

- Tips**
- Prioritize **structure over detail**
 - It's okay to leave gaps—this is about insight, not completeness.
 - Include things you found **confusing or hard**—those are valuable too.
 - Be open to revising as you go.

Agenda

1. Warm-Up (5 min)

Write down any major themes, concepts, or topics you remember from the course. Don't overthink—go with what stands out.

2. Agree on the Main Structure (10 min)

Together, choose 5–7 **main branches** for the map. These should be the “big ideas” or central threads of the course.

3. Build the Map (45 min)

Work together to expand each branch:

- Add subtopics, key ideas, tools, or examples.
- Discuss connections: how do different parts relate?
- You can each take the lead on 1–2 branches, or build collaboratively.

4. Reflect and Debrief (15–20 min)

As a group, discuss:

- What topics were easiest/hardest to recall?
- Were there surprises or aha moments?
- How might this structure help someone else (or your past self) understand the course?
- Any patterns or themes that weren't clear during the course but make sense now?

Chiara

- Functions
- mathematical concepts and formulas
- efficient code
- github
- terminal
- the logic behind problems
- loops
- optimization
- working together
- creating a website
- R can draw
- R projects
- make code faster
- extra R functions file
- main and branches
- Turing machine
- Markdown
- pull and push
-

Luca

- General features of coding: loops, functions
- Optimization
- Sharing and progressing on code together (GitHub)
- Debugging
- Cloud
- Learning Different programming languages
- terminal coding
- R projects, .rdm
- Creating a website
- turtle project
- the concept behind pi

Emily

- functions
- loops
- if-else
- optimization
- github
- forking
- terminal
- pi
- turtle
- math
- own website
- markdown
- benchmark
- while functions
- turtle package
- adjusting the colours in R program
- beginning video of pi and basics of machines

Hubi

- speedup Code
- vectorization
- generic functions
- functions in functions
- messure speed
- GitHub
- sharing code
- GitHubPages
- PI-Simulation
- Logo (simpleLanguage but powerfull)
- Problemsolving for improving skills
- ChatGPT as helper
- Book analogie to understand Programming

