

# Lecture II - First Steps in Julia

Applied Optimization with Julia

Dr. Tobias Vlček

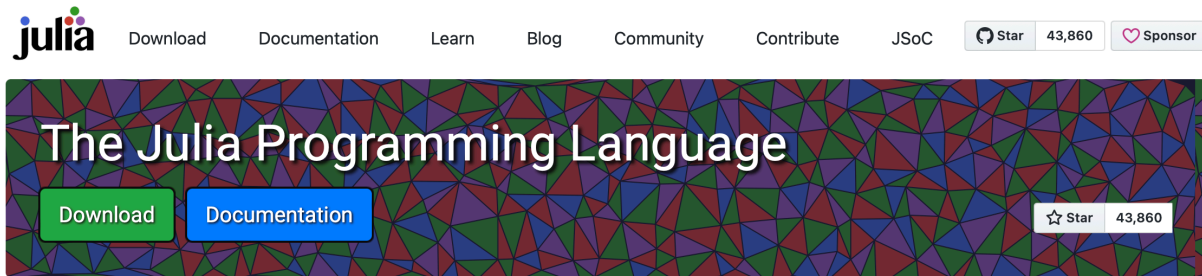
# Quick Recap on the Technical Setup

## Download and Install Julia



To prepare for the upcoming lectures, we start by installing the Julia Programming Language and an Integrated Development Environment (IDE) to work with Julia.

# Installing Julia



## Julia in a Nutshell

### Fast

Julia was designed for [high performance](#). Julia programs automatically compile to efficient native code via LLVM, and support [multiple platforms](#).

### Dynamic

Julia is [dynamically typed](#), feels like a scripting language, and has good support for [interactive](#) use, but can also optionally be separately compiled.

### Reproducible

[Reproducible environments](#) make it possible to recreate the same Julia environment every time, across platforms, with [pre-built binaries](#).

### Composable

Julia uses [multiple dispatch](#) as a paradigm, making it easy to express many object-oriented and [functional](#) programming patterns. The talk on the [Unreasonable Effectiveness of Multiple Dispatch](#) explains why it works so well.

### General

Julia provides [asynchronous I/O](#), [metaprogramming](#), [debugging](#), [logging](#), [profiling](#), a [package manager](#), and more. One can build entire [Applications and Microservices](#) in Julia.

### Open source

Julia is an open source project with over 1,000 contributors. It is made available under the [MIT license](#). The [source code](#) is available on GitHub.

- Head to [julialang.org](https://julialang.org) and follow the instructions.

...

### 💡 Tip

If you are ever asked to add something to your "PATH", do so!

## VS Code



- Next, we are going to install VS Code
- Alternatively, you can install VS Codium
- It is essentially VS Code but without any tracking by MS

## Installing VS Code

- Head to the website [code.visualstudio.com](https://code.visualstudio.com)
- OR to the website [vscodium.com](https://vscodium.com)
- Download and install the latest release

## Verify the Installation

- Start the IDE and take a look around
- Search for the field “Extensions” on the left sidebar
- Click it and search for “Julia”
- Download and install “Julia (Julia Language Support)”

## Create a new file

- Create a new file with a “.jl” ending
- Save it somewhere on your computer
- e.g., in a folder that you will use in the course

```
print("Hello World!")
```

Hello World!

- Run the file by clicking “run” in the upper right corner
- OR by pressing “Control+Enter” or “STRG+Enter”

## Everything working?

- If the terminal opens with a Hello World! 🎉 perfect!
- If not, it is likely that the IDE cannot find the path to Julia
- Try to determine the path and save it to VS Code
- After saving it, try to run the file again

### Tip

Don't worry if it is not running right away. We will fix this together!

# Learning Julia

## Julia as a Programming Language

- Following three lectures are dedicated to learning the basics
- Start with the very basics and gradually move on
- Focus in the first two lectures on the programming language
- Third lecture dedicated to Mathematical Optimization

# Working with IJulia

## IJulia

- IJulia is an interface between Julia and Jupyter Notebooks
- Popular tool for data analysis and visualization
- You can use IJulia to run Julia code in the notebooks

...

### Tip

You can also copy and paste code from the notebooks into your IDE!

## Installing IJulia

- Open the VS Code IDE and start a terminal
- Start Julia by typing `julia` in the terminal
- Install IJulia by typing `]` to open the package manager
- Install IJulia by typing `add IJulia`
- Press `Enter`

## Running IJulia

```
using IJulia; notebook()
```

- Start IJulia by typing the above code in the Julia prompt
- This will open a new browser window
- You can now run code in the notebooks

...

### Tip

You can also run the notebooks in VS Code, if you prefer!

## Downloading the Notebooks

- You will find the tutorial notebooks next to the tutorial pages

- On each page, you will find a button `Jupyter` on the right
- Click it to download the notebook and save it
- I'd recommend storing the notebooks in a separate directory for this course

## **Learning by doing**

- The best way to learn a programming language is by doing
- We will therefore solve problems the coming weeks
- The goal is to get you familiar with the language
- You can discuss the problems with your fellow students
- You can hand in your solutions to receive bonus points!



# Submission of Assignments

## Submission of Assignments

- You can work in groups of up to three people
- Submit the assignment via OpenOlat
- You will submit your assignment by uploading a notebook
- The assignment is due the day before the next tutorial

...

### Tip

Don't forget to save your notebook before uploading it to OpenOlat!

## Grading of Assignments

- Each assignment is worth 0.5 points
- You can get a maximum of 6.0 points from the assignments
- The points will be added to your exam points
- You need to pass the exam first, to receive any bonus points!

...

### Note

The assignments are **not** mandatory, but highly recommended!

# Five Tutorials for this Week

## Topics of the Tutorials

- **Variables:** Learn how to assign values to variables
- **Vectors:** Learn how to create and manipulate vectors
- **Comparisons:** Learn how to compare values
- **Loops:** Learn how to use loops to repeat code
- **Scope:** Learn about the scope of variables

## Get started with the tutorials

- Download the first notebook and open it
- Start with the first problem and solve it step by step
- You can find the tutorials here on the website
- You can ask questions anytime!

...

### Note

#### **And that's it for this lecture!**

The remaining time we will already start working on the first problems.

# Literature

## Literature

- Lauwens, B., & Downey, A. B. (2019). Think Julia: How to think like a computer scientist (First edition). O'Reilly®. [Link to the free book website](#).
- [Julia Documentation](#)

For more interesting literature to learn more about Julia, take a look at the [literature list](#) of this course.