Onderzoeksrapport LUA + Python





Icon

Description automatically generated

Geschreven door: Mathieu te Stroete.

Contents

[LUA 3](#_Toc101705507)

[Korte algemene beschrijving 3](#_Toc101705508)

[Kenmerken van LUA 3](#_Toc101705509)

[Wat wordt ondersteund door Lua? 3](#_Toc101705510)

[Basic types 3](#_Toc101705511)

[Advanced 3](#_Toc101705512)

[Code conventions: 3](#_Toc101705513)

[Manual 3](#_Toc101705514)

[Code voorbeelden in Lua 4](#_Toc101705515)

[Hello world. 4](#_Toc101705516)

[Variabelen aanmaken. 4](#_Toc101705517)

[Flow 4](#_Toc101705518)

[Functions 4](#_Toc101705519)

[Wat spreekt mij aan in de taal. 5](#_Toc101705520)

[Python 6](#_Toc101705521)

[Hello world. 6](#_Toc101705522)

[Duidelijke kenmerken van deze taal. 6](#_Toc101705523)

[Code conventions: 6](#_Toc101705524)

[Ondersteuning voor classes, arrays etc. 6](#_Toc101705525)

# Lua

## Korte algemene beschrijving

LUA is een programmeertaal oorspronkelijk verschenen in 1993 en ontworpen door Roberto Ierusalimschy, Waldemar Celes en Luiz Henrique de Figueiredo. De taal is voor het laatst geüpdatet op 29 Juni 2020. Daarnaast valt de taal sinds versie 5.0 onder de MIT-licentie, wat betekent dat het opensource is en gratis te gebruiken en aan te passen is voor commerciële doeleinden.

## Kenmerken van Lua

* Lua is dynamically typed. Dit betekend dat een variabele niet van tevoren beschreven hoeft te worden en alle vormen aan kan nemen.
* Lua wordt niet direct geïnterpreteerd maar wordt eerst naar bytecode geschreven. Dit wordt daarna uitgevoerd door de virtuele machine van Lua. Dit kan gedaan worden voor executie om het geheugengebruik te beperken.
* De interpreter van Lua is erg klein (ongeveer 150 kilobyte).
* Lua heeft zijn eigen garbage collector.
* Lua wordt voornamelijk gebruikt om embedded in een applicatie te draaien.

## Wat wordt ondersteund door Lua?

### Basic types

* Nil (null).
* Boolean (True or False)
* Number (Vergelijkbaar met double in Java)
* String (Array van characters)
* Function
* Userdata
* Thread
* table

### Advanced

* Lua ondersteund arrays in de vorm van een table
* Lua ondersteund indirect classes. Het maken van classes kan bereikt worden met gebruik van tabellen.
* Lua ondersteund standaard niet overerving. Dit kan wel bereikt worden door met tables.

## Code conventions:

* Lua is dynamically typed, een variabele kan dus alle vormen aannemen.
* Lua eindigt niet met ; aan het eind van een regel.
* Methodes in Lua gebruiken geen brackets om hun scope aan te geven. In plaats daarvan gebruikt het whitespacing voor het aangeven van een scope.
* Comments in Lua beginnen met een –

## Manual

Voor nog veel meer informatie over Lua: <https://www.lua.org/manual/5.4/manual.html>

## Code voorbeelden in Lua

### Hello world.

print("Hello, World!")

Alternatief:

print 'Hello, World!'

### Variabelen aanmaken.

**local** nummer = 1

**local** tekst = ‘tekst’

**local** boolean = False

### Flow

**if** condition **then**

*--statement body*

**elseif** condition **then**

*--statement body*

**else**

*--optional default statement body*

**end**

**while** condition **do**

*--statements*

**end**

**repeat**

*--statements*

**until** condition

**for** i = first, last, delta **do**

*--statements*

**end**

### Functions

**function** addto(x)

# Wat spreekt mij aan in de taal.

Ik vind het erg interessant dat Lua vaak embedded gebruikt wordt. Dit omdat de interpreter van Lua erg klein is. Hierdoor kan het makkelijk geïmplementeerd worden in al bestaande omgevingen. Daarnaast executeert de code zich snel en heeft het een relatief lage leercurve. Om deze redenen wordt Lua soms gebruikt in videogame development wat ook een interesse van mij is.

# Python

## Korte algemene beschrijving

Python is een programmeertaal oorspronkelijk verschenen in 1991 en ontworpen door Guide van Rossum, as vervolg op de ABC programmeertaal. Python wordt nog steeds uitgebreid, en krijgt meestal meerdere updates per jaar. Python is gratis om te gebruiken en valt onder de Python Software Foundation License.

## Kenmerken van Python

* Python is dynamically typed. Dit betekend dat een variabele niet van tevoren beschreven hoeft te worden en alle vormen aan kan nemen.
* De python interpreter zet de gemaakte code om naar bytecode. Deze wordt daarna door de python-virtual-machine omgezet in machine code.
* Python heeft zijn eigen garbage collector.

## Wat wordt ondersteund door Python?

### Basic types

* Int
* Float (vergelijkbaar met double in Java, dus met double precision)
* Str
* Bool
* Complex
* Dict (dictionary)
* Frozenset
* List
* Range
* Set
* Tuple

### Advanced

* Python ondersteund arrays natively in de vorm van lijsten. Daarnaast zijn er packages beschikbaar voor echte arrays (pandas / numpy bijvoorbeeld).
* Python ondersteund classes
* Python ondersteund overerving.

## Code conventions:

* Python volgt duck-typing, een variabele kan dus alle vormen aannemen.
* Een regel in Python eindigt niet met ; aan het eind van een regel.
* Methodes in Python gebruiken geen brackets om hun scope aan te geven. In plaats daarvan gebruikt het whitespacing voor het aangeven van een scope.
* Comments in Python beginnen met een #.

## Manual

Voor nog veel meer informatie over Python: <https://docs.python.org/3/>

## Code voorbeelden in Python

### Hello world.

Print(“Hello world”)

### Variabelen aanmaken.

nummer = 1

tekst = ‘tekst’

boolean = False

### Flow

**if** condition**:**

*#statement body*

**elseif** condition**:**

*#statement body*

**Else:**

*#optional default statement body*

**for** item **in** iterator:

*#statement body*

**while** condition:

*#statement body*

### Functions

**def** addto(x):

*#all the stuff in the function.*

return x

# Wat spreekt mij aan in de taal.

Python is voor veel mensen de eerste taal waarin ze beginnen te programmeren. Aangezien wij met de opleiding voornamelijk met Java hebben gewerkt vind ik dit een interessante taal om hiermee te vergelijking. In uitstraling is python duidelijk verschillend met Java of C#, door het duck-typing en gebruik maken van whitespacing in plaats van brackets voor de scoping.

Daarnaast maakt python veel gebruik van andere libraries om taken te doen. Denk hierbij aan bijvoorbeeld alle python packages beschikbaar op pypi.org waar 371 duizend verschillende packages beschikbaar voor allerlei doeleinden. Hiermee kan je in een zeer korte tijd applicaties bouwen met veel diepgang.