



# Parlement

Parlement is een programmeertaal afgeleid van [Chef](#) en [INTERCAL](#). De taal speelt met de ideeën uit de politiek.

## Inhoudsopgave

Variabelen & Declarations .....	3
Commentaar .....	4
Operators .....	5
Statements .....	6
If/else .....	6
Print .....	6
Functie .....	7
Loop .....	7

## Variabelen & Declarations

Parlement ondersteund drie typen variabelen. Namelijk int, string en bool.

Een int is altijd een positief en heel getal. Negatieven en komma's zijn niet toegestaan

Een string bevat tekst. De tekst staat tussen twee aanhalingstekens en de tekst staat op één lijn. De tekst mag `\n` en `\r` escapes bevatten.

Een bool accepteert de waarden:

- 0 -> als false
- 1 -> als true
- true -> als true
- false -> als false

Een variabele is te declareren door het type aan te geven en de naam van het variabel.

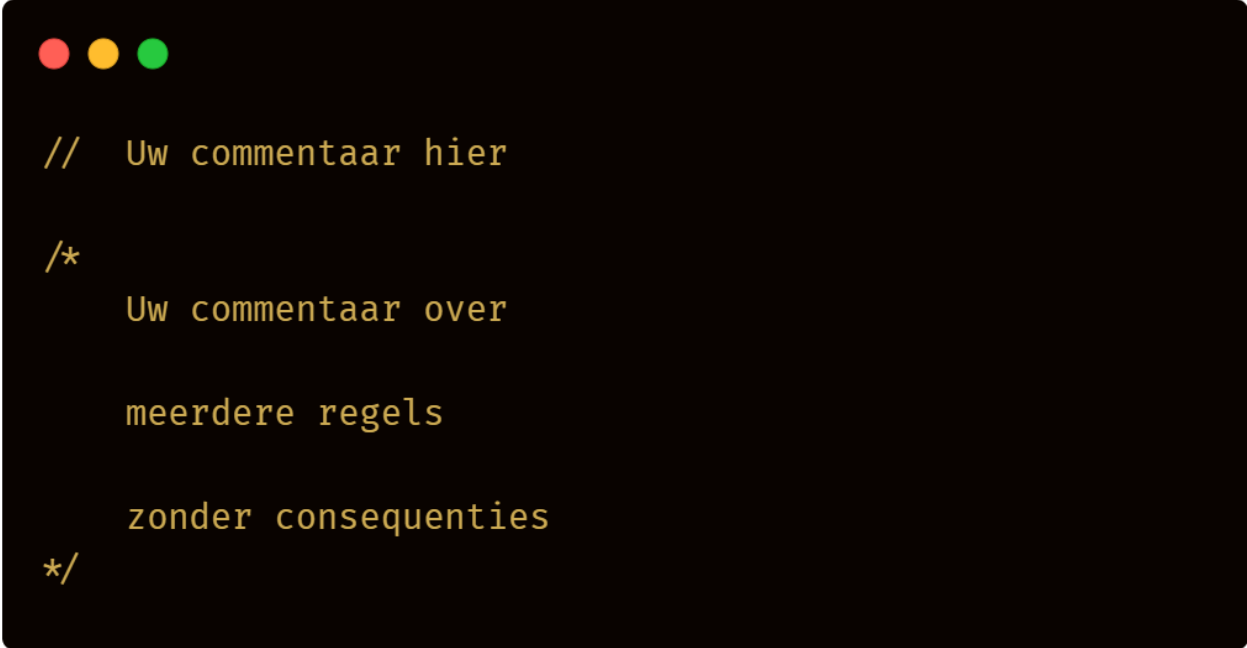
Het variabel is ook meteen toe te wijzen aan een waarde. Dit kan ook nadat het variabel al is gedeclareerd.



```
int var1 = 150
string var2 = "Hallo Wereld!"
bool var3 = true
```

## Commentaar

Parlement ondersteund commentaar per regel of per blokken. Om één regel van commentaar te voorzien zet u // voor het commentaar. Om meerdere regels te voorzien van commentaar begint u met /\* en eindigt u met \*/



```
// Uw commentaar hier

/*
  Uw commentaar over
  meerdere regels
  zonder consequenties
*/
```

## Operators

Zie onderstaande tabel voor de operators. Parlement ondersteund ook de rekenvolgorde.

Naam	Omschrijving	Notatie
Toewijzen	Wijst waarde aan variabel toe	=
Gelijk aan	Return boolean als expressie gelijk is	==
Niet gelijk aan	Return boolean als expressie niet gelijk is	!=
Groter dan	Return boolean als expressie groter is dan	>
Kleiner dan	Return boolean als expressie kleiner is dan	<
Vermenigvuldigen	Vermenigvuldigd expressie	*
Delen	Deelt de expressie	/
Optellen	Telt expressie op	+
Aftrekken	Trekt expressie van elkaar af	-
En	Return <i>waar</i> als beide expressies <i>waar</i> zijn	&&
Of	Return <i>waar</i> als een van de expressie <i>waar</i> is	

```
int a = 1
int b = 2

a = b           // = false
a ≠ b           // = true
a > b           // = false
a < b           // = true
a * b           // = 2
a / b           // = 0 (No decimals, rounded down)
(a + b) * b     // = 9
true && false    // = false
true || false   // = true
```

## Statements

### If/else

Als de expressie waar is dan wordt het eerste code blok uitgevoerd. Zie voorbeeld.


- *ANDERS* is niet verplicht.
- De accolades zijn niet verplicht als er zich maar één statement bevindt in de code block



```
ALS 1==1 DAN {  
    // Code als het waar is ...  
} ANDERS {  
    // Alternative code hier ...  
}
```

### Print

Koning Willem Alexander leest graag de troonrede op. Daarom print hij ook dingen naar de console. Het print statement ondersteunt int en string. Zie voorbeeld.



```
// Print "Hallo Wereld" naar console  
WILLIE WIL JE DIT OPLEZEN ALSJEBLIEFT "Hallo Wereld"
```

## Functie

Je kan een motie indienen bij het parlement. Dit kan later worden uitgevoerd door de motie naam aan te roepen en () toe te voegen. Voorbeeld: mijnFunctie()



```
MOTIE mijnFunctie
```

```
// Doe hier dingen
```

```
OKE STOP MAAR
```

## Loop

Mark rutte herhaalt graag dingen. Daarom kan je aangeven hoe vaak mark rutte iets moet herhalen. Zie voorbeeld.

- In voorbeeld: 15 kan ook een expressie zijn dat een int returned.



```
MARK RUTTE WIL JE DIT 15 KEER HERHALEN
```

```
mijnFunctie() // Doe hier dingen
```

```
OKE STOP MAAR
```