

# INFORMATICA

Prof. Dr. Helga Naessens

Bachelor of Science in de  
industriële wetenschappen

Academiejaar 2021 – 2022





## Informatica (E701053)

Wegens Covid19 kan mogelijk afgeweken worden van de onderwijs- en evaluatievormen. Dergelijke afwijkingen zullen via Ufora worden gecommuniceerd.

**Cursusomvang** (nominale waarden; effectieve waarden kunnen verschillen per opleiding)

**Studiepunten 6.0**

**Studietijd 180 u**

**Contacturen**

**60.0 u**

### Aanbodsessies en werkvormen in academiejaar 2022-2023

A (semester 2)

Nederlands

Gent

werkcollege: PC-  
klasoefeningen  
online hoorcollege

30.0 u

30.0 u

### Lesgevers in academiejaar 2022-2023

Naessens, Helga

TW05

Verantwoordelijk lesgever

Brouns, Leen

TW05

Medewerker

Denert, Marleen

TW05

Medewerker

### Aangeboden in onderstaande opleidingen in 2022-2023

**stptn**

**aanbodsessie**

Bachelor of Science in de industriële wetenschappen (afstudeerrichting bouwkunde)

6

A

Bachelor of Science in de industriële wetenschappen (afstudeerrichting chemie)

6

A

Bachelor of Science in de industriële wetenschappen (afstudeerrichting  
elektromechanica)

6

A

Bachelor of Science in de industriële wetenschappen (afstudeerrichting elektronica-ICT)

6

A

Bachelor of Science in de industriële wetenschappen (afstudeerrichting informatica)

6

A

Bachelor of Science in de industriële wetenschappen (gemeenschappelijk gedeelte)

6

A

Educatieve Master of Science in de maatschappijwetenschappen (afstudeerrichting  
communicatiewetenschappen)

6

A

Master of Science in de communicatiewetenschappen (afstudeerrichting nieuwe media en  
maatschappij)

6

A

### Onderwijstalen

Nederlands

### Trefwoorden

Programmeren, Python, HTML, CSS, databanken, SQL, computerwetenschappen (P170),  
informatica (P175), computertechnologie (T120).

### Situering

Het doel van dit opleidingsonderdeel is:

- inzicht geven in de rol en de werking van computers en netwerken;
  - leren programmeren, logisch leren denken, taken leren opsplitsen in deeltaken;
- Eenzijds heeft dit opleidingsonderdeel een praktisch nut:
- het leert de mogelijkheden van computers, netwerken en applicaties optimaler benutten;
  - het leert de studenten zelf programma's maken voor bijvoorbeeld berekeningen, gegevensverwerking of simulaties;
  - het leert de studenten een eenvoudige website te maken met behulp van HTML en CSS;
  - het leert de studenten informatie op te halen, toe te voegen, aan te passen en te verwijderen uit een eenvoudige databank.

Anderzijds heeft dit opleidingsonderdeel een algemeen vormende waarde: het geeft inzicht in abstracte structuren en processen, het ontwikkelt het analytisch vermogen, het leert de student modulair denken, het leert om problemen zelf op te lossen en hiervoor correcte oplossingen te formuleren.

Deze theoretische kennis en de opgebouwde competenties zijn toepasbaar in allerlei andere domeinen (ontwerp, planning, optimalisatie, ...).

## Inhoud

Het opleidingsonderdeel bestaat uit een aantal onderdelen.

Deel 1 besteedt aandacht aan zowel hardware- als softwarebeginselen van de computerwerking:

- Hoe werkt een computer en hoe wordt hij bestuurd?
- Hoe worden gegevens opgeslagen, georganiseerd en gestructureerd? Wat zijn de belangrijkste bestandsformaten, bv. voor foto, audio en video?
- Hoe communiceren computers met mekaar? Hoe werkt het internet?

Deel 2 zet de eerste stappen tot algoritmeopbouw: het leert de studenten programmeren in Python. Hierbij komen onder andere volgende onderwerpen aan bod: basisprincipes van gestructureerd programmeren (variabelen, bewerkingen, operatoren, sequentie, selectie, herhaling) en van datastructuren en algoritmen (functies, strings, lists, tuples, dictionaries, iteratie, zoeken, werken met bestanden).

Daarnaast wordt er ook een korte inleiding gegeven tot het maken van webapplicaties. Wat zijn de principes achter populaire webapplicaties? Hoe worden webpagina's en webformulieren ontwikkeld?

Ten slotte wordt er ook een inleiding tot databanken gegeven, waarbij onder andere de volgende onderwerpen aan bod komen: principes en begrippen relationele databank (tabellen, kolommen, rijen, relaties) en eenvoudige SQL (met nadruk op SELECT).

## Begincompetenties

Wetenschappelijke basiscompetenties verworven in het secundair onderwijs.

## Eindcompetenties

- 1 Inzicht hebben in de architectuur, de werking en de componenten van een computer- en netwerksysteem.
- 2 Inzicht hebben in de voorstelling van enkele belangrijke gegevenssoorten zowel intern in de computer als extern.
- 3 De basisconcepten inzake programmeren in Python kennen en kunnen toepassen.
- 4 Een eenvoudig probleem kunnen analyseren en dit kunnen vertalen naar een computerprogramma.
- 5 Zelfstandig eenvoudige webpagina's kunnen maken.
- 6 Een gegevensbank via eenvoudige SQL-opdrachten kunnen bevragen of aanpassen.

## Creditcontractvoorwaarde

Toelating tot dit opleidingsonderdeel via creditcontract is mogelijk mits gunstige beoordeling van de competenties

## Examencontractvoorwaarde

Dit opleidingsonderdeel kan niet via examencontract gevolgd worden

## Didactische werkvormen

Werkcollege: PC-klasoefeningen, online hoorcollege

## Toelichtingen bij de didactische werkvormen

- Theorie (30u): hoorcollege en demonstratie. Tijdens de hoorcolleges wordt mede aan de hand van voorbeelden stap voor stap de theorie uitgelegd.
- Labo (30u): werkcollege (zelfstandig werk aan een individuele PC).

## Leermateriaal

- Nederlandstalige syllabus "Basiskennis Informatica, Webpagina's en Inleiding tot databanken", te bekomen via de studentenkring Hermes (geschatte kostprijs: 5 euro)
- Boek "Practice of Computing Using Python, William F. Punch and Richard Enbody, Pearson." Aankoop vrijblijvend, te bekomen via de studentenkring Hermes (geschatte kostprijs 65 euro)
- De lesgevers bieden slides, voorbeeldprogramma's en oefeningen aan via het elektronisch leerplatform.
- Ter aanvulling zijn boeken over de behandelde onderwerpen ter beschikking in de bibliotheek.

## Referenties

- Practice of Computing Using Python, William F. Punch and Richard Enbody, Pearson
- Think Python, Allen B. Downey, O'Reilly
- Learning Python, Mark Lutz, O'Reilly
- An introduction to Computation and Programming using Python, John V. Guttag, MIT Press

## Vakinhoudelijke studiebegeleiding

De lesgevers zijn beschikbaar voor meer uitleg langs de daarvoor voorziene kanalen (tijdens en/of na de lessen of op afspraak).

## Evaluatiemomenten

periodegebonden en niet-periodegebonden evaluatie

## Evaluatievormen bij periodegebonden evaluatie in de eerste examenperiode

Schriftelijk examen

## Evaluatievormen bij periodegebonden evaluatie in de tweede examenperiode

Schriftelijk examen

## Evaluatievormen bij niet-periodegebonden evaluatie

Participatie, werkstuk, vaardigheidstest

## Tweede examenkans in geval van niet-periodegebonden evaluatie

Examen in de tweede examenperiode is niet mogelijk

## Toelichtingen bij de evaluatievormen

- PE (theorie en labo): schriftelijk examen. Het examen is een praktisch examen, voornamelijk bestaande uit oefeningen, aangevuld met theoretische vragen
- NPE: webpagina, quotatie ingediende oefeningen in Dodona, vrijblijvende vaardigheidstest Python

## Eindscoreberekening

Schriftelijk examen: 90%

Webpagina en quotatie ingediende oefeningen in Dodona voor labo: 10%

Bij deelname aan de (vrijblijvende) test van Python, kan de student zelf beslissen of hij de behaalde punten al dan niet wenst in te zetten voor een bepaalde vraag van het examen.

Indien de student de examenvraag **niet oplost**, wordt voor deze vraag de behaalde score op de (desbetreffende vraag van de) test overgenomen.

Indien de vraag **wel opgelost** wordt, wordt de score van de (desbetreffende vraag van de) test niet ingezet (de punten voor (die vraag van) de test worden dus niet in rekening gebracht).

## Faciliteiten voor werkstudenten

te bespreken met de titularis



# INFORMATICA

## WEBPAGINA'S (HTML EN CSS)







# Inhoudsopgave

<b>1</b>	<b>HTML</b>	<b>1</b>
1.1	Inleiding . . . . .	1
1.1.1	Wat is HTML? . . . . .	1
1.1.2	Terminologie . . . . .	1
1.1.3	Structuur HTML-document . . . . .	2
1.2	Standaard HTML-elementen . . . . .	5
1.2.1	Tekst . . . . .	7
1.2.2	Afbeeldingen . . . . .	8
1.2.3	Hyperlinks . . . . .	9
1.2.4	Lijsten . . . . .	11
1.2.5	Tabellen . . . . .	14
1.3	Structuurelementen . . . . .	18
1.4	Formulieren . . . . .	21
1.4.1	Het <form>-element . . . . .	21
1.4.2	Formulieronderdelen . . . . .	23
<b>2</b>	<b>CSS</b>	<b>29</b>
2.1	Een eerste voorbeeld . . . . .	29
2.2	Typografie . . . . .	32
2.3	Kleur . . . . .	33

2.4	Selectors . . . . .	33
2.4.1	Het attribuut id . . . . .	34
2.4.2	Het attribuut class . . . . .	34
2.4.3	Pseudoklassen . . . . .	35
2.4.4	Elementen binnen een element selecteren . . . . .	36
2.5	Paginaopmaak . . . . .	36
2.5.1	Randen . . . . .	37
2.5.2	Marges . . . . .	38
2.5.3	Padding . . . . .	38
2.5.4	Display . . . . .	38
2.5.5	Zwevende elementen . . . . .	39
2.6	Lijsten . . . . .	41

# Hoofdstuk 1

## HTML

### 1.1 Inleiding

#### 1.1.1 Wat is HTML?

HTML is de taal waarin elke webpagina is geschreven. Ze bepaalt de structuur van de pagina: hoofding, alinea's, links naar andere pagina's, figuren, formulieren, ...

HTML staat voor *Hypertext Markup Language* en verwijst enerzijds naar *hypertext* en anderzijds naar *markup language*. Een *hypertext* is een tekst die verbonden is met andere documenten via koppelingen of links. De term *markup language* geeft aan dat het vastleggen van de structuur van een webpagina gebeurt door tekst te markeren. Zoals iemand tekst kan markeren met een fluostift, zo markeert HTML tekst met *tags* of *labels* (<>). In het onderstaande voorbeeld wordt de tag <strong> gebruikt om het woord Hypertext te benadrukken.

---

<strong>Hypertext</strong> is een tekst met links.

---

De weergave van dit stukje tekst is de taak van de browser (bv. Chrome, Firefox, Internet Explorer, ...) (zie onderstaande figuur).

**Hypertext is een tekst met links.**

#### 1.1.2 Terminologie

##### HTML-document

De bestandsnaam van een webpagina heeft de extensie .html (soms ook .htm). De inhoud van dit bestand noemen we het HTML-document. Een HTML-document is een tekstdocument dat bestaat uit

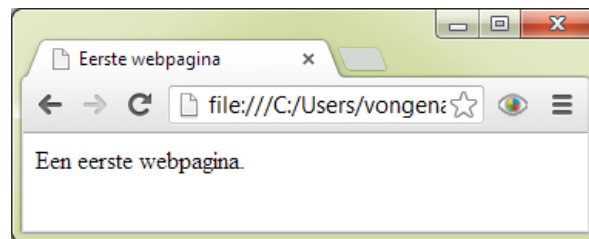
*HTML-elementen* en de eigenlijke *tekstinhoud* van de webpagina. Elk HTML-element begint met een *begintag* (<...>) en wordt meestal afgesloten met een gelijknamige *eindtag* (</...>). Een browser converteert de HTML-code naar een webpagina.

---

```
<!DOCTYPE html>
<html>
  <head>
    <title>Eerste webpagina</title>
  </head>
  <body>
    Een eerste webpagina.
  </body>
</html>
```

---

Listing 1.1: HTML-code



Figuur 1.1: Webpagina

## Tags

De basis van de HTML-syntax zijn *tags*. Een tag is een woord tussen de scherpe haken < en >. Tags komen meestal in koppels voor: een begintag (<...>) en een gelijknamige eindtag (</...>). De meeste tags zijn dus zoals haakjes: als je er één opent, dan moet je hem ook sluiten. Er zijn echter ook een aantal HTML-elementen die enkel een begintag hebben (en dus geen eindtag). Tags kunnen ook genest worden: een tag in een andere tag. In dat geval moet de laatst geopende tag eerst afgesloten worden.

---

```
<head><title>Eerste webpagina</title></head>
```

---

De inhoud tussen de begintag en eindtag samen met de beide tags noemen we een (HTML-)element. Hierboven staat dus het <head>-element. Dit element bevat het <title>-element

---

```
<title>Eerste webpagina</title>
```

---

De inhoud van het <title>-element is “Eerste webpagina”. Als een element geen inhoud heeft, dan wordt de eindtag vaak weggelaten.

### 1.1.3 Structuur HTML-document

De basisstructuur van een webpagina is altijd dezelfde:

---

```
<!DOCTYPE html>
<html>

<head>
  <title>Titel van de webpagina</title>
  <meta charset="UTF-8">
</head>

<body>
  Inhoud webpagina.
</body>

</html>
```

---

Listing 1.2: Basisstructuur HTML-document

De eerste regel `<!DOCTYPE html>` bepaalt het documenttype van de webpagina, namelijk HTML5. Hiermee geef je aan in welke versie van HTML het document opgesteld is. In dit voorbeeld en in de rest van deze cursus gebruiken we de laatste versie van HTML, namelijk HTML5.

In de vorige versie van HTML (HTML 4.01) was deze declaratie nog een stuk uitgebreider:

---

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
  "http://www.w3.org/TR/html4/strict.dtd">
```

---

Op de website <http://validator.w3.org/> nu kan je een webpagina valideren: controleren of ze voldoet aan de HTML5-standaard.

Op de tweede regel van een HTML-document staat steeds de tag `<html>`, die het begin van het HTML-document aangeeft. Op de laatste regel wordt het HTML-document afgesloten met `</html>`.

In de tag `<html>` is het ook mogelijk om aan te duiden in welke taal de webpagina opgesteld is. Hiervoor maak je gebruik van een taalcode bv. “nl” voor Nederlands, “nl-BE” voor de Vlaamse versie van het Nederlands, “en” voor Engels, ...

---

```
<html lang="nl">
```

---

Het opgeven van de taal van een HTML-document is nuttig voor zoekmachines, vertaalapplicaties, ...

`lang` noemen we een attribuut van het element `<html>`.

## Attributen

Attributen geven extra informatie over een element en bestaan uit een naam en een waarde. De syntax is als volgt:

---

```
naamAttribuut = "waarde attribuut"
```

---

Een attribuut moet uniek zijn binnen het element m.a.w. een element mag geen twee attributen hebben met dezelfde naam. De volgorde van de attributen in een element is onbepaald en vrij te kiezen. De waarden van een attribuut staat meestal tussen (enkele of dubbele) aanhalingstekens. De aanhalingsstekens mogen weggelaten worden als de waarde van het attribuut geen spaties bevat. Om verwarring en mogelijke fouten te vermijden raden we aan om steeds aanhalingstekens te schrijven.

## HTML-document

Het `<html>`-element bevat twee elementen: `<head>` en `<body>` die respectievelijk de hoofding en de inhoud van de webpagina bevatten.

## Hoofding

De hoofding van een webpagina bevat extra gegevens over de pagina, ook wel metadata genoemd, zoals een titel, trefwoorden, de gebruikte tekenverzameling en een beschrijving van de pagina.

---

```
<!DOCTYPE html>
<html lang="nl-BE">

<head>
  <title>Korfbalclub Floriant Merelbeke</title>
  <meta charset="UTF-8">
  <meta name="keywords" content="Floriant, korfbal, Merelbeke, sport">
  <meta name="description" content="Korfballen bij Floriant voor jong en oud,
    voor jongens en meisjes: sportief, recreatief, een gemengde ploegsport
    waar snelheid, behendigheid, samenspel en teamgeest belangrijk zijn">
</head>

<body>
  Webpagina korfbalclub Floriant te Merelbeke.
</body>

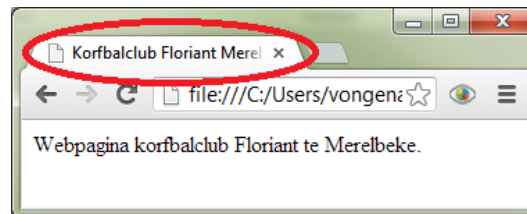
</html>
```

---

Listing 1.3: HTML-document met hoofding

**Titel** Het `<title>`-element geeft de webpagina een titel. Het is het enige onderdeel van de hoofding dat zichtbaar is in de browser, namelijk als titel van het tabblad (zie figuur 1.2). Je vindt het ook terug als

- naam van de bladwijzer voor de webpagina
- titel in de zoekresultaten van zoekmachines
- koptekst bij het afprinten van de pagina



Figuur 1.2: Titel van een webpagina

**Beschrijving** Een korte beschrijving van de webpagina kan ook deel uitmaken van de hoofding.

---

```
<meta name="description" content="Korfballen bij Floriant voor jong en oud,  
voor jongens en meisjes: sportief, recreatief, een gemengde ploegsport  
waar snelheid, behendigheid, samenspel en teamgeest belangrijk zijn">
```

---

Zoekmachines gebruiken beschrijvingen bij het bepalen van zoekresultaten. Een goeie omschrijving kan helpen om hoger te verschijnen in een zoekresultaat. Merk op dat het `<meta>`-element alleen uit een begintag bestaat. Aangezien dit element geen inhoud heeft, is er geen eindtag nodig. Dit element heeft wel attributen namelijk `name` en `content`.

**Tekenverzameling** De tekenverzameling bepaalt hoe de tekens van een document binair opgeslagen worden. Als je pagina's opmaakt in het Chinees of Arabisch dan zal je een andere tekenset gebruiken, dan voor pagina's in het Nederlands of het Engels. Voor die laatste pagina's is de meest aangewezen tekenverzameling "UTF-8".

---

```
<meta charset="UTF-8">
```

---

## 1.2 Standaard HTML-elementen

De inhoud van een webpagina vind je terug in het `<body>`-element van de HTML-code. Deze inhoud bevat o.a. tekst, afbeeldingen en hyperlinks. We bespreken de verschillende elementen aan de hand van een eenvoudige homepage voor de opleiding "Industrieel Ingenieur Informatica" (zie listing 1.4 en figuur 1.3).



Figuur 1.3: Homepagina opleiding



---

```

<!DOCTYPE html>
<html lang="nl-BE">

<head>
  <title>Toegepaste Ingenieurswetenschappen: Informatica |
    Academische Bachelor- en Masteropleidingen aan de Universiteit Gent</title>
  <meta charset="UTF-8">
  <meta name="keywords" content="Universiteit Gent, Toegepaste
    Ingenieurswetenschappen, Informatica, Industrieel Ingenieur, Gent">
  <meta name="description" content="Academische Bachelor- en Masteropleidingen
    aan de Universiteit Gent in Informatica">
</head>

<body>
  <h1>Toegepaste Ingenieurswetenschappen: Informatica</h1>
  <ul>
    <li><a href="#pijlers">Vier Pijlers</a></li>
    <li><a href="#tp">Theorie én praktijk</a></li>
  </ul>
  <h2 id="pijlers">Vier Pijlers</h2>
  <p>TIWI steunt op <strong>vier pijlers</strong>, die hieronder staan
    afgebeeld. Samen bereiden ze je uitstekend voor op een bloeiende carrière
    in de informatica.
  </p>
  
  <h2 id="tp">Theorie én praktijk</h2>
  <p>De opleiding TIWI biedt een uitgekiende balans tussen theoretisch
    fundament en praktische vaardigheden. Onze studenten
    spenderen zo'n 40% van hun tijd aan theorie. De overige 60% wordt
    verdeeld over praktische toepassing, en dit onder de vorm van individuele
    inoefening, groepswork en stage en eindwerk (<i>masterproef</i>).
  </p>
  <p>Neem voor meer informatie zeker een kijkje bij onze
    <a href="faq.html">vaak gestelde vragen</a>.
    Studiefiches kan je vinden op de website van
    onze <a href="http://tiwi.ugent.be/">faculteit</a>.
  </p>
</body>
</html>

```

---

Listing 1.4:

### 1.2.1 Tekst

HTML legt de structuur van een webpagina vast o.a. de opdeling in één of meerdere alinea's, kopteksten, ... .

Alinea's worden in HTML aangeduid met het `<p>`-element (*paragraph*).

---

```

<p>De opleiding TIWI biedt een uitgekiende balans tussen theoretisch fundament
en praktische vaardigheden. Onze studenten spenderen zo'n 40% van hun tijd aan theorie.
De overige 60% wordt verdeeld over praktische toepassing, en dit onder de vorm van individuele
inoefening, groepswork en stage en masterproef.
</p>

```

---

Koppen en subkoppen voor een webpagina maak je met de elementen `<h1>`, `<h2>`, ... , `<h6>` (*heading*).

Voor de belangrijkste kop, meestal de titel van de pagina, gebruik je `<h1>`. Een ondertitel wordt aangeduid met het element `<h2>` enz.

---

```
<h1>Toegepaste Ingenieurswetenschappen: Informatica</h1>
<h2>Vier Pijlers</h2>
<p>...</p>
<h2>Theorie én praktijk</h2>
<p>...</p>
```

---

Om tekst te benadrukken zijn er verschillende mogelijkheden: gewoon benadrukken met `<i>`, sterk benadrukken met `<strong>`, kernwoorden aanduiden met `<b>` en woorden aanduiden die de betekenis van een zin subtiel veranderen met `<em>`.

Het element `<i>` is bedoeld voor een stukje tekst met een andere toon of stemming dan de omringende tekst zoals bv. een technische term, een taxonomische aanduiding, een term in een vreemde taal, een dialectische term, ...

---

```
stage en eindwerk (<i>masterproef</i>)
```

---

Tekst die belangrijker is dan de omliggende tekst, die er moet uitspringen, wordt aangeduid met het element `<strong>`.

---

```
TIWI steunt op <strong>vier pijlers</strong>
```

---

## 1.2.2 Afbeeldingen

Afbeeldingen maken webpagina's aantrekkelijker. De meest gebruikte formaten voor afbeeldingen zijn jpeg, gif en png. Om een afbeelding toe te voegen maak je gebruik van `<img>`. Dit element bestaat alleen uit een begintag en heeft dus geen inhoud, maar wel enkel attributen. De attributen geven aan wat de locatie van de afbeelding is, welke tekst getoond moet worden als de afbeelding niet weergegeven kan worden, ...

---

```

```

---

De attributen `src` (*source*) en `alt` (*alternative*) zijn verplicht voor het `<img>`-element.

Het attribuut `src` bepaalt de locatie van de afbeelding. Mogelijke waarden zijn

**een relatief pad** Het bestand van de afbeelding bevindt zich in dezelfde map als het bestand van de webpagina of in een onderliggende map.

---

```


```

---

**een absoluut pad** Het bestand van de afbeelding bevindt zich op dezelfde server als de webpagina, maar in een andere map.

---

```

```

---

**een absolute URL** Een URL van een afbeelding op een andere server

---

```

```

---

Het attribuut `alt` speelt een rol als de browser de afbeelding niet kan of wil weergeven. In dat geval wordt de waarde van `alt` getoond.

Naast de twee verplichte attributen kan het `<img>`-element nog andere attributen hebben zoals `title`. Het attribuut `title` wordt gebruikt als tooltip voor de afbeelding (zie listing 1.5 en figuur 1.4).

---

```

```

---

Listing 1.5: Figuur met tooltip



Figuur 1.4: Figuur met tooltip

### 1.2.3 Hyperlinks

Hyperlinks (of links) zijn koppelingen tussen webpagina's die toelaten om te surfen op het web. Een link maak je met het `<a>`-element (*anchor*). De nieuwe pagina waarnaar de link verwijst vind je in het attribuut `href` (*hypertext reference*). Net zoals het `src`-attribuut van een `<img>`-element (zie 1.2.2) kan de waarde van het `href`-attribuut de volgende waarde hebben:

- een relatief pad

---

```
<a href="faq.html">vaak gestelde vragen</a>
```

---

- een absoluut pad

---

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

---

- een absolute URL

---

```
<a href="http://inwe.ugent.be/">faculteit</a>
```

---

Neem voor meer informatie zeker een kijkje bij onze [vaak gestelde vragen](#). Studiefiches kan je vinden op de website van onze [faculteit](#). Heb je nog meer vragen? Aarzel niet om [contact](#) op te nemen!

Figuur 1.5: Hyperlinks

Op de tekst tussen de begintag `<a>` en de eindtag `</a>` klik je om naar een nieuwe pagina te gaan (zie figuur 1.5). Je kan ook een afbeelding gebruiken als link. In dat geval is de inhoud van het `<a>`-element een `<img>`-tag:

---

```
<a href="http://www.ugent.be">
  
</a>
```

---

Het HTML-element `<a>` kan niet alleen gebruikt worden voor een link naar een andere webpagina, maar ook om links te voorzien binnen eenzelfde grotere webpagina. Grotere webpagina's voorzien meestal een inhoudsopgave: elk onderdeel van de webpagina krijgt een id toegewezen, die dan gekoppeld is aan een link in de inhoudsopgave. Dit wordt geïllustreerd in het onderstaande codefragment:

---

```
<ul>
  <li><a href="#pijlers">Vier Pijlers</a></li>
  <li><a href="#tp">Theorie én praktijk</a></li>
</ul>
<h2 id="pijlers">Vier Pijlers</h2>
<p>...</p>
<h2 id="tp">Theorie én praktijk</h2>
<p>...</p>
```

---

Om interne links te voorzien, geef je een waarde aan het `id`-attribuut van elk HTML-element dat overeenkomt met een onderdeel van de webpagina. Deze waarde moet uniek zijn binnen de webpagina, m.a.w. geen twee elementen in een zelfde HTML-document mogen dezelfde waarde hebben voor het `id`-attribuut. Bijvoorbeeld:

---

```
<h2 id="pijlers">Vier Pijlers</h2>
```

---

Vervolgens koppel je een link aan dit specifieke HTML-element door in het `href`-attribuut van de link `<a>` gebruik te maken van het `#`-teken gevolgd door de gegeven waarde aan het `id`-attribuut, bijvoorbeeld:

---

```
<a href="#pijlers">Vier Pijlers</a>
```

---

### 1.2.4 Lijsten

In HTML zijn er drie types opsommingen of lijsten:

- niet-genummerde lijsten
- genummerde lijsten
- definitielijsten

Een niet-genummerde lijst (*unordered list*) is een opsomming van een aantal items. Elk item start met een stip (*bullet*). In listing 1.6 en figuur 1.6 worden mogelijke studierichtingen opgelijst. De lijst zelf wordt gekenmerkt door het element `<ul>`, elk item in de lijst door het element `<li>`.

---

```
<ul>
  <li>bouwkunde</li>
  <li>chemie</li>
  <li>elektromechanica</li>
  <li>elektronica-ICT</li>
  <li>informatica</li>
</ul>
```

---

Listing 1.6: Niet-genummerde lijst

- bouwkunde
- chemie
- elektromechanica
- elektronica-ICT
- informatica

Figuur 1.6: Niet-genummerde lijst

Genummerde lijsten (*ordered list*) zijn volledig analoog. Ze worden beschreven door een `<ol>`-element. (Zie listing 1.7 en figuur 1.7)

---

```
<ol>
  <li>bouwkunde</li>
  <li>chemie</li>
  <li>elektromechanica</li>
  <li>elektrotechniek (afstudeerrichtingen elektrotechniek en automatisering)</li>
  <li>elektronica-ICT (afstudeerrichtingen elektronica en ICT)</li>
  <li>informatica</li>
</ol>
```

---

Listing 1.7: Genummerde lijst

1. bouwkunde
2. chemie
3. elektromechanica
4. elektrotechniek (afstudeerrichtingen elektrotechniek en automatisering)
5. elektronica-ICT (afstudeerrichtingen elektronica en ICT)
6. informatica

Figuur 1.7: Genummerde lijst

In een definitielijst (*definition list*) of begrippenlijst bestaat elk item uit twee delen: een begrip en de bijhorende uitleg. In listing 1.8 en figuur 1.8 wordt een lijst van TV-programma's met bijhorende beschrijvingen getoond. De lijst zelf is een `<dl>`-element en bestaat uit een opeenvolging van `<dt>`- en `<dd>`-elementen. Het eerste element bevat het begrip (*definition term*), het tweede de uitleg of omschrijving (*definition description*).

---

```

<dl>
  <dt>Canvascrack</dt>
  <dd>Gezellige, toegankelijke quiz onder leiding van Herman Van Molle.</dd>
  <dt>Band of brothers</dt>
  <dd>Prestigieuze tiendelige HBO-serie van Steven Spielberg en Tom Hanks
    over de belevenissen van de Amerikaanse 'Easy Company' tijdens
    de Tweede Wereldoorlog. De reeks won onder andere zes Emmy Awards. </dd>
  <dt>Breaking bad</dt>
  <dd>Amerikaanse serie over een leraar die drugsdealer wordt om zijn
    financiële problemen op te lossen. Maar van het een komt het ander
    en zo belandt hij helemaal in de criminaliteit.</dd>
  <dt>De bende van Wim</dt>
  <dd>Tiendelige reisreeks waarin acteur Wim Opbrouck samen met zijn
    vrienden, muzikant Jean Blaute en fotograaf Michiel Hendryckx,
    een lange motortocht maakt door Europa. Onderweg houden ze halt
    bij plaatsen die voor hen een speciale betekenis hebben.
    Plaatsen met een verhaal. </dd>
  <dt>Inspector Lynley Mysteries</dt>
  <dd>Britse misdaadreeks over de aristocratische Detective Superintendent
    Thomas Lynley en zijn 'working class' collega Detective Sergeant Barbara Havers. </dd>
</dl>

```

---

Listing 1.8: Definitielijst

Canvascrack  
Gezellige, toegankelijke quiz onder leiding van Herman Van Molle.

Band of brothers  
Prestigieuze tiendelige HBO-serie van Steven Spielberg en Tom Hanks over de belevenissen van de Amerikaanse 'Easy Company' tijdens de Tweede Wereldoorlog. De reeks won onder andere zes Emmy Awards.

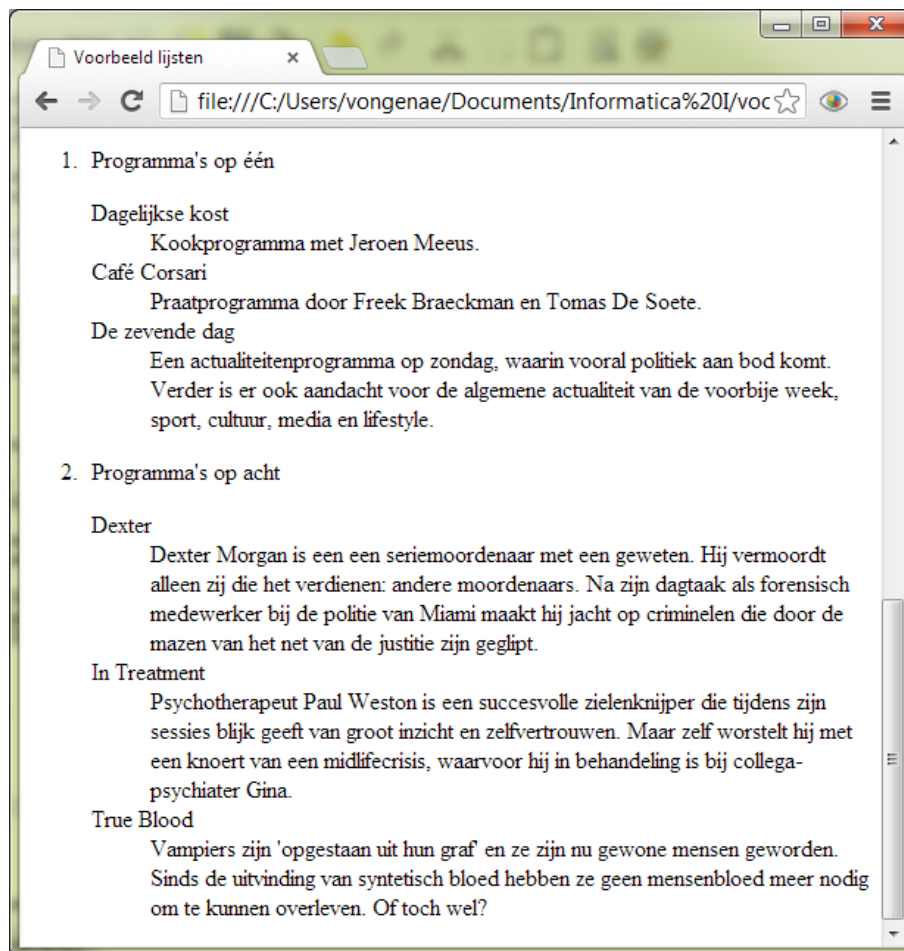
Breaking bad  
Amerikaanse serie over een leraar die drugsdealer wordt om zijn financiële problemen op te lossen. Maar van het een komt het ander en zo belandt hij helemaal in de criminaliteit.

De bende van Wim  
Tiendelige reisreeks waarin acteur Wim Opbrouck samen met zijn vrienden, muzikant Jean Blaute en fotograaf Michiel Hendryckx, een lange motortocht maakt door Europa. Onderweg houden ze halt bij plaatsen die voor hen een speciale betekenis hebben. Plaatsen met een verhaal.

Inspector Lynley Mysteries  
Britse misdaadreeks over de aristocratische Detective Superintendent Thomas Lynley en zijn 'working class' collega Detective Sergeant Barbara Havers.

Figuur 1.8: Definitielijst

Je kan lijsten ook nesten: een item in een lijst bestaat dan zelf ook uit een lijst (een lijst in een lijst). In listing 1.9 en figuur 1.9 wordt een genummerde lijst (een lijst van zenders) genest met een definitielijst (een lijst van programma's voor een zender met bijhorende uitleg).



Figuur 1.9: Geneste lijst

---

```

<ol>
  <li>Programma's op één
    <dl>
      <dt>Dagelijkse kost</dt>
      <dd>Kookprogramma met Jeroen Meeus.</dd>
      <dt>Café Corsari</dt>
      <dd>Praatprogramma door Freek Braeckman en Tomas De Soete.</dd>
      <dt>De zevende dag</dt>
      <dd>Een actualiteitenprogramma op zondag, waarin vooral politiek aan bod komt.
        Verder is er ook aandacht voor de algemene actualiteit van de voorbije week,
        sport, cultuur, media en lifestyle.</dd>
    </dl>
  </li>
  <li>Programma's op acht
    <dl>
      <dt>Dexter</dt>
      <dd>Dexter Morgan is een seriemoordenaar met een geweten. Hij vermoordt alleen
        zij die het verdienen: andere moordenaars. Na zijn dagtaak als forensisch
        medewerker bij de politie van Miami maakt hij jacht op criminelen die door de
        mazen van het net van de justitie zijn geglipt.</dd>
      <dt>In Treatment</dt>
      <dd>Psychotherapeut Paul Weston is een succesvolle zielenknijper die tijdens zijn
        sessies blijkt geeft van groot inzicht en zelfvertrouwen. Maar zelf worstelt
        hij met een knoert van een midlifecrisis, waarvoor hij in behandeling is bij
        collega-psychiater Gina.</dd>
      <dt>True Blood</dt>
      <dd>Vampiers zijn 'opgestaan uit hun graf' en ze zijn nu gewone mensen geworden.
        Sinds de uitvinding van syntetisch bloed hebben ze geen mensenbloed meer
        nodig om te kunnen overleven. Of toch wel?</dd>
    </dl>
  </li>
</ol>

```

---

Listing 1.9: Geneste lijst

### 1.2.5 Tabellen

Grotere hoeveelheden data of cijferdata worden vaak gepresenteerd in tabelvorm. Een tabel bestaat uit cellen die geordend zijn in een aantal rijen en kolommen (cfr. een matrix in de wiskunde). In HTML wordt een tabel gekenmerkt door het element `<table>`. Daarna worden de rijen gespecificeerd met een `<tr>`-element (*table row*). Per rij worden dan de gegevens van elke kolom aangeduid met het element `<td>` (*table data*). In listing 1.10 en figuur 1.10 worden vakken in tabelvorm getoond.



---

```

<table>
  <tr><td>Opleidingsonderdeel</td><td>Studiepnt.</td></tr>
  <tr><td>Wiskunde 1</td><td>6</td></tr>
  <tr><td>Mechanica 1</td><td>6</td></tr>
  <tr><td>Algemene chemie</td><td>6</td></tr>
  <tr><td>Elektriciteit</td><td>6</td></tr>
  <tr><td>Ontwerptools</td><td>3</td></tr>
  <tr><td>Materialen</td><td>3</td></tr>
  <tr><td>Wiskunde 2</td><td>6</td></tr>
  <tr><td>Elektrische en elektronische bouwstenen</td>
    <td>6</td></tr>
  <tr><td>Informatica 1</td><td>6</td></tr>
  <tr><td>Fysica 1</td><td>6</td></tr>
  <tr><td>Ingenieursproject</td><td>3</td></tr>
  <tr><td>Milieubeheer</td><td>3</td></tr>
</table>

```

---

Listing 1.10: Tabel

Opleidingsonderdeel	Studiepnt.
Wiskunde 1	6
Mechanica 1	6
Algemene chemie	6
Elektriciteit	6
Ontwerptools	3
Materialen	3
Wiskunde 2	6
Elektrische en elektronische bouwstenen	6
Informatica 1	6
Fysica 1	6
Ingenieursproject	3
Milieubeheer	3

Figuur 1.10: Tabel

In het bovenstaande voorbeeld wordt geen onderscheid gemaakt tussen de rij(en) die de hoofding van de tabel vormen en de rijen die de eigenlijke data bevatten. Je kan deze opsplitsing maken met de elementen `<thead>` (*table heading*) en `<tbody>` (*table body*). De verschillende kolommen in een hoofding worden niet aangeduid door het element `<td>`, maar door het element `<th>`. Listing 1.11 toont een aangepaste versie van de tabel met de vakken (figuur 1.11).

---

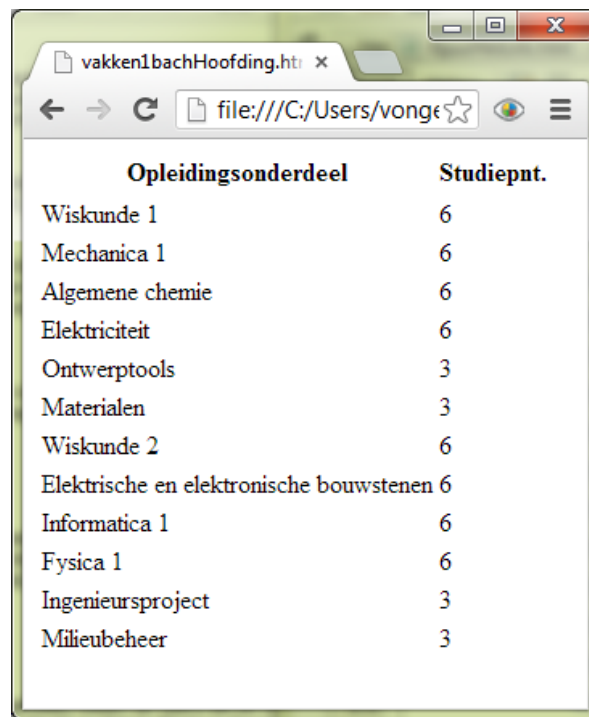
```

<table>
  <thead>
    <tr><th>Opleidingsonderdeel</th>
      <th>Studiepnt.</th></tr>
  </thead>
  <tbody>
    <tr><td>Wiskunde 1</td><td>6</td></tr>
    <tr><td>Mechanica 1</td><td>6</td></tr>
    <tr><td>Algemene chemie</td><td>6</td></tr>
    <tr><td>Elektriciteit</td><td>6</td></tr>
    <tr><td>Ontwerptools</td><td>3</td></tr>
    <tr><td>Materialen</td><td>3</td></tr>
    <tr><td>Wiskunde 2</td><td>6</td></tr>
    <tr><td>Elektrische en elektronische bouwstenen</td>
      <td>6</td></tr>
    <tr><td>Informatica 1</td><td>6</td></tr>
    <tr><td>Fysica 1</td><td>6</td></tr>
    <tr><td>Ingenieursproject</td><td>3</td></tr>
    <tr><td>Milieubeheer</td><td>3</td></tr>
  </tbody>
</table>

```

---

Listing 1.11: Tabel



Opleidingsonderdeel	Studiepnt.
Wiskunde 1	6
Mechanica 1	6
Algemene chemie	6
Elektriciteit	6
Ontwerptools	3
Materialen	3
Wiskunde 2	6
Elektrische en elektronische bouwstenen	6
Informatica 1	6
Fysica 1	6
Ingenieursproject	3
Milieubeheer	3

Figuur 1.11: Tabel

Cellen in een tabel samenvoegen tot grotere blokken kan met de attributen `colspan` en `rowspan` van een `<td>`- of `<th>`-element. De waarde van het attribuut `colspan`, respectievelijk `rowspan`, bepaalt hoeveel kolommen, respectievelijk rijen, samengenomen worden. Listing 1.12 toont een tabel met twee kolommen waarbij de eerste en laatste rij uit slechts één kolom bestaan. (zie figuur 1.12) In dit voorbeeld wordt ook een *footer* aan de tabel toegevoegd met het element `<tfoot>`.

---

```

<table>
  <thead>
    <tr><th colspan="2">Bachelor in de industriële wetenschappen: informatica</th></tr>
    <tr><th>Opleidingsonderdeel</th>
      <th>Studiept.</th></tr>
  </thead>
  <tbody>
    <tr><td>Wiskunde 1</td><td>6</td></tr>
    <tr><td>Mechanica 1</td><td>6</td></tr>
    <tr><td>Algemene chemie</td><td>6</td></tr>
    <tr><td>Elektriciteit</td><td>6</td></tr>
    <tr><td>Ontwerptools</td><td>3</td></tr>
    <tr><td>Materialen</td><td>3</td></tr>
    <tr><td>Wiskunde 2</td><td>6</td></tr>
    <tr><td>Elektrische en elektronische bouwstenen</td>
      <td>6</td></tr>
    <tr><td>Informatica 1</td><td>6</td></tr>
    <tr><td>Fysica 1</td><td>6</td></tr>
    <tr><td>Ingenieursproject</td><td>3</td></tr>
    <tr><td>Milieubeheer</td><td>3</td></tr>
  </tbody>
  <tfoot>
    <tr><td colspan="2">
      Meer informatie op de
      <a href="http://tiwi.be">website</a> van de vakgroep.
    </td></tr>
  </tfoot>
</table>

```

---

Listing 1.12: Tabel

Bachelor in de industriële wetenschappen: informatica	
Opleidingsonderdeel	Studiept.
Wiskunde 1	6
Mechanica 1	6
Algemene chemie	6
Elektriciteit	6
Ontwerptools	3
Materialen	3
Wiskunde 2	6
Elektrische en elektronische bouwstenen	6
Informatica 1	6
Fysica 1	6
Ingenieursproject	3
Milieubeheer	3
Meer informatie op de <a href="http://tiwi.be">website</a> van de vakgroep.	

Figuur 1.12: Tabel

## 1.3 Structuurelementen

Sinds HTML5 zijn er voor verschillende onderdelen van een webpagina HTML-elementen beschikbaar, zoals `<article>`, `<aside>`, `div`, `<figcaption>`, `<figure>`, `<footer>`, `<header>`, `<hgroup>`, `<mark>`, `<nav>`, `<section>`, `span` en `<time>`. Deze tags bepalen enkel de structuur van de pagina, maar niet de opmaak. Die wordt gerealiseerd in CSS (zie hoofdstuk 2). De onderstaande lijst geeft een korte beschrijving van de verschillende elementen. Listing 1.13 en figuur 1.13 illustreren het gebruik.

**article** Dit element omvat een deel van de webpagina die een afzonderlijke teksteenheid vormt. Bijvoorbeeld een nieuwsbericht, een artikel, een blogpost, ... Een `<article>`-element kan *heading*-elementen (`<h1>`, ...) bevatten, alinea's, ....

**aside** Dit element is bedoeld voor aanvullende inhoud m.a.w. tekst die zijdelings te maken heeft met de andere teksten op de webpagina. Bijvoorbeeld een citaat bij een interview.

**div** Dit element is specifiek bedoeld om te gebruiken in combinatie met CSS. Het wordt gebruikt om meerdere HTML-elementen te groeperen en daarvoor één stijl te definiëren in CSS. Bijvoorbeeld:

---

```
<div>
  <h3>...</h3>
  <p>...</p>
</div>
```

---

**figcaption** Dit element beschrijft een bijtekst voor een `figure`-element.

**figure** Dit element gebruik je om afbeeldingen, video's, maar ook stukjes code, songteksten, ... aan te duiden.

**footer** Een voettekst voor de webpagina of een onderdeel van de webpagina, bijvoorbeeld een artikel.

**header** Een koptekst voor de webpagina of een deel ervan.

**hgroup** Dit element groepeer verschillende *heading*-elementen (`<h1>`, ...) en eventueel andere elementen tot één hoofding.

**mark** Markeert een deel van de tekst.

**nav** Dit element legt een menu voor een webpagina vast. Een menu bestaat uit meerdere hyperlinks naar andere delen van de webpagina of naar andere pagina's op de website.

**section** Dit element laat toe om verschillende onderdelen van een webpagina inhoudelijk te groeperen. In listing 1.13 worden de verschillende nieuwsberichten gegroepeerd in één `section`-element.

**span** Ook dit element is specifiek voorzien om te gebruiken in combinatie met CSS. Het is bedoeld om een afwijkende stijl voor een aantal woorden te maken. Met behulp van het HTML-element `span` kan je deze woorden groeperen. Merk op: een `span`-element kan enkel voorkomen in een 'block-level'-element zoals `p`, `div`, `h1`, `h2`, ...



Figuur 1.13: Een webpagina met structuurelementen

**time** Dit element bepaalt een datum en/of tijdstip: éénmaal met het attribuut `datetime` en éénmaal als de tekstinhoud van het element. De waarde van het attribuut `datetime` moet de volgende vorm hebben `YYYY-MM-DD` voor een datum of `YYYY-MM-DDTHH:MM+HH:MM` voor een tijdstip. De tijd na het plusteken is het verschil met de UTC (*Coordinated Universal Time*).

---

```
<!DOCTYPE html>
<html>

<head>
  <title>Informatica aan de Universiteit Gent</title>
  <meta charset="UTF-8">
  <meta name="keywords" content="Universiteit Gent, Toegepaste
    Ingenieurswetenschappen, Informatica, Industrieel Ingenieur, Gent">
  <meta name="description" content="Academische Bachelor- en Masteropleiding
    aan de Universiteit Gent in Informatica">
</head>

<body>
  <header>
    <h1>Bachelor en Master in de industriële wetenschappen: Informatica</h1>
    <a href="http://www.ugent.be">
      <figure>
        <figcaption>Logo UGent</figcaption>
      </figure>
    </a>
  </header>
  <nav>
    <ul>
      <li><a href="#vacatures">Meer dan 9.000 ICT-vacatures niet ingevuld</a></li>
      <li><a href="#opendeur">Opendeurdag</a></li>
      <li><a href="#mp">Masterproef in de kijker</a></li>
    </ul>
  </nav>
  <section>
    <article>
      <header>
        <h2 id="vacatures">Meer dan 9.000 ICT-vacatures niet ingevuld </h2>
      </header>
      <p>In vier jaar tijd groeide de ICT-sector met bijna 10.000 mensen, met nog eens evenveel openstaande vacatures deze sector. Onze huidige studenten weten wat hen te wachten staat: werk, veel werk. Onze afgestudeerden voelen het ongetwijfeld ook: ervaring doorgeven en tegelijk zelf doorgroeien. Wij wensen hen allemaal veel succes en voldoening toe!</p>
    </article>
    <article>
      <header>
        <h2 id="opendeur">Opendeurdag</h2>
        <time datetime="2021-03-13">13 maart 2021</time>
      </header>
      <p>De Universiteit Gent verwelkomt je graag op haar jaarlijkse opendeurdag. Ook de vakgroep Informatica maakt je dan met plezier wegwijs door de opleiding TIWI. Kom gerust langs voor een blik achter de schermen en een fijne babbel met de docenten van de vakgroep. Tot dan!</p>
    </article>
  </section>
  <aside>
    <article>
      <h2 id="mp">Masterproef in de kijker</h2>
      <p>Vaak worden metadata toegevoegd aan multimediale bestanden, zodat deze gemakkelijker kunnen worden teruggevonden in databanken. In haar masterproef onderzoekt An De Moor technieken die kunnen bijdragen tot betere zoekresultaten. Door het gebruik van tag clouds en ontologieën, wordt bladeren en zoeken een fluitje van een cent. </p>
    </article>
  </aside>
</body>
</html>
```

```

        </article>
    </aside>
    <footer>
        Valentin Vaerwyckweg 1, 9000 Gent E-mail: info@tiwi.be
    </footer>
</body>
</html>

```

---

Listing 1.13: Voorbeeld gebruik structuurelementen

## 1.4 Formulieren

Op het web vind je allerlei formulieren terug: contactformulieren, inlogformulieren, formulieren om gebruikersgegevens in te voeren, ... Telkens wordt er informatie van de gebruiker door de browser doorgestuurd naar een webapplicatie op een webserver. De webapplicatie zal de info verwerken. Het verwerken van de informatie door een script of programma valt buiten het kader van deze cursus. Om het gebruik van formulieren toch te kunnen uittesten zullen we de gegevens via email verzenden.

HTML-formulieren bevatten typische GUI-elementen (*Graphic User Interface*) zoals drukknoppen, radioknoppen, tekstvelden en keuzelijsten, elk genoteerd als HTML-tags met bepaalde attributen. Zo heeft elk element, dat informatie verzamelt van de gebruiker, bijvoorbeeld een `name`-attribuut dat zijn naam bepaalt. De corresponderende waarde wordt dan door de gebruiker ingevuld of geselecteerd. Deze naam/waarde-paren worden in de regel pas naar de server doorgestuurd wanneer de gebruiker op een speciale *submitknop* drukt.

### 1.4.1 Het `<form>`-element

Listing 1.14 en figuur 1.14 toont een eenvoudige zoekformulier waar de gebruiker een voornaam en familienaam kan ingeven. De webapplicatie die op basis van deze gegevens een persoon zoekt, heeft als URL `http://www.ugent.be/nl/search`.

---

```

<form action="http://www.ugent.be/nl/search" method="GET">
    <label for="vnaam">Voornaam</label>
    <input id="vnaam" name="voornaam" type="text" />
    <label for="fnaam">Familienaam</label>
    <input id="fnaam" name="familienaam" type="text" />
    <input type="submit" value="Zoeken" />
</form>

```

---

Listing 1.14: Een eenvoudig Formulier

Een formulier wordt omsloten door `<form>` en `</form>`-tags. Daartussen worden de formuliercomponenten met passende HTML-tags aangeduid, namelijk `<label>` en `<input>`. Ook gewone HTML-elementen (hoofdingen, afbeeldingen, tabellen) mogen in een formulier worden gebruikt.



Figuur 1.14: Een eenvoudig Formulier

De `<form>`-tag bezit een aantal belangrijke attributen:

action	URL waarnaar de gegevens moeten worden verstuurd.
method	HTTP-aanvraagmethode: GET of POST

De waarde van het attribuut `method` wordt bepaald door de aard van het formulier. Het legt vast op welke manier met de webserver gecommuniceerd wordt.

**GET** Hiermee wordt aangegeven dat men een bepaalde bron wil raadplegen waarvan de inhoud door de webserver moet worden teruggestuurd. Een GET-aanvraag mag in principe echter nooit een verandering op de webserver als zij-effect hebben. (Dit betekent onder andere dat GET-aanvragen ook mogen beantwoord worden door proxy servers die de corresponderende gegevens hebben gecached.) Een voorbeeld hiervan is het opzoeken van een telefoonnummer, van informatie via een zoekmachine, ... De resultaten van vroegere zoekopdrachten mogen bewaard en hergebruikt worden bij een volgende opzoeking.

Bij deze methode worden de gegevens van het formulier aan het einde van de URL geplakt. In voorbeeld 1.14 werd als voornaam *Veerle* en familienaam *Ongena* ingetikt. De webpagina met de zoekresultaten zal dan de volgende URL hebben.

---

```
http://www.ugent.be/nl/search?voornaam=Veerle&familienaam=Ongena
```

---

**POST** Deze methode laat toe om bepaalde gegevens op de webserver aan te passen. Met POST kan je bijvoorbeeld gegevens toevoegen aan een databank of elektronische betalingen verrichten. In tegenstelling tot GET, wordt hier dus wel een zij-effect op de server verwacht en daarom worden POST-aanvragen doorgaans onderweg niet gecached.

Veranderen we in voorbeeld 1.14 de waarde van het attribuut `method` in POST, dan zal de URL van het zoekresultaat het volgende zijn.

---

```
http://www.ugent.be/nl/search
```

---

In dit geval worden de gegevens van het formulier dus niet toegevoegd aan de URL.



### 1.4.2 Formulieronderdelen

In listing 1.14 bestaat het formulier uit twee tekstvakken, twee labels en een knop. Net zoals de meeste GUI-elementen van een formulier worden de twee tekstvakken aangeduid met behulp van de `<input>`-tag. Deze tag kent onder andere de volgende attributen:

<code>type</code>	Geeft het soort GUI-element aan (knop, tekstveld, email, keuzelijst, enz.)
<code>id</code>	De unieke identificatie van dit element.
<code>name</code>	De naam van het invoerelement.
<code>value</code>	De initiële waarde van dit element.

Bij de tekstenvakken in voorbeeld 1.14 werd geen `value`-attribuut opgegeven omdat ze leeg mogen zijn.

Verder bevat het formulier twee labels. Een label is een stukje tekst voor een invoerelement, bv. een tekstvak, een keuzerondje, ... Het attribuut `for` legt de koppeling tussen het label en het invoerelement vast. De waarde van het `for`-attribuut verwijst naar de unieke identificatie van het invoerelement, namelijk de waarde van het `id`-attribuut. Die moet uniek zijn binnen de webpagina, m.a.w. geen twee elementen in een zelfde HTML-document mogen dezelfde waarde hebben voor het `id`-attribuut. Klikken op de tekst van het label heeft hetzelfde effect als klikken op het gekoppelde invoerelement. Bijvoorbeeld, klikken op het label *Voornaam* in voorbeeld 1.14 zal het eerste tekstvak selecteren.

We geven een kort overzicht van de verschillende GUI-elementen die met de `<input>`-tag worden aangeduid.

**`type="submit"`** Een submitknop. Wanneer je op deze knop drukt wordt de inhoud van het omsluitende formulier verstuurd naar de URL uit het `action`-attribuut van het formulier. Het `value`-attribuut verschijnt als opschrift op de knop.

**`type="image"`** Soms past het uiterlijk van een knop niet echt bij de rest van de layout van het formulier. Met deze `<input>`-tag kan je een eigen afbeelding introduceren als alternatief voor een submitknop. Behalve `name`- en `type`-attributen bevat deze tag ook nog het attribuut `src` met de URL van een afbeelding die moet worden getoond.

Net zoals bij de submitknop wordt de volledige inhoud van een formulier doorgestuurd op het moment dat je op de afbeelding klikt.

**`type="reset"`** Een resetknop. Wanneer je op deze knop drukt worden alle onderdelen van een formulier teruggezet op hun initiële waarden. (De informatie in het formulier wordt niet naar de server doorgestuurd.) Het `value`-attribuut bepaalt het opschrift van de knop.

**`type="text"`** Een tekstveld van één enkele lijn. Het `value`-attribuut van deze HTML-tag bevat de initiële inhoud van dit veld, maar het is de uiteindelijke inhoud van het tekstveld die als waarde naar de webserver wordt verstuurd. Met het attribuut `placeholder` kan je een suggestietekst in het tekstvak plaatsen. Deze tekst staat in het lichtgrijs en verdwijnt van zodra de gebruiker iets intikt. Andere attributen zijn `size`, die de zichtbare breedte van het tekstveld bepaalt (in lettertekens) en `maxlength` die aangeeft hoeveel lettertekens er maximaal in dit veld mogen worden

ingetikt. Is dit groter dan de breedte, dan zal de inhoud van het veld desnoods bij het intikken naar links of rechts worden verschoven.

**type="password"** Heeft dezelfde attributen als een tekstveld en gedraagt zich ongeveer gelijkwaardig. Het enige verschil is dat alle ingetikte lettertekens als sterretjes (of als bolletjes) op het scherm worden afgebeeld. Merk op dat de beveiling die je met dit soort velden bekomt slechts minimaal is. Wachtwoorden worden onversleuteld over het net verstuurd.

**type="email"** Een tekstveld waarin een emailadres ingevuld kan worden. Browsers zullen voor het versturen van de informatie naar de webserver controleren of de ingevoerde tekst een emailadres voorstelt. Indien dat niet het geval is, wordt er een foutbericht getoond en wordt er geen informatie naar de webserver gestuurd.

**type="url"** Een tekstveld waarin een URL ingevuld kan worden. Vergelijkbaar met het vorige element. De inhoud van het tekstveld moet nu een URL voorstellen.

**type="tel"** Een tekstveld waarin een telefoonnummer kan ingevuld worden. In tegenstelling tot het email- en urlveld wordt hier de structuur van de invoer niet gecontroleerd door de browser.

**type="checkbox"** Een checkbox is een element dat kan aan- of uitgevinkt worden. Enkel de naam/waarde-paren van de checkboxen die zijn aangevinkt, worden naar de webserver doorgegeven. Het `value`-attribuut heeft als defaultwaarde `on` en wordt niet op het scherm getoond. Soms worden verschillende checkboxen gebruikt met dezelfde naam maar met een andere waarde. In andere gevallen is de inhoud van dit attribuut eigenlijk niet belangrijk. Het attribuut `checked` (zonder gelijkheidsteken of waarde) geeft aan dat de checkbox reeds is aangevinkt wanneer het formulier voor het eerst op het scherm verschijnt.

---

```
<form method="POST" action="...">
  <input type="checkbox" name="veg"> Vegetarisch <br>
  <input type="checkbox" name="nrok" checked> Niet-roker <br>
  ...
</form>
```

---

**type="radio"** Net zoals checkboxen kunnen radioknoppen door de gebruiker worden aan- of uitgevinkt. De browser zorgt er echter voor dat er van een volledige groep radioknoppen slechts één tegelijkertijd is geselecteerd. Radioknoppen behoren tot dezelfde groep als ze hetzelfde `name`-attribuut hebben. Enkel de gegevens van de aangevinkte knoppen worden doorgestuurd. Het attribuut `checked` heeft dezelfde functie als bij checkboxen.

---

```
<form method="POST" action="...">
  <input type="radio" name="kamer" value="2" checked> 2-persoonskamer <br>
  <input type="radio" name="kamer" value="2bad"> 2-persoonskamer met bad<br>
  <input type="radio" name="kamer" value="1"> 1-persoonskamer <br>
  ...
</form>
```

---

**type="number"** Invoerveld voor een getal. Met de attributen `min` en `max` is het mogelijk om de minimale en maximale waarde van het getal vast te leggen. Met het attribuut `step` kan je bepalen dat de invoer een getal moet zijn uit een reeks van mogelijke waarden.

---

```
<input name="schoenmaat" type="number" min="29" max="50" step="1">
```

---

**type="time"** Invoerveld voor een tijdstip. Browsers tonen pijltjes in het invoervak om de tijd te verhogen of verlagen.

**type="date"** Invoerveld voor een datum. Net zoals bij het `time`-invoervak worden er pijltjes getoond om de waarde aan te passen. Bovendien bestaat ook de mogelijkheid om kalender te openen om de datum te selecteren.

**type="color"** Laat toe om een kleur te selecteren in een popup-venster.

Voor twee formulierelementen wordt er in plaats van `input` een eigen tag gebruikt: voor een tekstgebied (een tekstveld van meerdere lijnen) en voor een keuzelijst.

Je neemt een tekstgebied in je formulier op met behulp van de `<textarea>`-tag:

---

```
<textarea name="adresinfo" rows="5" cols="40">
  Naam Voornaam
  Straatnaam 42
  9000 GENT
</textarea>
```

---

Zoals gebruikelijk bevat het `name`-attribuut de naam van het element en van de bijhorende parameter. Wat de gebruiker in het veld intikt wordt teruggegeven als corresponderende waarde. De tekst die tussen de begin- en eindtags van de `textarea` staat, geldt als defaultwaarde. (Deze tekst wordt ook in het tekstveld ingevuld wanneer het formulier op het scherm verschijnt.) Andere attributen zijn `rows`, het aantal lijnen van het tekstveld, en `cols`, de breedte van het tekstveld uitgedrukt in aantal lettertekens.

Voor een keuzelijst gebruik je `<select>` en `<option>`.

---

```
<select name="fruit">
  <option value="A">Appel</option>
  <option value="P">Peer</option>
  <option value="S">Sinaasappel</option>
  <option value="K">Kokosnoot</option>
</select>
```

---

In bovenstaand voorbeeld wordt er een lijst afgebeeld van 4 fruitnamen waaruit de gebruiker er één kan kiezen. Als naam/waarde-paar wordt het `name`-attribuut van de omhullende `select` doorgestuurd (in ons geval `fruit`) en het `value`-attribuut van de gekozen optie (één van de letters A, P, S of K).

De `<select>`-tag ondersteunt een attribuut `multiple` (zonder waarde) die aangeeft dat er in de lijst ook tegelijkertijd meer dan één element mag worden aangeduid. In dit geval kunnen er dan meerdere naam/waarde-paren voorkomen met dezelfde naam.

Het `size`-attribuut voor `select` geeft het maximaal aantal elementen van de lijst aan dat tegelijkertijd wordt afgebeeld. Is het aantal opgegeven opties groter, dan worden scrollbars gebruikt.

---

```
<select name="fruit" size="1" multiple>
  <option value="A">Appel</option>
  <option value="P">Peer</option>
  <option value="S" selected>Sinaasappel</option>
  <option value="K">Kokosnoot</option>
</select>
```

---

Zoals je ziet kan de `<option>`-tag behalve het `value`-attribuut ook nog het attribuut `selected` dragen. Hiermee wordt de overeenkomstige optie per default geselecteerd.

In listing 1.15 en figuur 1.15 wordt het gebruik van invoerelementen geïllustreerd.

---

```
<form action="mailto:test@ugent.be" method="post">
  <fieldset>
    <legend>Persoonlijke gegevens</legend>
    <label for="naam">Naam:</label>
    <input id="naam" name="naam" type="text" placeholder="Voornaam Naam"><br>
    <label for="man">Man</label>
    <input id="man" name="geslacht" type="radio" value="M">
    <label for="vrouw">Vrouw</label>
    <input id="vrouw" name="geslacht" type="radio" value="V"><br>
    <label for="geboortedatum">Geboortedatum:</label>
    <input id="geboortedatum" name="geboortedatum" type="date" required><br>
    <label for="email">Email:</label>
    <input name="email" id="email" type="email" size="30"><br>
    <label for="telefoon">Telefoon:</label>
    <input name="telefoon" id="telefoon" type="tel"><br>
    <label for="website">Website:</label>
    <input name="website" id="website" type="url" value="http://www.ugent.be"><br>
    <label for="land">Land:</label>
    <select name="land" id="land">
      <option value="BE">België</option>
      <option value="DE">Duitsland</option>
      <option value="FR">Frankrijk</option>
      <option value="LU">Luxemburg</option>
      <option value="NL">Nederland</option>
    </select><br>
  </fieldset>
  <fieldset>
    <legend>Weetjes</legend>
    <input name="nrok" type="checkbox" id="nrok" checked>
    <label for="nrok">Niet-roker</label><br>
    <label for="kleur">Lieve-lings-kleur:</label>
    <input name="kleur" id="kleur" type="color"><br>
    <label for="schoenmaat">Schoenmaat:</label>
    <input name="schoenmaat" id="schoenmaat" type="number" min="29" max="50" step="1"><br>
    Interesses:
    <label for="sport">Sport</label>
    <input id="sport" name="interesses" type="checkbox" value="sport">
    <label for="nieuws">Actualiteit</label>
    <input id="nieuws" name="interesses" type="checkbox" value="actualiteit">
    <label for="muziek">Muziek</label>
    <input id="muziek" name="interesses" type="checkbox" value="muziek"><br>
    <label for="spreuk">Spreuk van de dag:</label>
    <textarea id="spreuk" name="spreuk" cols="30" rows="5"></textarea><br>
    <label for="slaaptijd">Slaaptijd om:</label>
    <input id="slaaptijd" name="slaaptijd" type="time">
  </fieldset>
  <input type="reset" value="Reset">
  <input type="submit" value="Verzenden">
</form>
```

---

Listing 1.15: Een formulier

The image shows a web browser window with a form. The browser's address bar shows a file path: `file:///C:/Helga/Info/cursus/2022/cursush`. The form is divided into two sections. The first section, titled 'Persoonlijke gegevens', contains the following fields: 'Naam' (with a placeholder 'Voornaam Naam'), 'Man' and 'Vrouw' radio buttons, 'Geboortedatum' (with a placeholder 'mm / dd / yyyy'), 'Email', 'Telefoon', 'Website' (with a placeholder 'http://www.ugent.be'), and 'Land' (a dropdown menu showing 'België'). The second section, titled 'Weetjes', contains: a checked checkbox for 'Niet-roker', a 'Liefelingskleur' color picker (showing black), a 'Schoenmaat' spinner, checkboxes for 'Sport', 'Actualiteit', and 'Muziek', a 'Spreuk van de dag' text area, and a 'Slaaptijd om:' time picker. At the bottom of the form are two buttons: 'Reset' and 'Verzenden'.

Figuur 1.15: Een formulier

In listing 1.15 worden `fieldset`-elementen gebruikt om het formulier op te delen in verschillende stukken met samenhangende elementen. Met het element `legend` kan je een titel geven aan een deel van het formulier bepaald door een `fieldset`-element.

Merk op dat het `action`-attribuut van het `form`-element nu niet verwijst naar de URL van een webapplicatie, maar naar een emailadres.

---

`mailto:test@ugent.be`

---

In dit geval zal er bij het klikken op de submitknop een mailvenster geopend worden (als er een mailclient geïnstalleerd is op de computer van de gebruiker). Het bovenstaande emailadres wordt ingevuld als bestemming en de waarden van het formulier vormen de inhoud van het mailbericht.



# Hoofdstuk 2

## CSS

HTML legt de structuur van een webpagina vast. De opmaak en layout van een webpagina wordt gedelegeerd naar een stijlblad. Een stijlblad bevat één of meerdere stijkenmerken (zoals kleur, plaats, lettertype, grootte, ...) voor HTML-elementen. Een stijlblad wordt beschreven in CSS (*Cascading Style Sheets*) en bewaard in een tekstbestand met extensie `.css`.

### 2.1 Een eerste voorbeeld

We hernemen listing 1.4 (figuur 1.3), een eenvoudige homepagina voor de opleiding Industrieel Ingenieur Informatica, en voegen er een stijlblad `tiwi.css` (zie listing 2.1) aan toe.

Om de koppeling te maken te maken tussen het HTML-bestand en het CSS-bestand moet je een `<link>`-element toevoegen in het `<head>`-element.

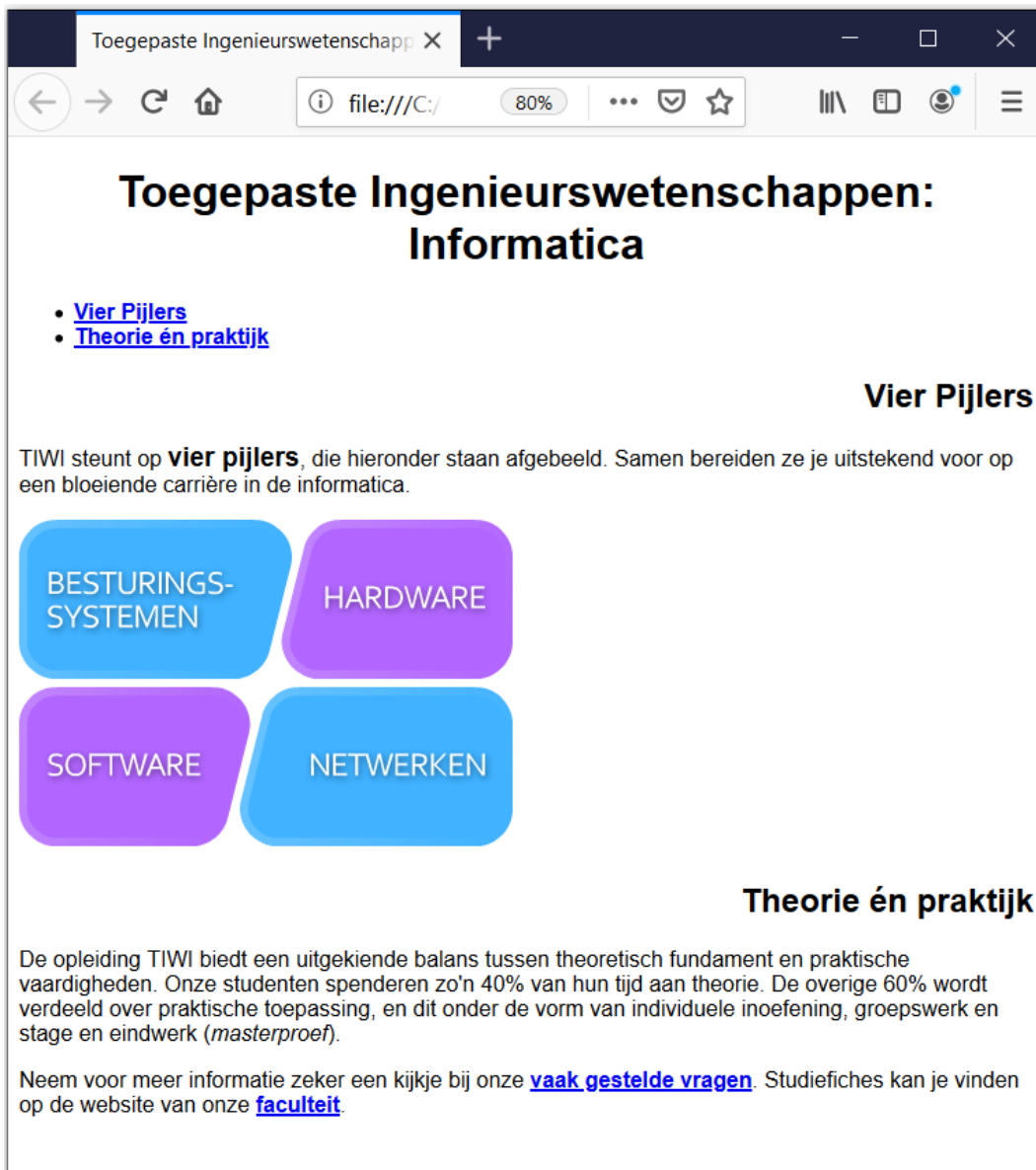
---

```
<head>
  <title>Toegepaste Ingenieurswetenschappen: Informatica |
    Academische Bachelor- en Masteropleiding aan de Universiteit Gent</title>
  ...
  <link rel="stylesheet" href="tiwi.css">
</head>
```

---

Het `<link>`-element definieert een koppeling naar een ander document. Als we het gebruiken om te koppelen naar een stijlblad, dan heeft het de volgende attributen:

- het attribuut `rel` beschrijft de relatie tussen het HTML-document en het stijlblad (altijd `stylesheet` voor een stijlblad);
- het attribuut `href` geeft de URL van het stijlblad, meestal een absoluut of een relatief pad. (zie 1.2.2)



Figuur 2.1: Eenvoudige homepage met stijlblad



---

```

/* font webpagina */
body {
    font-family: arial;
}
/* belangrijkste titel centreren*/
h1 {
    text-align: center;
}
/* ondertitel rechts aligneren*/
h2 {
    text-align: right;
}
/* belangrijke tekst iets groter */
strong {
    font-size: 120%;
}
/* hyperlinks vet */
a {
    font-weight: bold;
}

```

---

Listing 2.1: Eenvoudig stijlblad

Het resultaat in een browser wordt getoond in figuur 2.1.

Uit listing 2.1 kan je afleiden dat een stijlblad bestaat uit verschillende voorschriften (*rules*) van de volgende vorm:

---

```

selector {
    stijlkenmerk: waarde;
}

```

---

De `selector` bepaalt op welke HTML-onderdelen het stijlvoorschrift van toepassing is. In het voorbeeld zijn de selectoren namen van HTML-elementen.

De combinatie van een stijlkenmerk en een waarde wordt een *declaratie* genoemd. Voor één selector kan je meerdere declaraties opgeven.

---

```

selector {
    declaratie;
    declaratie;
    ...
}

```

---

In het voorbeeld hebben de voorschriften betrekking op het lettertype (*font*) en de alignatie van de tekst. In commentaar (`/* ... */`) staat het beoogde resultaat van de regel.

Indien verschillende elementen dezelfde CSS-eigenschap(en) hebben, kan je dit ook verkort noteren. Bijvoorbeeld

---

```

h1 {text-align: center;}
h2 {text-align: center;}
h3 {text-align: center;}

```

---

wordt in verkorte notatie:

---

```
h1, h2, h3 {text-align: center;}
```

---

Merk op dat het lettertype in listing 2.1 enkel gespecificeerd wordt voor het `body`-element. Alle kindelementen van `body` erven deze eigenschap over. De meeste CSS-eigenschappen worden overgeërfd door de kindelementen, behalve wanneer deze eigenschappen intrinsiek enkel voor één element bedoeld zijn bv. de hoogte van een element.

## 2.2 Typografie

De typografie van een pagina wordt o.a. bepaald door het lettertype, de lettergrootte, de regelafstand, ... We geven hier een beknopt overzicht van een aantal CSS-eigenschappen die de typografie van een pagina bepalen.

eigenschap	waarden	omschrijving
text-align	left, right, center, justify	Uitlijning van een tekst. Standaard is een tekst links uitgelijnd. Je kan een tekst centreren ( <code>center</code> ), rechts uitlijnen ( <code>right</code> ) of uitvullen ( <code>justify</code> ) door de eigenschap <code>text-align</code> in te stellen.
font-family	Arial, "Comic Sans MS", "Courier New", Georgia, ...	Lettertype van de tekst. Dit kan een lijst van lettertypes zijn gescheiden door komma's. Het eerst beschikbare lettertype wordt gebruikt. Als de naam van het font een spatie bevat, staat het tussen aanhalingstekens. Het laatste lettertype kan ook een <i>algemeen lettertype</i> zijn. In dit geval wordt het lettertype bepaald door de configuratie van de browser. Er zijn vijf algemene lettertypes: serif, sans-serif, cursive, fantasy en monospace.
font-size	xx-small, x-small, medium, large, x-large, xx-large, larger, smaller, <i>lengte, percentage</i>	De grootte van het lettertype. De waarden <code>xx-small</code> , ..., <code>xx-large</code> zijn relatief t.o.v. van het lettertype en de computer waarop de browser draait. De waarden <code>larger</code> en <code>smaller</code> maken gebruik van de vorige waarden en geven aan dat het element een stap groter of kleiner is dan het omliggende element. Het is ook mogelijk om de grootte van het lettertype uit te drukken in een lengte-eenheid of een percentage. Een veel gebruikte lengte-eenheid is <i>em</i> . De grootte van 1 em correspondeert met de grootte van het gekozen lettertype. Een alternatief is het gebruik van een percentage. 100% komt dan overeen met 1 em.
font-style	normal, italic, oblique	Deze eigenschap geeft aan of de tekst al dan niet schuingedrukt is.
font-variant	normal, small-caps	Als deze eigenschap ingesteld wordt op <code>small-caps</code> dan worden alle letters vervangen door kleine hoofdletters.
font-weight	normal, bold	Deze eigenschap laat toe om tekst in vetjes te tonen.

eigenschap	waarden	omschrijving
line-height	normal, <i>getal</i> , <i>lengte</i> , <i>percentage</i>	Regelafstand. Als de waarde van deze eigenschap een getal is (zonder lengtemaat of percentage) dan stelt het een vermenigvuldigingsfactor voor. De waarde 1.5 betekent dan anderhalve keer de fontgrootte.

## 2.3 Kleur

Een kleur in CSS wordt bepaald door de hoeveelheden rood, groen en blauw (RGB). Elke hoeveelheid wordt aangegeven door een getal tussen 0 en 255. Hoe groter het getal hoe lichter de kleur. De kleur rood is dus de combinatie van 255 rood, 0 groen en 0 blauw. Er zijn verschillende notaties om dit vast te leggen.

**rgb(255,0,0)** De kleur wordt bepaald door 3 getallen tussen 0 en 255.

**rgb(100%,0%,0%)** De kleur wordt bepaald door 3 percentages. Elk percentage stelt een deel van de waarde 255 voor.

**#FF0000** De kleur wordt bepaald door 3 hexadecimale getallen tussen 00 en FF.

Naast de RGB-notatie zijn er nog zestien kleuren in CSS die een naam hebben en waarvoor je dus niet de RGB-code hoeft te gebruiken: aqua (lichtblauw), black, blue, fuchsia, gray, green, lime, maroon (roodbruin), navy, olive, purple, red, silver, teal (blauwgroen), white en yellow.

De CSS-eigenschap `color` is de voorgrondkleur of tekstkleur van een element. De achtergrondkleur wordt vastgelegd met de eigenschap `background-color`. De standaardwaarde voor de achtergrondkleur is `transparent`, transparant. Dit betekent dat de kleur van het onderliggende element getoond wordt. De onderstaande instelling kleuren de achtergrond van de webpagina grijs en de tekst donkerblauw.

---

```
body {
    background-color: silver;
    color: navy;
}
```

---

## 2.4 Selectors

Tot nu toe werd de stijl voor de HTML-elementen bepaald op basis van hun naam (`p`, `body`, ...) m.a.w. de selector in een CSS-voorschrift was de naam van een element. Als meerdere elementen dezelfde stijl hebben dan kan je ze groeperen in een klasse. Individuele elementen kunnen een andere opmaak krijgen op basis van een unieke identificatie. Eén element kan ook verschillende verschijningsvormen hebben met elk een andere layout. Hiervoor gebruik je pseudoklassen. Tot slot is het ook mogelijk om met een selector elementen binnen een ander element te selecteren.

### 2.4.1 Het attribuut id

Met het attribuut `id` geef je een unieke identificatie aan een HTML-element. Deze identificatie mag uiteraard maar één keer voorkomen in de webpagina. Het laat toe om een specifieke stijl voor dit ene element vast te leggen. De waarde van dit attribuut mag geen spaties bevatten. De waarde moet ook beginnen met een letter, dus een getal als waarde kan niet.

Als je een `id` wilt gebruiken als selector, dan moet je voor de waarde een “#” plaatsen.

In listing 2.3 en 2.2 wordt enkel het `p`-element dat de copyrighttekst bevat kleiner en schuin getoond. Om dit te verwezenlijken wordt aan dit specifieke `p`-element een unieke identificatie toegekend.

---

```
<p id="copyright">&copy; Vakgroep Informatica, Universiteit Gent</p>
```

---

Listing 2.2: Id specificeren in HTML

---

```
#copyright {  
    font-size: small;  
    font-style: italic;  
}
```

---

Listing 2.3: Id in CSS

### 2.4.2 Het attribuut class

Om voor een aantal HTML-elementen (in de `body` van een webpagina) een zelfde stijl te definiëren kan je een klasse definiëren. Dit realiseer je door ze allemaal hetzelfde attribuut `class` te geven. Ook hier zijn spaties niet toegestaan. Als je met een selector alle elementen van een bepaalde klasse wil selecteren, dan start de selector met een punt (“.”) gevolgd door de naam van de klasse (d.i. de waarde van het `class`-attribuut).

In listing 2.5 en 2.4 wordt een klasse `waarschuwing` gebruikt om bepaalde elementen in vetjes te tonen en rood te kleuren.

---

```
<p class="waarschuwing">Bewaar geneesmiddelen en huishoudproducten buiten het bereik  
van kinderen.</p>  
  
<p>De meeste ongevallen gebeuren thuis, tijdens het gebruik van producten! Hou kinderen op  
afstand wanneer u <b class="waarschuwing">gevaarlijke producten</b> gebruikt zoals  
<span class="waarschuwing">ontstoppers, oplosmiddelen </span>...  
</p>
```

---

Listing 2.4: Klasse specificeren in HTML

---

```
.waarschuwing {  
    color: red;  
    font-weight: bold;  
}
```

---

Listing 2.5: Klasse in CSS

Je kan een klasse ook gebruiken om een selectie op basis van een HTML-naam te beperken. Zo zal de selector

---

`p.waarschuwing`

---

enkel die `p`-elementen selecteren waarvan het `class`-attribuut de waarde `waarschuwing` heeft.

### 2.4.3 Pseudoklassen

Sommige elementen, zoals bijvoorbeeld een hyperlink, kunnen verschillende verschijningsvormen hebben. Een link naar een pagina die je al bezocht hebt ziet er vaak anders uit dan een link die je nog niet aangeklikt hebt. Soms verandert de link als je er met de muis over beweegt of er op klikt. In CSS is het mogelijk om voor deze vier situaties een andere stijl te specificeren. Hiervoor zijn er vier pseudoklassen voorzien in CSS. Pseudoklassen herken je aan de dubbelpunt (:) in de selector.

**a:link** Een niet-bezochte link.

**a:visited** Een bezochte link. (Bv. een link waarop reeds geklikt is)

**a:hover** Een link die aangewezen wordt. (de muis beweegt erover)

**a:active** De actieve (net gekozen) link.

Sommige pseudoklassen kan je ook gebruiken voor andere HTML-elementen, bijvoorbeeld `li:hover` is de selector voor de lijstelementen waarover de muis beweegt.

Als je meerdere regels in een stijlblad hebt die van toepassing zijn op hetzelfde element, dan geldt het volgende:

- Een specifiekere selector heeft voorrang op een algemenere selector. Dit betekent bv. dat als je voor het `body`-element een stijleigenschap instelt en diezelfde eigenschap een andere waarde geeft voor een `p`-element, dan geldt voor het `p`-element de waarde die je instelde bij het `p`-element.
- Als beide selectors even specifiek zijn dan geldt het laatste genoemde CSS-voorschrift.

De volgorde waarin je de eigenschappen van de pseudoklassen van het `a`-element beschrijft in een stijlblad is dus belangrijk omdat het laatstgenoemde voorschrift voorgaande voorschriften overschrijft. Een link waarboven je beweegt is ook een al dan niet bezochte link is. Om het resultaat van de stijlkenmerken bij `a:hover` te zien moeten die dus na de stijlkenmerken van `a:visited` of `a:link` gespecificeerd worden. De meest aangewezen volgorde is dus:

- `a:link`
- `a:visited`
- `a:hover`
- `a:active`

#### 2.4.4 Elementen binnen een element selecteren

Soms wil je de layout kunnen vastleggen voor een element binnen een ander element. Stel dat je bijvoorbeeld een menu `nav` maakt gebruik makend van een lijst `ul`, en dat je wil dat de opmaak voor de lijst in het menu `nav` anders is dan de opmaak van andere `ul`-lijsten op de website (die zich dus niet in een `<nav>`-element bevindt). Dan moet je dus CSS-eigenschappen instellen voor elementen `ul` die zich bevinden in het `<nav>`-element. De selector voor de lijst is dan:

---

```
nav ul { ... }
```

---

Deze selector betekent: alle `ul`-elementen in een `nav`-element. De `ul`-elementen moeten dus tussen de begin- en eindtag van het `<nav>`-element voorkomen, maar ze hoeven geen direct kindelement te zijn, wat dus betekent dat er zich tussen het HTML-element `nav` en het HTML-element `ul` nog andere HTML-elementen mogen bevinden..

Indien je enkel de `ul`-elementen wenst te selecteren die directe kindelementen zijn van een `nav`-element, waarbij tussenliggende HTML-elementen dus niet toegestaan zijn, dan moet er nog een “>”-teken vermeld worden tussen het ouderelement en het kindelement, bijvoorbeeld:

---

```
nav > ul { ... }
```

---

### 2.5 Paginaopmaak

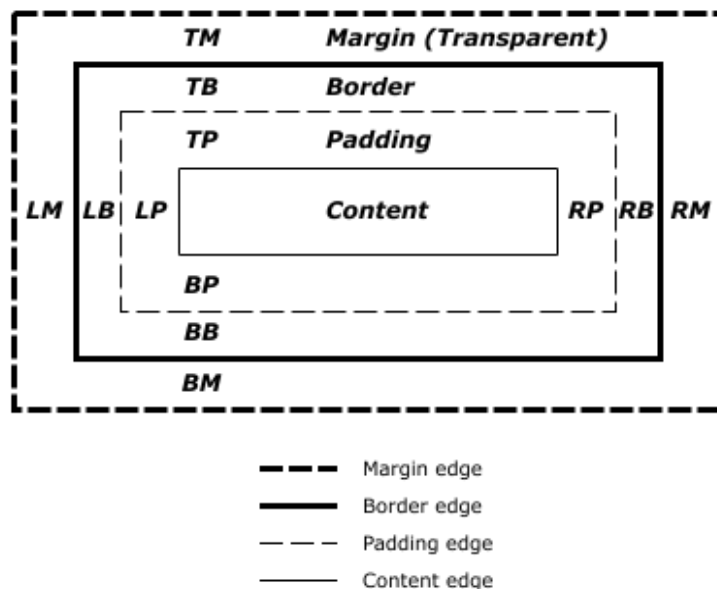
Elk HTML-element op een webpagina wordt weergegeven in een rechthoek (*box*). Dit noemt men het *boxmodel*. Dit model voorziet rond elke inhoud drie lagen: (zie ook figuur 2.2);

**padding** De witruimte tussen het element en zijn rand.

**border** De rand van het element.

**margin** De witruimte tussen de rand en andere elementen.

De afkortingen TM, LM, BM, RM, TB ... in figuur 2.2 staan respectievelijk voor *top margin*, *left margin*, *bottom margin*, *right margin*, *top border*, ...



Figuur 2.2: CSS-boxmodel (bron: W3C)

### 2.5.1 Randen

De rand van een HTML-element wordt bepaald door drie CSS-eigenschappen:

**border-width** de dikte van de rand

**border-color** de kleur van de rand

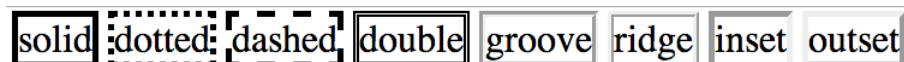
**border-style** de stijl of het type van de rand

De dikte van de rand kan je uitdrukken in absolute of relatieve eenheden.

**absolute eenheden** px (pixels), mm, ...

**relatieve eenheden** em, ... Het is ook mogelijk om een relatieve lijndikte te bepalen met de sleutelwoorden: `thin`, `medium` en `thick`

Voor het type van de rand zijn er acht verschillende opties: `solid`, `dotted`, `dashed`, `double`, `groove`, `ridge`, `inset` of `outset`. De verschillende types worden geïllustreerd in figuur 2.3.



Figuur 2.3: Randtypes

De stijl van de verschillende delen van een rand (links (*left*), rechts (*right*), boven (*top*) en onder (*bottom*)) kunnen ook apart ingesteld worden. Hiervoor voeg je aan de eigenschappen `border-width`, `border-color` of `border-style` het deel toe waarvan je de stijl wilt vastleggen.

Bijvoorbeeld: `border-top-width`, `border-bottom-width`, `border-left-color`, ...

### 2.5.2 Marges

Standaard zal een browser rond bijna alle elementen een marge (*margin*) voorzien. De marge van een element is de witruimte tussen het element en de elementen boven, links, rechts en onder. Om de marges zelf in te stellen maak je gebruik van de eigenschap `margin`. De verschillende delen van een marge worden apart geconfigureerd met de eigenschappen `margin-top`, `margin-right`, `margin-bottom` en `margin-left`.

Om marges in te stellen worden zowel absolute eenheden (bv. `px`) als relatieve eenheden (bv. percentages van de breedte van het venster) gebruikt.

### 2.5.3 Padding

Padding is de ruimte tussen de inhoud van het element en zijn rand. De corresponderende CSS-eigenschappen zijn `padding`, `padding-top`, `padding-right`, `padding-bottom` en `padding-left`.

De waarde van deze eigenschappen wordt meestal uitgedrukt in pixels of `em`, maar kan ook een percentage van het omliggende element zijn.

### 2.5.4 Display

In een webpagina worden inline-elementen (bv. `<img>`, tekst, ...) naast elkaar getoond en block-elementen (`<p>`, `<li>`, `<figure>`, ...) onder elkaar. Dit betekent dat standaard de CSS-eigenschap `display` voor inline-element de waarde `inline` heeft en voor block-elementen de waarde `block`.

Om één of meerdere block-elementen naast elkaar te laten staan, kan je de CSS-eigenschap `display` aanpassen naar de waarde `block-inline`. De inhoud van het element wordt opgemaakt als een block-element, maar het element zelf wordt als inline-element getoond.

In listing 2.6 zijn de `figure`-elementen block-elementen. Normaal worden die onder elkaar getoond in een webpagina (zie figuur 2.4). Als we de CSS-eigenschap `display` aanpassen naar de waarde `block-inline` (zie listing 2.7), dan worden ze naast elkaar afgebeeld (zie figuur 2.5).



---

```
<h1>Fotoalbum</h1>
  <p>Geef uw waardering voor onderstaande foto's:</p>
  <figure>
    <figcaption>tijgers.jpg</figcaption>
  </figure>
  <figure>
    <figcaption>koala.jpg</figcaption>
  </figure>
  <figure>
    <figcaption>papegaai.jpg</figcaption>
  </figure>
  <p>Vul het onderstaande formulier in.</p>
```

---

Listing 2.6: Voorbeeld block-elementen

## Fotoalbum

Geef uw waardering voor onderstaande foto's:



Vul het onderstaande formulier in.

Figuur 2.4: Foto's onder elkaar

---

```
figure {
  display: inline-block;
  margin: 5px;
}

img {
  width: 150px;
}

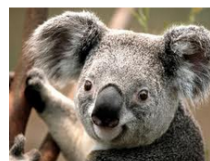
figcaption {
  text-align: center;
}
```

---

Listing 2.7: Aanpassen display block-elementen

## Fotoalbum

Geef uw waardering voor onderstaande foto's:



Vul het onderstaande formulier in.

Figuur 2.5: Foto's naast elkaar

### 2.5.5 Zwevende elementen

Om tekst naast of rond een foto te laten vloeien, moet je de foto zwevend maken. Dit betekent dat een element zover mogelijk naar boven en dan naar links of naar rechts zweeft in de huidige container en dat de ruimte aan de vrije kant beschikbaar is voor een ander element. Dit kan je realiseren met de

CSS-eigenschap `float`. Mogelijke waarden zijn `left` (naar links zweven), `right` (naar rechts zweven) of `none` (niet zweven).

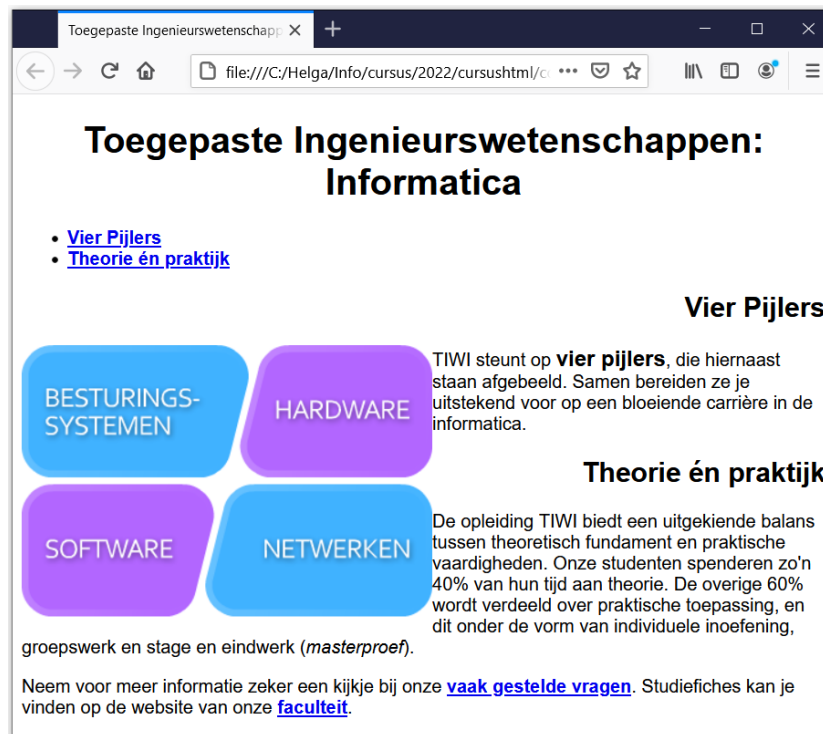
In figuur 2.1 staat de afbeelding onder de tekst “TIWI steunt ... informatica”. Om de afbeelding links (zie figuur 2.6) van de tekst te plaatsen, moet de volgende HTML aangepast worden:

---

```
<p>
  TIWI steunt op <strong>vier pijlers</strong>, die hieronder staan
  afgebeeld. Samen bereiden ze je uitstekend voor op een bloeiende carrière
  in de informatica.
</p>

```

---



Figuur 2.6: Zwevende figuur links

Eerst plaats je de afbeelding binnen hetzelfde `<p>-blok` (de container) als de tekst. De bovenstaande HTML-code wordt dan:

---

```
<p>
  
  TIWI steunt op <strong>vier pijlers</strong>, die hiernaast staan
  afgebeeld. Samen bereiden ze je uitstekend voor op een bloeiende carrière
  in de informatica.
</p>
```

---

Vervolgens voeg je in het bijhorende stijlblad de volgende regel toe om de figuur naar links te plaatsen:

---

```
img { float: left; }
```

---

Met behulp van de eigenschap `clear: both;` kan er voor gezorgd worden dat er naast een bepaald HTML-element geen zwevende elementen mogen voorkomen. Indien je dus de webpagina zoals in figuur 2.7 wilt bekomen, voeg je in het bijhorende stijlblad volgende regel toe:

---

```
h2 { clear: both; }
```

---

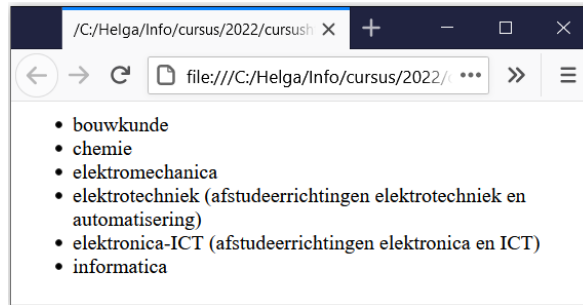


Figuur 2.7: Zwevende figuur links, clear voor h2

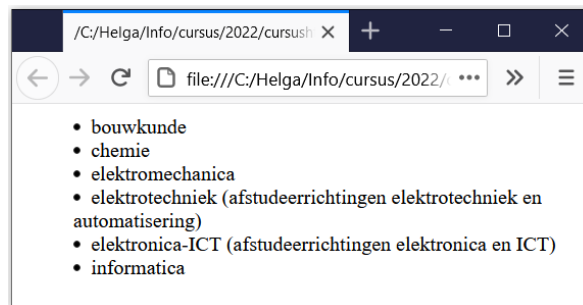
## 2.6 Lijsten

Voor de verschillende lijsttypes in HTML zijn er een aantal specifieke CSS-eigenschappen waarmee je het uiterlijk van de lijst kan aanpassen.

**list-style-position** De standaardwaarde voor `list-style-position` is `outside` (zie figuur 2.8). Dit betekent dat het opsommingsteken (bv. een bolletje) buiten het tekstblok staat. Figuur 2.9 illustreert wat er gebeurt als je deze eigenschap instelt op `inside`.



Figuur 2.8: Lijst met standaardwaarde voor `list-style-position`



Figuur 2.9: Lijst met `list-style-position` gelijk aan `inside`

**list-style-type** Deze eigenschap bepaalt de stijl van de lijst. Mogelijk stijlen zijn onder andere

**disc** Het opsommingsteken is een bolletje (•).

**circle** Het opsommingsteken is een kleine niet opgevulde cirkel (○).

**square** Het opsommingsteken is een klein opgevuld vierkantje.

**decimal** Het opsommingsteken is een getal: 1, 2, 3, ...

**lower-alpha** Het opsommingsteken is een kleine letter: a, b, c, ...

**lower-greek** Het opsommingsteken is een kleine Grieks letter:  $\alpha$ ,  $\beta$ ,  $\gamma$ , ...

**lower-roman** Het opsommingsteken is een romeins getal in kleine letters: i, ii, iii, ...

**upper-alpha** Het opsommingsteken is een hoofdletter: A, B, C, ...

**upper-greek** Het opsommingsteken is een Griekse hoofdletter: A, B,  $\Gamma$ , ...

**upper-roman** Het opsommingsteken is een romeins getal: I, II, III, ...

**list-style-image** Deze eigenschap laat toe om het opsommingsteken te vervangen door een afbeelding. Bv. `list-style-image: url(grijzeDriekhoek.png);` (zie figuur 2.10)

- › bouwkunde
- › chemie
- › elektromechanica
- › elektronica-ICT
- › informatica

Figuur 2.10: Lijst met figuur als opsommingsteken