DE

**BASIX1**

Programmeertaal

Jarno Witjes | EHI2VSe

Saxion Compilers en Operating Systems

Jarno Witjes

449590

EHI2VSe

Inhoud

[Inleiding 4](#_Toc6421253)

[Ontwerp van Basix1 5](#_Toc6421254)

[Features 5](#_Toc6421255)

[Beginnen met Basix1 6](#_Toc6421256)

[Voorbeeld 1: Hello World 6](#_Toc6421257)

[Voorbeeld 2: Variabelen 6](#_Toc6421258)

# Inleiding

De Basix1 programmeertaal is ontwikkeld als een opdracht voor Saxion’s module: Compilers en Operating Systems. Basix1 is een makkelijke taal waarmee je snel kleine basisprogramma’s kan schrijven en uit laten voeren. De eerste versie van de taal ondersteunt twee datatypes, conditional branching en loops. Deze eerste versie is niet object oriënterend.

Dit document bespreekt de taal aan de hand van voorbeelden.

# Ontwerp van Basix1

Basix1 is gebouwd voor programmeurs die snel een simpel programma willen bouwen voor verschillende doeleinden. De taal neemt wat weg van de Basic programmeertaal. Het programmeren met Basix1 gebeurd door ingebouwde functies aan te roepen en argumenten aan die functies te geven.

## Features

* Printen naar console
  + Tekst en integers
* Ondersteuning voor variabelen
* Conditional branching
* Loop

# Beginnen met Basix1

In de volgende pagina’s wordt uitgelegd hoe de taal werk. Er wordt uitgelegd aan de hand met voorbeelden.

## Voorbeeld 1: Hello World

We beginnen met de klassieke ‘Hello world!’ programma. Dit programma zal de tekst ‘Hello World’ printen naar de console en daarna afsluiten.

BASIX\_BEGIN  
  
print::text << “Hello World!” ;  
  
BASIX\_END

Regel voor regel:

BASIX\_BEGIN : Dit markeert het begin van een Basix1 programma.

print::text << “Hello World!” ; : Dit is de print functie. print geeft aan dat je iets wil printen naar de console. Na print moet je de aangeven wat je gaat printen, je kan tekst (text) of getallen (integer) printen. Nu dat je de print functie aan heb geroepen moet je aangeven wat je wilt printen je geeft dit aan met ‘<<’ met vervolgens je waarde.

Let op!

Je kan geen getal printen als je aan hebt gegeven dat je tekst gaat printen en andersom. voorbeeld:

BASIX\_BEGIN  
  
print::integer << “Hello World!” ;  
print::text << 2020;  
  
BASIX\_END

BASIX\_END : Markeert het einde van een Basix1 programma.

## Voorbeeld 2: Variabelen

Basix1 ondersteund variabelen maar alleen voor twee datatypes: text en integer. Er zijn verschillende functies voor het werken met variabelen:

make\_known::”type”::”name” << “value” ;

remake\_known::”name” << “value” ;

get\_known::”name” ;

De make\_known functie maakt een variable van het gegeven type, naam en waarde.

De remake\_known functie hangt een nieuwe waarde aan een al bestaande variabele.

De get\_known functie geeft de waarde terug van het gegeven al bestaande variabele. Voorbeeld:

BASIX\_BEGIN  
  
make\_known::integer::myint << 10 ;  
print::text << “Variable myint is now: ” ;  
print::integer << get\_known::myint; ;  
  
remake\_known::myint << 20 ;  
print::text << “Variable myint is now: ” ;  
print::integer << get\_known::myint; ;  
  
make\_known::integer::mathint << 20 + 12 ;  
print::text << “20 + 12 =” ;  
print::integer << get\_known::mathint ;  
  
make\_known::text::myname << “John Doe” ;  
print::text << “My name is: ” ;  
print::text << get\_known::myname; ;  
  
BASIX\_END

In het eerste blokje code wordt de variabele ‘myint’ aangemaakt. ‘myint’ Is van het type integer en heeft de waarde van 10. De waarde van ‘myint’ wordt uitgeprint naar de console.

In het tweede blokje wordt de variabele ‘myint’ aangepast, het heeft nu de waarde 20. Vervolgens wordt de nieuwe waarde geprint naar de console.

In het derde blokje code wordt een nieuwe variabele ‘mathint’ gemaakt. Deze variabele heeft de waarde van som 20 en 12.

In het vierde blokje wordt een text variabele ‘myname’ aangemaakt. Deze heeft de waarde ‘John Doe’. Vervolgens wordt deze geprint naar de console.