

Efficiënt zoeken en vinden op internet

Huub de Beer

Eindhoven, 4 juni 2011

Inhoudsopgave

1	Inleiding	1
1.1	Zoeken? Dat kan toch iedereen!	2
1.1.1	Opgaven	4
2	De basis van het zoeken	6
2.1	Eenvoudige zoekopdrachten	6
2.1.1	Opgaven	9
2.2	Propositie logica	10
2.2.1	Opgaven	13
3	De anatomie van een webpagina	15
3.1	Een webpagina zoals jij hem ziet	15
3.1.1	URLs	16
3.1.2	Opgaven	19
3.2	Een webpagina zoals de computer hem ziet	21
3.2.1	HTML	21
3.2.2	Opgaven	23
4	Zoeken vervolgd	25
4.1	Zoekopdrachten verder verfijnd: enkele speciale zoektermen uitgelicht	25
4.1.1	Opgaven	27
5	Zoekresultaten beoordeeld	30
5.1	De relevantie van een webpagina	30
5.2	De betrouwbaarheid van een webpagina	30
5.2.1	Opgaven	33

1 Inleiding

Het internet is tegenwoordig alom vertegenwoordigd. De meeste computers hebben continue een verbinding met het internet en steeds meer apparaten, van televisies tot mobiele telefoons volgen die trend. Als vanzelfsprekend ben je ook vertrouwd geraakt met het internet: je gebruikt het voor communicatie met familie en vrienden, voor het vinden en verspreiden van informatie, voor het rondhangen op sociale websites, enzovoorts.

Tegelijkertijd vertoef je maar op een heel klein deel van het internet. Je leeft als het ware in een digitaal dorp waar je iedereen kent en je goed je weg weet te vinden. Hoe groot het internet precies is, weet niemand. Een ding is zeker: er is een hele hoop informatie. Je moet het alleen zien te vinden.

Je zult de komende jaren veel en vaak zoeken op het internet. En vinden wat je zoekt, kost tijd. Het is belangrijk snel goede resultaten te vinden: daarmee bespaar je een hoop tijd. Het doel van deze cursus is daarom ook het *zo efficiënt mogelijk goede zoekresultaten vinden en verwerken*.

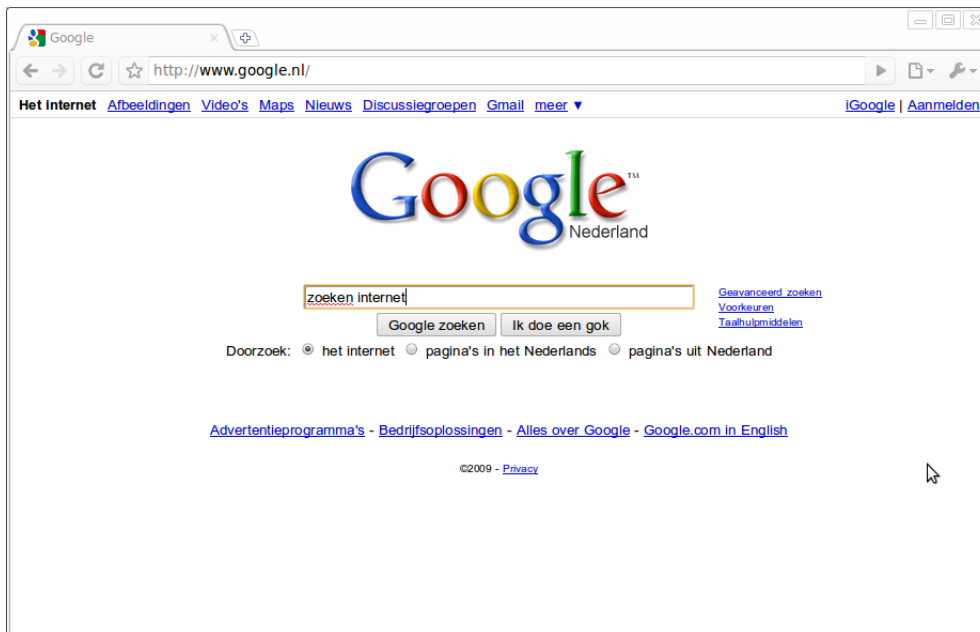
Naast praktische oefening met een zoekmachine (Paragrafen 2.1 en 4) leer je ook meer over de werking van zoekmachines (Paragraaf ??). Daarnaast komen ook theoretische onderwerpen aan bod, zoals de propositielogica (Paragraaf 2.2) en de anatomie van een webpagina (Paragraaf 3). Voordat we dieper ingaan op het zoekproces, kijken we naar het zoeken zoals je dat al kent.

1.1 Zoeken? Dat kan toch iedereen!

We zoeken vaak meerdere keren per dag op het internet. Snel even iets opzoeken: die ene website bijvoorbeeld of informatie over de film van vanavond. En zoeken is eenvoudig, je opent een webbrowsen en je gaat naar de website van een zoekmachine. Je voert enkele zoektermen in waarna je de zoekmachine opdracht geeft op zoek te gaan naar die zoektermen; vervolgens krijg je een hele hoop zoekresultaten voorgeschoteld.

Een voorbeeld: in Figuur 1 zie je een afbeelding van een webbrowsen die de website <http://www.google.nl> weergeeft. Google's zoekmachine is de meest gebruikte zoekmachine in Nederland¹. De gebruiker heeft al twee zoektermen ingevuld: **zoeken internet**. Als de gebruiker nu op de knop "Google zoeken" klikt, gaat Google's zoekmachine op zoek naar webpagina's die iets van doen hebben met beide zoektermen. In Figuur 2 zie je de eerste pagina met zoekresultaten.

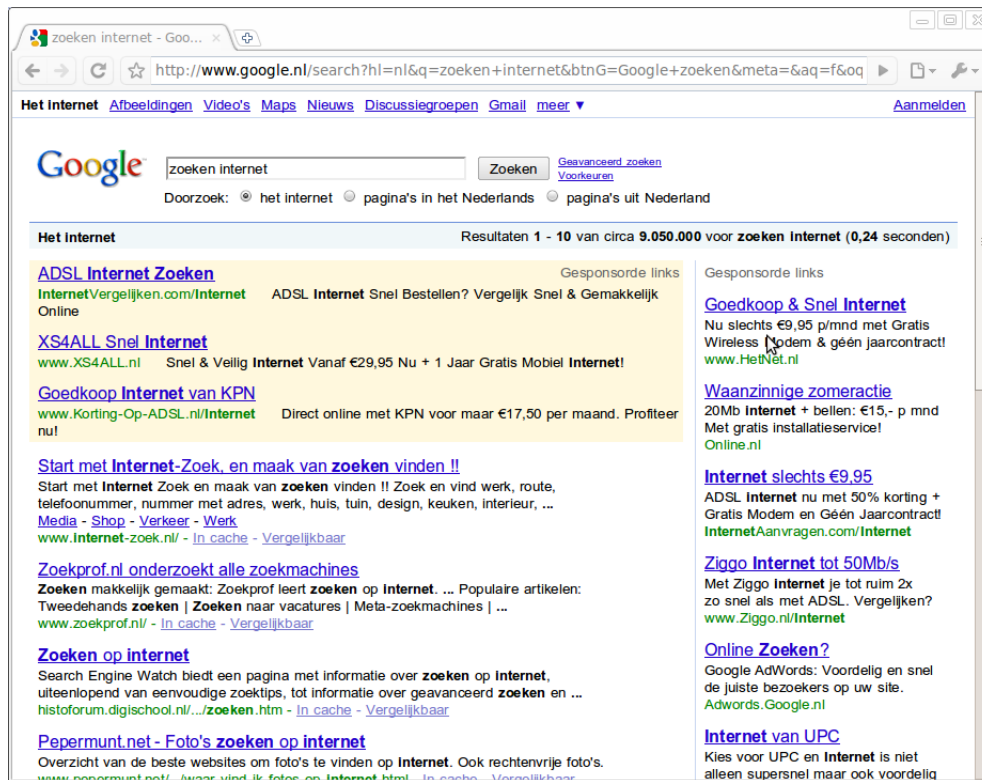
¹Zie <http://www.checkit.nl/nationaalsearchenginemonitor.html>.



Figuur 1: De zoekmachine van Google: zoeken naar **zoeken internet**.

We noemen een zoekresultaat van een zoekmachine een *hit*. Google geeft aan dat er ongeveer 9.050.000 hits zijn voor de zoekopdracht **zoeken internet**. De meeste hits zullen irrelevant zijn voor jouw zoekopdracht. Gelukkig sorteert de zoekmachine de hits op relevantie zodat de betere resultaten hoger in de lijst met zoekresultaten staan dan minder goede resultaten. Maar hoe weet de zoekmachine welke zoekresultaten voor jou relevant zijn? Trouwens, kun je zo'n zoekmachine eigenlijk wel vertrouwen? Hoe weet jij dat de eerste zoveel hits wel de beste hits zijn?

Deze vragen zijn lastig te beantwoorden. In Paragraaf ?? gaan we dieper in op de werking van een zoekmachine; een aantal van deze vragen wordt dan beantwoord. Een ding is zeker: verschillende zoekmachines zullen voor een gelijke opdracht verschillende zoekresultaten opleveren. Gebruik daarom ook eens een andere zoekmachine, misschien vind je andere, betere of meer zoekresultaten. Op http://en.wikipedia.org/wiki/List_of_search_engines vind je een grote lijst met verschillende zoekmachines.



Figuur 2: De zoekmachine van Google heeft een heleboel pagina's gevonden met zoektermen **zoeken internet**.

1.1.1 Opgaven

1.1.1.1 Houd een week lang bij hoe vaak je zoekt op internet. Elke zoekopdracht die je uitvoert, telt als een zoekopdracht. Ook als je de vorige zoekopdracht maar een klein beetje aanpast, bijvoorbeeld door een spelfout, telt dat als een zoekopdracht.

Houd die week ook bij hoe lang je in totaal met zoeken op internet bezig bent. Hierbij telt niet alleen het uitvoeren van een zoekopdracht, maar ook het bekijken van zoekresultaten.

1.1.1.2 Hoeveel verschillende zoekmachines ken je? Welke daarvan gebruik je wel eens?

★ **1.1.1.3** Je hebt al heel vaak gezocht op internet. Je bent een echte ervarings-

deskundige die weet zo snel mogelijk die zoekresultaten te vinden die je zoekt. Maar weet je ook *hoe* je zoekt? Wat is jouw strategie om goede zoekresultaten te verkrijgen?

Schrijf een korte handleiding over hoe *jij* op internet zoekt met behulp van Google's zoekmachine. Besteed extra aandacht aan de volgende punten:

- a. het formuleren van betere zoekwoorden
- b. het aantal zoekresultaten dat je bekijkt
- c. het beoordelen van de zoekresultaten

1.1.1.4 Er zijn drie grote zoekmachines op internet: <http://search.google.com>, <http://search.bing.com> en <http://search.yahoo.com> van respectievelijk Google, Microsoft en Yahoo!. Zoek met deze drie zoekmachines naar de volgende zoektermen en vergelijk de resultaten.

- a. beste zoekmachine
- b. linux windows
- c. Eindhoven

Wat valt je op? Welke zoekmachine vind je het best? Leg uit waarom.

1.1.1.5 Sommige zoekmachines zoeken met behulp van verschillende zoekmachines tegelijkertijd en verzamelen de zoekresultaten van al deze zoekmachines tot een nieuwe lijst met zoekresultaten. Dergelijke zoekmachines heten **metazoekmachines**. Voorbeelden van metazoekmachines zijn <http://www.zoeken.nl> en <http://www.metacrawler.com>.

Zoek ook met twee metazoekmachines naar:

- a. beste zoekmachine
- b. linux windows
- c. Eindhoven

Welke van deze twee metazoekmachine vind je het beste? Leg uit waarom. Werkt deze beste metazoekmachine beter dan de beste zoekmachine uit de vorige opgave? Leg uit waarom wel/niet.

2 De basis van het zoeken

2.1 Eenvoudige zoekopdrachten

Zoals we al hebben gezien geef je eenvoudig een zoekopdracht door een aantal zoektermen in een zoekmachine in te voeren. Als je op zoek bent naar informatie over keizerpenguïns kun je de zoekopdracht **keizerpenguin penguin** invoeren. De zoekmachine geeft je vervolgens een lijst met webpagina's die de zoekterm "keizerpenguin" *en* de zoekterm "penguin" bevatten.

Google koppelt automatisch losse zoektermen aan elkaar met de **AND**-operator. Operatoren ken je al uit de wiskunde, zoals $+$, $-$ en \times waarmee je getallen en onbekenden aan elkaar kunt koppelen tot ingewikkelder wiskundige expressies zoals $3 \times (5 - x)$. Bij het opstellen van ingewikkelde zoekopdrachten koppel je losse zoektermen aan elkaar met operatoren zoals de **AND**-operator, de **OR**-operator of de **-**-operator.

De AND-operator

zoekterm₁ AND zoekterm₂ AND ... AND zoekterm_n

De zoekmachine zoekt naar alle webpagina's waarin de zoektermen *zoekterm₁*, *zoekterm₂*, ..., *zoekterm_n* *allemaal* voorkomen. Omdat Google automatisch de **AND**-operator toevoegt tussen elke twee opeenvolgende losse zoektermen, is het niet nodig expliciet **AND** te schrijven. Met andere woorden, de zoekopdracht **zoekterm₁ AND zoekterm₂** is gelijk aan de zoekopdracht **zoekterm₁ zoekterm₂**.

Stel dat je een onderzoek doet naar pinguïns en dan met name naar de keizerpinguïn en de ezelspinguïn. Je kunt dan met de zoekopdracht **keizerpenguin ezelspenguin** zoeken naar alle webpagina's waarin zowel het woord "keizerpenguin" als het woord "ezelspenguin" voorkomt. Maar vind je dan ook pagina's waarin alleen het woord "keizerpenguin" voorkomt?

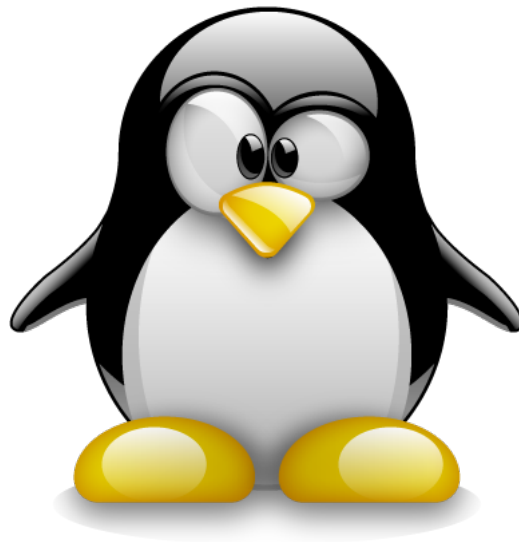
Natuurlijk kun je de zoekopdracht **keizerpenguin** uitvoeren en vervolgens de zoekopdracht **ezelspenguin** om alle pagina's te vinden waarin óf het woord "keizerpenguin" óf het woord "ezelspenguin" voorkomt. De kans is groot dat beide zoekopdrachten veel dezelfde resultaten opleveren, in beide gevallen ben je immers op zoek naar pagina's over pinguïns. Het is in dit geval aan jou om bij te houden of je een zoekresultaat al bent tegengekomen of niet. Daarbij moet je je ook twee keer door irrelevante zoekresultaten heen worstelen. Dit kan handiger met de **OR**-operator.

De zoekopdracht `keizerpinguin OR ezelspinguin` zoekt naar alle webpagina's waarin óf het woord "keizerpinguin" óf het woord "ezelspinguin" óf beide woorden voorkomen. In het algemeen:

De OR-operator

`zoekterm1 OR zoekterm2 OR ... OR zoektermn`

De zoekmachine zoekt naar alle webpagina's waarin een combinatie van een of meer zoektermen `zoekterm1`, `zoekterm2`, ..., `zoektermn` voorkomt. Let op: `OR` moet met hoofdletters geschreven worden. Je kunt `OR` ook vervangen met `|`. Dus de zoekopdracht `zoekterm1 OR zoekterm2` is gelijk aan de zoekopdracht `zoekterm1 | zoekterm2`.



Figuur 3: Tux, de Linux mascotte.

In Figuur 3 zie je Tux. Tux de pinguïn is de mascotte van het besturingssysteem Linux en komt vaak voor in logo's. Zoek je afbeeldingen van echte pinguïns, dan ben je waarschijnlijk niet geïnteresseerd in afbeeldingen van Tux of een pinguïnlogo. Je kunt een zoekmachine de opdracht geven om naar webpagina's te zoeken waar wel het woord "pinguïn" voorkomt, maar *niet* de woorden "tux", "linux", of "logo": `pinguin -tux -linux -logo`. Het niet mogen voorkomen van een zoekterm geef je aan met de operator `-`.

De --operator (de niet-operator)

-zoekterm₁

De zoekmachine zoekt naar alle webpagina's waar *zoekterm*₁ *niet* in voorkomt. Let op: na `–` volgt *direct* de zoekterm; er mag geen spatie tussen `–` en de zoekterm.

Tot nu toe hebben we enkel losse woorden als zoektermen gebruikt. Vaak is dat voldoende. Soms ben je echter geïnteresseerd in een bepaalde zinsnede, bijvoorbeeld een Oudnederlands woord met een vreemde spelling of de titel van een boek. Als je meer informatie wilt over het boek *De kip en de pinguïn* van Midas Dekkers kun je zoeken met: `kip pinguin`. Dit levert heel veel zoekresultaten op die niets met het boek te maken hebben. Je kunt echter ook zoeken naar zinsneden met behulp van het `"`-teken: `'de kip en de pinguin'`. Je plaatst de zinsnede waar je naar zoekt tussen dubbele aanhalingstekens.

Zouden keizerpinguïns en ezelspinguïns allebei op Antarctica voorkomen? En zo ja, wat is er bekend over het leven van die pinguïns aldaar? Een goede zoekopdracht om deze vragen te beantwoorden bestaat uit de zoekopdracht `Antarctica` en de zoekopdracht `keizerpinguin OR ezelspinguin`. Je combineert deze twee zoekopdrachten door ze gewoon achter elkaar te schrijven: `Antarctica keizerpinguin OR ezelspinguin`. Deze zoekopdracht vindt alle pagina's waarop het woord "Antarctica" en het woord "keizerpinguin" óf het woord "ezelspinguin" óf beide woorden voorkomen.

Op dezelfde manier kun je alle soorten zoekopdrachten die we tot nu toe zijn tegengekomen combineren tot langere zoekopdrachten die preciezer aangeven waar je naar zoekt. Hoe preciezer je zoekopdracht formuleert hoe beter de zoekresultaten zullen zijn. Voor de duidelijkheid kun je deelzoekopdrachten tussen haakjes (`(,)`) schrijven. Enkele voorbeelden:

- Er bestaan een aantal open source spelletjes waarin Tux de pinguïn, de mascotte van linux, een hoofdrol speelt. Je wilt een van deze spelletjes downloaden en spelen op jouw Microsoft Windows computer. Je geeft de volgende zoekopdracht: `tux (spel OR game) -linux windows download`. Je zoekt naar de zoekterm "tux" in combinatie met óf "spel" of het Engelse "game" waarbij wel het woord "windows" voorkomt maar niet "linux". Tot slot voeg je het woord "download" toe om juist die pagina's te vinden waarop je het spel kunt downloaden.
- Je wilt meer weten over het vwo in het voortgezet onderwijs. Je weet dat vaak "middelbaar onderwijs" wordt geschreven in plaats van

“voortgezet onderwijs”, beide zoektermen zijn goed. Omdat je meer informatie over enkel het vwo wilt hebben, zoek je naar webpagina’s waar het woord “vwo” voorkomt, maar de woorden “havo”, “mavo” en “vmbo” niet. De zoekopdracht: (‘voortgezet onderwijs’ OR ‘middelbaar onderwijs’) vwo -havo -mavo -vmbo.

Let op, de zoekopdracht (voortgezet OR middelbaar) onderwijs vwo -havo -mavo -vmbo levert andere resultaten op. De eerste paar hits zijn weliswaar hetzelfde, maar de rest van de resultaten en het totaal aantal zoekresultaten verschilt.

2.1.1 Opgaven

2.1.1.1 Voer de zoekopdrachten (‘voortgezet onderwijs’ OR ‘middelbaar onderwijs’) vwo -havo -mavo -vmbo en (voortgezet OR middelbaar) onderwijs vwo -havo -mavo -vmbo uit en vergelijk de eerste twintig à dertig resultaten. Wat is het verschil? Verklaar dit verschil.

2.1.1.2 Voorspel welke zoekresultaten de volgende zoekopdrachten opleveren:

- | | |
|--------------|------------------------|
| a. pinguin | d. pinguin -pinguin |
| b. ‘pinguin’ | e. ping -ping |
| c. -pinguin | f. pinguin OR -pinguin |

Voer deze zoekopdrachten vervolgens uit. Klopten jouw voorspellingen? Leg per zoekopdracht uit waarom de zoekmachine deze resultaten oplevert.

2.1.1.3 Formuleer een zo precies mogelijke zoekopdracht om de volgende vragen te kunnen beantwoorden. Schrijf deze zoekopdracht op en voer de zoekopdracht vervolgens uit. Probeer de vragen in zo min mogelijk zoekopdrachten te beantwoorden.

Je antwoord bevat de gebruikte zoekopdrachten, URL’s (webadressen) van de webpagina’s waarop je het antwoord hebt gevonden en het antwoord.

- a. Is er een boek met in de titel de zinsnede “koeien in de wei”? Zo ja, welke boeken bestaan er wie is de auteur?

Welke *gespecialiseerde* zoekmachine is beter geschikt om deze vraag te beantwoorden? Een **gespecialiseerde zoekmachine** is een zoekmachine die niet het hele internet afzoekt, maar die zich richt op een of enkele onderwerpen.

- b. Wat is de snelheid van een Afrikaanse antilope? En is die antilope sneller dan een ijsbeer?
- c. Wat betekent "Ager thu die wolvin, vore die avegront"?
- d. Wie was de eerste aartsbisschop van Utrecht na het herstel van de bisschoppelijke hiërarchie in 1853? Hoeveel bisschoppen hebben sindsdien dat ambt bekleed?
- e. Wie was de burgemeester van Eindhoven in jouw geboortjaar? En waar had hij zijn eerste burgemeesterschap? En zijn laatste?
- f. Je wilt een nieuwe telefoon kopen waarop je het besturingssysteem linux kunt draaien. Je wilt dat de telefoon ook verbinding kan maken met een draadloos computer netwerk (WiFi). Welke telefoons voldoen aan deze eisen? Zoek er mininmaal 5.

2.1.1.4 Formuleer een zo precies mogelijke zoekopdracht voor de volgende opgaven. Schrijf de zoekopdracht eerst op en voer die zoekopdrachten vervolgens ook uit. Hoe kun je je zoekopdrachten verbeteren? En leveren die verbeterde zoekopdrachten ook inderdaad betere zoekresultaten op?

Je antwoord bevat de gebruikte zoekopdrachten, URL's (webadressen) van de webpagina's waarop je het antwoord hebt gevonden en het antwoord.

- a. Je zoekt informatie over paardebloemen; je vraagt je af of die bloemen giftig zijn voor paarden, koeien of schapen. Is deze bloem giftig voor andere dieren? Zijn er (andere) planten die giftig zijn voor paarden?
- b. Deze Kerst kook jij het diner voor je familie. Je wilt wild klaarmaken en je twijfelt tussen konijn, haas en ree. Je voorkeur gaat uit naar een recept met kersen op sap. Vind enkele recepten.
- c. Je nodig een vriend uit voor een etentje. Je wilt spaghetti klaarmaken, maar je weet niet zeker of die vriend dat mag eten: hij is namelijk een Pastafari. Wat is een Pastafari? En mogen Pastafari pasta eten? Welke feestdagen vieren Pastafari? Is er een Nederlandse Pastafari geloofsgemeenschap?
- d. Je vraag je af of er ook kikkerbiljetjes te koop zijn die "geogst" worden nadat de kikkers gedood zijn. Waar kun je die kopen? Beschrijf het proces van het oogsten van kikkerbiljetjes.

2.2 Propositielogica

Zoekopdrachten met de operatoren **AND**, **OR** en **–** zijn zogenaamde Booleaanse expressies. Een **Booleaanse expressie** is opgebouwd uit proposities en operatoren zoals \wedge (én), \vee (óf) en \neg (niet). Alhoewel deze operatoren er anders uitzien dan in de zoekopdrachten die we eerder hebben gezien, zijn

het precies dezelfde operatoren.

Een **propositie** is een uitspraak die óf waar is óf onwaar is. De uitspraak “mijn band van mijn fiets is lek” is een voorbeeld van zo’n propositie: als mijn band inderdaad lek is, is deze uitspraak waar en anders is deze uitspraak onwaar. In het geval van een zoekopdracht kun je de losse zoekterm $zoekterm_1$ lezen als de propositie “De webpagina bevat het woord $zoekterm_1$ ”. Als deze propositie geldt dan zal de zoekmachine die webpagina opnemen in de lijst met zoekresultaten.

Het vakgebied dat gaat over het “rekenen” met proposities noemen we de **propositielogica**. In de propositielogica geven we proposities vaak aan met een enkele letter, bijvoorbeeld: $P =$ “mijn band van mijn fiets is lek” en $Q =$ “de stoel heeft drie poten”. We kunnen die letters vervolgens gebruiken in samengestelde Booleaanse expressies, zoals $P \wedge \neg Q$.

In de wiskunde reken je met getallen. Er zijn oneindig veel getallen waarmee je wiskundige expressies kunt maken. Booleaanse expressies zijn wat dat betreft een stuk eenvoudiger; er zijn maar *twee* Booleaanse waarden: **waar** en **onwaar**. Soms worden deze twee waarden voorgesteld door, respectievelijk, de twee getallen 1 en 0. Een propositie is dus een uitspraak die de waarde **waar** óf de waarde **onwaar** heeft.

De al bekende Booleaanse operatoren worden als volgt gedefiniëerd:

De negatie

$$\neg P$$

We zeggen: *niet* P .

De negatie van P is **waar** als P **onwaar** is en andersom. In een waarheidstabel:

P	$\neg P$
onwaar	waar
waar	onwaar

De conjunctie

$$P \wedge Q$$

We zeggen: P *en* Q .

De conjunctie van P en Q is **waar** als zowel P én Q **waar** zijn. In elk ander geval is deze expressie **onwaar**. In een waarheidstabel:

P	Q	$P \wedge Q$
onwaar	onwaar	onwaar
onwaar	waar	onwaar
waar	onwaar	onwaar
waar	waar	waar

De disjunctie

$$P \vee Q$$

We zeggen: P of Q .

De disjunctie van P en Q is waar als P óf Q óf zowel P en Q waar zijn. In een waarheidstabel:

P	Q	$P \vee Q$
onwaar	onwaar	onwaar
onwaar	waar	waar
waar	onwaar	waar
waar	waar	waar

Zoals je ziet, worden de Booleaanse operatoren gedefiniëerd met behulp van een waarheidstabel. Een *waarheidstabel* van een Booleaanse expressie is een tabel waarin alle mogelijke waarden van de proposities in die expressie en de expressie zelf worden opgesomd. Vaak worden als tussenstap in de berekening van de hele expressie deelexpressies ook opgenomen in de tabel. Een voorbeeld:

P	Q	$\neg Q$	$P \wedge \neg Q$
onwaar	onwaar	waar	onwaar
onwaar	waar	onwaar	onwaar
waar	onwaar	waar	waar
waar	waar	onwaar	onwaar

De $P \wedge \neg Q$ is erg abstract. Vind je dat moeilijk, dan kun je voor P en Q altijd een concrete uitspraak invullen, bijvoorbeeld: als $P =$ “de webpagina bevat het woord “pinguin”” en $Q =$ “de webpagina bevat het woord “logo””,

dan is de $P \wedge \neg Q$ gelijk aan de zoekopdracht `penguin -logo`. Met enige oefening leer je vanzelf deze abstracte Booleaanse expressies gebruiken.

Bij de expressie $P \wedge \neg Q$ is het duidelijk dat hier “P en niet Q” geschreven staat. Maar hoe lees je een Booleaanse expressie als $P \wedge Q \vee R$? Staat hier $P \wedge (Q \vee R)$ of $(P \wedge Q) \vee R$? Zoals je ziet, zijn haakjes belangrijk om aan te geven wat je *precies* bedoelt. Schrijf de expressies daarom altijd tussen haakjes zodat er geen twijfel bestaat over de betekenis van de volledige expressie. Sterker nog, zonder haakjes is dit voorbeeld zelfs geen geldige Booleaanse expressie.

2.2.1 Opgaven

2.2.1.1 Vertaal onderstaande zoekopdrachten naar een Booleaanse expressie in de propositielogica. Introduceer voor elk zoekwoord of zinsnede tussen aanhalingstekens een afkorting.

De eerste opgave gaat als volgt:

P = “de webpagina bevat het woord “penguin””

Q = “de webpagina bevat het woord “antarctica””

$P \wedge Q$

- `penguin antarctica`
- `fototoestel (accu OR batterij)`
- `besturingssysteem -windows`
- `''informatie met kunde''`
- `-logo -tux -linux`

2.2.1.2 P is **waar** en Q is **onwaar**. Wat is de waarde van onderstaande Booleaanse expressies?

- | | |
|--------------------------|-----------------------|
| a. $P \vee (Q \wedge P)$ | d. $\neg(P \wedge Q)$ |
| b. $P \wedge (Q \vee P)$ | e. $\neg(\neg P)$ |
| c. $\neg(P \vee Q)$ | f. $P \wedge \neg P$ |

2.2.1.3 Maak een waarheidstabel voor onderstaande Booleaanse expressies. Neem in die tabellen ook de expressies zoals $\neg P$ en $P \wedge Q$ op. Kort **onwaar** af tot 0 en **waar** tot 1.

- a. $P \vee (Q \wedge P)$ d. $\neg(P \wedge Q)$
 b. $P \wedge (Q \vee P)$ e. $\neg(\neg P)$
 c. $\neg(P \vee Q)$ f. $P \wedge \neg P$

2.2.1.4 P is **waar**, Q is **onwaar** en R is **waar**. Wat is de waarde van onderstaande Booleaanse expressies? Reken eerst de waarde van subexpressies uit.

- a. $P \vee \neg(Q \vee \neg R)$
 b. $Q \vee \neg(P \vee \neg R)$
 c. $P \wedge (\neg R \vee (Q \wedge R))$
 d. $R \vee (P \wedge (Q \vee (R \wedge Q)))$

2.2.1.5 Je hebt tot nu toe verschillende waarheidstabellen gezien met maximaal twee proposities. Een waarheidstabel met een propositie heeft twee rijen: een propositie heeft immers twee verschillende waarden. Een waarheidstabel met twee proposities heeft vier rijen. Daarmee zijn alle mogelijke combinaties van twee waarden per propositie opgenomen in de tabel.

In de volgende opgave ga je een waarheidstabel maken van een Booleaanse expressie met drie verschillende proposities. Hoeveel rijen heeft die tabel? En hoeveel rijen heeft een waarheidstabel met vier verschillende proposities?

Wat is de algemene formule voor het aantal rijen in een waarheidstabel met n het aantal verschillende proposities?

2.2.1.6 Maak een waarheidstabel voor onderstaande Booleaanse expressies. Neem in die tabellen ook deexpressies zoals $\neg P$ en $P \wedge Q$ op. Kort **onwaar** af tot 0 en **waar** tot 1. Let op, er zijn nu drie verschillende proposities!

- a. $P \vee \neg(Q \vee \neg R)$
 b. $P \wedge (\neg R \vee (Q \wedge R))$

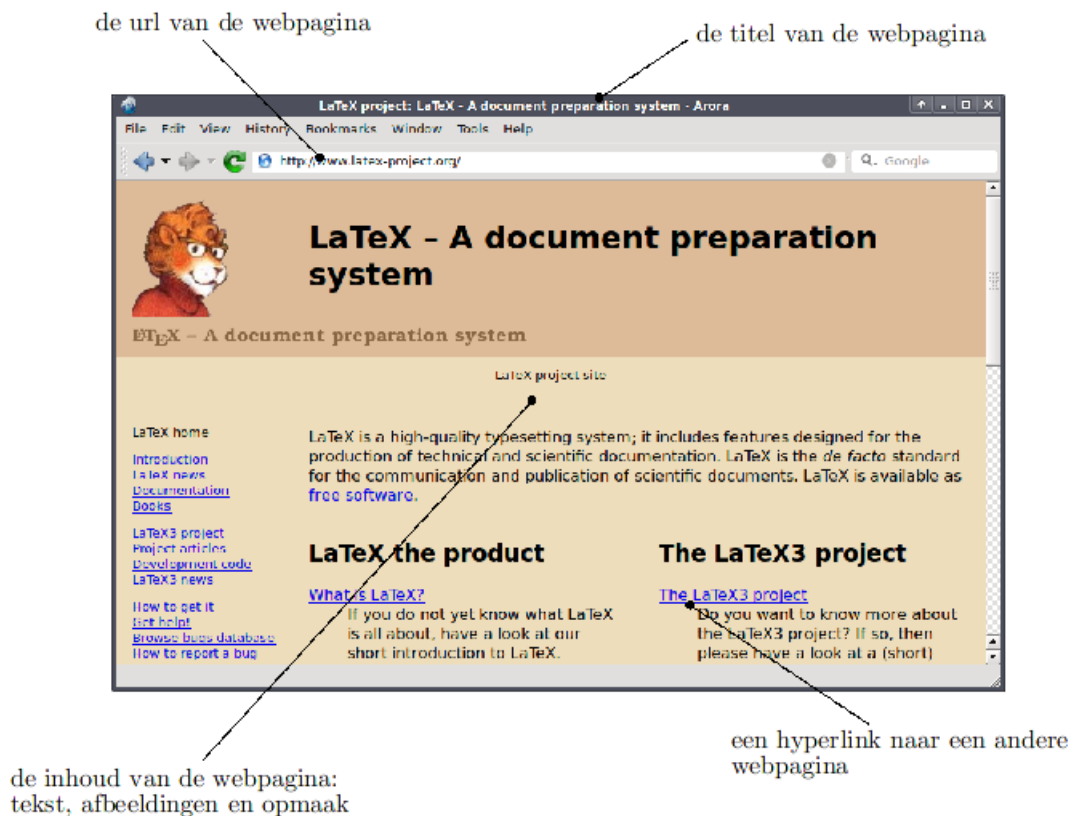
★ **2.2.1.7** Maak een waarheidstabel voor onderstaande Booleaanse expressies. Neem in die tabellen ook deexpressies zoals $\neg P$ en $P \wedge Q$ op. Kort **onwaar** af tot 0 en **waar** tot 1. Let op, er zijn nu drie verschillende proposities!

- a. $\neg P \vee \neg(P \vee \neg(Q \vee \neg(R \vee \neg Q)))$
 b. $\neg(P \wedge (\neg(\neg Q) \vee (R \wedge \neg P)))$

3 De anatomie van een webpagina

Voordat we verder gaan met zoeken en het beoordelen van zoekresultaten is het goed stil te staan bij de anatomie van een webpagina. Hoe ziet een webpagina er eigenlijk uit? Je hebt al vele duizenden webpagina's bekeken, maar wat weet je eigenlijk van webpagina's?

3.1 Een webpagina zoals jij hem ziet



Figuur 4: De website van het \LaTeX -project weergegeven in een browser; een aantal onderdelen zijn geannoteerd.

Als je met een browser naar <http://www.latex-project.org/> gaat, de website van het \LaTeX -project, ziet die website er in je browser ongeveer uit

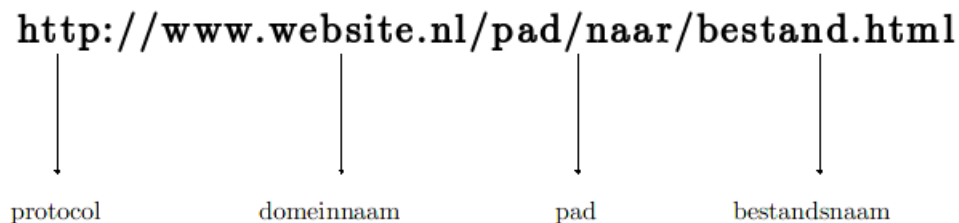
zoals in Figuur 4 (pagina 15). We onderscheiden een aantal verschillende onderdelen:

- De *titel* van de webpagina vind je in de titelbalk van de browser. In dit geval is de titel: *LaTeX project: LaTeX – Een document preparation system*. Let op, deze titel maakt géén deel uit van de inhoud van de webpagina. Een titel is onderdeel van zogenaamde *metadata* van een webpagina.
- De *URL* of het webadres van deze webpagina: <http://www.latex-project.org/>.
- De *inhoud* van deze webpagina. Je ziet een kleine afbeelding en veel tekst met hyperlinks; de pagina is netjes opgemaakt.
- *Hyperlinks* naar andere webpagina's. Een **hyperlink** is een verwijzing naar een andere webpagina waarop je kunt klikken. Na klikken op een hyperlink probeert de browser de aangeklikte pagina te laden. Veel hyperlinks worden weergegeven als een onderstreepte blauwgekleurde tekst. Niets weerhoudt een websitebouwer er echter van om een hyperlink een ander uiterlijk te geven.

3.1.1 URLs

Een **URL** is een webadres van een webpagina, afbeelding, of ander bestand op het internet. Alles wat je op het internet kunt vinden, heeft zo'n URL. De afkorting URL staat voor *Uniform Resource Locator*, wat zoveel betekent als: een algemene manier om “dingen op het internet” precies aan te duiden.

Over het algemeen ziet een URL er als volgt uit:



Figuur 5: De algemene opbouw van een URL (verkort).

Een URL bestaat uit een aantal verschillende onderdelen, achtereenvolgens:

- Een *protocol* waarmee het opgevraagde bestand naar de browser wordt gestuurd. Een **protocol** bestaat uit een aantal afspraken en regels waaraan verschillende partijen zich houden.

In het geval van het *HyperText Transfer Protocol* (**HTTP**) hebben we het over afspraken tussen een webbrowser en een webserver. HTTP legt precies vast hoe een webbrowser een webpagina opvraagt bij een webserver en hoe die webserver vervolgens die pagina naar de browser stuurt.

Naast HTTP kent jouw webbrowser nog een aantal andere protocollen, zoals FTP, FILE en HTTPS. In de opgaven zoek je uit waar deze protocollen voor dienen.

- Het *domein* waaronder de webserver bekend staat. Een **domeinnaam** bestaat meestal uit een naam en een zogenaamd *Top Level Domain* (**TLD**) zoals .com, .org of .nl. De domeinnaam `www.overheid.nl` bestaat dus uit de naam *www.overheid* en de TLD *.nl*.

Elke computer op het internet heeft een uniek nummer. Dat nummer, vier getallen tussen 0 en 255 gescheiden door een punt, noemen we het **IP-adres** van die computer. Zo heeft de website <http://www.overheid.nl> het IP-adres <http://62.112.232.27>.

Een IP-adres van je computer is te vergelijken met het telefoonnummer van je mobieltje: met behulp van een IP-adres kan een computer als het ware “opgebeld” worden of kun je ermee naar een andere computer “bellen”. Omdat telefoonnummers lastig te onthouden zijn, heb je in je mobieltje een telefoonboek waarin je de telefoonnummers van je vrienden en familie koppelt aan hun naam. Op vergelijkbare wijze worden IP-adressen van veel webserver gekoppeld aan een domeinnaam.

In plaats van een IP-adres kun je dus een eenvoudig te onthouden domeinnaam gebruiken om een webpagina bij een webserver op te vragen. Sommige webserver en de meeste gewone computers hebben echter geen domeinnaam. Wil je verbinding maken met deze computers dan zul je alsnog gebruik moeten maken van het IP-adres.

- Een *pad* naar een bestand op de webserver. Op je eigen computer bewaar je je bestanden waarschijnlijk in verschillende directories of subdirectories. Ook op een webserver staan bestanden vaak in verschillende directories.

Als het bestand `index.html` zich in de directory `algemeen` bevindt, dan is het pad van dat bestand `algemeen/`. Bevindt deze directory

zich zelf weer in een andere directory, bijvoorbeeld `web`, dan is het pad van dat bestand `web/algemeen/`.

Het pad geeft dus aan in welke subdirectory van de webserver een bestand zich bevindt.

- En een *bestandsnaam*. Vaak eindigen bestandsnamen in een betekenisvolle extensie zoals `.html` of `.htm` voor HTML bestanden en `.mp3` voor MP3-bestanden. Een URL geeft dus uiteindelijk de plaats van een bestand op het internet aan. Met behulp van die URL kun je dat bestand downloaden, bekijken, afspelen, enzovoorts.

Soms kom je URLs tegen waar na het pad geen bestandsnaam volgt. In dergelijke gevallen duidt de URL een standaardbestand aan: `index.html`.

Sommige URLs duiden speciale bestanden aan: een programma op een webserver. Zo'n programma noemen we een *script*. Als je via je browser zo'n script opvraagt, geef je de webserver de opdracht dat programma uit te voeren. De webserver stuurt dan het resultaat van dat programma, meestal gewoon een webpagina, naar je browser.

Soms worden er extra gegevens naar zo'n script gestuurd. We noemen dergelijke extra gegevens *parameters*. Een URL waarin parameters zijn opgenomen heet een vraag-URL. Je herkent een vraag-URL aan het ?-teken na de naam van het script. Vervolgens volgen de parameters. Een bekend voorbeeld:



Figuur 6: De URL van de zoekresultatenlijst na zoeken naar “url” via Googles zoekmachine.

In Figuur 6 zie je een voorbeeld van zo'n vraag-URL. In dit geval wordt het “search”-script van Googles zoekmachine aangeroepen met de parameter “q=url”. Met andere woorden, met deze URL zoek je met behulp van Googles zoekmachine naar alle webpagina's waarop het woord “url” voorkomt.

3.1.2 Opgaven

3.1.2.1 In elke onderstaande URL zit een of meer fouten. Geef per URL aan wat er fout is.

- a. www.kennisnet.nl
- b. <http://www.flickr.com>
- c. <http://www.eindhoven.nl/onderwijs//algemene-informatie//adressen-scholen.htm>
- d. <ftp://search.google.com>

3.1.2.2 Geef van onderstaande URLs aan wat het protocol, de domeinnaam, het pad en het bestand of script is. Geef, indien aanwezig, ook de parameters in de URL aan.

- a. <http://www.uitzendinggemist.nl/>
- b. <http://examen.kennisnet.nl/havo/oefenexamens/oudeexamens/tu>
- c. http://downloads.kennisnet.nl/oefenexamens/havo/2008/havo_tu12_2008_1_b.pdf
- d. <https://zoek.officielebekendmakingen.nl/Pages/UitgebreidZoeken.aspx?Publicatiesoort=2>

3.1.2.3 Het HTTP protocol wordt gebruikt om webpagina's over het internet te verzenden. Naast het HTTP protocol zijn er nog veel meer protocollen. Hieronder vind je afkortingen van een aantal van die protocollen. Zoek uit wat deze afkortingen precies betekenen en waarvoor deze protocollen gebruikt worden.

- a. HTTPS
- b. FTP
- c. FILE

3.1.2.4 Met behulp van het WHOIS-protocol kun je informatie opvragen over een website. Een manier om het WHOIS-protocol te gebruiken is de website: <http://whois.domaintools.com>. Nadat je een domeinnaam hebt ingevoerd, geeft deze website alle informatie over die domeinnaam dat bekend is in de WHOIS-database.

Zoek uit wie de volgende domeinnamen in beheer heeft. Geef een naam, een plaats en de datum waarop dit domein voor het eerst is geregistreerd.

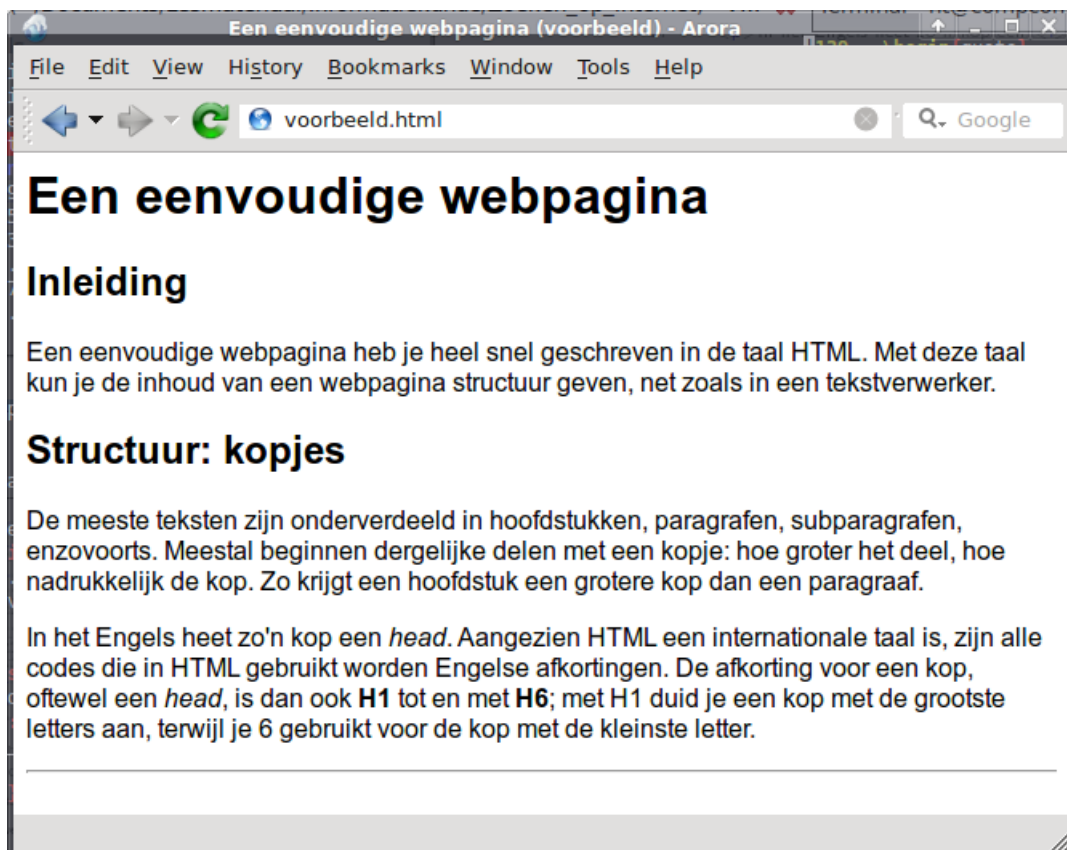
- | | |
|---|---|
| a. www.google.com | d. www.nu.nl |
| b. www.youtube.com | e. www.fok.nl |
| c. www.kennisnet.nl | f. www.tor.org |

3.1.2.5 elke computer op het internet heeft een (uniek) IP-adres. Zo ook de

computer die je nu gebruikt. Zoek met behulp van <http://tools.whois.net/yourip/> op welk IP-adres de computer heeft die je nu gebruikt. Zoek vervolgens uit van wie dat IP-adres is met behulp van de website: <http://whois.domaintools.com>.

3.1.2.6 Welke domeinnamen horen bij de volgende IP-adressen? Om het antwoord te vinden, type je het IP-adres, voorafgegaan van `http://`, in de adresbalk van je browser. Zoek vervolgens uit welke domeinnaam bij de weergegeven website hoort.

- | | |
|-------------------|-------------------|
| a. 80.237.159.15 | d. 145.58.30.33 |
| b. 213.239.154.35 | e. 209.85.227.147 |
| c. 86.59.21.36 | f. 69.63.187.12 |



Figuur 7: Een eenvoudige webpagina weergegeven in een browser.

3.2 Een webpagina zoals de computer hem ziet

In de vorige paragraaf hebben we gezien hoe jij een webpagina ziet: een webpagina heeft een titel, een URL en een inhoud. In Figuur 7 (pagina 20) zie je een weergave van een eenvoudige website in een browser. Maar hoe kijkt een browser tegen zo'n webpagina aan? Wat ziet een browser eigenlijk na ophalen van een webpagina?

In het voorbeeld van Figuur 7 zie je een webpagina getiteld *Een eenvoudige webpagina (voorbeeld)*. Deze pagina bestaat uit een korte tekst over een eenvoudige webpagina. Na de inleiding volgt een paragraaf over hoe je structuur in een HTML pagina kunt aanbrengen. Het geheel wordt afgesloten met een horizontale streep.

De browser geeft deze pagina weer na *interpretatie* van de broncode van de webpagina. In Listing 1 is de volledige broncode van deze pagina opgenomen. De broncode is geschreven in de taal HTML.

3.2.1 HTML

```

1 <html>
2 <head>
3 <!-- Metadata van deze webpagina -->
4 <title>Een eenvoudige webpagina (voorbeeld)</title>
5 <meta name="author" content="Mijnheer de Beer">
6 <meta name="keywords" lang="nl" content="voorbeeld, lesmateriaal, HTML,
7     informatiekunde">
8 </head>
9 <body>
10 <!-- De inhoud van deze webpagina -->
11 <h1>Een eenvoudige webpagina</h1>
12
13 <h2>Inleiding</h2>
14
15 <p>Een eenvoudige webpagina heb je heel snel geschreven in de taal
16 <acronym title="HyperText Markup Language">HTML</acronym>. Met deze taal
17 kun je de inhoud van een webpagina structuur geven, net zoals in een
18 tekstverwerker.</p>
19
20 <h2>Structuur: kopjes</h2>
21
22 <p>De meeste teksten zijn onderverdeeld in hoofdstukken, paragrafen,
23 subparagrafen, enzovoorts. Meestal beginnen dergelijke delen met een
24 kopje: hoe groter het deel, hoe nadrukkelijk de kop. Zo krijgt een
25 hoofdstuk een grotere kop dan een paragraaf.</p>
26
27 <p>In het Engels heet zo'n kop een <i>head</i>. Aangezien HTML een
28 internationale taal is, zijn alle codes die in HTML gebruikt worden
29 Engelse afkortingen. De afkorting voor een kop, oftewel een <i>head</i>,
30 is dan ook <b>H1</b> tot en met <b>H6</b>; met H1 duid je een kop met de
31 grootste letters aan, terwijl je 6 gebruikt voor de kop met de kleinste
32 letter.</p>
33

```

```

34 <hr>
35
36 </body>
37 </html>

```

Listing 1: De broncode van een eenvoudige website: geschreven in HTML.

HTML is een afkorting van *HyperText Markup Language* en is een zogenaamde *opmaaktaal* om webpagina's mee te maken. Als je naar Listing 1 kijkt zie je vanaf regel 11 bijna letterlijk de tekst die je ook in Figuur 7 ziet. Naast de tekst zijn er ook een aantal speciale codes zoals ``, `<h2>` en `</p>`. Dergelijke speciale codes noemen we HTML *tag* s.

Er zijn twee verschillende soorten tags

- *opentags* zoals `<h2>` en `<acronym title="HyperText Markup Language">`. Algemene vorm: `<naam attribuut1="waarde1"attribuut2="waarde2"...>`. Na het `<`-teken volgt de *naam* van de tag. Na de naam kunnen verschillende *attributen* volgen, gescheiden door spaties. Een attribuut heeft een attribuutnaam, een `=`-teken en een waarde. Een tag sluit je af met het `>`-teken.

Het voorbeeld `<acronym title="HyperText Markup Language">` gaat dus over de *acronym*-tag met een attribuut: *title* met waarde "HyperText Markup Language".

- *sluittags* zoals `</h2>` en `</acronym>`. Een opentag wordt bijna altijd afgesloten door een bijbehorende sluittag. De algemene vorm van een sluittag is: `</naam>`. Na het `<`-teken volgt het `/`-teken, de naam en tot slot het `>`-teken.

Een aantal tags heeft geen sluittag. Dergelijke tags noemen we *lege tags*. Een voorbeeld is de tag `<hr>` in Listing 1 (regel 34).

HTML is een internationale taal en de meeste tags hebben als naam een Engelse afkorting. Enkele voorbeelden:

- `<h1>` staat voor *heading 1*, oftewel *kop 1*
- `<h2>` staat voor *kop 2*
- `<hr>` staat voor *horizontal rule*, oftewel een horizontale lijn
- `<p>` staat voor *paragraph*, oftewel *alinea*

Op de website <http://www.handleidinghtml.nl/html/elementen/elementen.html> vind je de betekenis van alle HTML tags.

De inhoud van de webpagina staat in de `<body>`-tag (regels 9–36). Daarvoor zie je op regels 2–9 de `<head>`-tag. In de HEAD is ruimte voor zogenaamde *metadata*. De *metadata* van een webpagina bevat informatie over

de inhoud van de webpagina zoals de titel, de auteur en een aantal keywords. In het algemeen slaat de term metadata op informatie óver informatie.

Veel zoekmachines gebruiken de metadata van een webpagina om het belang van een webpagina vast te stellen of om een lijst met zoekresultaten te ordenen.

3.2.2 Opgaven

3.2.2.1 Je kunt de broncode van een webpagina eenvoudig bekijken: klik op het item “Bron weergeven” in het menu “Beeld” van je browser. Bekijk de broncode van de volgende webpagina’s en geef aan welke soorten metadata anders dan de titel, de auteur en keywords nog meer voorkomen.

- a. <http://www.latex-project.org>
- b. <http://www.kennisnet.nl>
- c. <http://www.handleidinghtml.nl/html/elementen/html.html>

3.2.2.2 Wat is het nut van metadata? Het gaat toch om de inhoud van een webpagina? En alle gegevens over de webpagina staat toch ook al in de inhoud? Geef antwoord op deze drie vragen en leg je antwoord uit.

3.2.2.3 Maak een lijstje met de volgende tags. Geef bij elke tag aan voor welke Engelse term de naam een afkorting is en geef vervolgens een Nederlandse vertaling. Omschrijf tot slot het effect dat de tag heeft op de weergave van de inhoud van de tag in een browser.

- | | |
|----------------------------|-----------------------------|
| a. <code><i></code> | e. <code><h3></code> |
| b. <code></code> | f. <code></code> |
| c. <code><h1></code> | g. <code><a></code> |
| d. <code><p></code> | h. <code><u></code> |

3.2.2.4 Bekijk de broncode van zowel <http://www.latex-project.org> en <http://www.google.nl>. Wat valt je op? Waarom zou Google zijn hoofdpagina er zo uit laten zien? Geef meer dan een reden.

3.2.2.5 Zoals gezegd, gebruiken sommige zoekmachines de metadata van een webpagina om zoekresultaten te ordenen en het belang van een pagina te bepalen. Wat is het nadeel van het gebruik van metadata?

Leg uit hoe je een willekeurige webpagina, bijvoorbeeld die uit Listing 1, hoger in een resultatenlijst van een zoekmachine kunt krijgen. Leg vervolgens uit hoe je dezelfde pagina hoger in een resultatenlijst kunt krijgen van een zoekopdracht die niets met de *inhoud* van de webpagina te maken heeft.

4 Zoeken vervolgd

In Hoofdstuk 2 heb je geleerd hoe je zoekopdrachten kunt opstellen die preciezer aanduiden waar je naar zoekt. Als je gewoon een of twee zoektermen invoert bij een zoekmachine krijg je meestal een lange lijst met zoekresultaten. Het is aan jou om uit die lange lijst de beste resultaten te vissen. Met behulp van de én-, óf- en niet-operator ben je in staat om die lijst al stevig uit te dunnen en betere resultaten over te houden.

Het vorige hoofdstuk handelde over de anatomie van een webpagina. Een belangrijk onderdeel van die anatomie is de zogenaamde metadata van een webpagina zoals de titel, de URL, hyperlinks, steekwoorden, enzovoorts. De metadata van een webpagina geeft je informatie over die webpagina en je kunt die informatie gebruiken om de kwaliteit van die webpagina in te schatten.

Veel zoekmachines zijn bekend met metadata en het is mogelijk om je zoekopdrachten te verfijnen met speciale zoektermen. Met behulp van een *speciale zoekterm* geef je een zoekmachine de opdracht om specifiek in de metadata van een webpagina te zoeken en niet (alleen) in de inhoud van de webpagina.

4.1 Zoekopdrachten verder verfijnd: enkele speciale zoektermen uitgelicht

De zoekmachine van Google kent vele verschillende speciale zoekopdrachten. Hieronder worden een aantal van die speciale zoekopdrachten besproken met enkele voorbeelden:

site:

`zoekterm1 site:domein.tld`

De zoekmachine vindt al die webpagina's waarop `zoekterm1` voorkomt én het domein in de URL gelijk is aan "domein.tld". Met andere woorden: je zoekt alleen op de website "domein.tld"; alle andere websites worden overgeslagen.

Als je op zoek bent naar een formulier van de gemeente Eindhoven ben je waarschijnlijk alleen geïnteresseerd in die formulieren die op de website van de gemeente Eindhoven te vinden zijn: `formulier site:eindhoven.nl`.

intitle:

`intitle:zoekterm1`

De zoekmachine vindt al die webpagina's waarin *zoekterm₁* in de titel voorkomt.

Normaal gesproken is een titel een erg bondige samenvatting van het document waarvan het de titel is. Als een woord in de titel voorkomt, dan is de kans groot dat de bijbehorende webpagina over dat woord gaat. Stel dat je een werkstuk maakt over pinguïns op Antarctica. De zoekopdracht `pinguin antarctica` levert heel veel pagina's op waarop de woorden pinguïn en Antarctica voorkomen, maar veel van die pagina's zullen niet specifiek over pinguïns op Antarctica gaan.

De zoekopdracht `intitle:pinguin intitle:antarctica` daarentegen levert enkel die webpagina's waarvan de titel de woorden pinguïn en Antarctica bevat. De kans is groot dat de meeste pagina's die deze zoekopdracht oplevert inderdaad over de pinguïns van Antarctica gaan.

filetype:

`filetype:extensie zoekterm1`

De zoekmachine vindt al die bestanden met bestandsextensie "extensie" waarin het woord *zoekterm₁* voorkomt. Een bestandsextensie geeft aan wat voor type bestand een bestand is. Zo heeft het MP3-bestand `muziek.mp3` de extensie "mp3" en het tekstbestand `oud-boek.txt` de extensie "txt". Bijna alle bestanden op jouw computer hebben een extensie.

Wereldwijd proberen regeringen speciale torrent-zoekmachines zoals <http://www.thepiratebay.org> te verbieden. Gelukkig biedt Googles zoekmachine uitkomst. Als je op zoek bent naar een torrent bestand van de laatste versie van Ubuntu kun je de volgende zoekopdracht gebruiken: `Ubuntu filetype:torrent`.

Op vergelijkbare wijze kun je op zoek gaan naar een Microsoft Powerpointpresentatie over pinguïns: `pinguin filetype:ppt`. Erg handig als je zelf een presentatie moet maken.

link:

`link:URL zoekterm1`

De zoekmachine vindt al die webpagina's waarop het woord *zoekterm*₁ voorkomt én die ook een hyperlink bevatten naar URL. Heb je eenmaal een goede webpagina gevonden over een onderwerp, dan kun je zo al die pagina's vinden die gebruik maken van die webpagina en deze goede webpagina als bron hebben opgenomen als een hyperlink.

Een voorbeeld. Na uitgebreid zoeken blijkt de wikipediapagina over pinguïns (<http://nl.wikipedia.org/wiki/Pinguïns>) een erg goede webpagina over pinguïns. Je verwacht dat anderen ook van deze wikipediapagina gebruik hebben gemaakt. Juist omdat ze deze goede pagina gebruiken, is de kans groot dat hun pagina's ook van goede kwaliteit zijn.

De zoekopdracht `link:http://nl.wikipedia.org/wiki/Pinguïns` levert echter voornamelijk pagina's van Wikipedia zelf op. Zoeken naar links naar een wikipediapagina werkt dus niet zo goed. Waarom eigenlijk niet?

4.1.1 Opgaven

4.1.1.1 In de tekst zijn al een aantal extensies genoemd, zoals mp3, torrent, txt en ppt voor respectievelijk muziekbestanden, torrentbestanden, tekstbestanden en Microsoft Powerpointbestanden. Er zijn nog veel meer verschillende bestandstypen.

Probeer voor onderstaande soorten bestanden drie verschillende extensies te geven.

- documenten
- muziek
- video
- afbeeldingen
- spreadsheets

4.1.1.2 Formuleer geschikte zoekopdrachten voor de volgende problemen. Maak gebruik van de vier speciale zoektermen die in dit hoofdstuk zijn geïntroduceerd. Probeer je zoekopdrachten ook uit en pas ze aan als ze niet het gewenste resultaat opleveren.

- zoek PDF-bestanden over het vak Informatiekunde
- zoek PDF-bestanden over het vak Wiskunde op <http://www.kennisnet.nl>
- zoek naar webpagina's die linken naar <http://www.wilhelmus.nl>
- zoek naar webpagina's die de zinsnede "Wilhelmus van Nassauwe" in de titel hebben

4.1.1.3 Formuleer zo precies mogelijke zoekopdrachten voor de volgende problemen. Maak gebruik van speciale zoektermen en de zoekoperatoren. Let op, noteer allen de zoekopdrachten, niet de resultaten. Controleer wel of je zinnige zoekopdrachten resultaten opleveren. Zo niet, pas je zoekopdracht dan aan. (Alle zoekopdrachten behoren resultaten op te leveren.)

- a. Je bent op zoek naar PDF-bestanden op de website van Greenpeace waarin over het schip "Rainbow Warrior" wordt gerept.
- b. Edsger Wybe Dijkstra was Nederlands eerste programmeur en is de beroemdste Nederlandse informaticus. Op <http://www.cs.utexas.edu/users/EWD> vind zijn archief. Jij maakt een werkstuk over Dijkstra en je maakt veel gebruik van dat archief. Anderen die ook over Dijkstra hebben geschreven hebben waarschijnlijk ook dat archief gebruikt.
Je bent voornamelijk geïnteresseerd in "gestructureerd programmeren", of "structured programming" op zijn Engels. Vind al die webpagina's over gestructureerd programmeren in het Nederlands en het Engels die gebruik maken van het digitale archief van Dijkstra.
- c. Zoek enkel PNG of JPG afbeeldingen over konijnen in relatie met een van de kerkelijke feestdagen Kerstmis, Pasen of Hemelvaart.
- d. Zoek naar torrentbestanden over "Star Trek" met Nederlandstalige ondertitels ("subtitles" op zijn Engels).

4.1.1.4

Google heeft verschillende soorten zoekmachines. Zo kun je zoeken naar enkel afbeeldingen, video's of in je e-mail. Al deze verschillende soorten documenten hebben ook verschillende soorten metadata. Je kunt met deze speciale zoekmachines ook zoeken in metadata die specifiek hoort bij het soort document.

Zo kent Googles GMail onder andere de volgende extra speciale zoektermen: `from:`, `to:`, `subject:`, `has:attachment`, `cc:`, `bcc:`, `before:2009/08/30` en `after:2009/08/30`. Deze speciale zoektermen verwijzen naar de welbekende onderdelen in je e-mail(programma): van wie het e-mail afkomstig is, naar wie het gesturd is, wat het onderwerp is, of het een bijlage bevat, wat het "CC"-veld of het "BCC"-veld is, en of het voor of na een bepaalde datum is verzonden. In plaats van zoeken naar webpagina's, zoek je nu naar e-mailberichten in je inbox.

Formuleer zoekopdrachten om de volgende e-mails te vinden:

- a. Alle e-mails waarin jouw e-mailadres als een CC-tje is toegevoegd.
- b. Alle e-mails met een subject met daarin de letterlijke tekst "[IK]".
- c. Alle e-mails die niet door jou zijn gezonden waaraan wel een bijlage zit bijgevoegd.

- d. Alle e-mails die jij tussen 01 – 01 – 2009 en 31 – 12 – 2009 hebt verzonden aan jezelf.

5 Zoekresultaten beoordeeld

In de vorige hoofdstukken heb je geleerd hoe je zo precies mogelijke zoekopdrachten kunt formuleren. Met behulp van zoekoperatoren en speciale zoektermen ben je in staat om betere zoekresultaten te verkrijgen. Maar goede zoekopdrachten kunnen geven alleen is niet genoeg om ook snel en efficiënt goede webpagina's te *vinden*. Je zult de zoekresultaten nader moeten bekijken en de gevonden webpagina moeten beoordelen: heb ik iets aan deze pagina en is de pagina betrouwbaar?

Het is belangrijk dat je zo snel mogelijk kunt bepalen of een gevonden webpagina voor jou relevant is én daarbovenop betrouwbaar. Om dat te bepalen volgen enkele vuistregels.

5.1 De relevantie van een webpagina

Een pagina waar je iets aan hebt, noemen we een *relevante* pagina. Een pagina is voor jou relevant als:

- er voor jou nieuwe informatie te vinden is die je nog niet eerder (in deze vorm) bent tegengekomen.

Er zijn een aantal websites die de lemma's van Wikipedia overnemen en daaromheen veel reclame plaatsen. Als je zo'n pagina vindt, dan is de kans groot dat je de Wikipediapagina over dit onderwerp ook al gevonden hebt. Deze "nieuwe" pagina is dan niet relevant.

- er een hoop (nieuwe) bronnen te vinden zijn.

Een webpagina kan best dezelfde informatie bevatten die je ook al op een aantal andere pagina's bent tegengekomen en toch relevant zijn als er verwezen wordt naar andere bronnen die jij nog niet kent. Bronnen vindt je meestal in de literatuurlijst, een lijst met hyperlinks, voetnoten en hyperlinks in de tekst.

5.2 De betrouwbaarheid van een webpagina

Een relevante pagina moet natuurlijk ook *betrouwbaar* zijn. De betrouwbaarheid van een pagina is niet altijd even gemakkelijk vast te stellen. Er zijn vele aspecten van een webpagina die de betrouwbaarheid beïnvloeden.

Hieronder worden een aantal van die aspecten besproken. Het zijn geen regels die je zonder meer toe kunt passen: er zijn altijd uitzonderingen. Ge-

bruik daarom altijd je gezond verstand bij het beoordelen van de betrouwbaarheid.

1. De auteur

- (a) Is de auteur bekend? Een anonieme webpagina is over het algemeen minder betrouwbaar.
- (b) Wie is de auteur: wat is zijn functie, autoriteit, relatie tot het onderwerp? Als de auteur op een universiteit werkt en onderzoek doet naar kwantummechanica, dan is een webpagina over de kwantummechanica van zijn hand betrouwbaar. Bij een webpagina over origami valt die betrouwbaarheid te betwijfelen.
Aan de andere kant kan een huisvrouw die jarenlange ervaring heeft met origami juist weer een betrouwbare webpagina over origami gemaakt hebben.

2. De organisatie

- (a) Van wie is de website? (Indien onduidelijk, gebruik de WHOIS database om uit te zoeken wie de eigenaar is)
- (b) Heeft deze bepaald webpagina enige inhoudelijke relatie tot de eigenaar van de website? Op een universiteit kunnen alle medewerkers een persoonlijke homepage opzetten; de universiteit bemoeit zich niet met de inhoud van deze homepage's.
- (c) Wat voor soort organisatie is het? Een bedrijf, onderzoeksinstelling, overheidsinstelling, non-profit instelling, ...
Als je op zoek bent naar objectieve informatie over een product of onderwerp, ben je bij een website van een bedrijf waarschijnlijk op het verkeerde adres. Ben je echter op zoek naar gedetailleerde productgegevens, dan is de website van een bedrijf de beste bron. Idem dito is een website van een overheids- of onderzoeksinstelling niet automatisch betrouwbaar: ook de overheid en onderzoekers hebben een agenda. Hoed je bij overheidswebsites voor propaganda.

3. De inhoud

- (a) Is de webpagina verzorgd? Ziet het er netjes uit? Zijn er weinig taalfouten?

- (b) Wat is het doel van de webpagina? Informerend of overtuigend?
Een commerciële website is over het algemeen overtuigend: het doel van de site is jou iets te laten kopen. Zo'n pagina is minder betrouwbaar dan een pagina die enkel informatie probeert over te brengen, zoals een wikipediapagina. Maar let op, een overtuigende pagina kan informerend overkomen en andersom.
Pas vooral op voor webpagina's die vanuit een enkel perspectief zijn geschreven.
- (c) Is het subjectief of objectief? Worden er verschillende perspectieven belicht? Komen zowel voor- als nadelen evenredig aan bod?

4. De bronnen

- (a) Worden er bronnen gebruikt? Hoeveel?
Over het algemeen is een webpagina die op meerdere bronnen gebaseerd is, betrouwbaarder. Tegelijkertijd zijn veel vergelijkbare bronnen of onbetrouwbare bronnen een teken van onbetrouwbaarheid.
- (b) Zijn de bronnen betrouwbaar?
Net zoals je de betrouwbaarheid van een webpagina moet beoordelen, zo moet je ook de betrouwbaarheid van bronnen beoordelen. Immers, als een webpagina gebaseerd is op onbetrouwbare bronnen, dan zal de webpagina zelf ook wel niet betrouwbaar zijn.
- (c) Wordt er netjes met bronnen omgegaan? Wordt er veel geciteerd of zelfs plagiaat gepleegd?
Het kopiëren of overmatige veel en lang citeren van bronnen verlaagt de betrouwbaarheid. Zo'n webpagina biedt weinig nieuws: bekijk zelf de bronnen en vergeet deze pagina.
- (d) Wordt deze pagina vaak als een bron gebruikt. Je controleert dit met de speciale zoekterm link:
Een betrouwbare webpagina wordt waarschijnlijk vaker gebruikt dan een onbetrouwbare webpagina.

5. De datum

- (a) Wanneer is de webpagina gemaakt?
Zeker voor actuele onderwerpen is het belangrijk dat je weet wanneer de pagina is gemaakt om vast te kunnen stellen of de informatie op de pagina nog wel up-to-date is.

- (b) Wanneer is de webpagina voor het laatst geüpdatet? Een webpagina die onlangs geüpdatet is, bevat waarschijnlijk betere informatie dan een pagina waar al tien jaar niets aan veranderd is.
- (c) Werken alle hyperlinks op de pagina nog. Niet werkende hyperlinks zijn een teken van verwaarlozing. Een verwaarloosde site is minder betrouwbaar dan een goed onderhouden site.

Het is erg lastig om een webpagina te beoordelen. Op de website van de *webdetective* vind je een HTML formulier waarmee je in 18 punten een webpagina kunt beoordelen. (De volledige URL is: <http://www.webdetective.nl/checklist.html>.) Met behulp van deze checklist kun je een score aan de betrouwbaarheid van een webpagina geven. Een hoge score betekent over het algemeen dat een pagina betrouwbaar is. Maar let op: *GEBRUIK JE GEZOND VERSTAND EN GA NIET ZONDER MEER AF OP DE SCORE.*

5.2.1 Opgaven

5.2.1.1 Ga naar de website <http://www.webdetective.nl> en volg de cursus. Je start de cursus door op de *Verder*-knop te klikken rechtsonder in het scherm. De cursus bestaat uit een aantal stukjes theorie en een twintigtal vragen. (De inhoud van deze cursus behoort ook tot de lesstof!)

Referenties

- Blachman, Nancy, 'Google Guide', [⟨URL: ⟩](#).
- Brin, S. en L. Page, *The Anatomy of a Large-Scale Hypertextual Web Search Engine*.
- Calishain, Tara en Rael Dornfest, *Google Hacks*, tweede druk (O'Reilly 2004).
- Cutts, Matt, 'How does Google collect and rank results?', *Librarian Central Newsletter* (19 december 2005), [⟨URL: ⟩](#).
- Google, 'Google search basics: Basic search help', [⟨URL: ⟩](#).
- Google, 'Google search basics: More search help', [⟨URL: ⟩](#).
- Google, 'Search Features', [⟨URL: ⟩](#).
- Schneider, Karen G., 'Beyond Algorithms: A Librarian's Guide to Finding Web Sites You Can Trust', *Librarian Central Newsletter* (19 januari 2006), [⟨URL: ⟩](#).
- Sherman, Chris en Gary Price, 'The Invisible Web: Uncovering Sources Search Engines Can't See', *Library Trends* 52:2 (2003), [⟨URL: ⟩](#), 282–298.
- Wall, Aaron, 'Search Engine History' (2007), [⟨URL: ⟩](#).