

**Parlement**

Parlement is een programmeertaal afgeleid van [Chef](http://progopedia.com/language/chef/) en [INTERCAL](https://en.wikipedia.org/wiki/INTERCAL). De taal speelt met de ideeën uit de politiek.

**Inhoudsopgave**

[Variabelen & Declarations 3](#_Toc69248966)

[Commentaar 4](#_Toc69248967)

[Operators 5](#_Toc69248968)

[Statements 6](#_Toc69248969)

[If/else 6](#_Toc69248970)

[Print 6](#_Toc69248971)

[Functie 7](#_Toc69248972)

[Loop 7](#_Toc69248973)

# Variabelen & Declarations

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

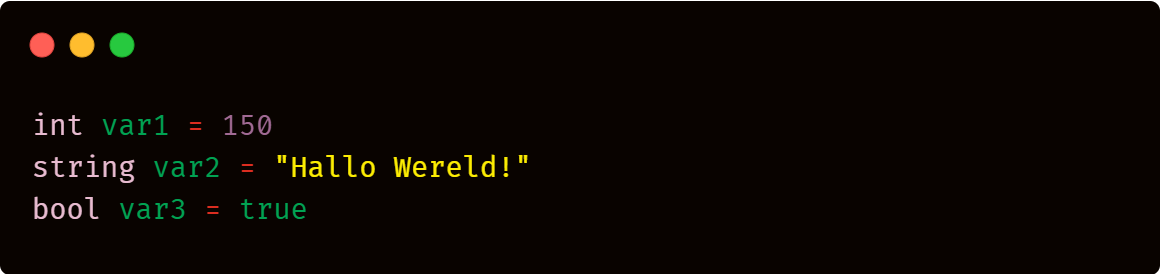
Een int is altijd een positief en heel getal. Negatieven en komma’s zijn niet toegestaan

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:

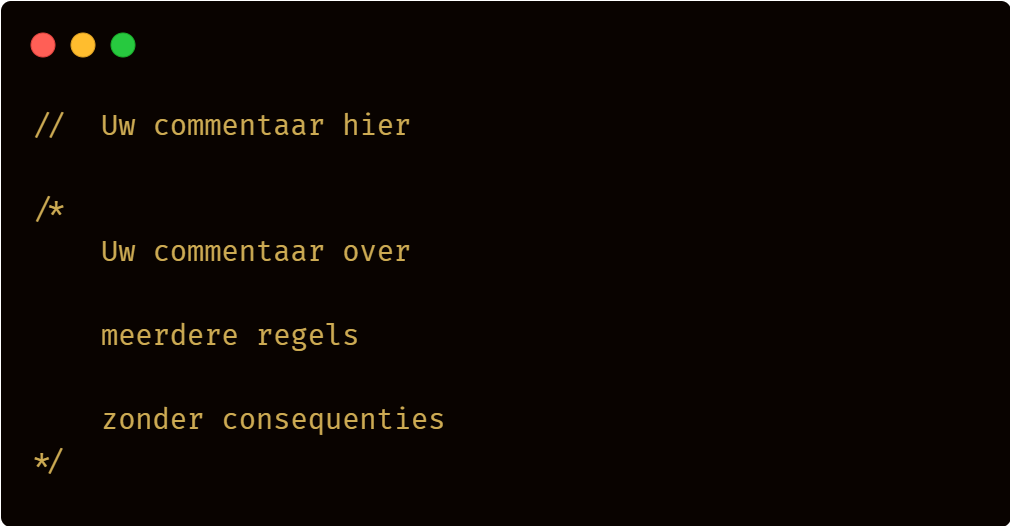
* 0 -> als false
* 1 -> als true
* true -> als true
* false -> als false

Een variabel 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.



# 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 **\*/**



# 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 *waar* als beide expressies *waar* zijn | && |
| Of | Return *waar* als een van de expressie *waar* is | || |

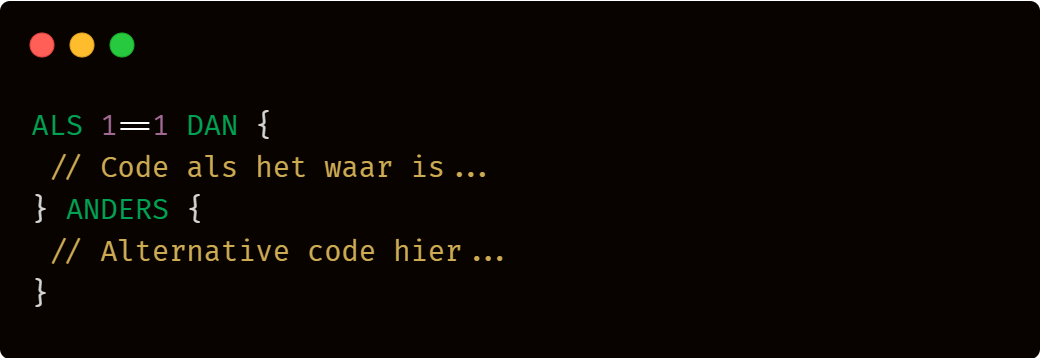


# Statements

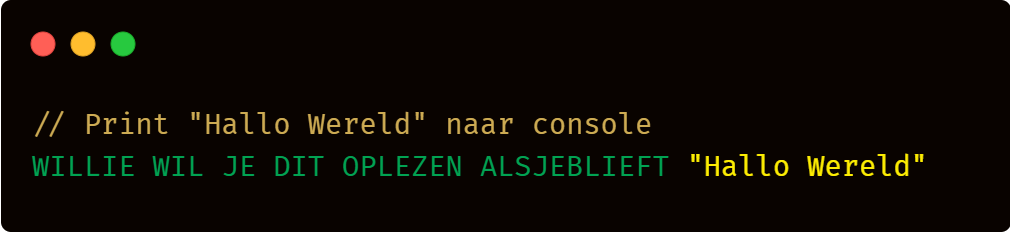
## If/else

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

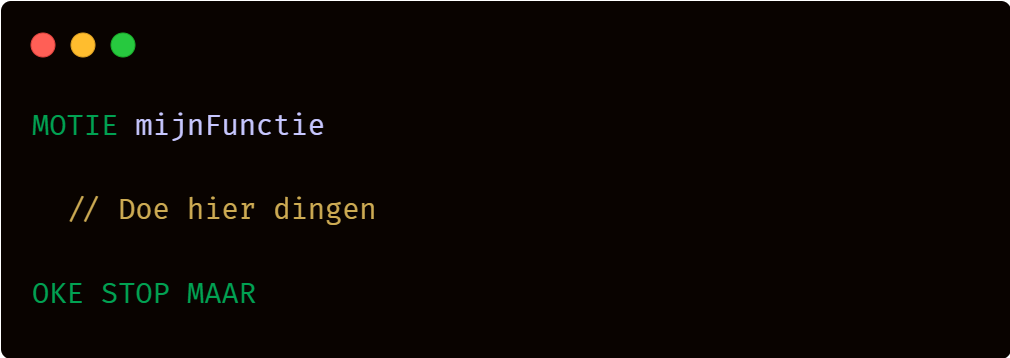


## Print

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

## Functie

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



## 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.

