

Lecture II - First Steps in Julia

Applied Optimization with Julia

Dr. Tobias Vlček
University of Hamburg

Technical Setup

Goals for Today

After this lecture, you will have:

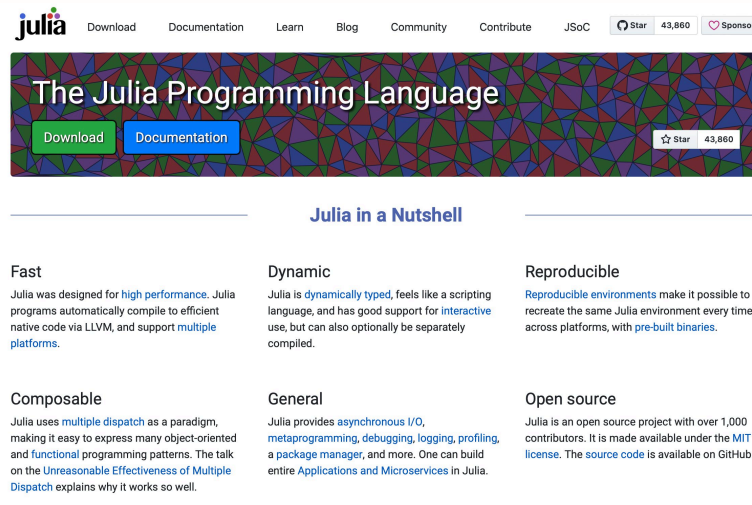
- A **working Julia and Zed setup** on your computer
- Made your first Git commit
- Learned how to submit assignments and earn bonus points
- Started with the first tutorials

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



- Head to julialang.org and follow the instructions.
- The easiest way to install Julia is via the shell/terminal
- Later, you can then manage Julia with `juliaup`



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

Zed

- Next, we are going to install Zed, the editor of this course
- macOS: `brew install --cask zed` (or the installer from zed.dev)
- Windows: installer from zed.dev/windows
- Linux: `curl -f https://zed.dev/install.sh | sh`



On Windows, Zed needs a DirectX-11-capable GPU. If it only shows a blank window, use VS Code with the “Julia (Julia Language Support)” extension instead: everything in this course works there too.

Verify the Installation

- Start Zed and take a look around
- Open the command palette: `Ctrl+Shift+P` (Windows/Linux) or `Cmd+Shift+P` (macOS)
- Choose `zed: extensions` and search for “Julia”
- Install the Julia extension for syntax 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

```
1 println("Hello World!")
```



Hello World!

Run the File

- Open a terminal in the folder where you saved the file
- Run it with `julia test.jl`
- Later, we will run files **cell by cell** directly from Zed — the first tutorial sets that up

Everything Working?

- If the terminal prints `Hello World!` → perfect!
- If not, the terminal likely **cannot find the path** to Julia
- Close and reopen the terminal after installing Julia, and say yes to any “PATH” question during installation
- Then try to run the file again



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

Learning Julia

Julia as a Language

- The 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**

A First Taste of Julia

Question: What will your savings be worth in 10 years?

```
1 savings = 1000          # Euros in your account
2 interest_rate = 0.03    # 3% interest per year
3 value = savings * (1 + interest_rate)^10
4 println("Value after 10 years: ", round(value; digits=2))
```

Value after 10 years: 1343.92

- Don't worry if this looks new — you will be able to **read and write** code like this after the tutorials!

Your Turn

Build the turbines with the most energy: tap a site. Bigger turbines give more, but no turbine may stand in another's wind shadow. Add sites to see how long a loop that tries every combination would need, at a billion per second.



+10 sites

2^{10} combinations: a loop needs
 $1 \mu\text{s}$



You

0 GWh

Ready: press
Optimize to compare
with the best plan.

Optimize

Reset

New

Julia in Zed

Tutorials in Zed

- The easiest way to work on the tutorials is a `.jl` file in Zed, run cell by cell
- For detailed instructions, just open the [first tutorial](#)
- It explains step-by-step how to set up the course environment and run the cells

 If you use `.jl` files, you can also put them under version control with Git, as you will see later in this lecture.

Downloading the Tutorials

- You will find the tutorial files next to the tutorial pages
- On each page, you will find a button `Julia` on the right
- Click it to download the `.jl` file and save it
- The file is split into cells by `# %%` markers, ready to run cell by cell
- I'd really recommend storing the files **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!

Working with Git

What is Git?

- Git is a **version control system** that tracks changes in your code
- Helps you collaborate and keep track of your work
- Saves “snapshots” of your project at different stages
- You can always go back to previous versions if something goes wrong
- No need to create files like `tutorial_v1.jl` and `tutorial_v2.jl`

Installing Git

- Head to git-scm.com and download Git
- Follow the installation instructions on the website for your OS



If you have any questions, feel free to ask!

Git in Zed

- Zed has **built-in Git support!**
- Open the **Git panel** with the Git icon in the status bar
- Staging, committing and file history need no extension



Git is not required for the course, but strongly recommended: once you get used to it, it becomes invaluable, especially if you are working with a lot of code!

Initialize a Repository

- Open your project folder (of our lecture) in Zed
- Open the Git panel and initialize a repository
- Or run `git init` in a terminal in that folder
- Your folder is now a Git repository!



You can also synchronize your repository with GitHub or other hosting services. Then, your code is saved in a remote location, making it accessible from anywhere and allowing collaboration with others.

Making Your First Commit

- Make changes to your files (e.g., work on a tutorial `.jl` file)
- Open the Git panel
- You'll see your changed files listed there
- Tick the checkbox next to files to **stage** them
- Add a commit message describing your changes
- Press `Cmd+Enter` (macOS) or `Ctrl+Enter` to **commit**

Viewing History

- Right-click a file and open its **File History**
- See every commit that changed this file
- Select a commit to see what changed in it



Start using Git from day one! Even for small projects, it's a good habit to develop.

Assignments

Find Your Group

- You hand in the assignments as a **group of up to three**
- Take a few minutes now: who do you want to work with?
- Tell me your group today and I will set it up in Moodle
- Your group stays the same for the whole semester



You would rather work alone? That is fine too: you are then a group of one.

Submission of Assignments

- Submit the assignment via Moodle: one upload counts for your group
- Upload the tutorial `.jl` files and keep their file names
- The assignment is due **at the start of the next tutorial**
- After that, Moodle does not accept uploads anymore



Don't forget to save your files before uploading them to Moodle!

Grading of Assignments

- Each assignment gives up to 5 points in Moodle
- Divided by 10, that is up to 0.5 bonus points for the exam
- You can get a maximum of 6.0 bonus points from the assignments
- You need to pass the exam first, to receive any bonus points!

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

What About AI?

- You may use AI tools: say so in one line in your submission
- But the bonus points only count if you pass the exam
- And in the exam, there is no AI to help you
- So it pays off to **solve the tutorials yourself**

 The full rules are in the [AI policy](#) on the course website.

Five Tutorials

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
- **Dictionaries:** Learn how to store and look up key-value pairs

 Dictionaries will become important later: they are how we will store and look up the data of our optimization models.

Get Started with the Tutorials

- Download the first tutorial file 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!**

 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.