Onboarding

Julia on HLRS Cluster/Laptops

HLRS Training Cluster

The cluster training.hlrs.de has two types of nodes.

CPU nodes

- “skl”
- 2x Intel Skylake
- 40 cores total

GPU nodes

- “clx-ai”
- 2x Intel Cascade Lake
- 36 cores total
- 8x NVIDIA V100

Jobs are scheduled with PBS Pro.

- Submit a job:
  - `qsub job_script.sh`

- Check on your queued/running jobs:
  - `qstat -nw`

VS Code → HLRS Cluster

Run VS Code on a cluster node via SSH.

Login node

- Works fine, just connect to
  - `accountname@training.hlrs.de`

Compute node

- At HLRS, possible but inconvenient
  - `SetEnv PBS_JOBID=…`
  - `SSH ProxyJump`

We will stay on Login nodes for the course.

To get Julia, load the necessary system modules.

- Modules on the HLRS training cluster
  - `module use julia`
  - `module use nvidia/nvhpc     # MPI+CUDA`
  - `module use compiler/nvidia # MPI+CUDA`

- Outside of the course: If there is no (working) system module, use standard binaries provided by `juliaup`.

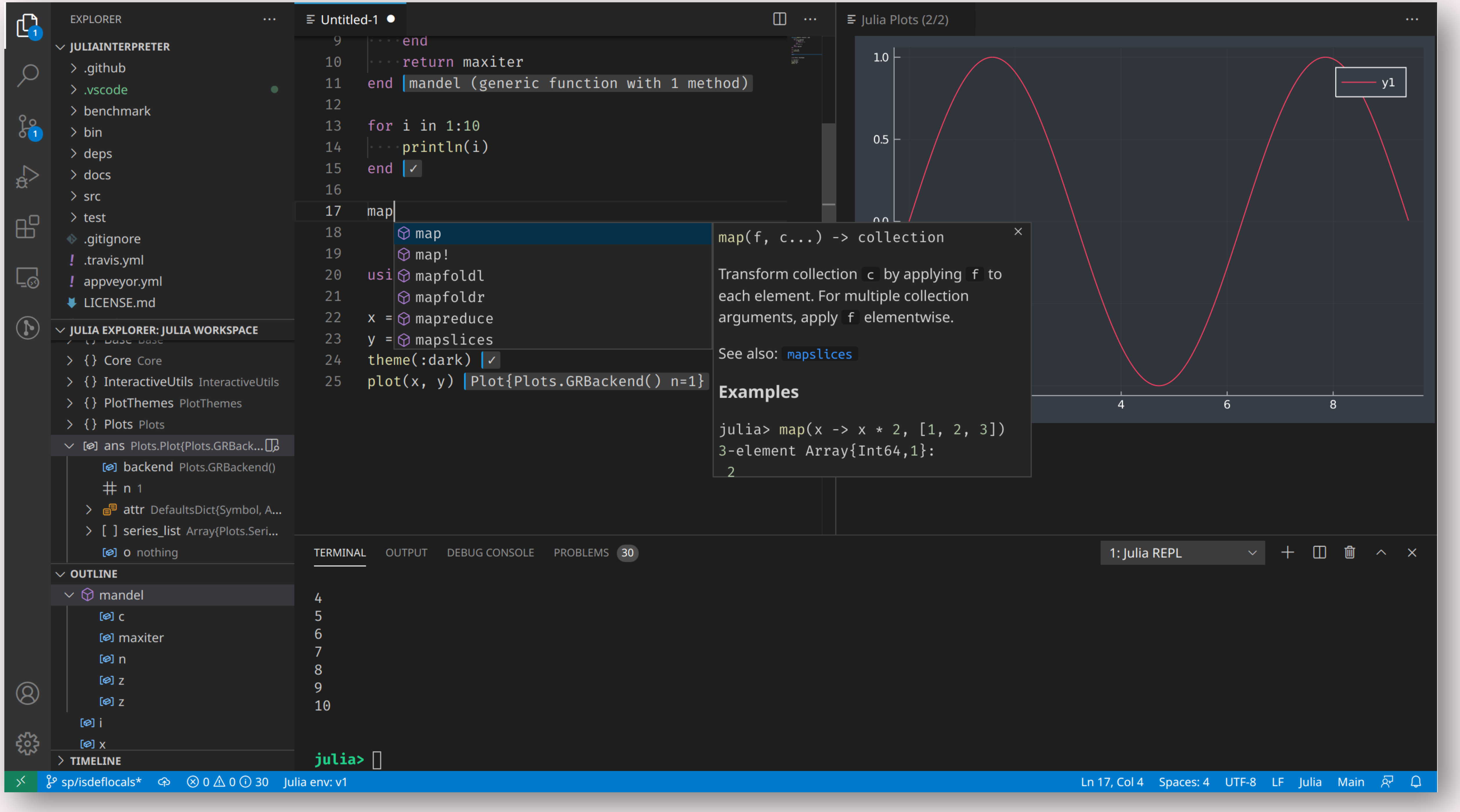
Comment: Julia depot is on the parallel file system.

- Julia depot = where Julia stores stuff
  - packages
  - binary dependencies
  - ...

- Environment variable: `JULIA_DEPOT_PATH`

- Why not `$HOME`?
  - Quotas
  - Read-only from compute jobs (sometimes)

Julia VS Code integration via extension.



```
    end
    return maxiter
end  mandel (generic function with 1 method)

for i in 1:10
    println(i)
end  ✓

map


usi

x =
y =
theme(:dark)  ✓
plot(x, y)  Plot{Plots.GRBackend() n=1}
```

`map(f, c...) -> collection`

Transform collection `c` by applying `f` to each element. For multiple collection arguments, apply `f` elementwise.

See also: `mapslices`

Examples

```
julia> map(x -> x * 2, [1, 2, 3])
3-element Array{Int64,1}:
 2
```

```
4
5
6
7
8
9
10


julia> 
```

On the cluster, the extension requires a wrapper.

- Julia: Executable Path should point to a wrapper script, like this one:

```
#!/bin/bash
[...]

# Load modules
module load julia
module load nvidia/nvhpc
module load compiler/nvidia

# Act like Julia (i.e. pass on all arguments)
exec julia "${@}"
```

Let's do it!

→ exercises/Day1/1_cluster_onboarding

HLRS Laptops

Most of the course can be completed on the laptops.

- Equipped with NVIDIA GPU

- Jupyter + VS Code
  - `jupyter lab`
  - `code`

- Course materials
  - `$HOME/JuliaHLRS24`

Let's start Jupyter!

→ cd JuliaHLRS24

→ jupyter lab

→ notebooks/Day1/1_julia_fundamentals.ipynb