

Masterclass programmeren op de GR TI-84 (les 1)

*

Kevin van As

December 13, 2015

Let me introduce myself...

- Kevin van As (23yr)
- PhD aan de TUDelft in de Natuurkunde
- Hobbies:
 - Les geven
 - Masterclasses organiseren
 - Computer programmeren
 - Computer games

Let me introduce myself...

- Kevin van As (23yr)
- PhD aan de TUDelft in de Natuurkunde
- Hobbies:
 - Les geven
 - Masterclasses organiseren
 - Computer programmeren
 - Computer games

Let me introduce myself...

- Kevin van As (23yr)
- PhD aan de TUDelft in de Natuurkunde
- Hobbies:
 - Les geven
 - Masterclasses organiseren
 - Computer programmeren
 - Computer games

Let me introduce myself...

- Kevin van As (23yr)
- PhD aan de TUDelft in de Natuurkunde *
- Hobbies:
 - Les geven
 - Masterclasses organiseren
 - Computer programmeren
 - Computer games

Let me introduce myself...

- Kevin van As (23yr)
- PhD aan de TUDelft in de Natuurkunde
- Hobbies:
 - Les geven
 - Masterclasses organiseren
 - Computer programmeren
 - Computer games

Wat is de cursus?

Leerdoelen

Je zult leren:

- Je Grafische Rekenmachine (GR) TI-84 beter leren kennen.
- Leren om programma's (PRGM) voor de GR te schrijven:
 - ABC-formule
 - Priemgetallen
 - Oplossen van (willekeurige) functies
 - Grafisch bewerken van functies
 - Eenheidscirkel
 - Exactor
- En dit leert je ook inzichten om op de computer te leren programmeren!



Wat is de cursus?

Leerdoelen

Je zult leren:

- Je Grafische Rekenmachine (GR) TI-84 beter leren kennen.
- Leren om programma's (**PRGM**) voor de GR te schrijven:
 - ABC-formule *
 - Priemgetallen
 - Oplossen van (willekeurige) functies
 - Grafisch bewerken van functies
 - Eenheidscirkel
 - Exactor
- En dit leert je ook inzichten om op de computer te leren programmeren!



Wat is de cursus?

Leerdoelen

Je zult leren:

- Je Grafische Rekenmachine (GR) TI-84 beter leren kennen.
- Leren om programma's (**PRGM**) voor de GR te schrijven:
 - ABC-formule *
 - Priemgetallen
 - Oplossen van (willekeurige) functies
 - Grafisch bewerken van functies
 - Eenheidscirkel
 - Exactor
- En dit leert je ook inzichten om op de computer te leren programmeren!



Wat is de cursus?

Leerdoelen

Je zult leren:

- Je Grafische Rekenmachine (GR) TI-84 beter leren kennen.
- Leren om programma's (**PRGM**) voor de GR te schrijven:
 - ABC-formule *
 - Priemgetallen
 - Oplossen van (willekeurige) functies
 - Grafisch bewerken van functies
 - Eenheidscirkel
 - Exactor
- En dit leert je ook inzichten om op de computer te leren programmeren!



Wat is de cursus?

Leerdoelen

Je zult leren:

- Je Grafische Rekenmachine (GR) TI-84 beter leren kennen.
- Leren om programma's (**PRGM**) voor de GR te schrijven:
 - ABC-formule *
 - Priemgetallen
 - Oplossen van (willekeurige) functies
 - Grafisch bewerken van functies
 - Eenheidscirkel
 - Exactor
- En dit leert je ook inzichten om op de computer te leren programmeren!



Wat is de cursus?

Leerdoelen

Je zult leren:

- Je Grafische Rekenmachine (GR) TI-84 beter leren kennen.
- Leren om programma's (**PRGM**) voor de GR te schrijven:
 - ABC-formule *
 - Priemgetallen
 - Oplossen van (willekeurige) functies
 - Grafisch bewerken van functies
 - Eenheidscirkel
 - Exactor
- En dit leert je ook inzichten om op de computer te leren programmeren!



Wat is de cursus?

Leerdoelen

Je zult leren:

- Je Grafische Rekenmachine (GR) TI-84 beter leren kennen.
- Leren om programma's (**PRGM**) voor de GR te schrijven:
 - ABC-formule *
 - Priemgetallen
 - Oplossen van (willekeurige) functies
 - Grafisch bewerken van functies
 - Eenheidscirkel
 - Exactor
- En dit leert je ook inzichten om op de computer te leren programmeren!



Wat is de cursus?

Leerdoelen

Je zult leren:

- Je Grafische Rekenmachine (GR) TI-84 beter leren kennen.
- Leren om programma's (**PRGM**) voor de GR te schrijven:
 - ABC-formule *
 - Priemgetallen
 - Oplossen van (willekeurige) functies
 - Grafisch bewerken van functies
 - Eenheidscirkel
 - Exactor



- En dit leert je ook inzichten om op de computer te leren programmeren!

Wat is de cursus?

Leerdoelen

Je zult leren:

- Je Grafische Rekenmachine (GR) TI-84 beter leren kennen.
- Leren om programma's (**PRGM**) voor de GR te schrijven:
 - ABC-formule *
 - Priemgetallen
 - Oplossen van (willekeurige) functies
 - Grafisch bewerken van functies
 - Eenheidscirkel
 - Exactor
- En dit leert je ook inzichten om op de computer te leren programmeren!



Wat is de cursus?

Opzet

- Wekelijks op Maandagavond 19:00-21:00.*
- Van DD/MM/YY tot DD/MM/YY, in totaal 6 keer.
- Les bestaat uit beetje uitleg, en veel zelf proberen.
- Je krijgt opdrachten mee naar huis om zelf te oefenen.

Wat is de cursus?

Opzet

- Wekelijks op Maandagavond 19:00-21:00.
- Van DD/MM/YY tot DD/MM/YY*, in totaal 6 keer.
- Les bestaat uit beetje uitleg, en veel zelf proberen.
- Je krijgt opdrachten mee naar huis om zelf te oefenen.

Wat is de cursus?

Opzet

- Wekelijks op Maandagavond 19:00-21:00.
- Van DD/MM/YY tot DD/MM/YY*, in totaal 6 keer.
- Les bestaat uit beetje uitleg, en veel zelf proberen.
- Je krijgt opdrachten mee naar huis om zelf te oefenen.

Wat is de cursus?

Opzet

- Wekelijks op Maandagavond 19:00-21:00.
- Van DD/MM/YY tot DD/MM/YY*, in totaal 6 keer.
- Les bestaat uit beetje uitleg, en veel zelf proberen.
- Je krijgt opdrachten mee naar huis om zelf te oefenen.

Notatie op deze slides

Op deze slides zullen we drie verschillende lettertypen vinden. Het huidige lettertype is normale tekst.

Dit lettertype wordt gebruikt om tekst van je rekenmachine te laten zien. `7→A:A≠6`

Tekens als `[PRGM]`, `[GRAPH]`, `[COS]`, `[*,T,θ,n]` en `[ENTER]` zijn fysieke knoppen van je rekenmachine.

En tekens als `[L1]`, `[COS-1]`, en `[MATRIX]` zijn knoppen die je met `[2nd]` kunt bereiken. Deze knoppen staan boven andere knoppen.

Bijvoorbeeld `[TEST]` is gelijk aan `[2nd][MATH]`.

Notatie op deze slides

Op deze slides zullen we drie verschillende lettertypen vinden. Het huidige lettertype is normale tekst.

Dit lettertype wordt gebruikt om tekst van je rekenmachine te laten zien. $7 \rightarrow A: A \neq 6$

Tekens als `[PRGM]`, `[GRAPH]`, `[QOS]`, `[*,T,θ,n]` en `[ENTER]` zijn fysieke knoppen van je rekenmachine.

En tekens als `[L1]`, `[COS-1]`, en `[MATRIX]` zijn knoppen die je met `[2nd]` kunt bereiken. Deze knoppen staan boven andere knoppen.

Bijvoorbeeld `[TEST]` is gelijk aan `[2nd][MATH]`.

Notatie op deze slides

Op deze slides zullen we drie verschillende lettertypen vinden. Het huidige lettertype is normale tekst.

Dit lettertype wordt gebruikt om tekst van je rekenmachine te laten zien. `7 → A: A ≠ 6`

Tekens als `PRGM`, `GRAPH`, `COS`, `X,T,θ,n` en `ENTER` zijn fysieke knoppen van je rekenmachine.

En tekens als `[L1]`, `[COS-1]`, en `[MATRIX]` zijn knoppen die je met `[2nd]` kunt bereiken. Deze knoppen staan boven andere knoppen.

Bijvoorbeeld `[TEST]` is gelijk aan `[2nd][MATH]`.

Notatie op deze slides

Op deze slides zullen we drie verschillende lettertypen vinden. Het huidige lettertype is normale tekst.

Dit lettertype wordt gebruikt om tekst van je rekenmachine te laten zien. $7 \rightarrow A: A \neq 6$

Tekens als `PRGM`, `GRAPH`, `COS`, `X,T,Θ,n` en `ENTER` zijn fysieke knoppen van je rekenmachine.

En tekens als `[L1]`, `[COS-1]`, en `[MATRIX]` zijn knoppen die je met `[2nd]` kunt bereiken. Deze knoppen staan boven andere knoppen.

Bijvoorbeeld `[TEST]` is gelijk aan `[2nd][MATH]`.

Notatie op deze slides

Op deze slides zullen we drie verschillende lettertypen vinden. Het huidige lettertype is normale tekst.

Dit lettertype wordt gebruikt om tekst van je rekenmachine te laten zien. `7 → A: A ≠ 6`

Tekens als `PRGM`, `GRAPH`, `COS`, `X,T,Θ,n` en `ENTER` zijn fysieke knoppen van je rekenmachine.

En tekens als `[L1]`, `[COS-1]`, en `[MATRIX]` zijn knoppen die je met `[2nd]` kunt bereiken. Deze knoppen staan boven andere knoppen.

Bijvoorbeeld `[TEST]` is gelijk aan `[2nd][MATH]`.

Nu...Laten we beginnen! Rekenmachines bij de hand...

Outline

- 1 Hoe open je een programma?
- 2 Variabelen en Datatypes
 - Getallen
 - Strings
 - Overige
- 3 Basic IO
 - Disp
 - Prompt
- 4 Exercises

*

NEW PRGM

We gaan ons eerste programma aanmaken.

- Druk op **PRGM**.
- Tenzij je eerder een programma hebt gemaakt, zie je alleen **EXEC EDIT NEW ***
- Blader met **▶** naar **NEW** en druk op **ENTER**.
- Typ een naam in. Een prgm naam is maximaal 8 karakters. Druk vervolgens op **ENTER**.

- Sluit het programma nu met [QUIT].



NEW PRGM

We gaan ons eerste programma aanmaken.

- Druk op **PRGM**.
- Tenzij je eerder een programma hebt gemaakt, zie je alleen **EDIT NEW**.*
- Blader met **▶** naar **NEW** en druk op **ENTER**.
- Typ een naam in. Een prgm naam is maximaal 8 karakters. Druk vervolgens op **ENTER**.

PROGRAM
NAME=MYPRGM01

Merk op dat je een **i**-cursor hebt: **[A-LOCK]** is geactiveerd.

- Sluit het programma nu met **[QUIT]**.



NEW PRGM

We gaan ons eerste programma aanmaken.

- Druk op **PRGM**.
- Tenzij je eerder een programma hebt gemaakt, zie je alleen **EXEC EDIT NEW**.*
- Blader met **▸** naar **NEW** en druk op **ENTER**.
- Typ een naam in. Een prgm naam is maximaal 8 karakters. Druk vervolgens op **ENTER**.

PROGRAM
NAME=MYPRGM01

Merk op dat je een **i**-cursor hebt: **[A-LOCK]** is geactiveerd.

- Sluit het programma nu met **[QUIT]**.



NEW PRGM

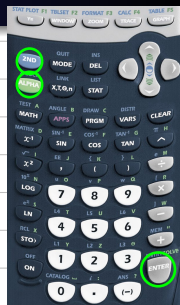
We gaan ons eerste programma aanmaken.

- Druk op **PRGM**.
- Tenzij je eerder een programma hebt gemaakt, zie je alleen **EXEC EDIT NEW**.*
- Blader met **▸** naar **NEW** en druk op **ENTER**.
- Typ een naam in. Een prgm naam is maximaal 8 karakters. Druk vervolgens op **ENTER**.

PROGRAM
NAME=MYPRGM01

Merk op dat je een **⏏**-cursor hebt: [A-LOCK] is geactiveerd.

- Sluit het programma nu met **QUIT**.



NEW PRGM

We gaan ons eerste programma aanmaken.

- Druk op **PRGM**.
- Tenzij je eerder een programma hebt gemaakt, zie je alleen **EXEC EDIT NEW**.*
- Blader met **▸** naar **NEW** en druk op **ENTER**.
- Typ een naam in. Een prgm naam is maximaal 8 karakters. Druk vervolgens op **ENTER**.

PROGRAM
NAME=MYPRGM01

Merk op dat je een **A**-cursor hebt: **[A-LOCK]** is geactiveerd.

- Sluit het programma nu met **[QUIT]**.



EDIT PRGM

Om het programma nu weer te openen, doe:

- Druk op PRGM.
- Je ziet nu een lijst met alle programma's die je kunt uitvoeren.

 *

```
EDIT NEW  
MYPRGM01
```

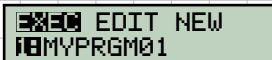
- Blader met ▶ naar EDIT.
- Hier zie je dezelfde lijst. Gebruik ▼ om naar je programma te bladeren en druk op ENTER.
- Als alternatief, kun je ook het nummer intoetsen wat voor je programma staat. Dit is een hotkey om je programma te openen. Ook kun je met ALPHA de eerste letter van je programmanaam intoetsen om snel je programma te vinden (indien je later een grotere lijst met programma's hebt dan 1)



EDIT PRGM

Om het programma nu weer te openen, doe:

- Druk op PRGM.
- Je ziet nu een lijst met alle programma's die je kunt uitvoeren.



EXEC EDIT NEW
MYPRGM01

*

- Blader met ▶ naar EDIT.
- Hier zie je dezelfde lijst. Gebruik ▼ om naar je programma te bladeren en druk op ENTER.
- Als alternatief, kun je ook het nummer intoetsen wat voor je programma staat. Dit is een hotkey om je programma te openen. Ook kun je met ALPHA de eerste letter van je programmanaam intoetsen om snel je programma te vinden (indien je later een grotere lijst met programma's hebt dan 1)





EDIT PRGM

Om het programma nu weer te openen, doe:

- Druk op **PRGM**.
- Je ziet nu een lijst met alle programma's die je kunt uitvoeren.

```
EXEC EDIT NEW
MYPRGM01
```

*

- Blader met  naar **EDIT**.
- Hier zie je dezelfde lijst. Gebruik  om naar je programma te bladeren en druk op **ENTER**.
- Als alternatief, kun je ook het nummer intoetsen wat voor je programma staat. Dit is een hotkey om je programma te openen. Ook kun je met **ALPHA** de eerste letter van je programmanaam intoetsen om snel je programma te vinden (indien je later een grotere lijst met programma's hebt dan 1).



EDIT PRGM

Om het programma nu weer te openen, doe:

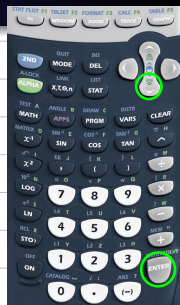
- Druk op PRGM.
- Je ziet nu een lijst met alle programma's die je kunt uitvoeren.



EXEC EDIT NEW
MYPRGM01

*

- Blader met ▶ naar EDIT.
- Hier zie je dezelfde lijst. Gebruik ▼ om naar je programma te bladeren en druk op ENTER.
- Als alternatief, kun je ook het nummer intoetsen wat voor je programma staat. Dit is een hotkey om je programma te openen. Ook kun je met ALPHA de eerste letter van je programmanaam intoetsen om snel je programma te vinden (indien je later een grotere lijst met programma's hebt dan 1)



EDIT PRGM

Om het programma nu weer te openen, doe:

- Druk op PRGM.
- Je ziet nu een lijst met alle programma's die je kunt uitvoeren.



EXEC EDIT NEW
MYPRGM01

*

- Blader met ▶ naar EDIT.
- Hier zie je dezelfde lijst. Gebruik ▼ om naar je programma te bladeren en druk op ENTER.
- Als alternatief, kun je ook het nummer intoetsen wat voor je programma staat. Dit is een hotkey om je programma te openen. Ook kun je met ALPHA de eerste letter van je programmanaam intoetsen om snel je programma te vinden (indien je later een grotere lijst met programma's hebt dan 1)



EDIT PRGM

Om het programma nu weer te openen, doe:

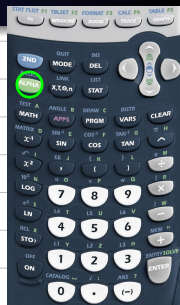
- Druk op PRGM.
- Je ziet nu een lijst met alle programma's die je kunt uitvoeren.



EXEC EDIT NEW
MYPRGM01

*

- Blader met ▶ naar EDIT.
- Hier zie je dezelfde lijst. Gebruik ▼ om naar je programma te bladeren en druk op ENTER.
- Als alternatief, kun je ook het nummer intoetsen wat voor je programma staat. Dit is een hotkey om je programma te openen. Ook kun je met ALPHA de eerste letter van je programmanaam intoetsen om snel je programma te vinden (indien je later een grotere lijst met programma's hebt dan 1).



Deleten/Archiveren van een **PRGM**

TODO?

*

Outline

1 Hoe open je een programma?

2 Variabelen en Datatypes

- Getallen
- Strings
- Overige

*

3 Basic IO

- Disp
- Prompt

4 Exercises

Datatype: Wat is een getal?

Onder het datatype 'getal', verstaan we alle numerieke waarden:

- 1 Gehele getallen / Integers - E.g., -4
- 2 Reeke getallen / Real Numbers - E.g., 3.141592654
- 3 Complexe getallen / Complex Numbers - E.g., $2+3i$

Voor je rekenmachine zijn al deze getallen hetzelfde datatype (in tegenstelling tot op de computer!), en daarom kan dezelfde variabele gebruikt worden voor elk van bovenstaande voorbeelden.

Datatype: Wat is een getal?

Onder het datatype 'getal', verstaan we alle numerieke waarden:

- 1 Gehele getallen / Integers - E.g., -4
- 2 Reeke getallen / Real Numbers - E.g., 3.141592654
- 3 Complexe getallen / Complex Numbers - E.g., $2+3i$

Voor je rekenmachine zijn al deze getallen hetzelfde datatype (in tegenstelling tot op de computer!), en daarom kan dezelfde variabele gebruikt worden voor elk van bovenstaande voorbeelden.

Datatype: Wat is een getal?

Onder het datatype 'getal', verstaan we alle numerieke waarden:

- 1 Gehele getallen / Integers - E.g., -4
- 2 Reeke getallen / Real Numbers - E.g., 3.141592654
- 3 Complexe getallen / Complex Numbers - E.g., $2+3i$.

Voor je rekenmachine zijn al deze getallen hetzelfde datatype (in tegenstelling tot op de computer!), en daarom kan dezelfde variabele gebruikt worden voor elk van bovenstaande voorbeelden.

Datatype: Wat is een getal?

Onder het datatype 'getal', verstaan we alle numerieke waarden:

- 1 Gehele getallen / Integers - E.g., -4
- 2 Reeke getallen / Real Numbers - E.g., 3.141592654
- 3 Complexe getallen / Complex Numbers - E.g., $2+3i$

Voor je rekenmachine zijn al deze getallen hetzelfde datatype (in tegenstelling tot op de computer!), en daarom kan dezelfde variabele gebruikt worden voor elk van bovenstaande voorbeelden.

Datatype: Wat is een getal?

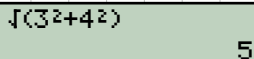
Onder het datatype 'getal', verstaan we alle numerieke waarden:

- ① Gehele getallen / Integers - E.g., -4
- ② Reeel getallen / Real Numbers - E.g., 3.141592654
- ③ Complexe getallen / Complex Numbers - E.g., $2+3i$

Voor je rekenmachine zijn al deze getallen hetzelfde datatype (in tegenstelling tot op de computer!), en daarom kan dezelfde variabele gebruikt worden voor elk van bovenstaande voorbeelden.

Ans

- Wanneer je een som intyped op je rekenmachine, zie je bijvoorbeeld:



$\sqrt{3^2+4^2}$
5

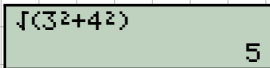
- Met `[ENTER]` herhalen we de laatste som.
Met `[ENTRY]=[2nd][ENTER]` halen* we de som terug op ons scherm. (Probeer het!)
- Met `[ANS][ENTER]` printen we ook hetzelfde antwoord. `Ans` is een “variabele” waarin het antwoord van de vorige bewerking is opgeslagen. Hier geldt dat `Ans` een andere naam is voor 5.
- Bijvoorbeeld, wat is hiervan de uitkomst?



$\sqrt{3^2+4^2}$
5
 $(Ans-4)^{Ans}+1$

Ans

- Wanneer je een som intyped op je rekenmachine, zie je bijvoorbeeld:



$\sqrt{3^2+4^2}$ 5

- Met **[ENTER]** herhalen we de laatste som.
Met **[ENTRY]** = **[2nd][ENTER]** halen* we de som terug op ons scherm. (Probeer het!)
- Met **[ANS][ENTER]** printen we ook hetzelfde antwoord. Ans is een “variabele” waarin het antwoord van de vorige bewerking is opgeslagen. Hier geldt dat Ans een andere naam is voor 5.
- Bijvoorbeeld, wat is hiervan de uitkomst?

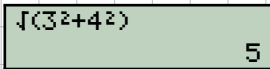


$\sqrt{3^2+4^2}$ 5
 $(Ans-4)^{Ans}+1$



Ans

- Wanneer je een som intyped op je rekenmachine, zie je bijvoorbeeld:



$\sqrt{3^2+4^2}$
5

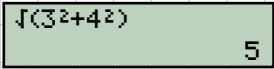
- Met **[ENTER]** herhalen we de laatste som.
Met **[ENTRY]** = **[2nd][ENTER]** halen* we de som terug op ons scherm. (Probeer het!)
- Met **[ANS][ENTER]** printen we ook hetzelfde antwoord. **Ans** is een “variabele” waarin het antwoord van de vorige bewerking is opgeslagen. Hier geldt dat **Ans** een andere naam is voor 5.
- Bijvoorbeeld, wat is hiervan de uitkomst?

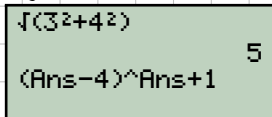


$\sqrt{3^2+4^2}$
5
 $(Ans-4)^{Ans}+1$



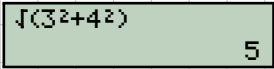
Ans

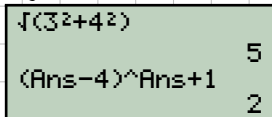
- Wanneer je een som intyped op je rekenmachine, zie je bijvoorbeeld: 
- Met `[ENTER]` herhalen we de laatste som.
Met `[ENTRY]=[2nd][ENTER]` halen* we de som terug op ons scherm. (Probeer het!)
- Met `[ANS][ENTER]` printen we ook hetzelfde antwoord. `Ans` is een “variabele” waarin het antwoord van de vorige bewerking is opgeslagen. Hier geldt dat `Ans` een andere naam is voor `5`.
- Bijvoorbeeld, wat is hiervan de uitkomst?



$\sqrt{3^2+4^2}$ 5
 $(Ans-4)^{Ans}+1$

Ans

- Wanneer je een som intyped op je rekenmachine, zie je bijvoorbeeld: 
- Met `[ENTER]` herhalen we de laatste som.
Met `[ENTRY]=[2nd][ENTER]` halen* we de som terug op ons scherm. (Probeer het!)
- Met `[ANS][ENTER]` printen we ook hetzelfde antwoord. `Ans` is een “variabele” waarin het antwoord van de vorige bewerking is opgeslagen. Hier geldt dat `Ans` een andere naam is voor 5.
- Bijvoorbeeld, wat is hiervan de uitkomst?



$\sqrt{3^2+4^2}$ 5
 $(Ans-4)^{Ans}+1$ 2

Vervang alle `Ans` door 5!

Ans

Meer voorbeelden

Om te testen of je het begrijpt, wat zijn van de volgende sommen de uitkomsten?

3+4

7

Ans-Ans*2

0

0

e^Ans

En wat moeilijker:

3

3

Ans+2

Ans-2

Ans

Meer voorbeelden

Om te testen of je het begrijpt, wat zijn van de volgende sommen de uitkomsten?

`3+4`

7

`Ans-Ans*2`

0

`e^Ans`

0

En wat moeilijker:

`3`

3

`Ans+2`

`Ans-2`

Ans

Meer voorbeelden

Om te testen of je het begrijpt, wat zijn van de volgende sommen de uitkomsten?

3+4

7

Ans-Ans*2

-7

0

0

e^Ans

En wat moeilijker:

3

3

Ans+2

Ans-2

Ans

Meer voorbeelden

Om te testen of je het begrijpt, wat zijn van de volgende sommen de uitkomsten?

3+4

7

Ans-Ans*2

-7

0

0

e^Ans

En wat moeilijker:

3

3

Ans+2

Ans-2

Ans

Meer voorbeelden

Om te testen of je het begrijpt, wat zijn van de volgende sommen de uitkomsten?

3+4

7

Ans-Ans*2

-7

0

0

e^Ans

1

En wat moeilijker:

3

3

Ans+2

Ans-2

Ans

Meer voorbeelden

Om te testen of je het begrijpt, wat zijn van de volgende sommen de uitkomsten?

3+4

7

Ans-Ans*2

-7

0

0

e^Ans

1

En wat moeilijker:

3

3

Ans+2

Ans-2

Ans

Meer voorbeelden

Om te testen of je het begrijpt, wat zijn van de volgende sommen de uitkomsten?

3+4	7
Ans-Ans*2	-7

0	0
e^Ans	1

En wat moeilijker:

3	3
Ans+2	5
Ans-2	3

Merk op dat hierbij **Ans** eerst gelijk is aan 3, maar vervolgens wordt overschreven door de nieuwe uitkomst: 5!

Hierdoor evalueert de laatste som als $5-2=3$.

Ans

Meer voorbeelden

En een laatste moeilijke:

$\sqrt{9}$

3

$\text{Ans}/3$

$\cos(\pi \text{Ans})$

$\text{Ans}+3$

*

Ans

Meer voorbeelden

En een laatste moeilijke:

$\sqrt{9}$	3
Ans/3	1
$\cos(\pi \text{Ans})$	-1
Ans+3	2

*

Net als in de vorige opgave, **Ans** begint op 3 en wordt daarna overschreven met de waarde 1.

De derde som is dan $\cos(\pi * 1)$. Merk hierbij op dat, net zoals je met $2x$ eigenlijk bedoelt $2 * x$, hetzelfde op je rekenmachine geldt: $\pi \text{Ans} = \pi * \text{Ans}$.

Als laatste stap is **Ans** gelijk aan -1, waardoor de laatste som gelijk is aan 2.

Ans

Meer voorbeelden

En een laatste moeilijke:

$\sqrt{9}$	3
Ans/3	1
$\cos(\pi \text{Ans})$	-1
Ans+3	2

*

Net als in de vorige opgave, **Ans** begint op 3 en wordt daarna overschreven met de waarde 1.

De derde som is dan $\cos(\pi * 1)$. Merk hierbij op dat, net zoals je met 2x eigenlijk bedoelt $2 * x$, hetzelfde op je rekenmachine geldt: $\pi \text{Ans} = \pi * \text{Ans}$.

Als laatste stap is **Ans** gelijk aan -1, waardoor de laatste som gelijk is aan 2.

Ans

Meer voorbeelden

En een laatste moeilijke:

$\sqrt{9}$	3
Ans/3	1
$\cos(\pi \text{Ans})$	-1
Ans+3	2

*

Net als in de vorige opgave, **Ans** begint op 3 en wordt daarna overschreven met de waarde 1.

De derde som is dan $\cos(\pi * 1)$. Merk hierbij op dat, net zoals je met 2x eigenlijk bedoelt $2 * x$, hetzelfde op je rekenmachine geldt: $\pi \text{Ans} = \pi * \text{Ans}$.

Als laatste stap is **Ans** gelijk aan -1, waardoor de laatste som gelijk is aan 2.

Andere variabelen voor getallen

Variabelen zijn doosjes.

- In het algemeen geldt dat een variabele een doosje is. In dat doosje kan je iets stoppen (een getal), en je kunt het er later weer uithalen. Een variabele* is dus niets anders dan een 'container' voor een getal.
- Net zagen we dat de inhoud van het doosje `Ans` automatisch vervangen werd, elke keer dat we een berekening uit voerde.
- Alle andere doosjes moeten we met de hand openen en vullen.

Andere variabelen voor getallen

Variabelen zijn doosjes.

- In het algemeen geldt dat een variabele een doosje is. In dat doosje kan je iets stoppen (een getal), en je kunt het er later weer uithalen. Een variabele* is dus niets anders dan een 'container' voor een getal.
- Net zagen we dat de inhoud van het doosje ~~fin~~ automatisch vervangen werd, elke keer dat we een berekening uit voerde.
- Alle andere doosjes moeten we met de hand openen en vullen.

Andere variabelen voor getallen

Variabelen zijn doosjes.

- In het algemeen geldt dat een variabele een doosje is. In dat doosje kan je iets stoppen (een getal), en je kunt het er later weer uithalen. Een variabele^{*} is dus niets anders dan een 'container' voor een getal.
- Net zagen we dat de inhoud van het doosje ~~fin~~ automatisch vervangen werd, elke keer dat we een berekening uit voerde.
- Alle andere doosjes moeten we met de hand openen en vullen.

Andere variabelen voor getallen

Variabelen zijn doosjes.

- In het algemeen geldt dat een variabele een doosje is. In dat doosje kan je iets stoppen (een getal), en je kunt het er later weer uithalen. Een variabele^{*} is dus niets anders dan een 'container' voor een getal.
- Net zagen we dat de inhoud van het doosje ~~Ans~~ automatisch vervangen werd, elke keer dat we een berekening uit voerde.
- Alle andere doosjes moeten we met de hand openen en vullen.

Andere variabelen voor getallen

Variabelen zijn doosjes.

- In het algemeen geldt dat een variabele een doosje is. In dat doosje kan je iets stoppen (een getal), en je kunt het er later weer uithalen. Een variabele^{*} is dus niets anders dan een 'container' voor een getal.
- Net zagen we dat de inhoud van het doosje ~~Ans~~ automatisch vervangen werd, elke keer dat we een berekening uit voerde.
- Alle andere doosjes moeten we met de hand openen en vullen.

Andere variabelen voor getallen

Het lezen van variabelen

- Met **[ALPHA]** kunnen we alle letters, inclusief θ , bereiken. Dit zijn alle variabelen die voor getallen bedoeld zijn.
- Bijvoorbeeld, **[ALPHA][MATH]=R**.
- Je kunt de waarde van een variabele zien door de variabele te evalueren met **[ENTER]**:

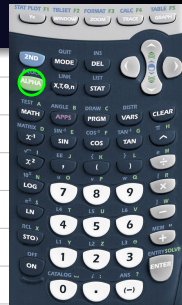
Y
2.585786438

Probeer zelf wat variabelen!

Merk op dat **X** en **Y** variabelen zijn die door de grafische functies (**[Y=]**) worden gebruikt.

- Als alternatief, kun je **[RCL](=[2nd][STO▶])+Y+[ENTER]** gebruiken:

Rcl Y 2.585786438



Andere variabelen voor getallen

Het lezen van variabelen

- Met **[ALPHA]** kunnen we alle letters, inclusief θ , bereiken. Dit zijn alle variabelen die voor getallen bedoeld zijn.
- Bijvoorbeeld, **[ALPHA][MATH]**= π .
- Je kunt de waarde van een variabele zien door de variabele te evalueren met **[ENTER]**:

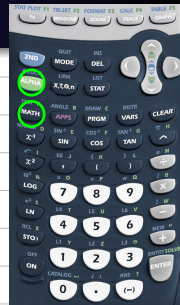
π
2.585786438

Probeer zelf wat variabelen!

Merk op dat X en Y variabelen zijn die door de grafische functies ($Y=$) worden gebruikt.

- Als alternatief, kun je **[RCL](=[2nd][STO])**+ π +**[ENTER]** gebruiken:

Rcl π 2.585786438



Andere variabelen voor getallen

Het lezen van variabelen

- Met **[ALPHA]** kunnen we alle letters, inclusief θ , bereiken. Dit zijn alle variabelen die voor getallen bedoeld zijn.
- Bijvoorbeeld, **[ALPHA][MATH]**= π .
- Je kunt de waarde van een variabele zien door de variabele te evalueren met **[ENTER]**:

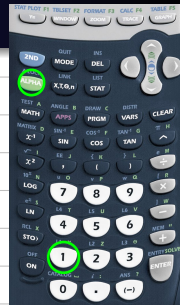
Y
2.585786438

Probeer zelf wat variabelen!

Merk op dat X en Y variabelen zijn die door de grafische functies (**[Y=]**) worden gebruikt.

- Als alternatief, kun je **[RCL (=2nd)][STO▶)]+Y+[ENTER]** gebruiken:

Rcl Y 2.585786438



Andere variabelen voor getallen

Het lezen van variabelen

- Met **[ALPHA]** kunnen we alle letters, inclusief θ , bereiken. Dit zijn alle variabelen die voor getallen bedoeld zijn.
- Bijvoorbeeld, **[ALPHA][MATH]** = π .
- Je kunt de waarde van een variabele zien door de variabele te evalueren met **[ENTER]**:

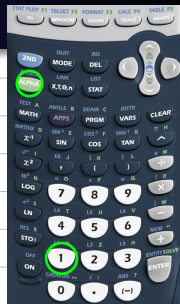
Y
2.585786438

Probeer zelf wat variabelen!

Merk op dat X en Y variabelen zijn die door de grafische functies (**[Y=]**) worden gebruikt.

- Als alternatief, kun je **[RCL (=2nd)][STO▶]** + **Y** + **[ENTER]** gebruiken:

Rcl Y 2.585786438



Andere variabelen voor getallen

Het lezen van variabelen

- Met **[ALPHA]** kunnen we alle letters, inclusief θ , bereiken. Dit zijn alle variabelen die voor getallen bedoeld zijn.
- Bijvoorbeeld, **[ALPHA][MATH]**=**π**.
- Je kunt de waarde van een variabele zien door de variabele te evalueren met **[ENTER]**:

Y
2.585786438

Probeer zelf wat variabelen!

Merk op dat **X** en **Y** variabelen zijn die door de grafische functies (**[Y=]**) worden gebruikt.

- Als alternatief, kun je **[RCL (=2nd)][STO▶)]+Y+[ENTER]** gebruiken:

Rcl Y 2.585786438



Andere variabelen voor getallen

Het lezen van variabelen

- Met **[ALPHA]** kunnen we alle letters, inclusief θ , bereiken. Dit zijn alle variabelen die voor getallen bedoeld zijn.
- Bijvoorbeeld, **[ALPHA][MATH]=R**.
- Je kunt de waarde van een variabele zien door de variabele te evalueren met **[ENTER]**:

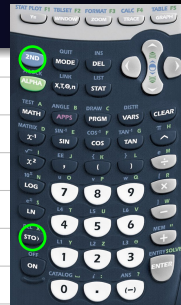
Y
2.585786438

Probeer zelf wat variabelen!

Merk op dat **X** en **Y** variabelen zijn die door de grafische functies (**[Y=]**) worden gebruikt.

- Als alternatief, kun je **[RCL](=[2nd][STO▶]) + Y + [ENTER]** gebruiken:

Rcl Y 2.585786438



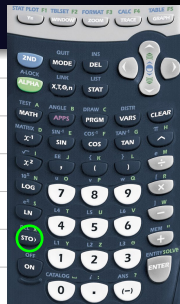
Andere variabelen voor getallen

Het storen van getallen in variabelen

Je kunt iets met **STO➡** als volgt in een doosje stoppen:

6➡A stored het getal 6 in de variabele A.*

“Store into a variable” is de programmeerjargon voor een getal in een doosje stoppen.



Andere variabelen voor getallen

Het storen van getallen in variabelen

Je kunt iets met **STO** als volgt in een doosje stoppen:
6→A stored het getal 6 in de variabele A.

“Store into a variable” is de programmeerjargon voor een getal in een doosje stoppen.



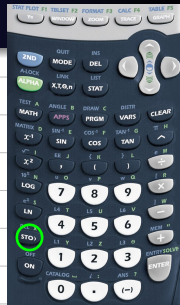
Andere variabelen voor getallen

Het storen van getallen in variabelen

Je kunt iets met **STO** als volgt in een doosje stoppen:

6 → A stored het getal 6 in de variabele A.

“Store into a variable” is de programmeerjargon voor een getal in een doosje stoppen.



Andere variabelen voor getallen

Het storen van getallen in variabelen

Je kunt iets met **STO➡** als volgt in een doosje stoppen:
6➡A stored het getal 6 in de variabele A.*

“Store into a variable” is de programmeerjargon voor een getal in een doosje stoppen.

En dat is alles wat je nodig hebt...!



Andere variabelen voor getallen

Voorbeelden

$3 \rightarrow A$	3
$2 \rightarrow B$	2
A^B	

*

$3 \rightarrow A$	3
$2 \rightarrow A$	2
A^A	

Andere variabelen voor getallen

Voorbeelden

3→A	
	3
2→B	
	2
A^B	
	9

*

Vervang A door 3 en B door 2 en de som is simpelweg $3^2 = 9$.

3→A	
	3
2→A	
	2
A^A	

Andere variabelen voor getallen

Voorbeelden

$3 \rightarrow A$	
	3
$2 \rightarrow B$	
	2
A^B	
	9

*

Vervang A door 3 en B door 2 en de som is simpelweg $3^2 = 9$.

$3 \rightarrow A$	
	3
$2 \rightarrow A$	
	2
A^A	

Andere variabelen voor getallen

Voorbeelden

3→A	
	3
2→B	
	2
A^B	
	9

*

Vervang A door 3 en B door 2 en de som is simpelweg $3^2 = 9$.

3→A	
	3
2→A	
	2
A^A	
	4

Vervang A door 2 en de som is simpelweg $2^2 = 4$. Merk hierbij op dat alleen het *laatste* wat je in een doosje stopt in het doosje zit!

Andere variabelen voor getallen

Ultiem voorbeeld

3→A

3

A→B

3

B+A→C

0→A

A+B+C

*

Andere variabelen voor getallen

Ultiem voorbeeld

$3 \rightarrow A$	3
$A \rightarrow B$	3
$B + A \rightarrow C$	6
$0 \rightarrow A$	0
$A + B + C$	9

*

Merk op dat op het moment dat C wordt uitgerekend, A nog gelijk is aan 3 (en niet 0).

Andere variabelen voor getallen

Ultiem voorbeeld

$3 \rightarrow A$	3
$A \rightarrow B$	3
$B + A \rightarrow C$	6
$0 \rightarrow A$	0
$A + B + C$	9

*

Merk op dat op het moment dat C wordt uitgerekend, A nog gelijk is aan 3 (en niet 0).

Speel zelf ook met wat gecompliceerde voorbeelden en kijk of je begrijpt wat je rekenmachine uitrekent!

Outline

1 Hoe open je een programma?

2 Variabelen en Datatypes

- Getallen
- **Strings**
- Overige

*

3 Basic IO

- Disp
- Prompt

4 Exercises

Datatype: Wat is een string?

Volgens “www.thefreedictionary.com”:

- 1 Material made of drawn-out, twisted fiber, used for fastening, tying, or lacing.
- 2 (Music) A cord stretched on an instrument and struck, plucked, or bowed to produce tones.*
- 3 (Physics) One of the extremely minute objects that form the basis of string theory.
- 4 A number of objects arranged in a line: a string of islands.

Datatype: Wat is een string?

Volgens “www.thefreedictionary.com”:

- 1 Material made of drawn-out, twisted fiber, used for fastening, tying, or lacing.
- 2 (Music) A cord stretched on an instrument and struck, plucked, or bowed to produce tones.*
- 3 (Physics) One of the extremely minute objects that form the basis of string theory.
- 4 A number of objects arranged in a line: a string of islands.

Datatype: Wat is een string?

Volgens “www.thefreedictionary.com”:

- 1 Material made of drawn-out, twisted fiber, used for fastening, tying, or lacing.
- 2 (Music) A cord stretched on an instrument and struck, plucked, or bowed to produce tones.*
- 3 (Physics) One of the extremely minute objects that form the basis of string theory.
- 4 A number of objects arranged in a line: a string of islands.

Datatype: Wat is een string?

Volgens “www.thefreedictionary.com”:

- 1 Material made of drawn-out, twisted fiber, used for fastening, tying, or lacing.
- 2 (Music) A cord stretched on an instrument and struck, plucked, or bowed to produce tones.*
- 3 (Physics) One of the extremely minute objects that form the basis of string theory.
- 4 A number of objects arranged in a line: a string of islands.

Datatype: Wat is een string?

Volgens “www.thefreedictionary.com”:

- 1 Material made of drawn-out, twisted fiber, used for fastening, tying, or lacing.
- 2 (Music) A cord stretched on an instrument and struck, plucked, or bowed to produce tones.*
- 3 (Physics) One of the extremely minute objects that form the basis of string theory.
- 4 A number of objects arranged in a line: a string of islands.

Datatype: Wat is een string?

Volgens “www.thefreedictionary.com”:

- 1 Material made of drawn-out, twisted fiber, used for fastening, tying, or lacing.
- 2 (Music) A cord stretched on an instrument and struck, plucked, or bowed to produce tones.*
- 3 (Physics) One of the extremely minute objects that form the basis of string theory.
- 4 A number of objects arranged in a line: a string of islands.
- 5 (Computers) A linear sequence of characters, words, or other data.

Een ‘string’ is gewoon een regel tekst!

Datatype: Wat is een string?

Volgens “www.thefreedictionary.com”:

- 1 Material made of drawn-out, twisted fiber, used for fastening, tying, or lacing.
- 2 (Music) A cord stretched on an instrument and struck, plucked, or bowed to produce tones.*
- 3 (Physics) One of the extremely minute objects that form the basis of string theory.
- 4 A number of objects arranged in a line: a string of islands.
- 5 (Computers) A linear sequence of characters, words, or other data.

Een ‘string’ is gewoon een regel tekst!

Waar heb je een string voor nodig?

Er zijn twee relevante toepassingen:

- 1 Het weergeven van tekst aan de gebruiker van je programma. *

- 1 De gebruiker uitleggen wat hij moet inputten.
- 2 De gebruiker een resultaat in woorden laten zien.
- 3 Etc.: Communiceren!

- 2 Vergelijkingen bij Y_1 zijn ook tekstregels:

"X²" → Y_1 verandert de Y_1 grafiek naar de functie $y = x^2$.
Daar gaan we in een later college op in.

Waar heb je een string voor nodig?

Er zijn twee relevante toepassingen:

- 1 Het weergeven van tekst aan de gebruiker van je programma. *

- 1 De gebruiker uitleggen wat hij moet inputten.
- 2 De gebruiker een resultaat in woorden laten zien.
- 3 Etc.: Communiceren!

- 2 Vergelijkingen bij Y_1 zijn ook tekstregels:

"X²" → Y_1 verandert de Y_1 grafiek naar de functie $y = x^2$.
Daar gaan we in een later college op in.

Waar heb je een string voor nodig?

Er zijn twee relevante toepassingen:

- 1 Het weergeven van tekst aan de gebruiker van je programma. *

- 1 De gebruiker uitleggen wat hij moet inputten.
- 2 De gebruiker een resultaat in woorden laten zien.
- 3 Etc.: Communiceren!

- 2 Vergelijkingen bij Y_1 zijn ook tekstregels:

"X²" → Y_1 verandert de Y_1 grafiek naar de functie $y = x^2$.
Daar gaan we in een later college op in.

Waar heb je een string voor nodig?

Er zijn twee relevante toepassingen:

- 1 Het weergeven van tekst aan de gebruiker van je programma. *

- 1 De gebruiker uitleggen wat hij moet inputten.
- 2 De gebruiker een resultaat in woorden laten zien.
- 3 Etc.: Communiceren!

- 2 Vergelijkingen bij Y_1 zijn ook tekstregels:

"X²" → Y_1 verandert de Y_1 grafiek naar de functie $y = x^2$.
Daar gaan we in een later college op in.

Waar heb je een string voor nodig?

Er zijn twee relevante toepassingen:

- 1 Het weergeven van tekst aan de gebruiker van je programma. *

- 1 De gebruiker uitleggen wat hij moet inputten.
- 2 De gebruiker een resultaat in woorden laten zien.
- 3 Etc.: Communiceren!

- 2 Vergelijkingen bij Y_1 zijn ook tekstregels:

"X²" → Y_1 verandert de Y_1 grafiek naar de functie $y = x^2$.
Daar gaan we in een later college op in.

Waar heb je een string voor nodig?

Er zijn twee relevante toepassingen:

- 1 Het weergeven van tekst aan de gebruiker van je programma.
*
 - 1 De gebruiker uitleggen wat hij moet inputten.
 - 2 De gebruiker een resultaat in woorden laten zien.
 - 3 Etc.: Communiceren!
- 2 Vergelijkingen bij $Y=$ zijn ook tekstregels:

"X²"→Y₁

 verandert de Y₁ grafiek naar de functie $y = x^2$.
Daar gaan we in een later college op in.



Het storen van een string in een variabele

- Gebruik dezelfde knop als voor getallen: **[STO➡]**.
- Voor het string datatype, moeten we de **Str#** variabelen gebruiken.
Deze staan in het **[VARS]**-menu onder "String...".

- Bijvoorbeeld:

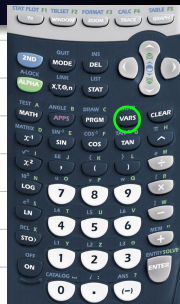
"HELLO WORLD"→Str1

- Merk hierbij op dat een string tussen quotatie-tekens (") moet staan: "="**[ALPHA][+]**.
De spatie maak je met **␣**=**[ALPHA][0]**.



Het storen van een string in een variabele

- Gebruik dezelfde knop als voor getallen: **[STO▶]**.
- Voor het string datatype, moeten we de **Str#** variabelen gebruiken.
Deze staan in het **[VARS]**-menu onder "String...".



- Bijvoorbeeld:

```
"HELLO WORLD"→Str1
```
- Merk hierbij op dat een string tussen quotatie-tekens (") moet staan: "**[ALPHA][+]**".
De spatie maak je met **[ALPHA][0]**.

Het storen van een string in een variabele

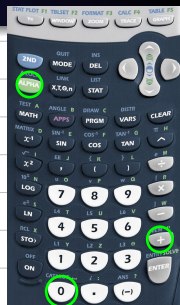
- Gebruik dezelfde knop als voor getallen: **[STO▶]**.
- Voor het string datatype, moeten we de **Str#** variabelen gebruiken.

Deze staan in het **[VARS]**-menu onder "String...".

- Bijvoorbeeld:

```
"HELLO WORLD"→Str1
```

- Merk hierbij op dat een string tussen quotatie-tekens (") moet staan: "="**[ALPHA]****[+]**.
De spatie maak je met **␣**=**[ALPHA]****[0]**.



Het storen van een string in een variabele

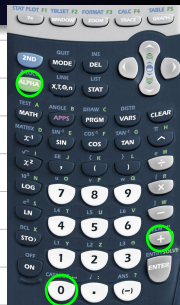
- Gebruik dezelfde knop als voor getallen: **[STO▶]**.
- Voor het string datatype, moeten we de **Str#** variabelen gebruiken.

Deze staan in het **[VARS]**-menu onder "String...".

- Bijvoorbeeld:

```
"HELLO WORLD"→Str1
```

- Merk hierbij op dat een string tussen quotatie-tekens (") moet staan: **"**=**[ALPHA][+]**.
De spatie maak je met **␣**=**[ALPHA][0]**.



Strings

Voorbeelden

```
"HELLO WORLD"→Str1  
HELLO WORLD  
Str1+"?"
```

*

```
"HELLO WORLD"→Str1  
HELLO WORLD  
"Str1 = "+Str1
```

Strings

Voorbeelden

```
"HELLO WORLD"→Str1  
HELLO WORLD  
Str1+"?"  
HELLO WORLD?
```

De tekst HELLO WORLD wordt opgeslagen in `Str1`. Vervolgens voegen we er een vraagteken, `?`, aan toe en weergeven we de uitkomst.

```
"HELLO WORLD"→Str1  
HELLO WORLD  
"Str1 = "+Str1
```

Strings

Voorbeelden

```
"HELLO WORLD"→Str1  
HELLO WORLD  
Str1+"?"  
HELLO WORLD?
```

De tekst HELLO WORLD wordt opgeslagen in `Str1`. Vervolgens voegen we er een vraagteken, `?`, aan toe en weergeven we de uitkomst.

```
"HELLO WORLD"→Str1  
HELLO WORLD  
"Str1 = "+Str1
```

Strings

Voorbeelden

```
"HELLO WORLD"→Str1  
HELLO WORLD  
Str1+"?"  
HELLO WORLD?
```

De tekst HELLO WORLD wordt opgeslagen in `Str1`. Vervolgens voegen we er een vraagteken, `?`, aan toe en weergeven we de uitkomst.

```
"HELLO WORLD"→Str1  
HELLO WORLD  
"Str1 = "+Str1  
Str1 = HELLO WORLD
```

Merk op dat we ook *functies en variabelen* tussen quotatie-tekens kunnen zetten, om de naam van die functie of variabele om te zetten naar een string.

Strings

Oprachten

2+2	4
2+2	5

Hoe kun je deze tekst op je scherm weergeven?

*

```
"HI"→Str1  
"3*2="→Str2  
Str1+", KEVIN"  
Str2+6
```

Wat is de output van je rekenmachine voor deze som? En waarom?

Strings

Opdrachten

2+2	4
2+2	5

Hoe kun je deze tekst op je scherm weergeven?

De eerste regel is de standaard output van de rekenmachine voor de som $2+2$.

De tweede regel is een string met genoeg spaties om 5 rechts te zetten.

```
"HI"→Str1  
"3*2="→Str2  
Str1+", KEVIN"  
  
Str2+6
```

Wat is de output van je rekenmachine voor deze som? En waarom?

Strings

Opdrachten

```
2+2      4  
2+2      5
```

```
"HI"→Str1  
"3*2="→Str2  
Str1+", KEVIN"  
HI, KEVIN  
Str2+6
```

Hoe kun je deze tekst op je scherm weergeven?

De eerste regel is de standaard output van de rekenmachine voor de som $2+2$.
De tweede regel is een string met genoeg spaties om 5 rechts te zetten.

Wat is de output van je rekenmachine voor deze som? En waarom?

Strings

Opdrachten

```
2+2      4  
2+2      5
```

```
"HI"→Str1  
"3*2="→Str2  
Str1+", KEVIN"  
HI, KEVIN  
Str2+6  
ERR: DATA TYPE
```

Hoe kun je deze tekst op je scherm weergeven?

De eerste regel is de standaard output van de rekenmachine voor de som $2+2$.

De tweede regel is een string met genoeg spaties om 5 rechts te zetten.

Wat is de output van je rekenmachine voor deze som? En waarom?

6 is een getal, niet een string. Je kunt een string en een getal niet bij elkaar optellen.

`Str2+"6"` had wel gewerkt.

Outline

1 Hoe open je een programma?

2 Variabelen en Datatypes

- Getallen
- Strings
- Overige

*

3 Basic IO

- Disp
- Prompt

4 Exercises

Andere datatypen

Andere datatypen verdienen ook een korte benoeming:

- **Lists.** Een collectie getallen.
Nuttig om een berekening in 1 keer op heel veel getallen tegelijk uit te voeren. Ook kun je gebruik maken van functies van de rekenmachine, zoals `mean` (gemiddelde), op lijsten.
- **Matrix.** Een soort 2D list. Geen middelbare school wiskunde.
- **Boolean.** "Juist" of "Onjuist"

Deze datatypen/variabelen zullen we deels later behandelen.

Andere datatypen

Andere datatypen verdienen ook een korte benoeming:

- **Lists.** Een collectie getallen.
Nuttig om een berekening in 1 keer op heel veel getallen tegelijk uit te voeren. Ook kun je gebruik maken van functies van de rekenmachine, zoals ~~mean~~^{*} (gemiddelde), op lijsten.
- **Matrix.** Een soort 2D list. Geen middelbare school wiskunde.
- **Boolean.** "Juist" of "Onjuist"

Deze datatypen/variabelen zullen we deels later behandelen.

Andere datatypen

Andere datatypen verdienen ook een korte benoeming:

- **Lists.** Een collectie getallen.
Nuttig om een berekening in 1 keer op heel veel getallen tegelijk uit te voeren. Ook kun je gebruik maken van functies van de rekenmachine, zoals ~~mean~~^{*} (gemiddelde), op lijsten.
- **Matrix.** Een soort 2D list. Geen middelbare school wiskunde.
- **Boolean.** "Juist" of "Onjuist"

Deze datatypen/variabelen zullen we deels later behandelen.

Andere datatypen

Andere datatypen verdienen ook een korte benoeming:

- **Lists.** Een collectie getallen.
Nuttig om een berekening in 1 keer op heel veel getallen tegelijk uit te voeren. Ook kun je gebruik maken van functies van de rekenmachine, zoals ~~mean~~^{*} (gemiddelde), op lijsten.
- **Matrix.** Een soort 2D list. Geen middelbare school wiskunde.
- **Boolean.** “Juist” of “Onjuist”

Deze datatypen/variabelen zullen we deels later behandelen.

Andere datatypen

Andere datatypen verdienen ook een korte benoeming:

- **Picture.** Je kunt met de functies in het [DRAW]-menu tekeningen op je rekenmachine maken. Deze kun je in `Pic#` variabelen opslaan. *
- **Graph.** Je kunt alle functies die je bij `Y=` hebt staan opslaan in een `GDB#` variabele (`GDB` = Graph Database). Daarmee kun je op een later tijdstip de functies/grafieken weer tevoorschijn toveren.

Deze datatypen/variabelen zullen we deels later behandelen.

Andere datatypen

Andere datatypen verdienen ook een korte benoeming:

- **Picture.** Je kunt met de functies in het [DRAW]-menu tekeningen op je rekenmachine maken. Deze kun je in **Pic#** variabelen opslaan. *
- **Graph.** Je kunt alle functies die je bij $Y=$ hebt staan opslaan in een **GDB#** variabele (GDB = Graph Database). Daarmee kun je op een later tijdstip de functies/grafieken weer tevoorschijn toveren.

Deze datatypen/variabelen zullen we deels later behandelen.

Andere datatypen

Andere datatypen verdienen ook een korte benoeming:

- **Picture.** Je kunt met de functies in het [DRAW]-menu tekeningen op je rekenmachine maken. Deze kun je in **Pic#** variabelen opslaan. *
- **Graph.** Je kunt alle functies die je bij $\boxed{Y=}$ hebt staan opslaan in een **GDB#** variabele (GDB = Graph Database). Daarmee kun je op een later tijdstip de functies/grafieken weer tevoorschijn toveren.

Deze datatypen/variabelen zullen we deels later behandelen.

Outline

1 Hoe open je een programma?

2 Variabelen en Datatypes

- Getallen
- Strings
- Overige

*

3 Basic IO

- Disp
- Prompt

4 Exercises

Het PRGM-I/O menu: input & output.

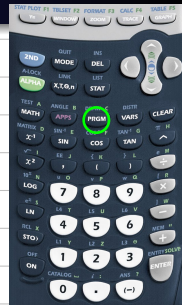
We gaan nu een programma tekst laten weergeven.

- Maak zelf een nieuw programma aan: `DISP1`.
- Open het programma (`PRGM` `▶` `DISP1`).
- Druk op `PRGM` om het programmeermenu te openen. *
- Bij menu `I/O` staat alle Input en Output.
- We zijn nu geïnteresseerd in `Disp`: "Display".

Het PRGM-I/O menu: input & output.

We gaan nu een programma tekst laten weergeven.

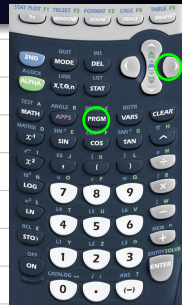
- Maak zelf een nieuw programma aan: **DISP1**.
- Open het programma (**PRGM** **▶** **DISP1**).
- Druk op **PRGM** om het programmeermenu te openen. *
- Bij menu **I/O** staat alle Input en Output.
- We zijn nu geïnteresseerd in **Disp**: "Display".



Het PRGM-I/O menu: input & output.

We gaan nu een programma tekst laten weergeven.

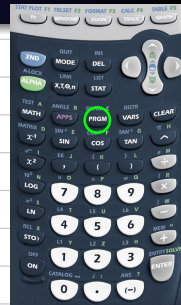
- Maak zelf een nieuw programma aan: **DISP1**.
- Open het programma (**PRGM** **▶** **DISP1**).
- Druk op **PRGM** om het programmeermenu te openen. *
- Bij menu **I/O** staat alle Input en Output.
- We zijn nu geïnteresseerd in **Disp**: "Display".



Het PRGM-I/O menu: input & output.

We gaan nu een programma tekst laten weergeven.

- Maak zelf een nieuw programma aan: **DISP1**.
- Open het programma (**PRGM** **▶** **DISP1**).
- Druk op **PRGM** om het programmeermenu te openen.
- Bij menu **I/O** staat alle Input en Output.
- We zijn nu geïnteresseerd in **Disp**: "Display".

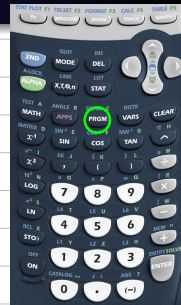


```
PRGM I/O EXEC
1:If
2:Then
3:Else
4:For(
5:While
6:Repeat(
7:End
```


Het PRGM-I/O menu: input & output.

We gaan nu een programma tekst laten weergeven.

- Maak zelf een nieuw programma aan: **DISP1**.
- Open het programma (**PRGM** **▶** **DISP1**).
- Druk op **PRGM** om het programmeermenu te openen. *
- Bij menu **I/O** staat alle Input en Output.
- We zijn nu geïnteresseerd in **Disp**: "Display".

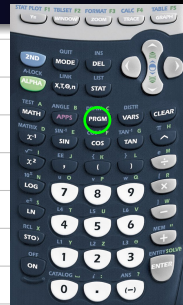


```
CTL I/O EXEC
1:Input
2:Prompt
3:Disp
4:DispGraph
5:DispTable
6:Output(
7:getKey
```

Het PRGM-I/O menu: input & output.

We gaan nu een programma tekst laten weergeven.

- Maak zelf een nieuw programma aan: **DISP1**.
- Open het programma (**PRGM** **▶** **DISP1**).
- Druk op **PRGM** om het programmeermenu te openen. *
- Bij menu **I/O** staat alle Input en Output.
- We zijn nu geïnteresseerd in **Disp**: “Display”.



```
CTL I/O EXEC
1:Input
2:Prompt
3:Disp
4:DispGraph
5:DispTable
6:Output(
7:getKey
```

Hello World!!

We maken nu een "Hello World" programma:

- Kies de **Disp** functie uit het **I/O** menu.
- Zet je rekenmachine op **[A-LOCK]**, zodat je letters kunt typen, zonder herhaaldelijk op **[ALPHA]** te drukken.
- Typ vervolgens "HELLO WORLD" achter **Disp**.
- Done! Execute het programma om te testen!



```
PROGRAM:DISP1
:
```

Hello World!!

We maken nu een “Hello World” programma:

- Kies de **DISF** functie uit het **I/O** menu.
- Zet je rekenmachine op **[A-LOCK]**, zodat je letters kunt typen, zonder herhaaldelijk op **[ALPHA]** te drukken. *
- Typ vervolgens **"HELLO WORLD"** achter **DISF**.
- Done! Execute het programma om te testen!



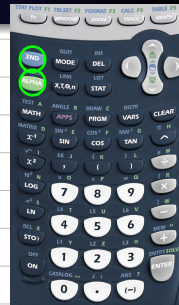
```
CTL  EXEC
1:Input
2:Prompt
3:Disp
4:DispGraph
5:DispTable
6:Output(
7:getKey
```

```
PROGRAM: DISP1
:Disp
```

Hello World!!

We maken nu een "Hello World" programma:

- Kies de **Disp** functie uit het **I/O** menu.
- Zet je rekenmachine op [A-LOCK], zodat je letters kunt typen, zonder herhaaldelijk op **[ALPHA]** te drukken.
- Typ vervolgens "HELLO WORLD" achter **Disp**
- Done! Execute het programma om te testen!



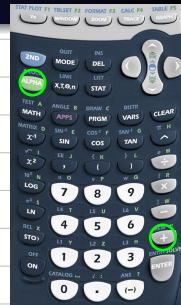
```
CTL [I/O] EXEC
1:Input
2:Prompt
[DISP]Disp
4:DispGraph
5:DispTable
6:Output(
7↓getKey
```


```
PROGRAM:DISP1
:Disp 0
```

Hello World!!

We maken nu een "Hello World" programma:

- Kies de **Disp** functie uit het **I/O** menu.
- Zet je rekenmachine op [A-LOCK], zodat je letters kunt typen, zonder herhaaldelijk op **[ALPHA]** te drukken.
- Typ vervolgens "HELLO WORLD" achter **Disp**.
- Done! Execute het programma om te testen!



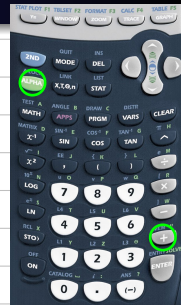
```
CTL  EXEC
1:Input
2:Prompt
3:Disp
4:DispGraph
5:DispTable
6:Output(
7:getKey
```

```
PROGRAM:DISP1
:Disp "HELLO WORLD"
```

Hello World!!

We maken nu een "Hello World" programma:

- Kies de **Disp** functie uit het **I/O** menu.
- Zet je rekenmachine op [A-LOCK], zodat je letters kunt typen, zonder herhaaldelijk op **[ALPHA]** te drukken.
- Typ vervolgens "HELLO WORLD" achter **Disp**.
- Done! Execute het programma om te testen!



```
CTL  EXEC
1:Input
2:Prompt
3:Disp
4:DispGraph
5:DispTable
6:Output(
7↓getKey
```

```
PROGRAM:DISP1
:Disp "HELLO WORLD"
```

Outline

1 Hoe open je een programma?

2 Variabelen en Datatypes

- Getallen
- Strings
- Overige

*

3 Basic IO

- Disp
- Prompt

4 Exercises

Input: Prompt

- Nice...Maar het is veel leuker als de rekenmachine iets meer kan dan alleen "Hello World" zeggen.
- `Prompt` vertelt de rekenmachine iets: Input.
- Bijvoorbeeld, `Prompt A` vraagt de gebruiker om de waarde van de variabele `A`: een getal*
- Bijvoorbeeld, `Prompt Str1` vraagt de gebruiker om de waarde van de variabele `Str1`: een string.

Input: Prompt

- Nice...Maar het is veel leuker als de rekenmachine iets meer kan dan alleen "Hello World" zeggen.
- **Prompt** vertelt de rekenmachine iets: Input.
- Bijvoorbeeld, `Prompt A` vraagt de gebruiker om de waarde van de variabele `A`: een getal*
- Bijvoorbeeld, `Prompt Str1` vraagt de gebruiker om de waarde van de variabele `Str1`: een string.

```
CTL ☒ EXEC  
1:Input  
☒ Prompt  
3:Disp  
4:DispGraph  
5:DispTable  
6:Output(  
7:getKey
```

Input: Prompt

- Nice...Maar het is veel leuker als de rekenmachine iets meer kan dan alleen "Hello World" zeggen.
- **Prompt** vertelt de rekenmachine iets: Input.
- Bijvoorbeeld, **Prompt A** vraagt de gebruiker om de waarde van de variabele **A**: een getal.*
- Bijvoorbeeld, **Prompt Str1** vraagt de gebruiker om de waarde van de variabele **Str1**: een string.

```
CTL ☒ EXEC  
1:Input  
☒ Prompt  
3:Disp  
4:DispGraph  
5:DispTable  
6:Output(  
7:getKey
```

Input: Prompt

- Nice...Maar het is veel leuker als de rekenmachine iets meer kan dan alleen "Hello World" zeggen.
- **Prompt** vertelt de rekenmachine iets: Input.
- Bijvoorbeeld, **Prompt A** vraagt de gebruiker om de waarde van de variabele **A**: een getal.*
- Bijvoorbeeld, **Prompt Str1** vraagt de gebruiker om de waarde van de variabele **Str1**: een string.

```
CTL ☒ EXEC
1:Input
☒ Prompt
3:Disp
4:DispGraph
5:DispTable
6:Output(
7:getKey
```

Input: Prompt

Meerdere argumenten

- Merk op dat je meerdere argumenten aan **Prompt** kunt geven.
- `Prompt X,Y,Z` vraagt de gebruiker om de waarde van de variabelen `X`, `Y` en `Z`. *
- Dit geeft een kleiner, overzichtelijker programma dan drie aparte **Prompt** commando's.
- Hetzelfde geldt voor **Disp**.

```
PROGRAM: BAD
:Prompt X
:Prompt Y
:Prompt Z
```

```
PROGRAM: GOOD
:Prompt X,Y,Z
```

Input: Prompt

Meerdere argumenten

- Merk op dat je meerdere argumenten aan `Prompt` kunt geven.
- `Prompt X,Y,Z` vraagt de gebruiker om de waarde van de variabelen `X`, `Y` en `Z`. *
- Dit geeft een kleiner, overzichtelijker programma dan drie aparte `Prompt` commando's.
- Hetzelfde geldt voor `Disp`.

```
PROGRAM: BAD
:Prompt X
:Prompt Y
:Prompt Z
```

```
PROGRAM: GOOD
:Prompt X,Y,Z
```

Input: Prompt

Meerdere argumenten

- Merk op dat je meerdere argumenten aan `Prompt` kunt geven.
- `Prompt X,Y,Z` vraagt de gebruiker om de waarde van de variabelen `X`, `Y` en `Z`. *
- Dit geeft een kleiner, overzichtelijker programma dan drie aparte `Prompt` commando's.
- Hetzelfde geldt voor `Disp`.

```
PROGRAM: BAD
:Prompt X
:Prompt Y
:Prompt Z
```

```
PROGRAM: GOOD
:Prompt X,Y,Z
```

Input: Prompt

Meerdere argumenten

- Merk op dat je meerdere argumenten aan `Prompt` kunt geven.
- `Prompt X,Y,Z` vraagt de gebruiker om de waarde van de variabelen `X`, `Y` en `Z`. *
- Dit geeft een kleiner, overzichtelijker programma dan drie aparte `Prompt` commando's.
- Hetzelfde geldt voor `Disp`.

```
PROGRAM: BAD
:Prompt X
:Prompt Y
:Prompt Z
```

```
PROGRAM: GOOD
:Prompt X,Y,Z
```


“Hello World” is zo onpersoonlijk...

We breiden “Hello World” uit: “Hello ‘naam’”

- Maak een nieuw programma aan: `DISP2`.
- Vraag de gebruiker om zijn naam en sla deze op in `Str1`.
- Print vervolgens de `Str1` variabele met `Disp`, samen met de tekst “HELLO ”. *
- Run het programma en laat het jouw naam weergeven.
Let op: Zet je naam tussen quotes (“)

```
CTL ☒ EXEC
1:Input
☒ Prompt
3:Disp
4:DispGraph
5:DispTable
6:Output(
7:getKey
```

“Hello World” is zo onpersoonlijk...

We breiden “Hello World” uit: “Hello ‘naam’”

- Maak een nieuw programma aan: **DISP2**.
- Vraag de gebruiker om zijn naam en sla deze op in `Str1`.
- Print vervolgens de `Str1` variabele met `Disp`, samen met de tekst “HELLO ”.
- Run het programma en laat het jouw naam weergeven.
Let op: Zet je naam tussen quotes (“)

```
CTL ☒ EXEC
1:Input
☒ Prompt
3:Disp
4:DispGraph
5:DispTable
6:Output(
7:getKey
```

```
PROGRAM: DISP2
:
:
```

“Hello World” is zo onpersoonlijk...

We breiden “Hello World” uit: “Hello ‘naam’”

- Maak een nieuw programma aan: **DISP2**.
- Vraag de gebruiker om zijn naam en sla deze op in **Str1**.
- Print vervolgens de **Str1** variabele met **Disp**, samen met de tekst “HELLO ”. *
- Run het programma en laat het jouw naam weergeven.
Let op: Zet je naam tussen quotes (“)

```
CTL 11/00 EXEC
1:Input
2:Prompt
3:Disp
4:DispGraph
5:DispTable
6:Output(
7:getKey
```

```
PROGRAM:DISP2
:Prompt Str1
:
```

“Hello World” is zo onpersoonlijk...

We breiden “Hello World” uit: “Hello ‘naam’”

- Maak een nieuw programma aan: `DISP2`.
- Vraag de gebruiker om zijn naam en sla deze op in `Str1`.
- Print vervolgens de `Str1` variabele met `Disp`, samen met de tekst “HELLO ”.
- Run het programma en laat het jouw naam weergeven.
Let op: Zet je naam tussen quotes (“)

```
CTL ☒ EXEC
1:Input
☒ Prompt
3:Disp
4:DispGraph
5:DispTable
6:Output(
7:getKey
```

```
PROGRAM:DISP2
:Prompt Str1
:Disp "HELLO "+Str1
```

“Hello World” is zo onpersoonlijk...

We breiden “Hello World” uit: “Hello ‘naam’”

- Maak een nieuw programma aan: **DISP2**.
- Vraag de gebruiker om zijn naam en sla deze op in **Str1**.
- Print vervolgens de **Str1** variabele met **Disp**, samen met de tekst “HELLO ”. *
- Run het programma en laat het jouw naam weergeven.
Let op: Zet je naam tussen quotes (“).

```
CTL ☒ EXEC
1:Input
☒ Prompt
3:Disp
4:DispGraph
5:DispTable
6:Output(
7:getKey
```

```
PROGRAM:DISP2
:Prompt Str1
:Disp "HELLO "+Str1
```

Exercises

- 1 **DISP2** is nu onduidelijk voor de gebruiker...Laat de gebruiker m.b.v. **Disp** weten wat hij moet doen!
- 2 Schrijf een programma, **NATCHSTS**, dat enkele natuurconstanten in variabelen stored (**STO**^{*}). Bijvoorbeeld, store $6.67384 \cdot 10^{-11} (\text{Nm}^2\text{kg}^{-2})$ in **G**. Bedenk zelf welke je wilt storen. (Gebruik de Binas.) Doe er minstens 5.
- 3 Schrijf een programma, **DIST2P**, wat de afstand tussen twee punten uitrekent.
Input: $P_1 = (x_1, y_1)$ en $P_2 = (x_2, y_2)$. Output: $d(P_1, P_2)$.

Exercises

- 1 Schrijf een programma, **ABCPD**, die de oplossing geeft van de vergelijking $ax^2 + bx + c = 0$. Gebruik de ABC-formule en ga ervanuit dat a , b en c een *positieve* discriminant $D = b^2 - 4ac > 0$ geven. (Volgend college zullen we rekening houden met een negatieve discriminant.)
Input: a , b en c . Output: x_1 en x_2 (twee oplossingen!).

Exercises (optioneel)

- 1 Schrijf ook een programma, **DISTPL**, wat de afstand tussen een punt en een lijn uitrekent.
Input: $P = (x, y)$ en a, b in $l: y = ax + b$. Output: $d(P, l)$.
- 2 Schrijf een programma wat de gebruiker in de maling neemt: **FOOLYOU**. Genereer de volgende output:

2+2	4
2+2	5

*

Om het interessant te maken: Doe dit zonder gebruik te maken van de spatie ()!

- 3 Breid **FOOLYOU** uit (**FOOLYOU2**) zodat hij input van de gebruiker accepteert, zodat de gebruiker kan bepalen waar $2 + 2$ gelijk aan is. Leef je uit! Gebruik nu gerust de spatie () weer, mocht je met strings willen werken.

Antwoorden

Hieronder staan mogelijke antwoorden. Uiteraard is het mogelijk om een programma op oneindig veel manieren te schrijven. Zo lang als het programma dezelfde functie volbrengt, is het correct.

```
PROGRAM:DISP2  
:Disp "HOE HEET JE?"  
:Prompt Str1  
:Disp "HELLO "+Str1
```

*

```
PROGRAM:NATCNSTS  
:6.67384E-11→G  
:1.602176565E-19→E  
:9.10938215E-31→M  
:6.02214129E-23→N  
:299792458→C  
:6.62606896E-34→H  
:1.3806488E-23→K
```

Antwoorden

```
PROGRAM: DIST2P
:Disp "POINT 1: "
:Prompt X,Y
:X→A:Y→B
:Disp "POINT 2: "
:Prompt X,Y
:J((X-A)²+(Y-B)²)→D
:Disp "DISTANCE=",D
```

Pythagoras!

*

```
PROGRAM: ABCPD
:Disp "SOLVING AX²+BX+C=0"
:Prompt A,B,C
:B²-4AC→D
:Disp "THERE ARE TWO SOLUTIONS: "
:(-B+√(D))/(2A)→X
:Disp X
:(-B-√(D))/(2A)→X
:Disp X
```

Antwoorden (optioneel)

```
PROGRAM: DISTPL  
:Disp "POINT: "  
:Prompt X,Y  
:Disp "LINE Y=AX+B: "  
:Prompt A,B  
:abs(-AX+Y-B)/J(A^2+1)→D  
:Disp "DISTANCE=",D
```

Vrijwel hetzelfde als DIST2P,
maar een andere formule.

```
PROGRAM: FOOLYOU  
:Disp "2+2",4  
:Disp "2+2",5
```

```
PROGRAM: FOOLYOU2  
:Disp "WAT IS 2+2  
VOLGENS JOU?"  
:Prompt Str1  
:Disp "2+2",4  
:Disp "2+2",Str1  
:Disp "", ""
```