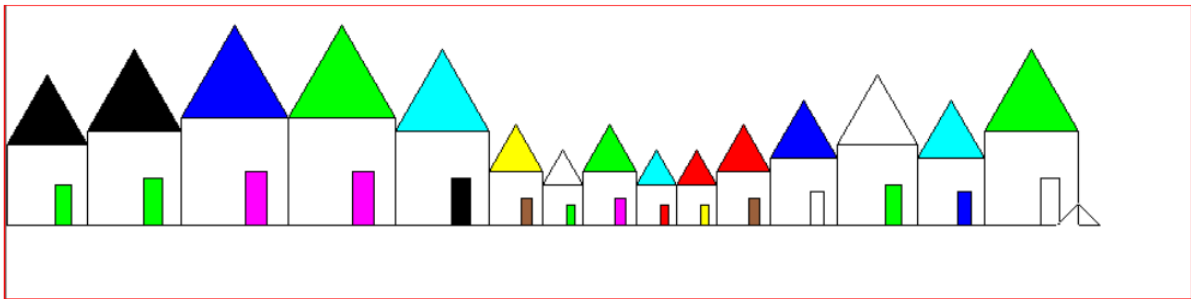

FMS LOGO

Practicum



1 JANUARI 2020
LEIDSCHER RIJN COLLEGE
Informatica

Inhoud

| | |
|---|----|
| Uitleg Workspace | 3 |
| Opslaan Workspace | 3 |
| Herhalen. | 5 |
| Procedures maken. | 5 |
| Combineren van procedures. | 6 |
| Procedures variabel maken. | 7 |
| Inkleuren van (gesloten) figuren. | 7 |
| Random. | 8 |
| Fouten opsporen. | 8 |
| Programmeer regels | 10 |
| Efficient programmeren:..... | 10 |
| Overzicht veelgebruikte JavaLogoCodes | 11 |
| Vaardighedenlijst: | 12 |

In het hoofdstuk Grondslagen wordt er stilgestaan bij het begrip Algoritme.

Om wat gevoel te krijgen bij dit begrip ga je wat programmeren in FMSlogo. Bij dit programmeren geef je opdrachten aan een 'teken-turtle'.

Het doel van dit practicum is

- Werken met begrippen als herhaling, sequentie, variabelen.
- Een efficiënt algoritme bedenken om een opdracht uit te voeren.
- Het algoritme zo efficiënt mogelijk beschrijven en programmeren

Het programma waar we mee werken kan je thuis ook installeren. In opdracht 1 staat de link waar het te vinden is.

In dit practicum staat alleen de hoog nodige uitleg. Op het internet is ook veel te vinden over dit programma en in de planner staat een link naar een workshop. Het is de bedoeling dat je jezelf de vaardigheden eigen maakt die op blz 12 staan beschreven. Je docent zal dit regelmatig komen checken.

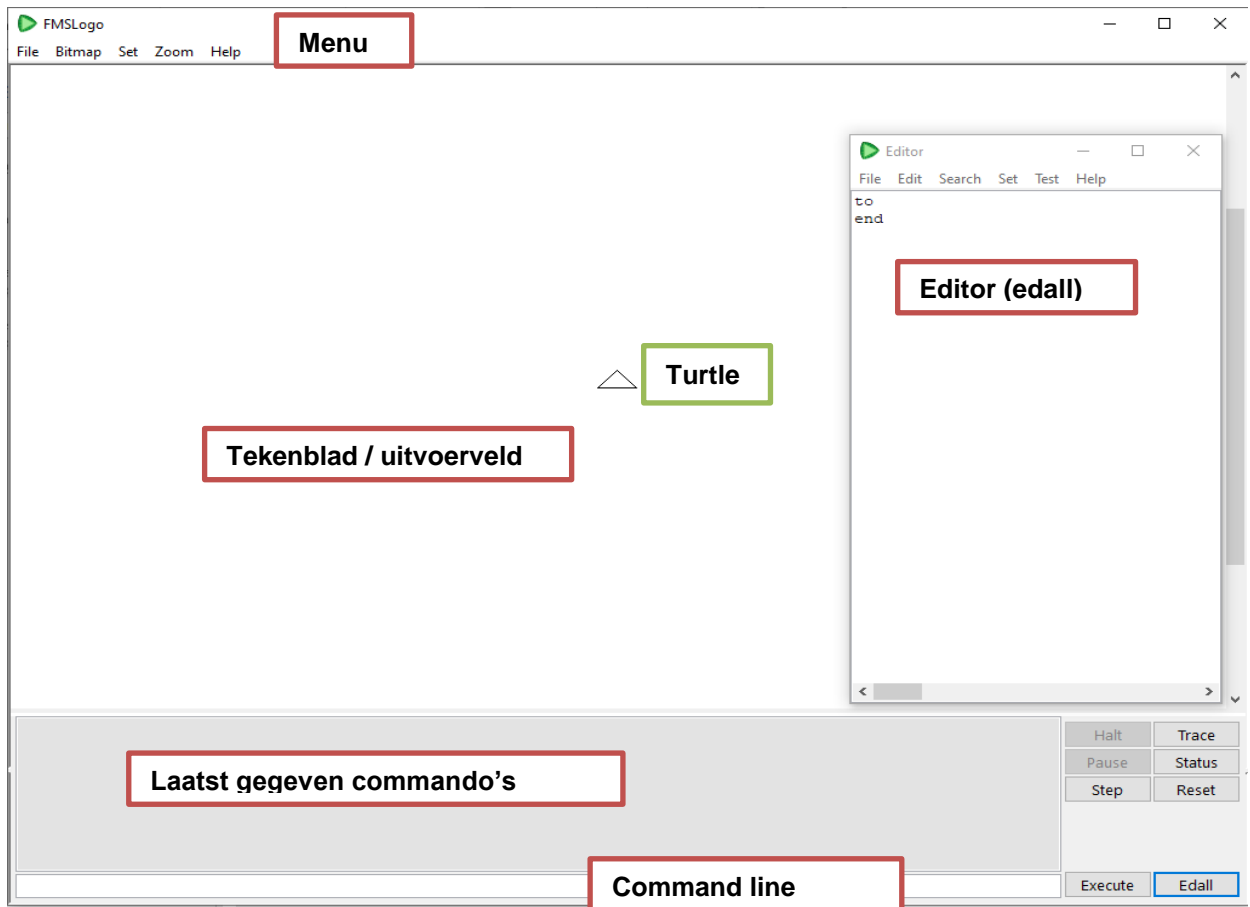
Veel plezier en creativiteit.

Opdracht 1: Thuisopdracht.

Download het programma FMSlogo en installeer dat op je PC.
Helaas werkt dit niet op een Apple of Chromebook

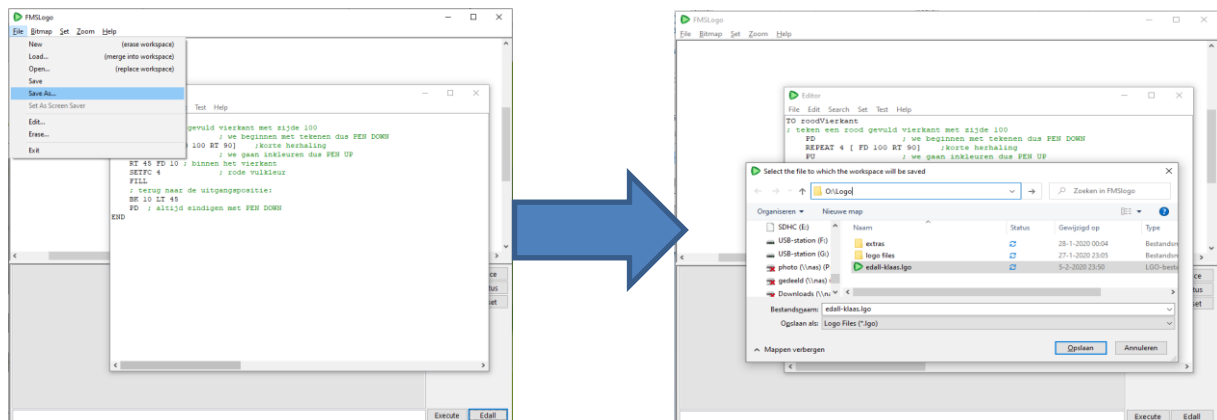
Downloadlink: <https://sourceforge.net/projects/fmslogo/>

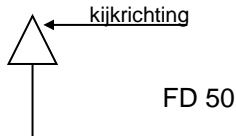
Uitleg Workspace



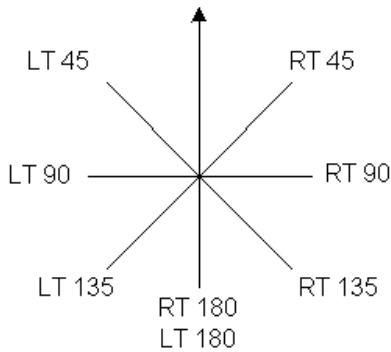
Opslaan Workspace.

Als je klaar bent met je programmeer sessie moet je je hele workspace nog opslaan, anders ben je je werk kwijt. Maak in je OneDrive een map LoGo in de map Informatica en sla hier je bestanden op. Dan kan je thuis verder werken.





De schildpad heeft de vorm van een kleine driehoek in het midden van het scherm, de 'kop' geeft de richting aan waarin hij 'kijkt'. Wanneer de schildpad beweegt tekent hij een lijn achter zich. In dit geval is hem verteld dat hij 50 'stappen' naar voren moet gaan door de opdracht FORWARD 50, afgekort tot FD 50



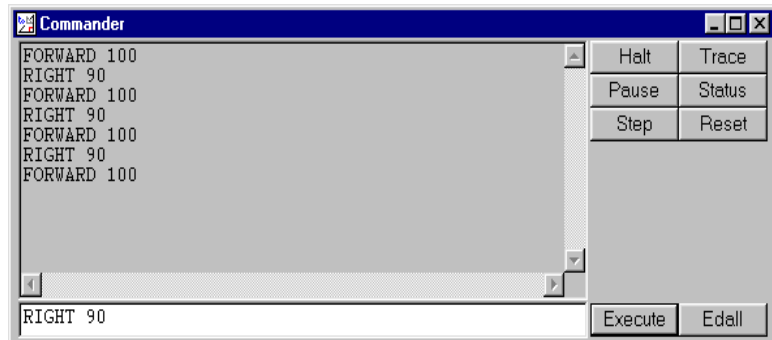
Je kunt de schildpad laten draaien door hem te 'vertellen' naar rechts of links te gaan. De afbeelding hiernaast toont je hoe je de schildpad de verschillende richtingen op kunt laten gaan. In plaats van het typen van RIGHT of LEFT kun je ook hier afkortingen gebruiken.

De opdrachten geef je in het Commander venster, in het invoer vak (commandoregel).

Opdracht 2:

Probeer het volgende eens uit:

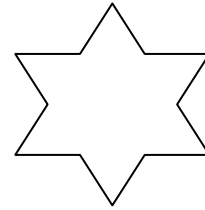
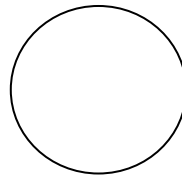
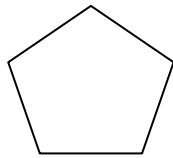
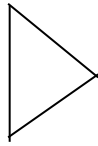
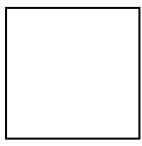
```
FORWARD 100
(FD 100)
RIGHT 90 (RT
90)
FORWARD 100
RIGHT 90
FORWARD 100
RIGHT 90
FORWARD 100
```



Gebruik CS (Wis scherm) tussen elke opdracht om met een schoon scherm verder te gaan, je kunt ook CLEARSCREEN gebruiken. Gebruik CT (ClearTekst) om je opdrachten scherm te wissen

Opdracht 3:

Maak de volgende figuren



Herhalen.

Om een vierkant te tekenen gebruik je nu
FD 100 RT 90 FD 100 RT 90 FD 100 RT 90 FD 100 RT 90

Hier staat eigenlijk 4 maal FD 100 RT 90 . Het is efficiënter om hiervoor een herhaling te gebruiken. In FMSlogo gaat dit als volgt:

```
repeat 4 [FD 100 RT 90 ]
```

Opdracht 4

Teken de figuren uit opdracht 3 met behulp van een herhaling.

Procedures maken.

In het programma FMSlogo is een set commando's meegeprogrammeerd (API).

Deze basisset kan je zelf uitbreiden en je eigen procedures toevoegen. Een procedure is een aantal losse commando's achter elkaar die je een naam geeft. Door deze naam als opdracht te geven in de commandoregel worden de commando's in één keer uitgevoerd. Super handig voor veel gebruikte opdrachten.

Opdracht 5: (uitvoeren)

Maak een eigen procedure waarmee een vierkant getekend wordt.

1. Open de Edall (rechts onderin)
2. TO vierkant (nooit spaties gebruiken in de naam !!)

3. Geef losse commando's:(nieuwe regel) repeat 4 [fd 100 rt 90]
4. Sluit af met END (nieuwe regel)
5. Sla je edall op met ctrl-D
6. Test je procedure vierkant in de commando regel

```
;teken een vierkant  
TO vierkant  
repeat 4 [ fd 100 rt 90]  
END
```

Als de procedure goed werkt voeg je voeg commentaar in, zodat je weet wat deze procedure doet. B.v. ;teken een vierkant

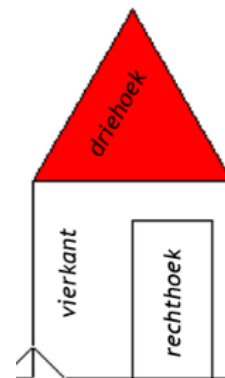
Opdracht 6

Maak van alle basisfiguren uit opdracht 3 en eigen procedure. Maak daarbij ook een procedure voor een rechthoek. Gebruik een logische naam voor je procedure.

Combineren van procedures.

Je kan de procedures die je zelf gemaakt hebt weer hergebruiken (nesten) in nieuwe procedures. Als je bv een huis wil tekenen kan je de procedure vierkant, rechthoek en driehoek hergebruiken.

```
TO huis      ;teken een huis  
  Vierkant  ;teken de muren  
  ...verplaats  
  Driehoek  ;teken het dak  
  ...verplaats  
  Rechthoek ;teken de deur  
  ...verplaats  
END
```



Opdracht 7

Maak een procedure huis waarbij je de procedures rechthoek, vierkant en driehoek (her) gebruikt en sla deze procedure op.

Procedures variabel maken.

Het is onhandig om voor ieder vierkantje dat we willen tekenen en die een andere grootte heeft, steeds de procedure vierkant te moeten kopiëren en dan één ding aan te passen. Dit zou er dan zo uitzien: repeat 4[FD 100 RT 90] naar repeat 4 [FD 50 RT 90]. Het enige verschil is de waarde achter het FD commando.

Om dit efficiënt te kunnen programmeren kunnen we gebruik maken van een variabele. Op de plek waar iets veranderd vullen we nu een letter/woord in, bv :zijde.

Aandachtspunten bij een variabele:

- De naam van de variabele mag van alles zijn, ook één letter.
- Een variabele begint altijd met een dubbele punt.
- De variabele naam staat ook achter de procedure naam.
- Het is mogelijk om meerdere variabele te gebruiken bij één procedure

Een variabele procedure vierkant ziet er dan als volgt uit.

```
TO vierkant :zijde
repeat 4 [FD :zijde RT 90]
END
```

De procedure "vierkant :zijde" heeft vanaf nu altijd een waarde nodig bij het invoeren in de commandoregel.

Opdracht 8

Maak alle basisfiguren uit eerdere opdrachten variabel in grootte. Ook de procedure huis.

Inkleuren van (gesloten) figuren.

De turtle tekent nu alleen de buitenlijnen van een figuur. We kunnen een gesloten figuur echter ook inkleuren.

Hiervoor moeten een aantal stappen gedaan worden.

- De turtle moet in het gesloten figuur gaan staan. (pu rt 45 fd 10)
- De vulkleur moet bepaald worden (setfc [255 0 0])
- Figuur moet gevuld worden (fill)
- De turtle weer terug naar startpositie (bk 10 lt 45 pd)

Kleurcodes kunnen op 2 manieren gegeven worden. Of met RGB waarde tussen blokhaken of een index nummer. De kleurcodes vind je in de bijlage achterin.

Opdracht 9

Maak een procedure "kleur" waarmee je de stappen om in te kleuren in één keer maakt. Gebruik een variabele om de kleur te bepalen (met index nummer)

Random.

Het commando Random genereert willekeurig een getal. Bij bv random 8, wordt er een heel getal tussen 0 en 7 gegenereerd.

Hiervan kan je bv gebruik maken bij het inkleuren van figuren met een random kleur.

Opdracht 10

Kopieer de procedure kleur en noem deze randomkleur. Pas de code zo aan dat er een random kleur wordt gekozen als deze procedure wordt aangeroepen.

Opdracht 11:

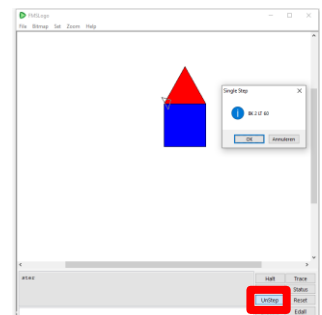
Pas de procedures van je basisvormen zo aan , dat ze direct ingekleurd worden.

Maak ook een procedure voor een halve cirkel die ingekleurd wordt. Gebruik hier voor ARC2 'graden' 'radius'.

Fouten opsporen.

Om fouten in snel uitgevoerde procedures te kunnen achterhalen, kan je gebruik maken van STEP (rechts onderin). Hiermee worden programma regels 1 voor 1 uitgevoerd, nadat jij ze bevestigd met Enter. Zo kan je zien waar een programma niet doet wat jij bedacht had.

Een tweede mogelijkheid biedt de knop "Trace" Hiermee laat het programma zien welke (deel) procedures aangeroepen worden.



Opdracht 12

- A: Maak een straatje van 6 huizen, grootte 70.
- B: Maak een straatje huizen met een variabel aantal die door de gebruiker kan worden aangegeven. Bv straatje 5 voor een straat met 5 huizen.
- C: Maak een straatje met een variabel aantal huizen en grootte van de huizen. bv straat 5 100 voor een straat met 5 huizen met een grootte 100.
- D: Maak een straatje met een variabel aantal huizen met een random grootte en kleur.

Programmeer regels

- ▶ Procedures en variabelen een toepasselijke naam in camelCase:
 - ▶ Bijvoorbeeld: roodVierkant of kleinHuis
- ▶ Inspringen:
 - ▶ Binnen procedures en soms binnen REPEAT
- ▶ Nieuwe regel:
 - ▶ PU, PD en verplaatsingen op een aparte regel
 - ▶ Bij korte herhalingen: REPEAT op één regel
 - ▶ Bij lange herhalingen: inspringen en opdrachten op een nieuwe regel
- ▶ LOGO sleutelwoorden in HOOFDLETTERS, zoals FD, FILL en REPEAT.
- ▶ Voeg commentaar toe aan procedures (; dit is commentaar)
- ▶ Eindig een procedure altijd met een PU en ga terug naar de startplek.
- ▶ Verwijder alle overbodige code.

Voorbeeld code volgens regels

```
TO roodVierkant
; teken een vierkant met zijde 100
; eindigt op de startplek
  PD                ; we beginnen met tekenen dus PEN DOWN
  REPEAT 4 [ FD 100 RT 90] ; korte herhaling
  PU                ; we gaan inkleuren dus PEN UP
  RT 45 FD 10       ; binnen het vierkant
  SETFC 4           ; rode vulkleur
  FILL
; terug naar de uitgangspositie
  BK 10 LT 45
  PD                ; altijd eindigen met PEN DOWN
END
```

Efficiënt programmeren:

- ▶ Zo min mogelijk regels.
 - ▶ herhaling waar het kan
 - ▶ Deelprocedures gebruiken
 - ▶ Ook direct laten inkleuren
 - ▶ Turtle op goede plek laten uitkomen
 - ▶ Nooit overbodige code in edall
 - ▶ Procedures verwijderen die je niet gebruikt
 - ▶ Dubbelingen samenvoegen, bijv. FD 45 FD 10 = FD 55
- ▶ Netjes > syntax lay-out:
 - ▶ gebruik HOOFD- en kleine letters en inspringen !!

Overzicht veelgebruikte JavaLogoCodes

| | | |
|------------------------|----------------------------------|-----------------------|
| Forward (FD) | ga vooruit | bijv: forward 100 |
| Back (bk) | ga achteruit | bijv: bk 50 |
| RT | draai naar rechts | bijv: rt 80 |
| LT | draai naar links | bijv: lt 45 |
| Clearscreen (CS) | 'Tekenblad' leeg maken | bijv: cs |
| Cleartext (CT) | Maakt commander scherm leeg | bijv: ct |
| Pennormal | Laat turtle weer normaal tekenen | bijv: pennormal |
| Penpaint (PPT) | laat turtle weer schrijven | bijv: ppt |
| Penerase (PE) | Turtle wordt 'Gum' | bijv: pe |
| SetPenColor (SETPC) | verander de penkleur | bijv: setpc [255 0 0] |
| Setpensize | verander pendikte | bijv: Setsize [10 |
| Penup (PU) | verplaatsen zonder schrijven | bijv: pu |
| Pendown (PD) | turtle kan weer schrijven | bijv: pd |
| SetFloodColor (SETFC) | verander vulkleur | bijv: setfc[0 0 255] |
| Setscreencolor (setsc) | verander de achtergrondkleur | bijv: setsc[0 0 255] |
| Fill | vul een gesloten vorm | bijv: fill |
| ARC 'graden' 'radius' | maakt een boog | bijv: arc 180 100 |
| ARC2 'graden' 'radius' | maakt boog vanaf positie turtle | bijv: arc2 180 100 |
| Home | Turtle naar startpositie | bijv: home |
| HT (hide turtle) | Turtle niet meer zichtbaar | |
| ST (show turtle) | Turtle zichtbaar | |

Ifelse <voorwaarde>[waar][niet waar] bijv: ifels :x>50
 [vierkant :x][driehoek :x]

Random 8: genereerd heel getal tussen 0 en 7
 PopUp boodschap: Messagebox [titel][boodschap] geeft pop-up met melding
 Repcount: geeft waarde van het aantal herhalingen dat is gedaan.

Kleurentabel

| Name | Index | RGB values | Name | Index | RGB values |
|--------|-------|---------------|-----------|-------|---------------|
| Black | 0 | [0 0 0] | Brown | 8 | [155 96 59] |
| Blue | 1 | [0 0 255] | Light | 9 | [197 136 18] |
| Green | 2 | [0 255 0] | brown | 10 | [100 162 64] |
| Cyan | 3 | [0 255 255] | Mid-green | 11 | [120 187 187] |
| Red | 4 | [255 0 0] | Blue- | 12 | [255 149 119] |
| Magent | 5 | [255 0 255] | green | 13 | [144 113 208] |
| a | 6 | [255 255 0] | Salmon | 14 | [255 163 0] |
| Yellow | 7 | [255 255 255] | Blue-ish | 15 | [183 183 183] |
| White | | | Orange | | |
| | | | Silver | | |

Vaardighedenlijst:

| | |
|---|---|
| 1 | Sequentie |
| 2 | Herhaling |
| 3 | Procedures maken |
| 4 | Programmeerregels toepassen |
| 5 | Procedures hergebruiken |
| 6 | Procedures variabele maken |
| 7 | Inkleuren |
| 8 | Toepassen Random (kleur) |
| 9 | |
| | Extra mogelijkheden: |
| X | <ul style="list-style-type: none">• Recursie• Variabelen globaal, lokaal• Keuze (if-else) |