deel2">deel 2</a></li>
</ul>
<article id="deel1">
  <h3>Deel 1: Inleiding</h3>
</article>
<article id="deel2">
  <h3>Deel 2: Structuur</h3>
</article>
```

html

Koppeling naar een andere pagina:

```
<a href="./index.html#deel2">deel 2</a>
```

html

Linken naar andere pagina

Op de pagina waar je naar verwijst moet het anker (`id` -attribuut) aanwezig zijn.

Koppeling naar een e-mailadres

Bij een koppeling naar een e-mailadres voegen we een `mailto` -opdracht toe aan het `href` -attribuut, zodat je een e-mailprogramma kan openen met een vooraf ingevuld e-mailadres. De `mailto` -opdracht kunnen we naar wens uitbreiden met een `cc` , `bcc` en `subject` .

```
<a href="mailto:persoon1@bedrijf.be">...</a>
<a
  href="mailto:persoon1@bedrijf.be?
  cc=persoon2@bedrijf.be& bcc=persoon3@bedrijf.be& subject
```

html

```
>...</a  
>
```

- De ampersand `&` is geschreven als `&` omdat er geen speciale tekens kunnen gebruikt worden in HTML
- De spatie is geschreven als `%20`, omdat er geen spaties mogen voorkomen in een URL.

▼ Oefening

Maak een link die:

- Een e-mail opent naar info@bedrijf.be.
- Het onderwerp instelt op "Vraag over product".
- Een cc toevoegt naar support@bedrijf.be.

▼ Engelstalige video: hyperlinks - OPTIONEEL

HTML - Links - W3Schools.com



Lijsten

Lijsten worden gebruikt om gerelateerde items overzichtelijk weer te geven. HTML kent drie soorten lijsten:

De geordende lijst (``)

Een geordende lijst of **ordered list** geeft items in een **bepaalde volgorde** weer (genummerd).

```
<ol>
  <li>item A</li>
  <li>item B</li>
  <li>item C</li>
  <li>item D</li>
  <li>item D</li>
</ol>
```

html

1. item A
2. item B
3. item C
4. item D
5. item D

Ongeordende lijst (``)

Een ongeordende lijst of **unordered list** geeft items weer **zonder specifieke volgorde** (met opsommingstekens).

```
<ul>
  <li>item A</li>
  <li>item B</li>
  <li>item C</li>
  <li>item D</li>
</ul>
```

html

- item A
- item B
- item C
- item D

Definitie lijst (`<dl>`)

Een definitielijst of **d**escription **l**ist wordt gebruikt voor termen en hun beschrijving.

- `<dl>`: description list
- `<dt>`: description term
- `<dd>`: description

```
<dl>
  <dt>Term</dt>
  <dd>Definitie van woord</dd>
</dl>
```

html

Term
Definitie van woord

Geneste lijsten

Lijsten kunnen genest worden (Engels: nesting), wat betekent dat een lijst binnen een andere lijst kan worden geplaatst. Dit kan handig zijn om sub-items of gerelateerde items te groeperen.

```
<ul>
  <li>item A</li>
  <li>
    item B
    <ul>
      <li>item B1</li>
      <li>
        item B2
        <ul>
          <li>item B2.1</li>
          <li>item B2.2</li>
        </ul>
      </li>
    </ul>
  </li>
  <li>item C</li>
</ul>
```

- item A
- item B
 - item B1
 - item B2
 - item B2.1
 - item B2.2
- item C

Nesten van lijsten

Een `` - of `` -element kan alleen `` -elementen bevatten. Als je een geneste lijst wilt maken, moet je een nieuwe `` of `` binnen een bestaand `` -element plaatsen.

▼ Oefening

Maak een nieuwe webpagina `recept.html` aan en zoek online op hoe je vanillecake maakt.

Voeg een ongeordende boodschappenlijst (``) toe aan de webpagina met alle nodige ingrediënten om de cake te maken. Voeg een geordende lijst (``) toe met de stappen die je moet volgen om de cake te maken.

▼ Engelstalige video: lijsten - OPTIONEEL

HTML - Lists - W3Schools.com



Media

Afbeeldingen toevoegen

Om een afbeelding toe te voegen aan een webpagina, gebruik je het

`` -element. Dit element heeft:

- een `src` -attribuut dat het pad naar de afbeelding specificeert
- een `alt` -attribuut dat een alternatieve tekst biedt. De `alt` -tekst heeft volgende doelen:
 - *Toegankelijkheid*: schermlezers lezen de `alt` -tekst voor zodat slechtziende of blinde gebruikers weten wat er op de afbeelding staat.
 - *Fallback*: als het bestand ontbreekt, kapot is of de internetverbinding traag is, wordt de `alt` -tekst getoond in plaats van de afbeelding.

- *SEO (zoekmachineoptimalisatie)*: zoekmachines gebruiken de `alt` - tekst om beter te begrijpen waar de afbeelding over gaat. Dit kan helpen om je afbeelding en/of pagina hoger te laten scoren in de zoekmachine.

```

```

html



Afbeeldingsformaten

Bij het werken met afbeeldingen op het web is het belangrijk om het juiste formaat te kiezen. Elk formaat heeft zijn eigen voor- en nadelen afhankelijk van het type afbeelding en het beoogde gebruik. De drie meest gebruikte formaten zijn GIF, JPEG en PNG.

	GIF	JPEG	PNG
Aantal kleuren	256	16 miljoen	16 miljoen
Type compressie	Lossless (behoudt exacte kleuren)	Lossy (kwaliteitsverlies)	Lossless
Bestandsgrootte	Groter	Kleiner	Kleiner
Downloadtijd	Langer	Korter	Korter
Decompressietijd	Korter	Langer	Langer
Animatie	Aanbevolen	Afgeraden	Afgeraden

	GIF	JPEG	PNG
Transparantie	Mogelijk (vanaf Gif89a)	Niet mogelijk	Mogelijk

De keuze van formaat moet je geval per geval beoordelen. Heb je bijvoorbeeld een afbeelding met veel kleuren en een transparante achtergrond dan kies je voor PNG.

Een algemene leidraad:

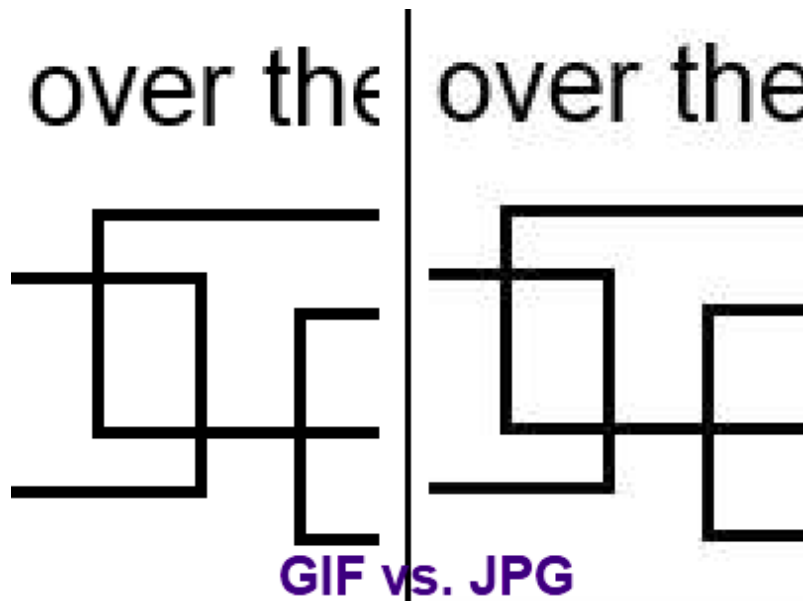
	GIF	JPEG	PNG
Logo's	Aanbevolen	Afgeraden	Afgeraden
Computertekening	Aanbevolen	Afgeraden	Afgeraden
Foto's	Afgeraden	Aanbevolen	Aanbevolen

Andere formaten

Naast GIF, JP(E)G en PNG bestaan er nog vele andere formaten.

- SVG: een vectormaat, schaalbaar zonder kwaliteitsverlies.
- WebP: Een modern formaat, met een focus op kleinere bestandsgroottes en transparantie.
- ...

Enkele voorbeelden die het verschil tussen de formaten illustreren:



Afbeeldingen groeperen

Via het `<figure>` - en `<figcaption>` -element kan je een afbeelding met een bijschrift groeperen. Dit is vooral handig voor afbeeldingen die een specifieke context of uitleg nodig hebben.

```
<figure>
  
  <figcaption>Een voorbeeld van een afbeelding met
```

html

```
bijschrift.</figcaption>  
</figure>
```



Een voorbeeld van een afbeelding met bijschrift.

Afbeelding als koppeling

Via een `<a>`-element kan je aangeven dat tekst een koppeling is naar een andere pagina. Dit kan ook met afbeeldingen. Om een afbeelding als koppeling te gebruiken, plaats je het ``-element binnen een `<a>`-element.

```
<p>Een gewone link naar de PXL-website: <a  
href="https://pxl.be">PXL</a></p>  
<p>  
  Een koppeling naar de PXL-website via een afbeelding:  
  <a href="https://pxl.be">  
      
  </a>  
</p>
```

html

Een gewone link naar de PXL-website: [PXL](#)



Een koppeling naar de PXL-website via een afbeelding:

⋮ Engelstalige video: afbeeldingen - OPTIONEEL

HTML - Images - W3Schools.com



⋮

Andere media: iframe

Met een `<iframe>`-element kan je andere webpagina's of externe content insluiten in je eigen pagina.

Let op

Deze code kan niet live gewijzigd worden.

html

```
<iframe src="https://www.youtube.com/embed/y6TZHLA zg5o"
width="300"></iframe>
```

Classical Piano Music & F...



▼ Oefening

Maak een webpagina `muziek.html` aan en voeg een YouTube-video toe via een `<iframe>`.

- Zoek op hoe je een `embed` -link van een YouTube-video verkrijgt om te gebruiken in het `src` -attribuut van het `<iframe>` -element.
- Stel de hoogte en breedte van het `<iframe>` -element in op 600 en 800 pixels respectievelijk.
- Zoek op welke andere mogelijke attributen er zijn voor het `<iframe>` -element.

::: Engelstalige video: iFrames - OPTIONEEL

HTML - Iframes - W3Schools.com



...

HTML in 5 minuten

Je kan onderstaande Engelstalige video gebruiken om de basis van HTML te herhalen.

HTML in 5 minutes



Previous page

[Overzicht](#)

Next page

[Oefeningen](#)