

Basis statische website

Realiseren

hoofdstuk

1

Introductie HTML





Algemene informatie

Onderwerp	Introductie HTML
Leerdoel(en)	<ol style="list-style-type: none">1. De student kan door middel van een handleiding de werkomgeving installeren die nodig is voor het maken van een eenvoudige statische website.2. De student kan minimaal 2 verschillen benoemen tussen een statische website en een dynamische website.3. De student kan benoemen met behulp van welke talen statische websites gemaakt worden en kan daarbij aangeven waar welke taal voor gebruikt wordt.4. De student kan benoemen hoe structuur wordt aangebracht in een artikel, een website en in HTML-code.5. De student kan benoemen wat de voordelen zijn van een HTML-editor zoals Visual Studio Code ten opzichte van een Kladblok.6. De student kan de broncode bekijken van bestaande webpagina's door gebruik te maken van de Developer Tools.7. De student kan benoemen hoe het internet werkt en daarin de begrippen client, ISP, DNS, IP-adres en server te benoemen en te beschrijven.8. De student kan benoemen uit welke elementen en bijbehorende attributen binnen de openingstags daarvan een HTML-code bestaat en kan de functie daarvan in eigen woorden benoemen.
Vereiste voorkennis	De student heeft voor dit hoofdstuk geen specifieke voorkennis nodig.
Kwalificatiedossier	<ul style="list-style-type: none"><input type="checkbox"/> B1-K1-W1: Plant werkzaamheden en bewaakt de voortgang<input type="checkbox"/> B1-K1-W2: Ontwerpt software<input checked="" type="checkbox"/> B1-K1-W3: Realiseert (onderdelen van) software<input type="checkbox"/> B1-K1-W4: Test software<input type="checkbox"/> B1-K1-W5: Doet verbetervoorstellen voor de software <input type="checkbox"/> B1-K2-W1: Voert overleg<input type="checkbox"/> B1-K2-W2: Presenteert het opgeleverde werk<input type="checkbox"/> B1-K2-W3: Reflecteert op het werk



Inhoudsopgave

Algemene informatie	2
Inhoudsopgave	3
Introductie	4
Inhoud	4
Een statische webpagina	4
Hoe het internet werkt	6
HTML-code bekijken	6
De ontwikkelomgeving	7
Installeren Visual Studio Code	9
Mappenstructuur laden in Visual Studio Code	15
Structuur	16
Structuur in Word	16
Structuur in Websites	17
Elementen en tags	17
Lege elementen	18
Attributen	18
De opbouw van HTML	18
Doctype	19
Webpagina bestanden organiseren	20



Introductie

Bij het onderdeel realiseren ga je vooral aan de slag met het maken van software. Software is echter een breed begrip waar we allerlei verschillende soorten programma's mee kunnen bedoelen. Bijvoorbeeld een programma dat je installeert op een computer, een app op je telefoon, of zelfs een apparaat waarvan een chip geprogrammeerd is om bijvoorbeeld een huis te automatiseren. Al deze verschillende soorten van software ga je uiteindelijk leren maken gedurende deze opleiding. Dat is echter niet makkelijk en dat zal je langzaam moeten gaan leren. We beginnen daarom met het realiseren van eenvoudige kleine websites.

Kleine websites worden vaak met alleen HTML, CSS en Javascript gemaakt. Hierbij ga je de inhoud (teksten, afbeeldingen en dergelijke) van de webpagina zelf coderen. Dit zijn vaak de websites waar al voor een lange tijd dezelfde informatie op staat en heel af en toe een keer wordt bijgewerkt. Denk bijvoorbeeld aan een website van een kleine sportvereniging. Dit soort websites noemen we **statische** websites.

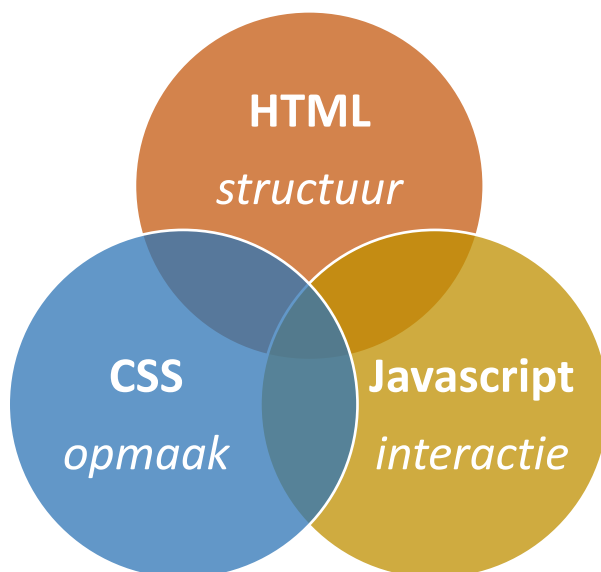
Grotere websites maken vaak gebruik van lastigere technologieën die worden gebruikt om automatisch HTML en CSS te produceren. Denk bijvoorbeeld aan een webwinkel. Dit soort grote en complexe websites gebruiken wellicht een database om gegevens op te slaan, en programmeertalen als PHP, ASP.Net, Java of Ruby. Dit soort grotere en complexere websites die zelf HTML en CSS produceren noemen we ook wel **dynamische** websites.

In dit thema ga je beginnen met het maken van de statische websites. Dat betekent dat we in dit thema bij realiseren gebruik gaan maken van HTML, CSS en Javascript. Later in de opleiding zal je ook aan de slag gaan met het maken van de dynamische website en de daarbij behorende technieken en talen.

Inhoud

Een statische webpagina

Een statische webpagina wordt bijna altijd gebouwd met de HyperText Markup Language (HTML), Cascading StyleSheets (CSS) en Javascript. Iedere taal heeft zijn eigen functie.





HTML staat voor Hypertext Markup Language en wordt gebruikt om de structuur op te zetten van de website. Denk daarbij onder andere aan de Kopteksten, paragrafen, afbeeldingen, links naar andere webpagina's en video's. Een website met alleen HTML ziet er ontzettend saai uit. Daarom ga je er ook CSS aan toevoegen.

CSS staat voor Cascading Stylesheet en wordt gebruikt om de HTML te voorzien van een opmaak. Standaard is een link naar een andere webpagina blauw. Als je dit een andere kleur, lettertype, grootte, positie wilt geven doe je dat door middel van CSS.

Hieronder zie je het verschil tussen de Koning Willem I College website zonder CSS en met CSS.

Zonder CSS:

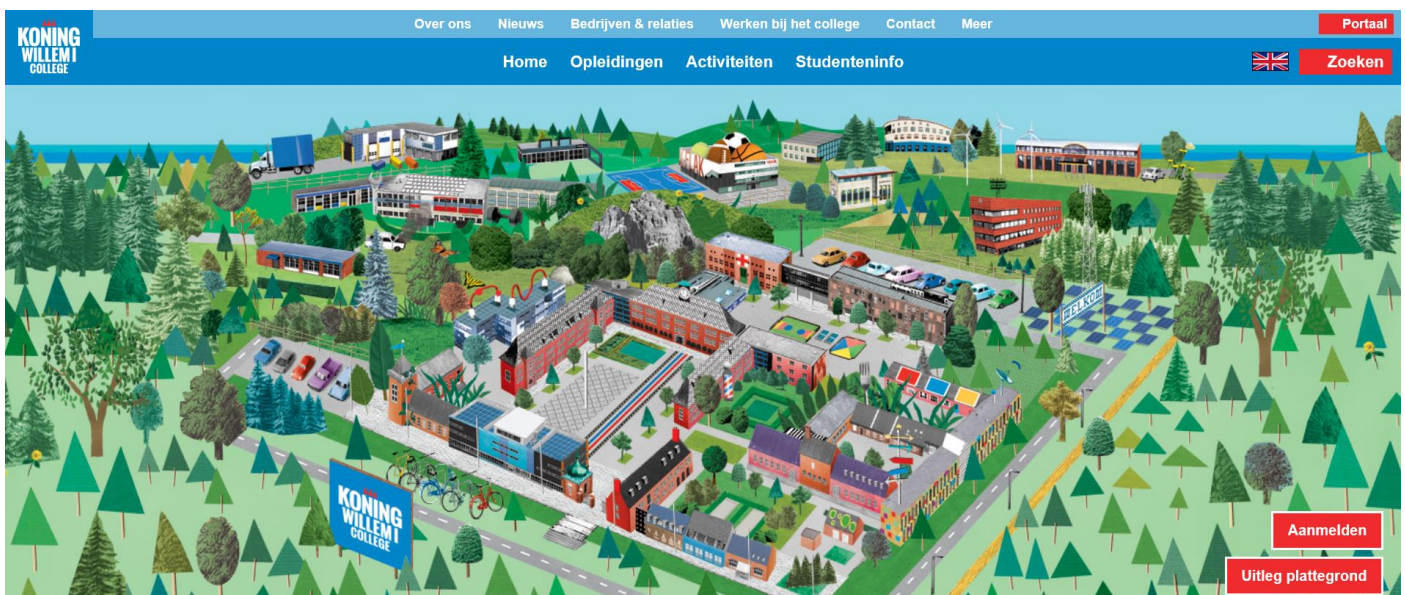
- [Over ons](#)
 - [Community college](#)
 - [Jaar in Beeld](#)
 - [Wie zijn wij?](#)
 - [Jaarverslagen](#)
 - [Organogram](#)
 - [Afdelingen](#)
 - [Missie, visie, kern- en merkwwaarden](#)
 - [Formele visitatie](#)
 - [UNESCO-school](#)
 - [ANBI-status](#)
- [Nieuws](#)
 - [Agenda](#)
 - [Jaar in Beeld](#)
 - [Nieuwsberichten](#)
 - [Persberichten](#)
- [Bedrijven & relaties](#)
 - [Ondernemerschap](#)
 - [Het InnovatieLab](#)
 - [Maatwerk](#)
 - [Projecten](#)
 - [Sturen Leeren](#)
- [Werken bij het college](#)
 - [Vacatures](#)
 - [Open sollicitatie](#)
 - [Studeerlaten lerarenopleiding](#)
- [Contact](#)
 - [Locaties](#)
 - [Routebeschrijving](#)
 - [Stuur ons een e-mail](#)
- [Meer](#)
 - [School voor de Toekomst](#)
 - [Aanbestedingen](#)
 - [Studenteninfo van A tot Z](#)
- [Home](#)
- [Opleidingen](#)
- [Activiteiten](#)
- [Studenteninfo](#)



Welkom bij het Koning Willem I College



Met CSS:



Javascript wordt gebruikt om de webpagina interactiever te maken. Denk daarbij aan bijvoorbeeld een diavoorstelling waarbij foto's elkaar afwisselen of aan een eenvoudige leeftijdschecker op basis van een ingevoerde geboortedatum.



Hoe het internet werkt

Een website met alle bijbehorende bestanden zoals plaatjes en video's wordt (gehost) op een webserver. De webserver die de website host kan zich overal in de wereld bevinden. Om ervoor te zorgen dat die webserver gemakkelijk gevonden kan worden krijgt die webserver een IP-adres (een unieke code). Alle apparaten die verbonden zijn met het internet hebben zo'n IP-adres, dus ook jouw computer, telefoon en misschien wel printer. Op het moment van schrijven staat de website van Bol.com gehost op een server met het IP-adres 185.14.169.113. Als je dat IP-adres intypt in je browser kom je uit op de website van bol.com (let op, dit werkt niet altijd zo. Dit ligt ook aan allerlei andere instellingen die je kunt doen op de server. Je kunt daarnaast ook meerdere websites op dezelfde server hosten). Het moeten invoeren van een IP-adres in de browser om een website te bezoeken is voor een bezoeker niet erg praktisch en gebruikersvriendelijk. Daarom komt er nog wat meer bij kijken als je een website wilt bezoeken.

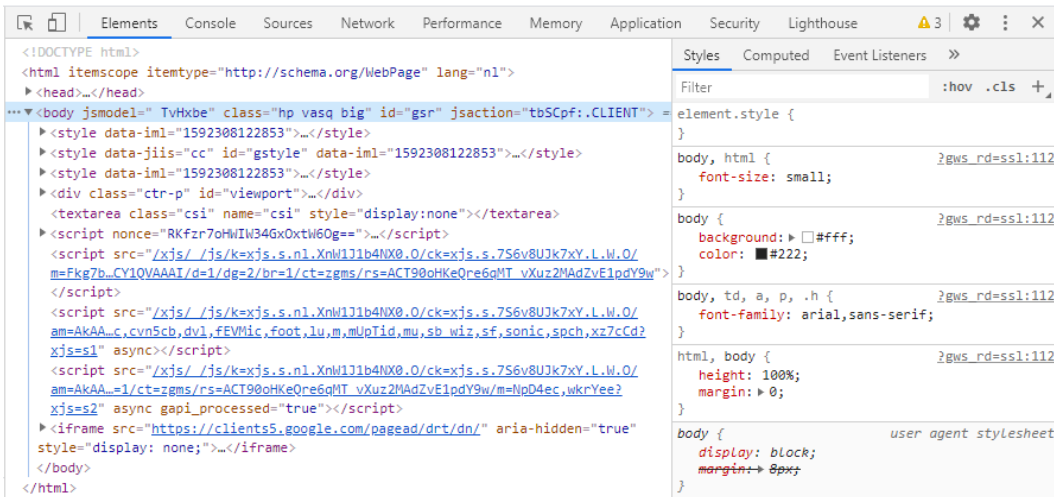
Als je op jouw eigen computer of telefoon (clients) een website wilt bezoeken maak je (vaak via een router of accesspoint) verbinding met jouw modem. Jouw modem geeft aan de internet service provider (ISP), zoals KPN of Ziggo, door welke website je wilt bezoeken. Vervolgens wordt het verzoek doorgestuurd naar een DNS-Server (Domain Name System). In dit systeem staat geregistreerd welk domeinnaam hoort bij welk IP-adres. Daarin staat bijvoorbeeld vastgelegd dat het IP-adres van de server waar de website van bol.com gehost wordt 185.14.169.113 is. Van deze DNS-server krijg jij dan het IP-adres dat jij nodig hebt. Deze wordt teruggestuurd naar jouw computer of telefoon. Nu weet je computer het juiste IP-adres. Deze wordt door de browser automatisch weer naar het internet gestuurd. Nu weet je ISP precies naar welke webserver je verzoek moet.

Het verzoek dat jij die website wilt bezoeken komt aan op de juiste server die vervolgens de benodigde bestanden doorstuurt naar de browser van jouw computer. De browser vertaalt de ontvangen code zodat de bezoeker de website te zien krijgt.

HTML-code bekijken

Je kunt van een bestaande website ook de HTML- CSS-, en Javascriptcode bekijken. Daarbij moet er wel worden aangegeven dat de code die je dan vaak ziet erg rommelig is. Dit komt omdat de meeste websites dynamisch zijn. Oftewel de code wordt automatisch geproduceerd. Als je een statische website zou bekijken, zie je rechtstreeks de code die de Software Developer heeft geschreven.

Als je een webpagina vanuit Google Chrome bezoekt kun je de achterliggende code bekijken door op F12 te drukken. Hieronder een voorbeeld als je F12 indrukt op www.google.nl. We noemen dit venster de Developer Tools



HTML-code wordt meestal netjes gestructureerd zodat de code overzichtelijk blijft. In onderstaand voorbeeld zie je 2 HTML-bestanden die hetzelfde resultaat opleveren. Echter is het bij de overzichtelijke variant veel makkelijker om later dingen aan te passen of om fouten te ontdekken. Daarom moet je de geschreven code altijd goed structureren.

Gestructureerd:

```
<! Doctype html>
<html>
  <head>
    <title>
      Voorbeeld structureren
    </title>
    <meta charset = "UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
  </head>
  <body>
    <p>
      HTML-code moet je altijd goed structureren.
    </p>
  </body>
</html>
```

Niet gestructureerd:

```
<! Doctype html><html><head><title>Voorbeeld structureren</title><meta charset = "UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0"></head><body><p>HTML-
code moet je altijd goed structureren.</p></body></html>
```

De ontwikkelomgeving

Voor het maken van een statische website heb je software nodig. De meest eenvoudige software om dit te doen is kladblok. Dat lees je goed! Met kladblok (Notepad) kun je complete webpagina's bouwen! Omdat kladblok daar in eerste instantie niet specifiek voor gemaakt is zitten er heel veel nadelen aan. Je moet echt alles zelf doen en de kans dat je dan een fout maakt is ontzettend groot. Als je software gaat gebruiken die specifiek gemaakt is voor het maken van een website zitten er vaak handige hulpmiddelen in zodat je sneller en makkelijker je code kunt schrijven.



Hieronder zie je dezelfde HTML-code. De ene gemaakt in Kladblok, de ander gemaakt met behulp van de software Visual Studio Code. Dit is een gratis softwarepakket voor het ontwikkelen van statische websites. Visual Studio Code ga je ook gebruiken tijdens de lessen realiseren.

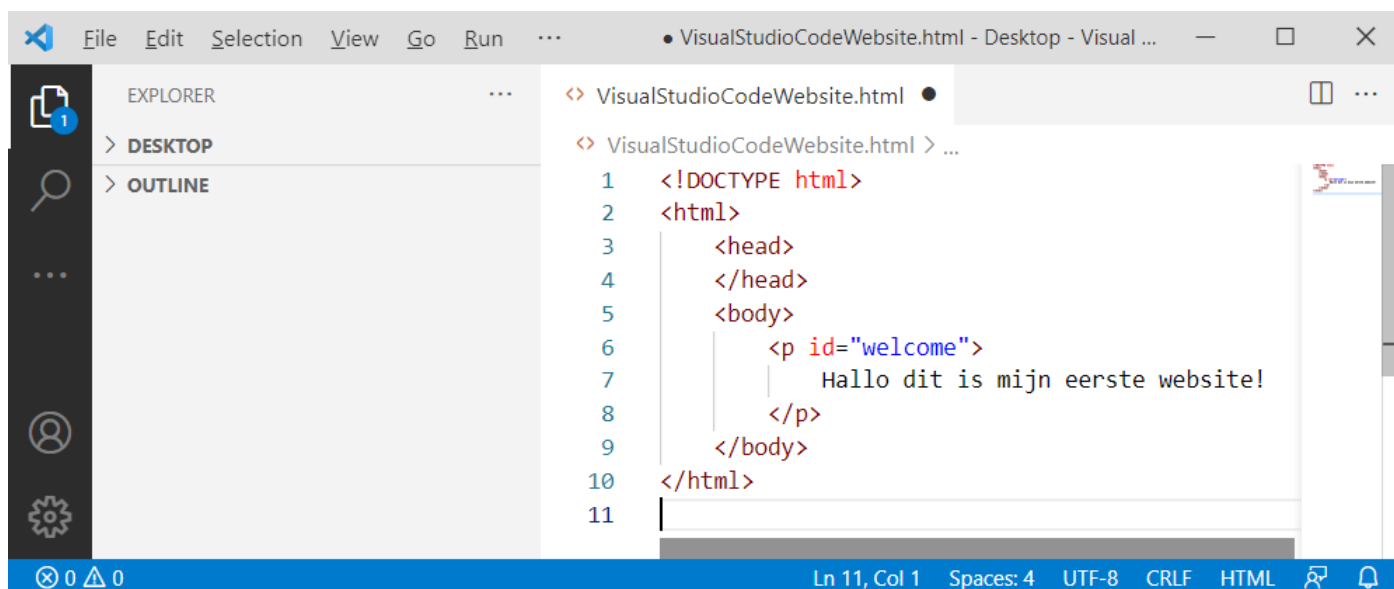


HTML-code in Kladblok:



```
kladblokWebsite.html - Notepad
File Edit Format View Help
<!doctype html>
<html>
  <head>
  </head>
  <body>
    <p id="welcome">
      Hallo! Dit is mijn eerste website!
    </p>
  </body>
</html>
```

HTML-code in Visual Studio Code:



```
VisualStudioCodeWebsite.html - Desktop - Visual ...
File Edit Selection View Go Run ...
EXPLORER
> DESKTOP
> OUTLINE
VisualStudioCodeWebsite.html
VisualStudioCodeWebsite.html > ...
1 <!DOCTYPE html>
2 <html>
3   <head>
4   </head>
5   <body>
6     <p id="welcome">
7       Hallo dit is mijn eerste website!
8     </p>
9   </body>
10 </html>
11
```

Je ziet dat in Visual Studio Code gebruik wordt gemaakt van kleuren, waardoor de code overzichtelijker is. Daarnaast zorgt het er ook voor dat je minder snel fouten maakt. Als je ergens wat vergeet in de code zoals aanhalingstekens dan worden bepaalde delen van de code anders gekleurd.

Daarnaast vult Visual Studio Code ook zelf code aan, kun je een preview openen van de webpagina zodat je direct het resultaat kunt zien, kun je delen van codes tijdelijk verbergen (handig als je een hele lange code hebt waar je even niets meer mee wilt doen) en nog veel meer handige hulpmiddelen voor ontwikkelaars. Allerlei dingen die in Kladblok niet mogelijk zijn.

 Je kunt nu **Oefening 1.1** gaan maken met behulp van onderstaande informatie.

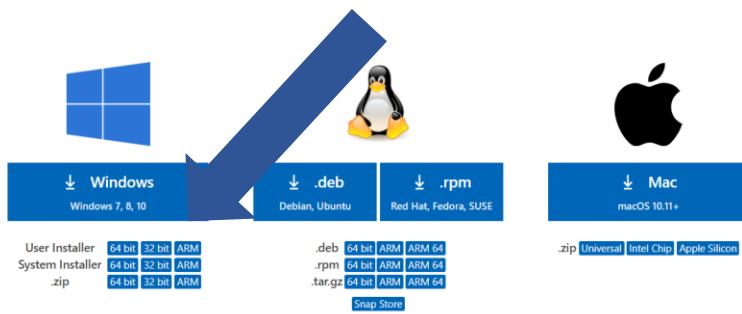
Installeren Visual Studio Code

- Ga naar de website <https://code.visualstudio.com/download>.
- Download de laatste versie van deze software op de gegeven website en installeer het programma volgens de stappen op de volgende bladzijden.

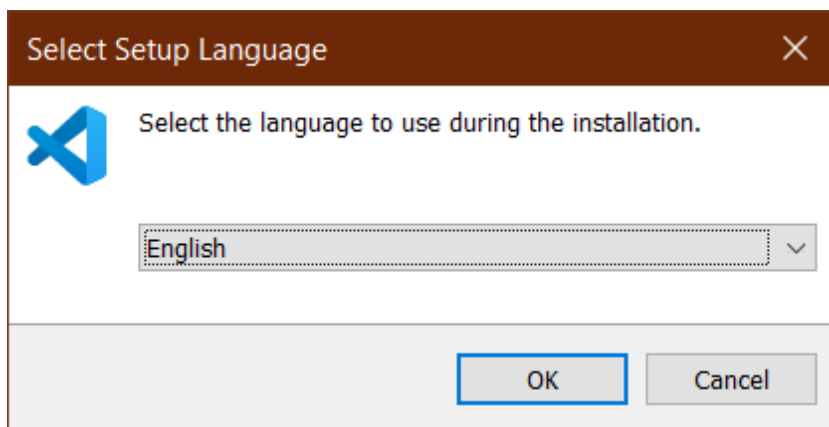


Download Visual Studio Code

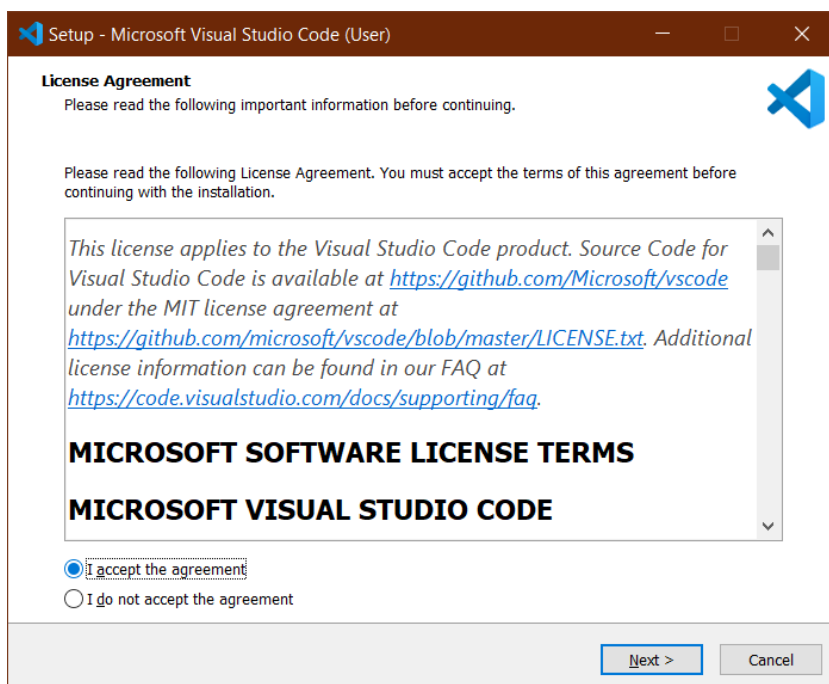
Free and built on open source. Integrated Git, debugging and extensions.



- Open het gedownloade installatiebestand.
- Controleer of de taal op Engels staat en klik op **OK**.

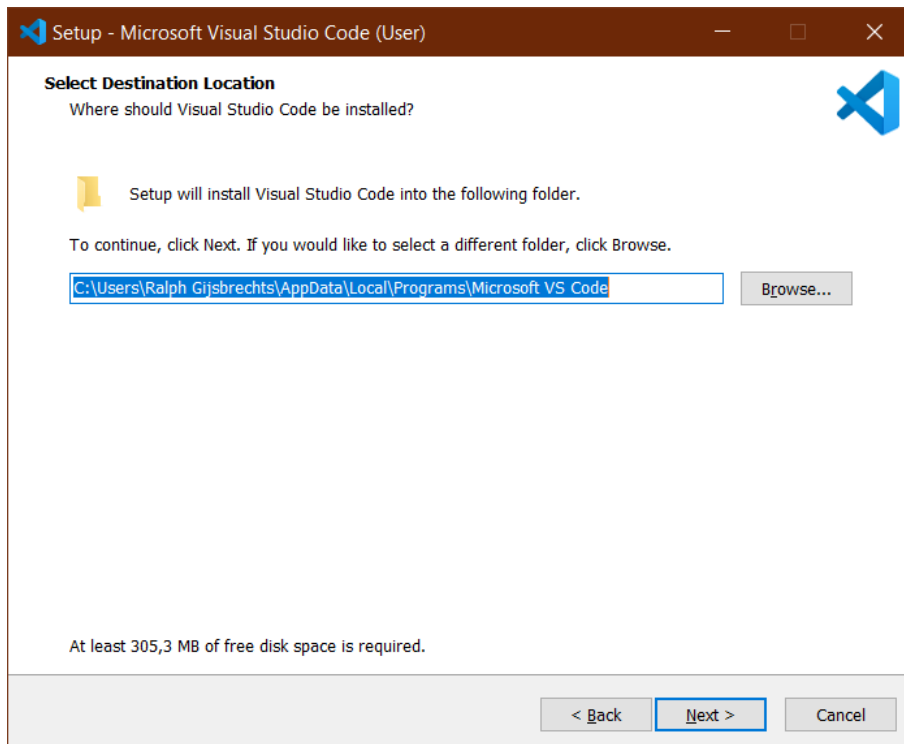


- Accepteer de overeenkomst en klik op **Next**.

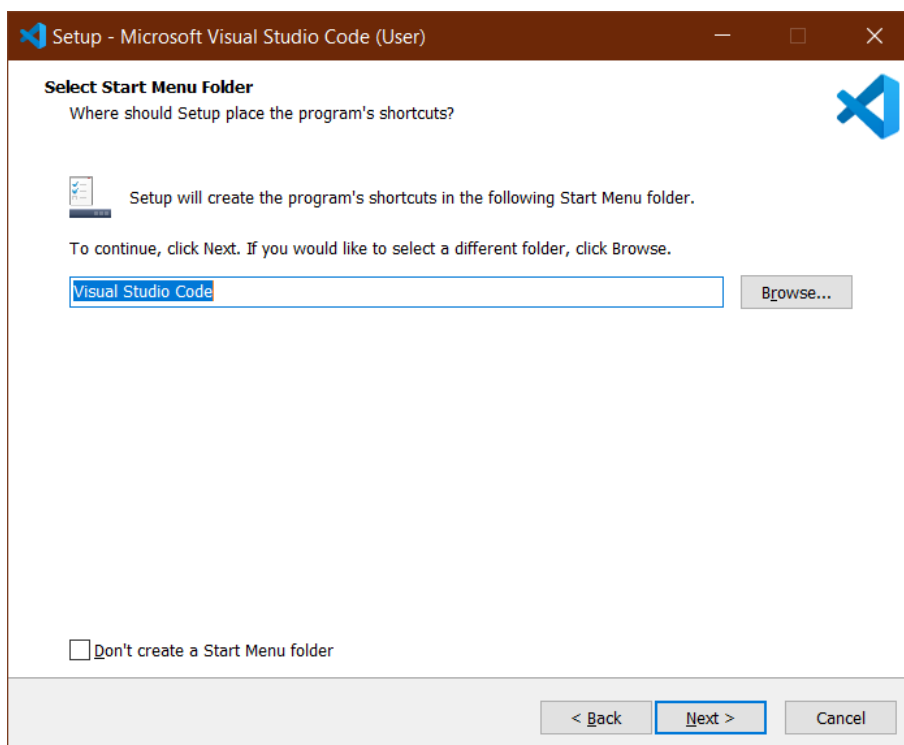




→ Klik op **Next**.



→ Klik op **Next**.





→ Selecteer alle opties onder **Other** en klik op **Next**.

Select Additional Tasks

Which additional tasks should be performed?

Select the additional tasks you would like Setup to perform while installing Visual Studio Code, then click Next.

Additional icons:

☐ Create a desktop icon

Other:

☒ Add "Open with Code" action to Windows Explorer file context menu

☒ Add "Open with Code" action to Windows Explorer directory context menu

☒ Register Code as an editor for supported file types

☒ Add to PATH (requires shell restart)

< Back Next > Cancel

→ Klik op **Install**.

Ready to Install

Setup is now ready to begin installing Visual Studio Code on your computer.

Click Install to continue with the installation, or click Back if you want to review or change any settings.

Destination location:
C:\Users\Ralph Gijsbrechts\AppData\Local\Programs\Microsoft VS Code

Start Menu folder:
Visual Studio Code

Additional tasks:

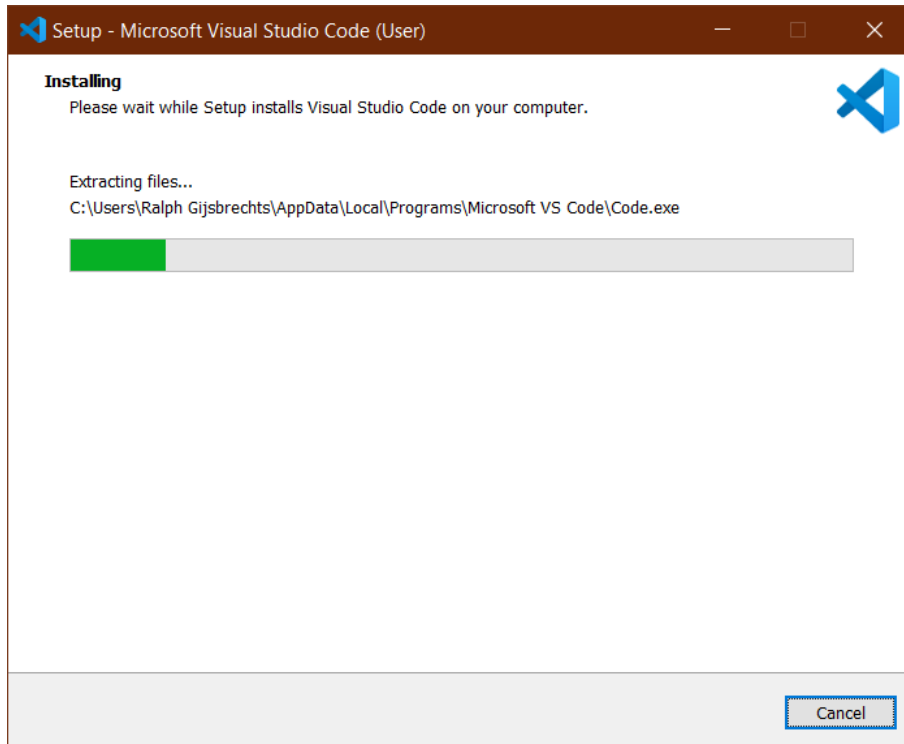
Other:

- Add "Open with Code" action to Windows Explorer file context menu
- Add "Open with Code" action to Windows Explorer directory context menu
- Register Code as an editor for supported file types
- Add to PATH (requires shell restart)

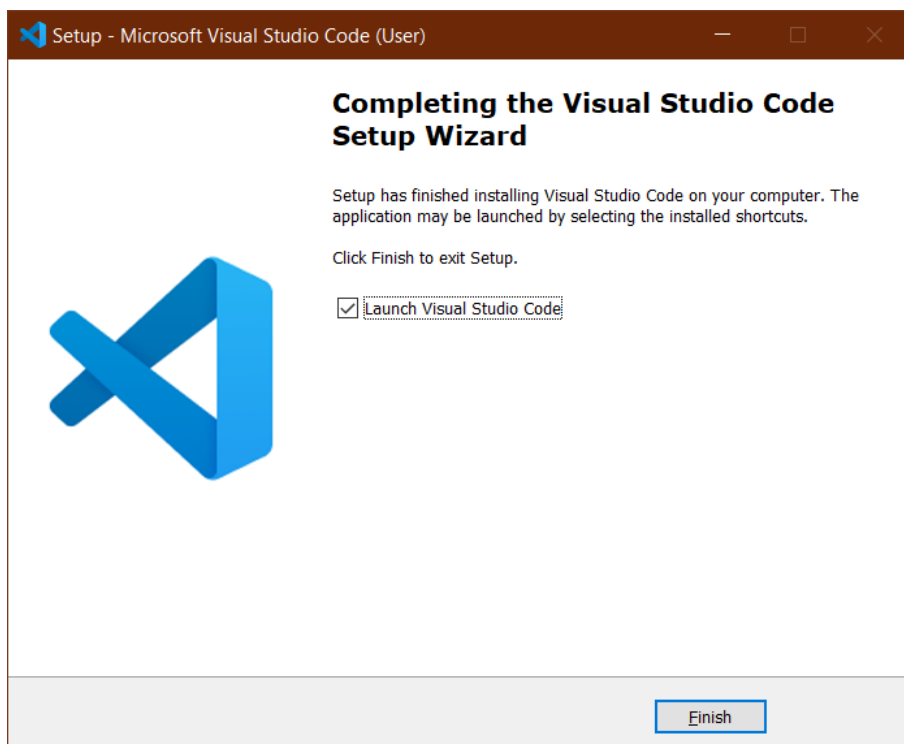
< Back Install Cancel



De software wordt nu geïnstalleerd.

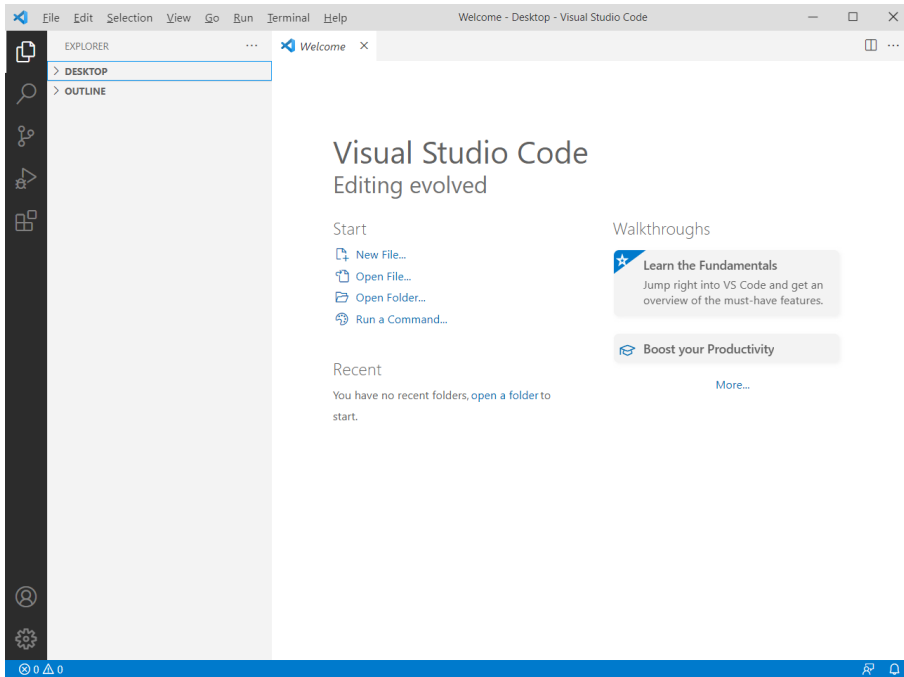


→ Klik op **Finish**.





Het programma wordt nu geopend en is klaar voor gebruik!



Installeren Live Preview

→ Klik op onderstaande link

<https://marketplace.visualstudio.com/items?itemName=ms-vscode.live-server>

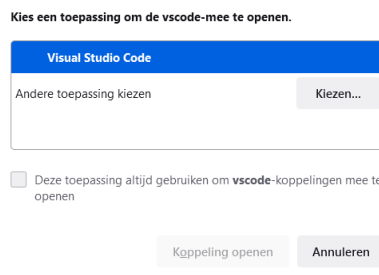
→ Klik op de groene knopje Install

Je krijg deze popup te zien:

→ Zorg dat Visual Studio Code staat geselecteerd en klik op:
Koppeling openen

Visual Studio Code opent zich

→ Klik op het blauwe knopje Install

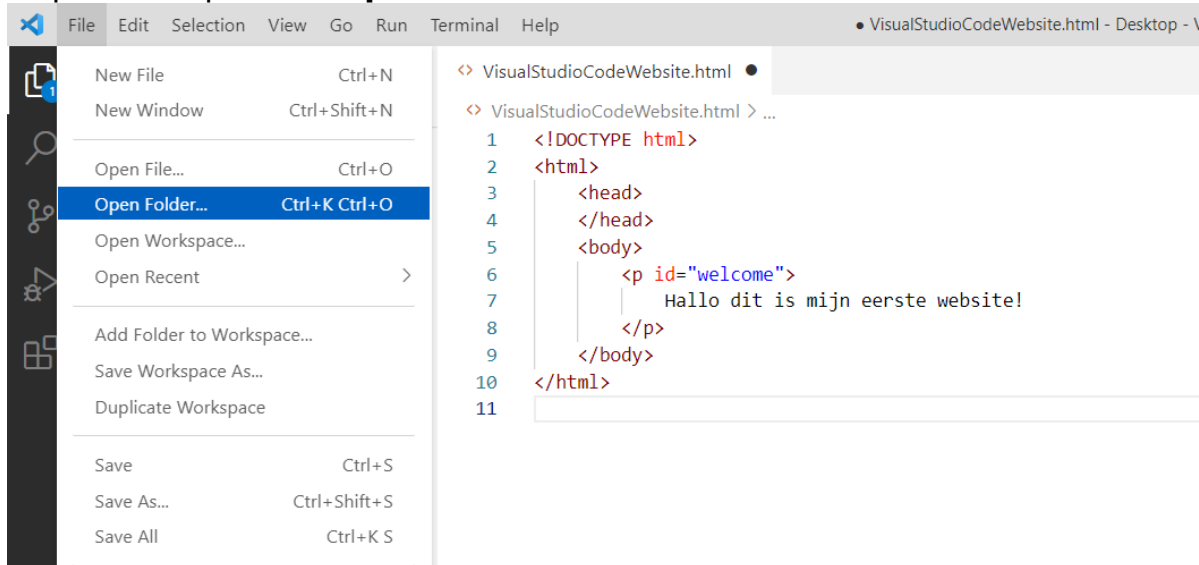




Mappenstructuur laden in Visual Studio Code

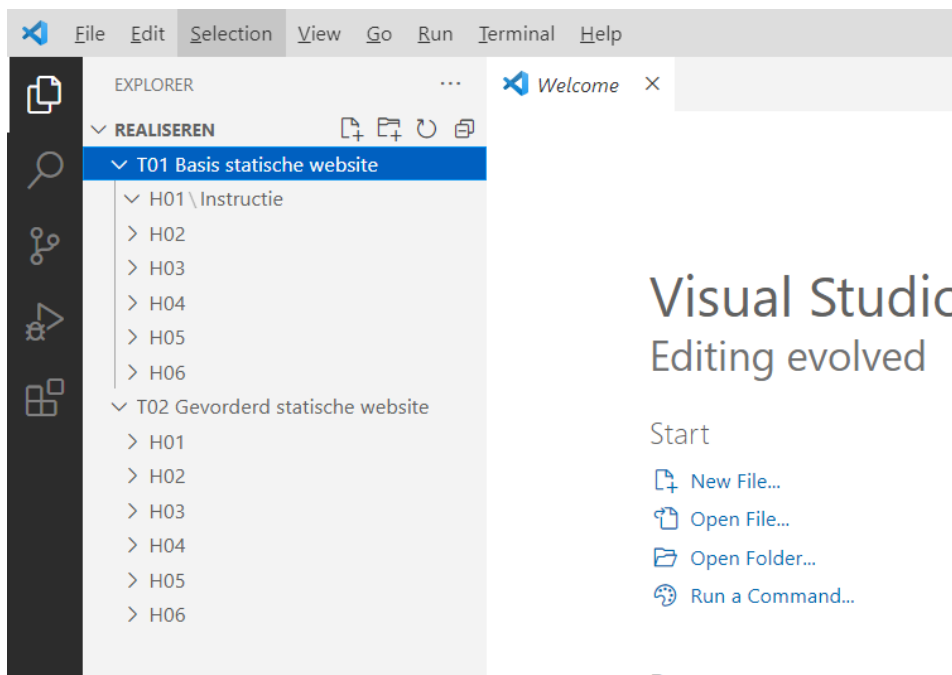
Stap 1: Open het programma Visual Studio code.

Stap 2: Klik op **File → Open Folder...**



Stap 3: Zoek de mappenstructuur op die je bij taak 2 van T1_REA_Oefening_1.1 gemaakt hebt en selecteer de map realiseren (1x klikken!). Klik vervolgens op de button met de tekst **Map selecteren**.

Stap 4: Je ziet nu dat aan de linkerkant van het programma je mappenstructuur zichtbaar is geworden. Je kunt nu je opdrachten hier steeds in uitwerken. De bestanden die je hier aanmaakt komen nu automatisch op je OneDrive ook te staan





Structuur

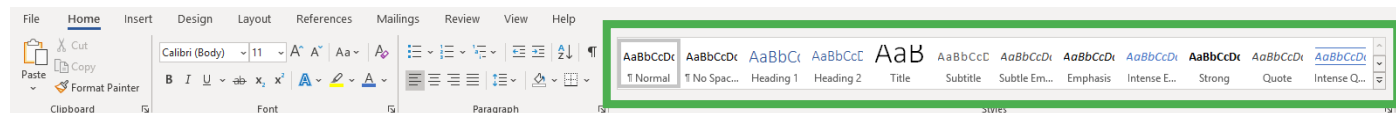
Denk eens na over de artikelen die je in een tijdschrift leest: voor elk artikel is er een kop, paragraaftekst, en eventueel een of meer afbeeldingen. Als het artikel lang is kan het zijn opgedeeld met tussenkoppen of zijn er citaten afgedrukt. Structuur helpt lezers de verhalen in de tijdschriften te begrijpen. Bij een website is dat niet anders. Kijk bijvoorbeeld maar eens op www.tweakers.net daar zie je dat de makers hetzelfde hebben gedaan!



Je ziet dat op de webpagina ook gebruik gemaakt is van een koptekst, paragraaftekst, links, afbeeldingen etc. Dit draagt allemaal bij aan de structuur van de website.

Structuur in Word

Teksten in Word kun je structureren door gebruik te maken van verschillende structuurbeschrijvingen. Zo kun je bijvoorbeeld in Word hoofdkoppen, subkoppen en paragrafen aanmaken om je tekst goed te structureren.



Je ziet dat iedere structuurbeschrijving er vaak net wat anders uit ziet. Echter is dat niet het enige wat je verandert. Je verandert namelijk ook de structuur van het document. Dat kun je het beste zien als je Word een inhoudsopgave laat maken. Die maakt gebruik van de structuur in het document. Zo wordt een tekst van het type hoofdkop of subkop standaard opgenomen in de inhoudsopgave. Als je een normale tekst groter maakt, of dikgedrukt maakt is dat dus geen structuurbeschrijving maar opmaak.



Structuur in Websites

Net zoals in Word kun je ook de inhoud van jouw website structureren. Nu maak je op een website geen inhoudsopgave, maar toch is het beschrijven van de structuur ook hier belangrijk. Zo kun je de website bijvoorbeeld gemakkelijker opmaken met CSS. Je kunt in onderstaande voorbeeld zien dat de zwarte tekst (de tekst die op de pagina te zien is) tussen de HTML-code (blauwe tekst) staat om de structuur van die tekst te beschrijven.

Structuur in Word:

Hoofdkop

Een paragraaf

Subkop niveau 2

Nog een paragraaf

Structuur in HTML-Code:

```
<h1>
    Hoofdkop
</h1>
<p>
    Een paragraaf
</p>
<h2>
    Subkop niveau 2
</h2>
<p>
    Nog een paragraaf
</p>
```

Structuur in browser:

Hoofdkop

Een paragraaf

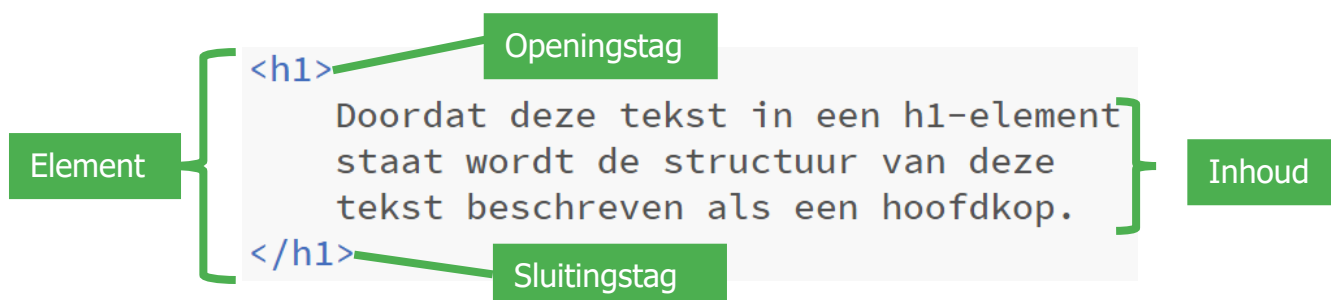
Subkop niveau 2

Nog een paragraaf

Elementen en tags

Structuur wordt in HTML beschreven door gebruik te maken van elementen. Elementen bestaan meestal uit 3 onderdelen:

1. Openingstag
2. De inhoud
3. Sluitingstag



Een element wordt geopend met een openingstag. Je herkent een openingstag aan de <>-tekens. Vervolgens wordt er een inhoud geplaatst. Een element wordt afgesloten met een sluitingstag. Je sluitingstag is hetzelfde als een openingstag, maar dan wordt er een /-teken geplaatst na het kleiner dan-teken (<).



Lege elementen

Er bestaan ook elementen die alleen een openingstag hebben. Deze elementen noemen we lege elementen. Zo bestaat er een element `
` om een tekst op een nieuwe regel te zetten. Dat ziet er dan als volgt uit:

```
<p>
  Deze zin staat op regel 1 <br>
  En deze zin staat op regel 2
</p>
```

Leeg element

Je ziet dat er dan geen sluitingstag gebruikt hoeft te worden. Dit komt omdat je aan het element geen inhoud kan mee te geven.

Attributen

Dan zijn er nog attributen. Attributen kom je alleen tegen in openingstags en geven extra informatie over elementen.

Attribuutnaam Attribuutwaarde

```
<p id="introduction">
  Inleidende tekst voor een artikel
</p>
```

De attribuutnaam geeft aan wat voor soort extra informatie over de inhoud van het element wordt gegeven. Het moet met kleine letters worden geschreven. De attribuutwaarde is de informatie over het attribuut of de instelling van het attribuut. Het moet tussen dubbele aanhalingstekens geplaatst worden. Verschillende attributen kunnen verschillende waarden bevatten. In bovenstaand voorbeeld wordt er een attribuut `id` toegevoegd aan een `p`-element (paragraaf). De waarde van het attribuut is `introduction`. Dit kunnen we later gebruiken in CSS om alleen het element op te maken met het `id` `introduction`. De voorkeur gaat uit naar Engelstalige attribuutwaarden.

De opbouw van HTML

Ieder webpagina bevat in de HTML-code een doctype, een html-element, een head-element, een title-element, meerdere meta-elementen en een body-element.

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

Op de volgende bladzijde wordt deze code verder toegelicht.



Element	Attribuut	Betekenis
html		Alles dat binnen het html-element geplaatst is wordt gelezen als HTML-code.
head		Voor het body-element zie je het head-element. Het head-element bevat informatie <i>over</i> de pagina. Deze informatie is niet zichtbaar in het hoofdvenster van de browser.
title		Het title element plaats je in het head element. De inhoud van het element <title> wordt ofwel bovenin de browser getoond, boven de plaats waar je normaal de URL van de pagina die je wilt bezoeken intypt, of op het tabblad voor die pagina (als je browser tabbladen heeft zodat je meerdere pagina's tegelijk open kunt hebben).
meta		Het element <meta> is een leeg element en bevindt zich binnen het head-element. Het bevat informatie over de webpagina. Bezoekers kunnen deze informatie niet zien. Je kunt hier bijvoorbeeld in opgeven welke karakterset je wilt gebruiken op de pagina, maar je kunt ook instellingen opgeven die ervoor zorgen dat je website op verschillende apparaten goed wordt weergegeven.
	charset	Met het charset-attribuut kun je aangeven welke set van karakters je wilt gebruiken op je webpagina. In HTML5 is dit standaard UTF-8. Klik hier om te zien welke karakters in met UTF-8 gebruikt kunnen worden.
	name	De name-attribuut en het content-attribuut zijn veel voorkomende attributen binnen het meta-element. De waarde van het name-attribuut is de naam van de instelling (eigenschap) die je wilt opgeven.
	content	Met het content-attribuut geef je de waarde op van de eigenschap die je bij het name-attribuut hebt opgegeven.
body		Alles binnen het body-element wordt in het hoofdvenster van de browser getoond. Dit is dus hetgeen dat de bezoeker te zien krijgt.

Doctype

In ieder HTML-bestand begin je met het opgeven van het type bestand. Er zijn verschillende versies en varianten van HTML. In iedere versie is een set van elementen opgenomen. Zo bestaat er bijvoorbeeld in HTML5 een header-element, maar in eerdere versies van HTML niet. Je moet dus eerst aangeven in welke versie van HTML je wilt gaan werken. Dat doe je door een doctype toe te voegen. Hieronder de verschillende doctypes van een aantal HTML versies.

HTML5 (gebruiken we in de les):

```
<!doctype html>
```

HTML 4.0.1:

```
<!doctype html public "-//w3c//dtd html 4.01 transitional//en"
"http://www.w3.org/TR/html4/loose.dtd">
```



XHTML 1.1:

```
<!doctype html public "-//w3c//dtd xhtml 1.1//en"  
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
```

Je ziet dat de doctype van HTML5 veel eenvoudiger is dan die van oudere versies.

Webpagina bestanden organiseren

Als je aan een nieuwe oefening begint maak je voor die oefening een map aan in het bijbehorende hoofdstuk. Je geeft die map de naam van de oefening. Voor de spaties en punten gebruik je een underscore (_). Voor oefening 1.3 maak je dan een map aan die oefening_1_3 heet. De startpagina geef je vervolgens altijd de naam index.html. Als er een bestand op een webserver staat die de naam index.html heeft, wordt dit namelijk standaard gezien als de startpagina. Maak in de map van de oefening ook onderstaande mappen aan indien van toepassing.

Onderdeel	Mapnaam
Webpagina's (behalve de index)	pages
JavaScript bestanden	scripts
Afbeeldingen	images
Lettertypes	fonts

Eventuele andere type bestanden worden in mappen geplaatst middels een zelfde soort naamgeving.



Je kunt nu **Oefening 1.2** maken.