

# Non-English based Programming



Credit: User: [Dmellas](#)  
CC BY SA 3.0

An Introduction and presentation of some concepts for  
doing it

# Overview

- Background
- Problem statement
- Examples
- How to translate functions



# **Non-English-Based Programming languages**

Programming Languages that do not use keywords taken from or inspired by English vocabulary including comments, variables, functions and other elements.

# Creating a function in Python

- Define a function with the ***def*** keyword
- Write the function identifier (name) followed by parentheses and a colon.
- **To call this function**, write the name of the function followed by parentheses:

***myfunction()***

```
def functionName():  
    # What to make the function do
```

```
def myfunction():  
    print("Hello World")
```

# Creating a function/Methods in Ruby

Functions are declared using the `def` keyword:

```
1  def greeting
2    puts "Hello Ruby"
3  end
4
5  greeting()
6    => Hello Ruby
```

- Functions in Ruby are created using the `def` keyword (short for define).
- Functions that exist in an object are typically called methods.
- Functions and methods are the same, except one belongs to an object.
- Objects are created from classes using the `.new` method

# Visual Programming Languages

Based on short sentences with gaps for variables

Blocks have different colors and shapes and can be

Combined using Drag & Drop

Scratch & Snap! Are translated into many languages

There are also other visual programming language and different Frameworks

For defining own Block Based User Interfaces, for example Blockly



# Where can you use Non-English-Programming?

- Spreadsheet functions
- Visual Programming Languages, like Scratch or Snap!, LabView, Visual Basic, Scala, MATLAB etc
- Translating existing functions or defining new functions in a programming language of your choice

# How to define and translate R functions

- Functions make calculation easier
- Function to calculate the area of a Rectangle, with English and German function name:

```
Rectangle <- function(g, h){g*h}
```

```
Rechteck <- function(g, h){g*h}
```

Replacing the assignment of the name changes the function name

Can be also used for predefined functions.



# Spreadsheet functions

- Spreadsheet functions are translated into many languages
- For example the following two Functions:

MID(A2,3,5)

TEIL(A2;3;5)

Both functions do the same thing, extracting out of Cell A2 starting at character 3 a five characters long string

# Function definition in other Programming languages

- You can find a overview about how to define functions in many programming languages at

[https://rosettacode.org/wiki/Function\\_definition](https://rosettacode.org/wiki/Function_definition)

- The allowed characters for function names depend on the way how the Operating System handles internationalization and on the allowed input of the Programming Language
- Experienced not-allowed characters in the past when defining R-Functions in an older R-Version

## Questions

- **Hogü-456**

**tim.herb@gmx.de**

- **Dnshitobu**

**fuseinikamaldeen@gmail.com**

