

$\text{\LaTeX}$ -cursus 2

$\text{\TeX}$ niCie

18 november 2025

s.v.p. alvast inloggen op

[overleaf.com](https://www.overleaf.com)

(Maak een account aan als er nog geen hebt)

Math mode displays

```

1 Wiskunde kan inline
2 met $f(x) = \int_0^x y^2 dy$
3 of in een display met
4 \[ f(x) = \int_0^x y^2 dy. \]
5 Alternatief kan inline wiskunde
6 ook met \(f(x) = \frac{1}{3}x^3\)
7 en een display kan ook met
8 \begin{equation*}
9     f(x) = \frac{1}{3}x^3
10 \end{equation*}

```

Wiskunde kan inline met $f(x) = \int_0^x y^2 dy$
of in een display met

$$f(x) = \int_0^x y^2 dy.$$

Alternatief kan inline wiskunde ook met $f(x) = \frac{1}{3}x^3$ en een display kan ook met

$$f(x) = \frac{1}{3}x^3.$$

Laatste dingetjes

Bij berekeningen wil je vaak vergelijkingen onder elkaar zetten. Je doet dit met de `align` omgeving van `amsmath`-package.

```

1 Zo zet je het onder elkaar
2 \begin{align}
3     f(x) &= (x+1)(x-1) - (x+1)^2 \\
4     &= (x^2 - 1) - (x^2 + 2x + 1) \\
5     &= -2x - 2
6 \end{align}

```

Zo zet je het onder elkaar

$$\begin{aligned}
 f(x) &= (x+1)(x-1) - (x+1)^2 & (1) \\
 &= (x^2 - 1) - (x^2 + 2x + 1) & (2) \\
 &= -2x - 2 & (3)
 \end{aligned}$$

Laatste dingetjes

Als je het nummertje bij de vergelijking niet wil, dan gebruik je de `align*`-omgeving:

```
1 \begin{align*}
2   f(x) &= (x+1)(x-1) - (x+1)^2 \\
3   &= (x^2 - 1) - (x^2 + 2x + 1) \\
4   &= -2x - 2
5 \end{align*}
```

$$\begin{aligned} f(x) &= (x+1)(x-1) - (x+1)^2 \\ &= (x^2 - 1) - (x^2 + 2x + 1) \\ &= -2x - 2 \end{aligned}$$

Laatste dingetjes

Om **matrices** te maken doe je het volgende:

Laatste dingetjes

Om **matrices** te maken doe je het volgende:

```
1 \begin{pmatrix}
2   a & b & c \\
3   d & e & f \\
4   g & h & i
5 \end{pmatrix}
```

Laatste dingetjes

Om **matrices** te maken doe je het volgende:

```
1 \begin{pmatrix}
2   a & b & c \\
3   d & e & f \\
4   g & h & i
5 \end{pmatrix}
```

Dit geeft

$$\begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix}$$

Eigen commando's (Agenda)

We gaan kijken hoe je je eigen commando's kunt aanmaken. We gaan kijken naar:

Eigen commando's (Agenda)

We gaan kijken hoe je je eigen commando's kunt aanmaken. We gaan kijken naar:

- Nieuwe commando's

Eigen commando's (Agenda)

We gaan kijken hoe je je eigen commando's kunt aanmaken. We gaan kijken naar:

- Nieuwe commando's
- Nieuwe commando's met één of meerdere *inputs*

Eigen commando's (Agenda)

We gaan kijken hoe je je eigen commando's kunt aanmaken. We gaan kijken naar:

- Nieuwe commando's
- Nieuwe commando's met één of meerdere *inputs*
- Herdefiniëren van commando's

Eigen commando's (Agenda)

We gaan kijken hoe je je eigen commando's kunt aanmaken. We gaan kijken naar:

- Nieuwe commando's
- Nieuwe commando's met één of meerdere *inputs*
- Herdefiniëren van commando's
- Operatoren

Eigen commando's (Agenda)

We gaan kijken hoe je je eigen commando's kunt aanmaken. We gaan kijken naar:

- Nieuwe commando's
- Nieuwe commando's met één of meerdere *inputs*
- Herdefiniëren van commando's
- Operatoren
- Oefeningetje

Eigen Commando's

Je kunt je **eigen commando's** aanmaken om veel typwerk te voorkomen. Je doet dit als volgt:

1

```
\newcommand{\naam}{wat je wil dat je commando doet}
```

Eigen Commando's

Je kunt je **eigen commando's** aanmaken om veel typwerk te voorkomen. Je doet dit als volgt:

```
1 \newcommand{\naam}{wat je wil dat je commando doet}
```

Je zet `\newcommand` meestal voor je `\begin{document}`:

```
1 \documentclass[a4paper]{article}
```

```
2 \newcommand{\R}{\mathbb{R}}
```

```
3 \begin{document}
```

```
4 We kunnen nu ons nieuwe commando aanroepen:  $x \in \mathbb{R}$ .
```

```
5 Dit is sneller dan  $x \in \mathbb{R}$ 
```

```
6 \end{document}
```

Eigen Commando's

Je kunt je **eigen commando's** aanmaken om veel typwerk te voorkomen. Je doet dit als volgt:

```
1 \newcommand{\naam}{wat je wil dat je commando doet}
```

Je zet `\newcommand` meestal voor je `\begin{document}`:

```
1 \documentclass[a4paper]{article}
```

```
2 \newcommand{\R}{\mathbb{R}}
```

```
3 \begin{document}
```

```
4 We kunnen nu ons nieuwe commando aanroepen:  $x \in \mathbb{R}$ .
```

```
5 Dit is sneller dan  $x \in \mathbb{R}$ 
```

```
6 \end{document}
```

We kunnen nu ons nieuwe commando aanroepen: $x \in \mathbb{R}$. Dit is sneller dan $x \in \mathbb{R}$

Eigen Commando's

Je kunt ook je eigen commando's maken die nog een **input** nodig hebben. Je doet dit met `\newcommand{\naam}[1]{wat je wil dat het doet}`. Bijvoorbeeld

1

```
\newcommand{\bb}[1]{\mathbb{#1}}
```

Eigen Commando's

Je kunt ook je eigen commando's maken die nog een **input** nodig hebben. Je doet dit met `\newcommand{\naam}[1]{wat je wil dat het doet}`. Bijvoorbeeld

1

```
\newcommand{\bb}[1]{\mathbb{#1}}
```

Let op dat je hier één argument *moet* invoeren!

Eigen Commando's

Je kunt ook je eigen commando's maken die nog een **input** nodig hebben. Je doet dit met `\newcommand{\naam}[1]{wat je wil dat het doet}`. Bijvoorbeeld

```
1 \newcommand{\bb}[1]{\mathbb{#1}}
```

Let op dat je hier één argument *moet* invoeren! Dus bijvoorbeeld:

```
1 \documentclass[a4paper]{article}
2 \newcommand{\bb}[1]{\mathbb{#1}}
3 \begin{document}
4 De reële getallen zijn  $\mathbb{R}$ , de breuken zijn  $\mathbb{Q}$ .
5 \end{document}
```

Eigen Commando's

Je kunt ook je eigen commando's maken die nog een **input** nodig hebben. Je doet dit met `\newcommand{\naam}[1]{wat je wil dat het doet}`. Bijvoorbeeld

```
1 \newcommand{\bb}[1]{\mathbb{#1}}
```

Let op dat je hier één argument *moet* invoeren! Dus bijvoorbeeld:

```
1 \documentclass[a4paper]{article}
2 \newcommand{\bb}[1]{\mathbb{#1}}
3 \begin{document}
4 De reële getallen zijn  $\mathbb{R}$ , de breuken zijn  $\mathbb{Q}$ .
5 \end{document}
```

De reële getallen zijn $\mathbb{R}$, de breuken zijn $\mathbb{Q}$.

Eigen Commando's

Voor een commando met **meerdere argumenten** veranderen we de [1] naar bijvoorbeeld [2]:

```
1 \documentclass[a4paper]{article}
2 \newcommand{\der}[2]{\frac{d #1}{d #2}}
3 \begin{document}
4 De afgeleide van  $f$  is  $\frac{d f}{d x}$ 
5 \end{document}
```

Eigen Commando's

Voor een commando met **meerdere argumenten** veranderen we de [1] naar bijvoorbeeld [2]:

```

1 \documentclass[a4paper]{article}
2 \newcommand{\der}[2]{\frac{d #1}{d #2}}
3 \begin{document}
4 De afgeleide van  $f$  is  $\frac{d f}{d x}$ 
5 \end{document}
```

Dit geeft:

De afgeleide van f is $\frac{df}{dx}$

Eigen Commando's

Soms is een commando al gedefinieerd. Je kunt het commando dan '**herschrijven**', mits het niet al te belangrijk is. Een voorbeeld is `\P`. Als je probeert om dit te definiëren:

```
1 \newcommand{\P}{\mathbb{P}}
```

dan krijg je een foutmelding

Command `\P` already defined

Eigen Commando's

Soms is een commando al gedefinieerd. Je kunt het commando dan '**herschrijven**', mits het niet al te belangrijk is. Een voorbeeld is $\mathbb{P}$. Als je probeert om dit te definiëren:

```
1 \newcommand{\P}{\mathbb{P}}
```

dan krijg je een foutmelding

Command $\mathbb{P}$ already defined

Om het commando te herschrijven:

```
1 \renewcommand{\P}{\mathbb{P}}
```

Eigen Commando's

Soms is een commando al gedefinieerd. Je kunt het commando dan '**herschrijven**', mits het niet al te belangrijk is. Een voorbeeld is `\P`. Als je probeert om dit te definiëren:

```
1 \newcommand{\P}{\mathbb{P}}
```

dan krijg je een foutmelding

Command `\P` already defined

Om het commando te herschrijven:

```
1 \renewcommand{\P}{\mathbb{P}}
```

Een veelvoorkomend voorbeeld is `\renewcommand{\vec}{\mathbf}` om $\mathbf{v}$ in plaats van $\vec{v}$ te krijgen.

Eigen Commando's (operatoren)

Sommige functies, zoals $\sin(\cdot)$ zijn – in tegenstelling tot normale letters in mathmode – rechtgedrukt. Als je ook een commando wil maken voor iets wat rechtgedrukt moet staan, dan kan je dat doen met de `\operatorname{}` command.

Eigen Commando's (operatoren)

Sommige functies, zoals $\sin(\cdot)$ zijn – in tegenstelling tot normale letters in mathmode – rechtgedrukt. Als je ook een commando wil maken voor iets wat rechtgedrukt moet staan, dan kan je dat doen met de `\operatorname{}` command.

```
1 \documentclass[a4paper]{article}
2 \newcommand{\Sp}{\operatorname{Span}}
3 \begin{document}
4 De span noteren we als  $\$ \Sp(\vec{v}_1, \dots, \vec{v}_n) \$$ .
5 \end{document}
```

Eigen Commando's (operatoren)

Sommige functies, zoals $\sin(\cdot)$ zijn – in tegenstelling tot normale letters in mathmode – rechtgedrukt. Als je ook een commando wil maken voor iets wat rechtgedrukt moet staan, dan kan je dat doen met de `\operatorname{}` command.

```

1 \documentclass[a4paper]{article}
2 \newcommand{\Sp}{\operatorname{Span}}
3 \begin{document}
4 De span noteren we als  $\$ \Sp(\vec{v_1}, \dots, \vec{v_n}) \$$ .
5 \end{document}
```

De span noteren we als $\text{Span}(\mathbf{v_1}, \dots, \mathbf{v_n})$.

Oefening

Maak een commando zodat `\t(A)` het volgende weergeeft¹:

$$\mathrm{Tr}(A)$$

Let op, `\t` is standaard al gedefinieerd! Bovendien is `Tr` *rechtgedrukt*!

¹`Tr` is een notatie voor het *spoor* van een matrix

Oefening

Maak een commando zodat `\t(A)` het volgende weergeeft¹:

$$\mathrm{Tr}(A)$$

Let op, `\t` is standaard al gedefinieerd! Bovendien is `Tr` *rechtgedrukt*!

Antwoord

```
\renewcommand{\t}{\operatorname{Tr}}
```

¹`Tr` is een notatie voor het *spoor* van een matrix

Comments (Agenda)

Een **comment** is een stukje code dat genegeerd zal worden bij het uitvoeren van je code. Dit kan soms erg handig zijn.

We gaan het volgende bespreken:

Comments (Agenda)

Een **comment** is een stukje code dat genegeerd zal worden bij het uitvoeren van je code. Dit kan soms erg handig zijn.

We gaan het volgende bespreken:

- Hoe maak je comments

Comments (Agenda)

Een **comment** is een stukje code dat genegeerd zal worden bij het uitvoeren van je code. Dit kan soms erg handig zijn.

We gaan het volgende bespreken:

- Hoe maak je comments
- Waarvoor is dit handig?

Comments (Hoe Maak Je Comments?)

In $\text{\LaTeX}$ kun je – net als in andere programmeertalen – comments maken binnen je code. Je geeft aan wat een comment moet zijn door een %-teken aan het begin van de regel te zetten. Je kunt in Overleaf ook `Ctrl + ?` typen. Dan komt er een % aan het begin van de regel.

Comments (Hoe Maak Je Comments?)

In $\text{\LaTeX}$ kun je – net als in andere programmeertalen – comments maken binnen je code. Je geeft aan wat een comment moet zijn door een %-teken aan het begin van de regel te zetten. Je kunt in Overleaf ook `Ctrl + ?` typen. Dan komt er een % aan het begin van de regel. $\text{\LaTeX}$ negeert *alles* na het %-teken. Voorbeeld:

```
1 \documentclass[a4paper]{article}
2 \begin{document}
3 Hier is tekst die wel weergegeven wordt.
4 % Deze tekst wordt niet weergegeven.
5 \end{document}
```

Comments (Hoe Maak Je Comments?)

In $\text{\LaTeX}$ kun je – net als in andere programmeertalen – comments maken binnen je code. Je geeft aan wat een comment moet zijn door een %-teken aan het begin van de regel te zetten. Je kunt in Overleaf ook `Ctrl + ?` typen. Dan komt er een % aan het begin van de regel. $\text{\LaTeX}$ negeert *alles* na het %-teken. Voorbeeld:

```
1 \documentclass[a4paper]{article}
2 \begin{document}
3 Hier is tekst die wel weergegeven wordt.
4 % Deze tekst wordt niet weergegeven.
5 \end{document}
```

Dit geeft dan:

Hier is tekst die wel weergegeven wordt.

Comments (Waarvoor?)

Comments zijn nuttig voor twee belangrijke redenen:

Comments (Waarvoor?)

Comments zijn nuttig voor twee belangrijke redenen:

- Tekst **opslaan**;

Comments (Waarvoor?)

Comments zijn nuttig voor twee belangrijke redenen:

- Tekst **opslaan**;
 - ▶ Als je een stuk van een opdracht hebt uitgetypt, en je komt erachter dat het niet helemaal klopt, dan is het vaak handig om dit niet allemaal te verwijderen, maar het gewoon in een comment te zetten. Zo blijft het toch opgeslagen in je bestand, zonder dat het weergegeven wordt.

Comments (Waarvoor?)

Comments zijn nuttig voor twee belangrijke redenen:

- Tekst **opslaan**;
 - ▶ Als je een stuk van een opdracht hebt uitgetypt, en je komt erachter dat het niet helemaal klopt, dan is het vaak handig om dit niet allemaal te verwijderen, maar het gewoon in een comment te zetten. Zo blijft het toch opgeslagen in je bestand, zonder dat het weergegeven wordt.
- Om **foutmeldingen** te verhelpen;

Comments (Waarvoor?)

Comments zijn nuttig voor twee belangrijke redenen:

- Tekst **opslaan**;
 - ▶ Als je een stuk van een opdracht hebt uitgetypt, en je komt erachter dat het niet helemaal klopt, dan is het vaak handig om dit niet allemaal te verwijderen, maar het gewoon in een comment te zetten. Zo blijft het toch opgeslagen in je bestand, zonder dat het weergegeven wordt.
- Om **foutmeldingen** te verhelpen;
 - ▶ Bij een foutmelding die je niet snapt → comment je code *helemaal* en compile je document. Als je nu geen errors krijgt, dan heb je het stuk met de fout te pakken. Je kunt nu langzaam regels gaan uncommenten en je bestand weer compileren om zo te vinden waar de error zit.

Stellingen en Bewijzen (Agenda)

Om een stelling in te voegen heb je een “theorem-environment” nodig. We gaan kijken naar het volgende:

- Hoe maak ik een Stelling, Lemma, Bewijs etc.
- Hoe pas ik de opmaak hiervan aan?

Stellingen en Bewijzen

In veel documenten zitten speciale omgevingen zoals stellingen, lemma's, bewijzen, opmerkingen enzovoort. Deze omgevingen zijn *niet* standaard gedefinieerd in $\text{\LaTeX}$.

Stellingen en Bewijzen

In veel documenten zitten speciale omgevingen zoals stellingen, lemma's, bewijzen, opmerkingen enzovoort. Deze omgevingen zijn *niet* standaard gedefinieerd in $\text{\LaTeX}$. Om deze omgevingen te krijgen heb je de `amsthm` package nodig. Zet dus in je preamble `\usepackage{amsthm}`.

Stellingen en Bewijzen

In veel documenten zitten speciale omgevingen zoals stellingen, lemma's, bewijzen, opmerkingen enzovoort. Deze omgevingen zijn *niet* standaard gedefinieerd in $\text{\LaTeX}$. Om deze omgevingen te krijgen heb je de `amsthm` package nodig. Zet dus in je preamble `\usepackage{amsthm}`. Vervolgens kun een omgeving aanmaken:

```
1 \newtheorem{stl}{Stelling}
```

Stellingen en Bewijzen

In veel documenten zitten speciale omgevingen zoals stellingen, lemma's, bewijzen, opmerkingen enzovoort. Deze omgevingen zijn *niet* standaard gedefinieerd in $\text{\LaTeX}$. Om deze omgevingen te krijgen heb je de `amsthm` package nodig. Zet dus in je preamble `\usepackage{amsthm}`. Vervolgens kun een omgeving aanmaken:

```
1 \newtheorem{st1}{Stelling}
```

- Het eerste argument `st1` is de naam van je omgeving.

Stellingen en Bewijzen

In veel documenten zitten speciale omgevingen zoals stellingen, lemma's, bewijzen, opmerkingen enzovoort. Deze omgevingen zijn *niet* standaard gedefinieerd in $\text{\LaTeX}$. Om deze omgevingen te krijgen heb je de `amsthm` package nodig. Zet dus in je preamble `\usepackage{amsthm}`. Vervolgens kun een omgeving aanmaken:

```
1 \newtheorem{stl}{Stelling}
```

- Het eerste argument `stl` is de naam van je omgeving.
- Het tweede argument `Stelling` is de naam die vetgedrukt in je document zal komen te staan als je de omgeving aanroept.

Stellingen en Bewijzen

In veel documenten zitten speciale omgevingen zoals stellingen, lemma's, bewijzen, opmerkingen enzovoort. Deze omgevingen zijn *niet* standaard gedefinieerd in $\text{\LaTeX}$. Om deze omgevingen te krijgen heb je de `amsthm` package nodig. Zet dus in je preamble `\usepackage{amsthm}`. Vervolgens kun een omgeving aanmaken:

```
1 \newtheorem{stl}{Stelling}
```

- Het eerste argument `stl` is de naam van je omgeving.
- Het tweede argument `Stelling` is de naam die vetgedrukt in je document zal komen te staan als je de omgeving aanroept.

Je roept je nieuwe omgeving aan met `\begin{stl} ... \end{stl}`.

Stellingen en Bewijzen

Bijvoorbeeld:

```
1 \documentclass[a4paper]{article}
2 \usepackage{amsthm}
3 \newtheorem{stl}{Stelling}
4 \begin{document}
5 \begin{stl}
6   Dit is mijn stelling.
7 \end{stl}
8 \end{document}
```

geeft:

Stelling 1. *Dit is mijn stelling*

Stellingen en Bewijzen (Nummering)

Je kunt ook stellingen maken die **geen nummertje** hebben. Hiervoor moet je een nieuwe omgeving definiëren:

Stellingen en Bewijzen (Nummering)

Je kunt ook stellingen maken die **geen nummertje** hebben. Hiervoor moet je een nieuwe omgeving definiëren:

```
1 \newtheorem*{stl*}{Stelling}
```

Stellingen en Bewijzen (Nummering)

Je kunt ook stellingen maken die **geen nummertje** hebben. Hiervoor moet je een nieuwe omgeving definiëren:

```
1 \newtheorem*{stl*}{Stelling}
```

Je krijgt dan:

Stelling. *Dit is mijn stelling*

Stellingen en Bewijzen (Bewijzen)

Het invoegen van een **bewijs** is erg makkelijk als je `amsthm` hebt geïmporteerd.

```
1 \documentclass{article}
2 \usepackage[dutch]{babel}
3 \usepackage{amsthm}
4 \newtheorem{stelling}{Stelling}
5 \begin{document}
6 \begin{proof}
7     Dit is mijn bewijs.
8 \end{proof}
9 \end{document}
```

Stellingen en Bewijzen (Bewijzen)

Het invoegen van een **bewijs** is erg makkelijk als je `amsthm` hebt geïmporteerd.

```
1 \documentclass{article}
2 \usepackage[dutch]{babel}
3 \usepackage{amsthm}
4 \newtheorem{stelling}{Stelling}
5 \begin{document}
6 \begin{proof}
7     Dit is mijn bewijs.
8 \end{proof}
9 \end{document}
```

Je krijgt dan:

Bewijs. Dit is mijn bewijs. $\square$

Stellingen en Bewijzen (Bewijzen)

De taal waarin het woord ‘Bewijs’ staat, hangt af van de `babel` package.

Als je `\usepackage[english]{babel}` gebruikt, dan zal er ‘*Proof.*’ staan.

Stellingen en Bewijzen (Bewijzen)

De taal waarin het woord 'Bewijs' staat, hangt af van de babel package.

Als je `\usepackage[english]{babel}` gebruikt, dan zal er '*Proof.*' staan.

```
1 \documentclass{article}
2 \usepackage[dutch]{babel}
3 \usepackage{amsthm}
4 \newtheorem{stelling}{Stelling}
5 \begin{document}
6 \begin{stelling}
7     Dit is mijn stelling
8 \end{stelling}
9 \begin{proof}
10     Dit is mijn bewijs.
11 \end{proof}
12 \end{document}
```

Stellingen en Bewijzen (Nummering)

Vaak heb je dat je zowel lemma's als stellingen in je bestand hebt.

Stellingen en Bewijzen (Nummering)

Vaak heb je dat je zowel lemma's als stellingen in je bestand hebt.

Standaard hebben deze verschillende nummering zodat je Stelling 1 en Lemma 1 krijgt.

Stellingen en Bewijzen (Nummering)

Vaak heb je dat je zowel lemma's als stellingen in je bestand hebt.

Standaard hebben deze verschillende nummering zodat je Stelling 1 en Lemma 1 krijgt.

Meestal wil je dat de nummering doorloopt, dus je wil Lemma 1, Stelling 2, etc. Om dit te doen gebruiken we:

```
1 \newtheorem{stl}{Stelling}  
2 \newtheorem{lem}[stl]{Lemma}
```

Nu loopt de teller van 'Lemma' mee met die van Stelling.

Stellingen en Bewijzen (Nummering)

Je kunt ook je stellingen meenummeren met je hoofdstukken.

Stellingen en Bewijzen (Nummering)

Je kunt ook je stellingen meenummeren met je hoofdstukken. Je doet dit met:

```
1 \newtheorem{stelling}{Stelling}[section]  
2 \newtheorem{lemma}[stelling]{Lemma}
```

Kijk goed waar de [...] staat!

Stellingen en Bewijzen (Nummering)

Je kunt ook je stellingen meenummeren met je hoofdstukken. Je doet dit met:

```
1 \newtheorem{stelling}{Stelling}[section]  
2 \newtheorem{lemma}[stelling]{Lemma}
```

Kijk goed waar de [...] staat! Nu krijg je in section 1 bijvoorbeeld Stelling 1.1, Lemma 1.2, etc.

Stellingen en Bewijzen (Opmaak)

Er zijn 3 standaard ‘vormen’ voor een omgeving:

Stellingen en Bewijzen (Opmaak)

Er zijn 3 standaard ‘vormen’ voor een omgeving:

- `plain`: schuingedrukte tekst, extra ruimte om de omgeving heen (vaak gebruikt in stellingen, lemmas, proposities)

Stellingen en Bewijzen (Opmaak)

Er zijn 3 standaard ‘vormen’ voor een omgeving:

- `plain`: schuingedrukte tekst, extra ruimte om de omgeving heen (vaak gebruikt in stellingen, lemmas, proposities)
- `definition` rechtgedrukte tekst, extra ruimte om de omgeving heen (vaak gebruikt in definities, voorbeelden, opgaves)

Stellingen en Bewijzen (Opmaak)

Er zijn 3 standaard ‘vormen’ voor een omgeving:

- `plain`: schuingedrukte tekst, extra ruimte om de omgeving heen (vaak gebruikt in stellingen, lemmas, proposities)
- `definition` rechtgedrukte tekst, extra ruimte om de omgeving heen (vaak gebruikt in definities, voorbeelden, opgaves)
- `remark` Rechtgedrukte tekst, *geen* extra ruimte om de omgeving heen (vaak gebruikt in opmerkingen, samenvattnigen, conclusies)

Stellingen en Bewijzen (Opmaak)

Er zijn 3 standaard ‘vormen’ voor een omgeving:

- `plain`: schuingedrukte tekst, extra ruimte om de omgeving heen (vaak gebruikt in stellingen, lemmas, proposities)
- `definition` rechtgedrukte tekst, extra ruimte om de omgeving heen (vaak gebruikt in definities, voorbeelden, opgaves)
- `remark` Rechtgedrukte tekst, *geen* extra ruimte om de omgeving heen (vaak gebruikt in opmerkingen, samenvattnigen, conclusies)

Je zet een omgeving in een van deze stijlen door:

```
1 \theoremstyle{plain}
2 \newtheorem{stl}{Stelling}
```

Stellingen en Bewijzen (Opmaak)

Om dus een aantal omgevingen goed te zetten, zowel in nummering als in stijl:

```
1 \theoremstyle{plain}
2 \newtheorem{stl}{Stelling}[section]
3 \newtheorem{lem}[stelling]{Lemma}
4 \newtheorem{prop}[stelling]{Propositie}
5
```

Stellingen en Bewijzen (Opmaak)

Om dus een aantal omgevingen goed te zetten, zowel in nummering als in stijl:

```
1 \theoremstyle{plain}
2 \newtheorem{stl}{Stelling}[section]
3 \newtheorem{lem}[stelling]{Lemma}
4 \newtheorem{prop}[stelling]{Propositie}
5
6 \theoremstyle{definition}
7 \newtheorem{def}[stelling]{Definitie}
8 \newtheorem{voorb}[stellin]{Voorbeeld}
9
```

Stellingen en Bewijzen (Opmaak)

Om dus een aantal omgevingen goed te zetten, zowel in nummering als in stijl:

```
1 \theoremstyle{plain}
2 \newtheorem{stl}{Stelling}[section]
3 \newtheorem{lem}[stelling]{Lemma}
4 \newtheorem{prop}[stelling]{Propositie}
5
6 \theoremstyle{definition}
7 \newtheorem{def}[stelling]{Definitie}
8 \newtheorem{voorb}[stellin]{Voorbeeld}
9
10 \theoremstyle{remark}
11 \newtheorem*{opm}{Opmerking}
```

Packages (Inleiding)

We gaan een aantal packages bespreken die handig zijn om te kennen en te gebruiken. Een paar hiervan zijn al besproken. De overigen zullen we bespreken.

- Handige packages
- Korte bespreking van enumitem
- Korte bespreking van hyperref en cleveref
- Probeer het zelf eens!

Packages

Packages staan in de **preamble** van een document om meer commando's in het document te kunnen gebruiken. Handige packages zijn onderanderen:

- **Voor wiskunde:** amssymb, amsmath, amsthm, mathrsfs
- **Voor taal:** babel, inputenc, fontenc
- **Voor afbeeldingen:** graphicx, float, tikz
- **Voor lijsten:** enumitem
- **Voor linkjes:** hyperref, cleveref
- **Voor layout:** geometry, fancyhdr

Enumitem

```
1 Een genummerde lijst kan
2 gemaakt worden met
3 \begin{enumerate}[label=\arabic*.]
4     \item pen
5     \item papier
6     \item laptop
7 \end{enumerate}
8 en een ongenummerde lijst met
9 \begin{itemize}[label=\textbullet]
10     \item lunch
11     \item waterfles
12 \end{itemize}
```

Een genummerde lijst kan
gemaakt worden met

1. pen
2. papier
3. laptop

en een ongenummerde lijst met

- lunch
- waterfles

Enumitem (short labels)

```
1 Een genummerde lijst kan  
2 gemaakt worden met  
3 \begin{enumerate}[1.]  
4     \item pen  
5     \item papier  
6     \item laptop  
7 \end{enumerate}
```

Een genummerde lijst kan
gemaakt worden met

1. pen
2. papier
3. laptop

Hiervoor moet je enumitem als volgt in de preamble zetten:

```
\usepackage[shortlabels]{enumitem}
```

Hyperref en cleveref

```

1 Stel dat ik wil terug refereren naar
2 \begin{align}\label{eq: cirkel}
3     x^2 + y^2 = a
4 \end{align}
5 dan kan ik nu gemakkelijk terug
6 refereren naar een formule
7 van de cirkel met \cref{eq: cirkel}.
8 In plaats hiervan kan je
9 ook \ref{eq: cirkel} gebruiken.
10 Dit kan je ook doen
11 voor figuren, sections en meer.

```

Stel dat ik wil terug refereren naar

$$x^2 + y^2 = a \quad (1)$$

dan kan ik nu gemakkelijk terug refereren naar een formule van de cirkel met verg. (1). In plaats hiervan kan je ook 1 gebruiken. Dit kan je ook doen voor figuren, sections en meer.

Pas op: eerst hyperref in je preamble introduceren en daarna cleveref!

Probeer het zelf

Probeer zelf eens:

- i. een genummerde lijst te maken,
- ii. een ongenummerde lijst te maken,
- iii. een label te geven aan een section, figuur, stelling of iets anders,
- iv. als je een label hebt, probeer er nu naar te refereren,

Tekstopmaak

Resultaat	Code	Resultaat	Code
Tekst	<code>\textbf{Tekst}</code>		
<i>Tekst</i>	<code>\textit{Tekst}</code>		
TEKST	<code>\textsc{Tekst}</code>		
<u>Tekst</u>	<code>\underline{Tekst}</code>		

Tekstopmaak

Resultaat	Code	Resultaat	Code
Tekst	<code>\textbf{Tekst}</code>	Tekst	<code>\tiny Tekst</code>
<i>Tekst</i>	<code>\textit{Tekst}</code>	Tekst	<code>\small Tekst</code>
TEKST	<code>\textsc{Tekst}</code>	Tekst	<code>\large Tekst</code>
<u>Tekst</u>	<code>\underline{Tekst}</code>	Tekst	<code>\huge Tekst</code>

Tekstopmaak

Resultaat	Code	Resultaat	Code
Tekst	<code>\textbf{Tekst}</code>	Tekst	<code>\tiny Tekst</code>
<i>Tekst</i>	<code>\textit{Tekst}</code>	Tekst	<code>\small Tekst</code>
TEKST	<code>\textsc{Tekst}</code>	Tekst	<code>\large Tekst</code>
<u>Tekst</u>	<code>\underline{Tekst}</code>	Tekst	<code>\huge Tekst</code>
Tekst	<code>\mathbf{Tekst}</code>		
$\mathbb{T}$	<code>\mathbb{T}</code>		
$\mathcal{T}$	<code>\mathcal{T}</code>		
$\mathfrak{T}$	<code>\mathfrak{T}</code>		
α	<code>\boldsymbol{\alpha}</code>		

Zelf oplossingen vinden

$\text{\LaTeX}$ biedt veel opties, wat zowel handig als frustrerend kan zijn. Daarom is het belangrijk om te leren zoeken naar wat je wil bereiken in je document.

Het helpt om **duidelijke zoekopdrachten** te hebben, zoals bijvoorbeeld 'tabellen maken latex', 'citing in latex', 'error ... latex'.

Handige websites om te bezoeken:

- <https://tex.stackexchange.com/>
- https://www.overleaf.com/learn/latex/Learn_LaTeX_in_30_minutes
- https://nl.wikibooks.org/wiki/LaTeX/Wiskundige_formules
- <https://detexify.kirelabs.org/classify.html>

Afsluiting

Leuk dat jullie er waren!

Voor vragen zijn wij bereikbaar op `texnicie@a-eskwadraat.nl`