

# Programming

In this section, we present some references on the basics of programming in different languages. This course does not specify which language to use, so you are free to choose the one that suits you best for the various exercises and your project. In any case, there are a set of best practices that will help you write code that is readable, understandable, and reusable. You can find a (quick) guide to best practices in programming at this [link](#).

## 1 R

R is a programming language specialized in statistical analysis and data visualization. It is free, open-source, and available on Windows, macOS, and Linux. Regularly updated, new packages are available every day (see [CRAN](#)).

Here is a guide that introduces the basics of R: [Learn R in Y Minutes](#) and a guide to [good practices in R](#). Finally, an important feature of the R language is the *pipe* operator. This makes the code clearer. You can find an explanation of this operator [here](#).

## 2 Python

Python is a general-purpose programming language. It is free, open-source, and available on Windows, macOS, and Linux. Although it is a general-purpose language, there is a large community around data analysis and machine learning in Python. Thus, the usual methods have already been implemented, e.g. in the [sklearn](#) package, but if necessary, it is possible to view the code and modify it.

Here is a guide that introduces the basics of Python: [Learn Python in Y Minutes](#) and a guide to [good practices in Python](#).

### 3 Julia

Julia is also a general-purpose programming language. It is a much newer language than its two counterparts, R and Python (Julia was created in the early 2010s, while R and Python were created in the early 1990s). It may therefore seem less mature in some respects, but most traditional data analysis methods have been implemented. Like R and Python, this language is free, open-source, and available on all major operating systems. Julia has the advantage of being faster than R and Python. It also includes native support for calling libraries in C or Fortran, and non-native support for calling libraries in R and Python.

Here is a guide that introduces the basics of Julia: [Learn Julia in Y minutes](#) and a guide to [best practices in Julia](#).

### 4 SAS

SAS is a proprietary programming language specialized in data analysis. Although used in certain industries, e.g. pharmaceuticals, its use is declining. Furthermore, the license must be renewed annually, there is no online community (apart from the official documentation, there is not much help available), and it is impossible to view the code for the various procedures. For these reasons, I do not recommend using SAS for this course. However, if you wish, the university can provide licenses for a fee.