

In deze derde les van Unit 3 leer je de clausule **Else** en het gebruik van **Elself**.

Doelen:

- **If...Then...Else** opdrachten ontwerpen zodat er acties zijn voor zowel de waarde waar als voor de waarde onwaar van de conditie.
- De clausule **Elself** gebruiken om verschillende mogelijke voorwaarden binnen één blokstructuur af te handelen

Soms is het nodig om een actie te laten uitvoeren als een voorwaarde *waar* is en een andere actie als dezelfde voorwaarde *onwaar* is. Dit is het moment dat de clausule **Else** in beeld komt.

Bij het selecteren van de structuur **If...Then...Else...EndIf** uit met menu Besturing worden alle vier de sleutelwoorden in het programma geplakt en kun je de ontbrekende stukken van de programmacode invullen..

```

Define slaag_zak()=
Prgm
[ ]
If Then
Else
EndIf
[ ]
EndPrgm
[ ]
    
```

Het programma:

Voor een bepaalde cursus zijn er drie toetsen en voor de cursus kun je slagen of zakken. Er is een gemiddeld cijfer van 6,5 (ofwel een score van 65%) nodig om te slagen. Schrijf een programma dat de drie toetsscores ontvangt, het gemiddelde ervan berekent en uiteindelijk het bericht “gezakt” of “geslaagd” weergeeft.

- Bepaal eerst welke methode he wilt gebruiken om de drie scores in het programma te krijgen: argumenten of Request-opdrachten?
- Hoe wordt het gemiddelde berekend?
- Voor de uitvoer van het programma gaan we de opdracht **Text** uit het menu I/O gebruiken

Voor het gemak zullen we drie argumenten voor de drie toetsscores gebruiken.

```

Define slaag_zak(t1,t2,t3)=
Prgm
gem:=(t1+t2+t3)/3
[ ]
If Then
Else
[ ]
EndIf
[ ]
    
```

Docenten Tip: voor de andere methode zijn geen argumenten nodig maar drie Request-opdrachten:

```

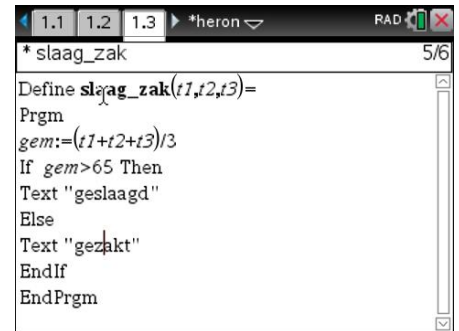
Request "Toets 1 score?",t1
Request "Toets 2 score?",t2
Request "Toets 3 score?",t3
    
```

Het gemiddelde van de drie scores is het totaal van de scores gedeeld door 3.

$$gem:=(t1+t2+t3)/3.$$

Gebruik een decimale punt na de 3 (of een komma afhankelijk van de instellingen) om je ervan te verzekeren dat het resultaat een decimale benadering is.

Voltooi tenslotte de **If**-structuur en gebruik **Text**-opdrachten om ofwel “geslaagd” ofwel “gezakt” weer te geven. (*Hoe zou je ervoor kunnen zorgen dat ook het gemiddelde in het tekstvak wordt getoond?*)

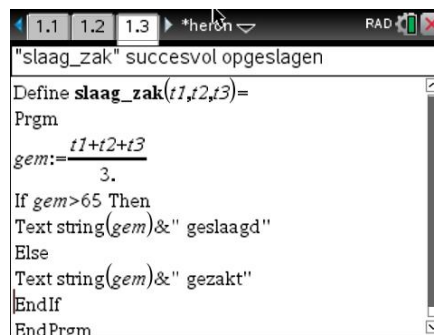


```

1.1 1.2 1.3 *heron RAD
* slaag_zak 5/6
Define slaag_zak(t1,t2,t3)=
Prgm
gem:=(t1+t2+t3)/3
If gem>65 Then
Text "geslaagd"
Else
Text "gezakt"
EndIf
EndPrgm
    
```

Docenten Tip: Om het gemiddelde en het woord “geslaagd” of “gezakt” weer te geven in hetzelfde tekstvak moet je het getal **gem** omzetten in een tekenreeks (string) en dan aaneenschakelen:

Text string(gem) & “geslaagd”; Een alternatief zou kunnen zijn om
Disp gem,”geslaagd” te gebruiken



```

1.1 1.2 1.3 *heron RAD
"slaag_zak" succesvol opgeslagen
Define slaag_zak(t1,t2,t3)=
Prgm
gem:= $\frac{t1+t2+t3}{3}$ 
If gem>65 Then
Text string(gem)&" geslaagd"
Else
Text string(gem)&" gezakt"
EndIf
EndPrgm
    
```

Docenten Tip: Overweeg om een programma te schrijven om de discriminant van de abc-formule ($b^2 - 4ac$) te onderzoeken. Hoeveel nulpunten zal de kwadratische functie hebben?