

[🏠](#) / Collaborative notes archives from workshops

Collaborative notes archives from workshops

August/September 2024

Day1: Session 1 (13.08.24) - About lesson design, deployment and iterative improvement

:calendar: Schedule

Time (CEST)	Title	EEST (UTC+3)	BST (UTC+1)
8.45 - 9.00	Connecting time	9.45 - 10.00	7.45 - 8.00
9.00 - 9.15	Intro and Icebreaker	10.00 - 10.15	8.00 - 8.15
9.15 - 10.00	Lesson design and development	10.15 - 11.00	8.15 - 9.00
10.00 - 10.15	Break	11.00 - 11.15	9.00 - 9.15
10.15 - 11.00	Lessons with version control	11.15 - 12.00	9.15 - 10.00
11.00 - 11.15	Break	12.00 - 12.15	10.00 - 10.15
11.15 - 12.00	How we collect feedback and measure impact	12.15 - 13.00	10.15 - 11.00

:icecream: Icebreaker

Check-in

Which emoji best describes your current state of mind? [Emoji cheat sheet](#)

- :Decaffed ... and going strong ...
- :coffee: 😞
- :100: (ironically!) :coffee: intake happening
- :smile:
- :tired_face:
- :coffee: :party:
- :coffee: :happy:
- 😞
- just got zoom installed / before my coffee
- :smil-
- :sweat_smile: (+)
- :yawning_face:

- :is it morning already?
- :smile:
- :yawning_face:
- :nerded_face:
- 🙄
- :smile:
- :sleeping:
- :sleepy: :coffee: :smiley_cat:
- 🥱
- :coffee:
- :coffee:

Introduction in breakoutrooms.

- Name / Affiliation / Location

About teaching

What is the hardest thing about teaching for you?

- Knowing how “it’s going”, whether learners are happy or not (especially when teaching online) :+1:
- The preparation the night before. It is always much more than I would hope, no matter how prepared I want it to be :+1: :+1:
- Managing groups with vastly different academic backgrounds
- Teaching wide spectrum of skill level at the same time. :+1: :+1: :+1: :+1:
- Time management! (Knowing how much can fit in a session) :+1: :+1:
- General preparation time, how to fit it into the regular work schedule, and estimate how much time is needed for prep.
- How to reduce too much text into just the right amount
- Preparation of the session material and estimating the right amount of time for each section
- Getting learners to take the first step: sign up for and attend a workshop, when they don't think programming is a skill they can learn
- Getting learners to take the *second* step: translating what they've learned in a workshop into something they can apply
- preparation and time management
- Engaging with the students which was much easier for me when I used to coach
- Finding the right depth for an unknown audience for “my topic”
- education background of participants and their learning objectives

What is the best thing about teaching for you?

- Seeing when learner gets interested/curious about something and takes inspiration from the course. :+1:

- Seeing it works and someone can do something new.
- Motivated students grasping new stuff
- Students picking up and running with the material and skills I give them and using it for their own work. :+1:
- The “ah-ha!” moment when a student gets it! :+1::100: :+1:
- Learning from students who know about some topic more than you do. :+1:
- Being able to help people reach their goals, by showing them something new.
- Teaching is optimism acted out in the hope of making a difference both ways I suppose.
- The feeling of accomplishment when you see learning grow and implement the learnings :book:
- mutual learning and impact on learners :+1:
- Results and feedbacks / Mutual interests and interesting discussions
- mutual learning and I can also learn lots of things and new ideas from participants

:question: Questions

General / Practicalities

- Will the material be made available?
 - Materials are and will stay available at <https://coderefinery.github.io/train-the-trainer/>
 - Can I share them with a colleague
 - Yes please :)
 - Can I reuse them in my own teaching?
 - Yes please :)
- Will these sessions be recorded?
 - No, but the materials (see above) should include everything needed for self-learning.
- Can I get a certificate for this workshop?
 - You can get a certificate of attendance after the workshop by sending an e-mail to support@coderefinery.org, telling us a bit about what you have learned and which sessions you attended.
- How often will we be talking in breakout rooms? I'm not in a place where I can talk, so if we're going to talk a lot I'll need to move offices
 - We'll have about one breakoutroom session per episode
-

Episode 1: Lesson design and development

Materials: <https://coderefinery.github.io/train-the-trainer/lesson-development/> **Discussion:**

- When you start preparing a new lesson or training material, where do you start?
 - I look at existing materials on the same topic, but chose/refine topics based on the perceived needs/interests of the presumed audience
 - Outline of structure and Material collection for a new lesson :+1:
 - What material I have already? (what other material is already out there?)
 - Start by looking for related resources (documents, videos, courses, etc) to have a base

data and better understanding of what have been done around the topic

- Know your audience
- Review existing material
- Think about learning objectives (to keep focus on essential things)
- Think of three things simple enough that they will be remembered the next day.
Design around that.
- I write down the thoughts that I have and then go back and structure it.
- What tricks help you with “writer’s block” or the empty page problem?
 - Write anything down that comes to mind, sometimes draw something, looking out the window :)
 - Do a mind map - what concepts do I want to get across?
 - Starting small, for example a list of headings and then building around it. :+1:
 - What does one know about the target audience and why should they be spending longer than 1 minute listening to what they will eventually get to hear?
 - Get some inspiration from another source.
 - Start with the three things above
 - Start with the plan and the overview design
 - Some kind of outline / main message(s)
 - Start with three things. Three supporting points for each of these.
- Maybe you haven’t designed training material yet. But how do you start when creating a new presentation?
 - Think about the learning objectives and try to break them down into steps
 - Title, Objectives, target audience and Plan
 - Some kind of outline / main message(s). Get as many images as I can, instead of words
 - Draft an outline and use chatgpt to fill in details
 - Example of how the teaching content is applied in a real-world context
- If your design process has changed over time, please describe what you used to do and what you do now instead.
 - I look more at existing materials and try to get more information about the audience. Unfortunately getting information about the audience before the event is hard
 - I used to start from the beginning and get from there but that often meant a very polished start and a rushed end. Now I try an overview first, then fill out sparsed details at every section
 - Updating the data and tweak the presentation
 - If I am teaching a small group (or one to one) talk to them before hand - find out what they know already, what they want to learn.
- What do you know now about preparing lessons/training/presentations that you wish you knew earlier?
 - less is more. It’s better to have 2-3 main messages rather than trying to show everything in one go :+1: :+1: :+1: :+1:
 - how much practice time the learners need to master what’s taught
 - Don’t worry about something going wrong. It often makes the lesson (and thus the material) more presistent in memory. :+1:
 - Try and remove everything except what you want the person to learn
 - That’s a very tough part for me in the sense that I never know how much of an

underlying “black box” is still ok...

- Designing intermediate materials is hard, and requires putting some “gatekeeping” making sure that learners are directed to appropriate courses
- When I see a cool graphic, concept, slide, etc., download it and save it in my ‘new-materials’ folder to use later on!
- I have come to the conclusion that perhaps a more “agile” approach to developing materials (try to do design/teaching iterations quickly) might be the best way to go, but there are risks with this approach too

Questions

- From zoom chat: Can you expand on what a learner persona is?
 - A defined example person described in terms of the materials you are preparing, making up a story about the background and interest of t
 - Answer from chat: A learner persona is a blackbox which has a very simple algorithm inside: GIGO.
 - Answer from chat: I think it’s the same as ICP (Ideal customer persona) where you describe the learner as a customer
 - Answer from chat: I’m a big fan of “How Learning Works” by Ambrose et al <https://bookshop.org/p/books/how-learning-works-eight-research-based-principles-for-smart-teaching-michele-dipietro/18640868?ean=9781119861690>
 - See also book: [Teaching tech together](#)

CodeRefinery lessons: <https://coderefinery.org/lessons/>

- From chat: Do you have an example of an instructor guide?
 - One example from our [reproducible research lesson](#)
 - Another from git intro: <https://coderefinery.github.io/git-intro/guide/>
- From zoom chat: Do you have a “measure” of how better are the new lesson compared to the old ones? Does it make sense to talk about measurements here?
 - So far we only have daily feedback that we collect at the end of the day and we can compare that to earlier daily feedbacks. More about that in the last episode of today. But measuring this is indeed not easy and it also takes time to capture the effect during longer-term surveys. Some of the changes were more leap-of-faith than based on data.
 - I guess that sometimes one can use “compelling arguments” instead of data to justify a decision
- Software Carpentry does teach using the command line, and gives a session on the command line before. What do you think about this approach? Are you stating that a 3-hour lesson about CLI is not sufficient to make people comfortable enough to use that for version control?
 - Yes, good question. For CR we consider it a more advanced step: we do think people should learn command line, but it’s for a different course. We accept some people might not know it or want to know it so adapt to that.
 - In general, we’ve found the “you need X, but have to learn A, B, and C first” approach

Exercise in breakoutrooms

- Room 1
 - Research data management
 - Objectives
 - Research life cycle
 - FAIR principles (Findable, Accessible, Interoperable, Reusable)
 - Exercise
 - develop a data management plan (DMP) that outlines how their research data will be handled throughout its lifecycle.
- Room 2: What is the difference between eating breakfast and sitting through a lecture? (Tip: Start by listing the similarities ...)
- Room2 : Making papers in LaTeX - What is LaTeX and how is it different from editors like Word - Basic Structures - How to find and use a template - Including figure & table - How to use references and labels - Exercise: to create a new LaTeX document & edit it - Exercise: common error messages: can you find the error in this code? - Exce
- Room 3
 - GPU Programming (1 hour intro)
 - Learning objectives:
 - What is GPU programming.
 - When is it usable, beneficial to use GPU programming?
 - What are technologies to do GPU programming?
 - [What are and how to manage typical issues]
- Room 4
 - Git / Figures / Project management / data cleaning
 - Chose data cleaning
 - Three learning objectives:
 1. What do we mean by “clean”? How to identify it?
 2. Identify some common problems in a dataset
 3. Identifying and handling missing values/fields
 - Exercise
 1. Discuss problems, find the most common ones
 - In small groups
 - What problems have you run into with datasets you've worked with?
 - What problems can you imagine, or have you heard about from colleagues/news/social media?
 - Report back, instructor collects all problems into a list
 2. Have a dataset to clean; a clean and a messy example
 - (identifying) Using a visualisation or overviewing tool?
- Room 5
 - Jupyter Notebooks
 - Learning Objectives

- Can setup an environment you can reuse / can share with others / a project.
- The order you run the commands in needs to match the order of the code in the Notebook - otherwise you end up in a pickle.
- Why use a Jupyter Notebook? What are the advantages? Easy to use environment.
- Exercise
 - Print variable assignments from different cells - show that the order you run the code is important.
 - hand out a domain specific notebook that does some calculations and visualization to run and modify, to show that you can share easily
 - show !pip install to add dependencies/features
- Room 6
 - Using GitHub without the command line
 - Learning Outcomes:
 - Be able to discuss changes before merging changes into the main repository.
 - Publish a personal repository page (I think its called intro repository)
 - Share their script/notebook/code/small data on GitHub
 - Homepage using GitHub pages or the README that becomes the “index page” of the GitHub user account page
 - Learner personas:
 - Check the Personas of the learners
 - Somebody who has seen/heard of GitHub but hasn’t used it yet
 - Someone using it for their own work but struggling with collaboration
 - Exercises:
 - Upload/share an example dataset or script
 - Review another collaborators pull request.
 - Take part in the discussion on a pull request.
- Room 7 Linux shell basics: why do we want to teach them? (this took most of the time for the discussion)
 - Learning objectives:
 1. Filesystem: Directory (CLI) - Folder (GUI) analogy
 2. Basic commands (which commands are basic? make a survey of shell histories and get the most common ones)
 3. Basic constructs (e.g., for loops, pipes, while loops... which ones are basic? See above)

Note from chat: Zoom timer: https://support.zoom.com/hc/en/article?id=zm_kb&sysparm_article=KB0068677

Episode 2: Lessons with version control

Materials: <https://coderefinery.github.io/train-the-trainer/lessons-with-git/>

Poll options: I want to hear about:

- A: Why use version control for teaching materials

- B: Different template options
- C: How CodeRefinery does it; CodeRefinerys lesson template

Poll vote (multi-select): add a to your answer - A: ooo - B: oooooo - C: oooooooo :ghost:

Question to audience: if someone wants to make a tiny fix to your material, how hard is it?

- If my material is only in pdf format which is not online, then it is hard.
- I hope it is easy - my material is in GitHub. Nobody has ever done that though!
- Slides: hard to find, easy to edit; Git repos somewhat easy, unless they have to be built locally or with an action; HackMD easiest to edit.
 - if you use something like google slides they are not that difficult to find
 - known permanent address. Google docs et al fail on that/same for many pads.
- For this material today - very easy! e.g. <https://github.com/coderefinery/train-the-trainer/pull/128>
- Template with an edit button in the HTML pointing to the source-code in the repo.

Questions/Discussion topics

Upcoming CodeRefinery workshop in September: <https://coderefinery.github.io/2024-09-10-workshop/> Please share with your networks, and let us know if you would like to co-teach a lesson (to support@coderefinery.org or in our [Zulip chat](#))

- Changes to the material are sometimes not just tiny fixes. For example, one wants to present a slightly different selection of topics, but maybe some topics are “coupled” (perhaps naturally or perhaps just because of how they are presented), or change the general “theme” (e.g., life sciences vs theoretical physics) while keeping the actual topics the same. How could a lesson be developed to avoid having to do a huge rewrite?
 - You can try to modularize your material and keep non topic specific parts non themed. E.g. where you explain the functionality, don’t make reference to the example (not simple), so the example can be changed depending on the audience.
- What are the pros/cons of renaming a course / changing a repo name in GitHub?
 - It should be relatively unproblematic since GitHub will forward to the new name.
 - How long does Github keep the “old” name linked? Is there a max time it’s blocked, or changed as soon as a new one appears with the name?
 - in my experience it forwards “forever” until I create a new repo with the old name which will break the forward
- what should be in the readme for the git repo? (canonical address? contact points? license!)
 - <https://coderefinery.github.io/mini-workshop/2/documentation/#often-a-readme-is-enough-checklist>
 - I miss the url to the (main) repo in the readme
 - Do you mean the link to the rendered (lesson) page?
 - Link to rendered page: We often have this in the “about” section of the readme (see up

right: <https://github.com/coderefinery/train-the-trainer>)

- this gets lost in a fork, doesn't it? so 20 folks forking and or reuploading the repo somewhere means the url gets lost?
 - True, but they might want to have their own version rendered?
 - yes, I have had issues finding and contributing to the “master” repo so all benefit. Else it gets cluttered. Both is valid
 - Good point though, the link to rendered page should probably also be included in the readme for completeness; ah and the link to the repository too, yes
 - so realistically we need a content folder and some more next to the readme as a best practice. sounds good!
- Is myst (markdown renderer) enabled by default in sphinx now?
 - to my knowledge no. we add the “myst_parser” extension to conf.py.

Cicero: <https://cicero.readthedocs.io/> Sphinx documentation: <https://www.sphinx-doc.org/en/master/> Carpentries workbench for creating Carpentries lessons: <https://carpentries.github.io/workbench/> This is the testing lesson Richard showed: <https://coderefinery.github.io/testing/> And the corresponding repository: <https://github.com/coderefinery/testing> You can lean back and watch, exercise coming in a bit :) Richard will now edit our train the trainer material repository: <https://github.com/coderefinery/train-the-trainer> The Sphinx extension for CodeRefinery lessons: <https://github.com/coderefinery/sphinx-lesson> Empty lesson template to build your own lesson: <https://github.com/coderefinery/sphinx-lesson-template>

More questions

- Does the coderefinery.org page built on sphinx?
 - It's built also from markdown but it uses <https://www.getzola.org/> (which is more optimized to render websites, but we could do teaching also with Zola or create the project page using Sphinx)
- Is there a way to build and preview CR lessons without the command line?
 - The GitHub workflow we use creates HTML from all branches. This means that one can push a branch to GitHub, preview it there, and then open a pull request. One thing that would be nice but we don't have yet is if a bot automatically posts the link to the preview to a pull request.
- For VCS-2, for the Carpentries Linux shell example, Lesson, Github repo, to me it seemed as though the Sandpaper setup they have makes a very nice interface, complicates the relationship between source and output. E.g on the soucre page <https://github.com/swcarpentry/shell-novice/blob/main/index.md> I could only see to 'Prerequisites' and not 'Download files' on the page <https://swcarpentry.github.io/shell-novice/>. Perhaps I have missed something?
 - I am sorry I don't know the new Sandpaper setup well enough to answer this question.
- Do we have the instructions to build the lessons available somewhere to try out later?
 - <https://github.com/coderefinery/sphinx-lesson>

Exercise VCS-1 (and VCS-2 (if time)) (<https://coderefinery.github.io/train-the-trainer/lessons-with-git/#exercises>) in breakoutrooms until xx:05

then short summary in main room and then break.

Use the notes below. Collect a list of lesson formats and discuss what you like about each of them.

- PDF slides by a presentation program
- pdf slides with beamer in latex under source control.
- Markdown slides via Github
- CodeRefinery template
- Carpentries template
- Google Slides
- git-book
- [Jupyter Book](#) - allows creating a structured reference website using RST/Markdown, including Jupyter Notebooks to show result of running code
- [mdbook](#) - create simple/minimal website using Markdown
- [Binder](#) - allows learners to run their own live version of a github repository with Jupyter Notebooks, with only a web browser
- Jupyter notebooks and jupyterhub platform
- Jupyter + Nbviewer and custom css (read only) + Binder link (hands on)
- RMarkdown
- Quarto Slides
- [Reveal.js](#) - write slides in HTML (also supports markdown)
 - Example: <https://bl-presentations.erambler.co.uk/2024-02-21-idcc24/>
- [Remark.js](#) - write slides in markdown
 - Example: <https://bl-presentations.erambler.co.uk/2024-02-19-idcc-pids-workshop/>

Room discussions:

- Room 1
 - (<https://smc-aaucph.github.io/SPIS/README.html>)
 - CodeRefinery template
 - Shared notes & blogs
- Room 2
 - demo the exercise
- Room 3
 - Mix of JupyterLab, Terminal within that, and traditional slides
 - Jupyter + RISE (benefit: slides that are editable and executable live)
 - Challenges
 - present code in an interactive way
 - handle lots of images/ figures without too much additional overhead when setting up the material

- Room 4
 - Material and tools depend on instructors
 - Some slide building tools
 - <https://revealjs.com/>
 - <https://remarkjs.com/>
 - <https://github.com/rust-lang/mdBook>
 - Using flat git repos without fancy styling to make it easier to edit but still version controlled
- Room 5
 - Using R as a GIS: <https://github.com/nickbearman/intro-r-spatial-analysis>
 - Workbook: RMarkdown, Slides: Quarto
 - No continuous intergration
 - Having a jupyter nbinder link to test the notebook
 - repo lives on codeberg in this example
- Room 7
 - Ex 1:
 - Sphinx, similar to CR
 - Sometimes pdfs generated with beamer package (or from pptx and odt), but blows up repository size if in version control
 - Ex 2:
 - Content
 - CR: Content commonly found int the content folder
 - SC: Mostly under episodes
 - Question that came up:
 - is CR/Sphinx approach mostly for technical topics? What about language.
 - I guess by our nature it's focused there. But probably could be used for others (I guess it it's much easier to use git in a technical audience.)

Episode 3: How we collect feedback and measure impact

Materials: <https://coderefinery.github.io/train-the-trainer/feedback-and-impact/>

Questions to audience

- What tricks/techniques have you tried in your teaching or seen in someone else's teaching that you think have been particularly effective in collecting feedback from learners?
 - preparing a survey and emailing it to participants as soon as the event ends (usually get <50% of answers but it's a representative enough sample)
 - do engage with the audience, give them the time to get the courage to speak up
 - Be the audience yourself
 - yes! some problems/issues I don't notice as instructor, only as listener
 - Have time (e.g. 5 min) in the session to fill out the feedback form
 - If the course/workshop has several days/sections do a couple questions every change, then a full one afterwards

- “Traffic light” feedback (e.g. for pacing or progress on exercises): give each learner two different coloured post-it notes for in-person, or use emoji for online, to indicate a current status
- Can you give tips or share your experiences about how to convert feedback into changes or actionable items?
 - ask questions about things you know you don’t know
 - “Don’t let the noise of others’ opinions drown out your own inner voice.” — Steve Jobs
- When people ask questions / get stuck with a specific step. Is there a typo I can change there and then in the workbook?
 - Being able to fix small things on the fly helps sometimes, otherwise open issues if working on github
 - If multiple people have the same question, then this is an indication that I should look at this in more detail
- How do you measure the impact of your teaching? Any tips or experiences about what you have tried or seen other courses do?
 - “How likely are you to recommend this workshop to others? (0/5)”
 - Check the ‘Garbage Out’ bin, what you find there will be what went across. The rest will be history...
- Anybody knows of good resources on survey design? Please link them here.
 - “Our job is to figure out what they’re going to want before they do... Our task is to read things that are not yet on the page.” - Steve Jobs

Questions/Discussion points

- Feedback is necessary to design good lessons. Improvement can only be done iteratively. So the faster the iteration (design/teach), the better. But on the other hand, I cannot constantly bomb our community with training event emails. So I would like to try proposing a course many times a year, but I don’t want to spam everybody many times a year.
 - :+1:
 - With CR we also have a calendar that can be imported to different calendar applications, where we add all workshops: <https://github.com/coderefinery/calendar/>
- Does an even split of too fast / too slow mean speed is about right?!
 - I take it to mean “it’s roughly where it should be”. Would be better to accomodate the sides better though... (we do try to design for a broad audience: some easy stuff, some advanced stuff, so people can make their own path)
 - It could also indicate that the prerequisites/scope are not well defined :+1:
 - Based on long term surveys we also know though that even though the workshop might have been to fast, people appreciate having “heard things before and know where to find them later”
 - perhaps the even split between too fast/too slow is ok only if the majority of votes goes for “just right” :+1:
- One of the most important disciplines in journalism is to challenge your working premises. (Bill Keller) This applies equally to learning as well.
 - Participants can be encouraged to write some posts on LinkedIn. If it is helpful some

blurbs can be created for pre and post-workshop to generate more traction for Code Refinery and future participation.

- Ask participants if they use Dev.to, hashnode etc to blog their work or use twitter etc to have a memory of their work.
- Do you have any suggestions for prerequisites? I try to make them clear, but occasionally people attend who clearly do not meet them. :+1:
 - I guess it depends how much it matters for the workshop itself. If you have group exercises etc, it probably matters? , lecture style maybe not so much?; important is that the participants themselves know that prerequisites exist and if they meet them
 - Sometimes people that do not meet the prerequisites join to get to “know what they don’t know”, find out where to find information on topics they might get interested in future etc
 - Other suggestions? Viewpoints? :)
 - Thank you - things for me to think about there :-)
 - With CR’s latest ideas, we try to make the prerequisites clear, but there is no capacity limit: it’s OK if people drop by and are less than prepared: it’s livestream, anyone can come. We try to have something for these people, too: learn what you are missing, why you might want to study more in the future.
 - Any thoughts on applying this to a more practical focused course, with a total capacity limit (i.e. not like CodeRefinery).
- Everytime you do a course on CodeRefinery, think of it as trying to bake a new cake with an old recipe with no idea about the individual taste preferences of the one you bake the cake for. Then, you will not experience too many surprises. Fine-tuning the recipe for one cohort still remains the ‘old recipe’ as far as the next ‘new cohort’ will be concerned. C’est la vie.
 - nice analogy!
- Question from chat: For examples 2 & 4, would you also consider using something like a Likert scale? E.g. Rate how well you agree with the statement “My code is more reusable as a result of attending a CodeRefinery workshop” (from 1 Strongly disagree to 5 Strongly agree)
 - Thanks! I am new to Likert scale and will read up on it. In my experience numbered answers can be useful for reporting. But I also like questions where we ask for what actually to change.
- Comment from chat: Richard, your point crowns the point of having CodeRefinery, With Radovan’s and Samantha’s acquiescence assured, I would like to suggest you wear a crown going forward, especially for the CodeRefinery session. Do we have a deal ?
 - I don’t quite understand this
 - I *think* it’s a tongue-in-cheek-comment (joke) but I am not sure.....
- What is the ideal way to allow people to interact while course is going on in the main online session, i.e. without a breakout room? Taking a pause or discussion in the background?
 - I’ve had some success with WhatsApp groups for a course - particularly ones that run over 4 (or 2) sessions.
 - A quick survey/quiz
 - Question to the audience that we take time to answer but it helps to show the

Wrapup

- Thank you for active participation :)
 - We'll continue answering questions after the workshop and make the cleaned up (no names etc) available from the materials
- Next session **Tuesday August 20, 9 CEST: Tools and techniques adopted in CodeRefinery workshops**
 - we will talk about on-boarding, install help, roles, screensharing, sound, collaborative notes
- All workshop materials will stay available (after the full workshop also on Zenodo :tools:)
- CodeRefinery community: <https://coderefinery.zulipchat.com/>
- If you want to co-teach with us in the upcoming (or next) CodeRefinery workshop (<https://coderefinery.github.io/2024-09-10-workshop/>); or just want to learn more about this opportunity: Send your name and e-mail address to support@coderefinery.org; we will contact you!

Feedback about today's session

What one thing was the most valuable/useful thing you learned today? Or generally one thing you liked about this session:

- Lesson design approach
- I liked the backwards lesson design, gives a name to a practice I've already been doing :+1: :+1:
- Lessons from code refinery on lesson design. Learning about Sphinx.
- Collaborative notes and anonymized archive of notes.
- it's nice to have a community to discuss these kinds of problems
- Hearing the variety of tools people use for slides (PDF, Quarto, Beamer etc.)
- Good intro of the tools, the jargon and the pedagogical philosophy of teaching tech.
- The absolute openness in having everything publicly available - materials ofc but also questions, discussions, feedback, ...
- Thinking about all content being available publicly and as a git repo
- Thanks for explaining the HedgeDoc interface, with view and edit options
 - Is there anyway to come back to this document later (e.g. tomorrow) and see what has changed since now? So I can see if any other questions have been asked/answered?
 - There is a "revision" point under "Menu" up right, which shows you different versions of this document
- Modesty: the gentle art of enhancing your charm by pretending not to be aware of it. - Oliver Herford
-

What one thing would you improve about this session?

- Perhaps some more interactivity but can be hard to plan and time
 - We'll see what we can do.
- *Maybe* mix the breakout groups up for each breakout session (although there are advantages to consistency too) :+1:
 - yes, pros and cons! For me, having consistency worked well this time.
 - Likely we will have different participants next week, and a bit of mixing will happen naturally.
- The pace of the presentations could be a little tighter (the pace is good for a discussion though)
- Richard's voice was only 80% audible
 - Sorry to hear that. It seemed fine on my end. Will check better for next session.
- Hands on simulation and collaborative notes
- I found it difficult to follow the presentation and the collaborative notes at the same time
 - We will highlight the collaborative notes more in the next session. Our strategy usually is to say to focus on the "teaching part" and use the notes only when needed. Since you can go back and also check the notes later.
 - My problem is that I want to follow both since I don't want to miss out on anything. As such, I have both windows (zoom and the notes) open side-by-side. Since there is near-constant real-time typing in the editor, and as such visible movement on screen, I get constantly distracted. This makes following the event quite exhausting.
- Heads up to participants on using the break out rooms on voice interaction and screen sharing.
 - Do you mean that we should have mentioned it more clearly in the pre workshop email?
- I found it hard to follow with so many things going on at once...
 - A collaborative document being changed at the same time as...
 - Someone speaking and explaining content at the same time as...
 - Chat in a different application at the same time as...
 - Content changing in a termin application.
 - Thank you for the feedback. We will remove the Zoom chat out for the next session and explain the use of the collaborative notes more.
- Make more use of defining and then making sure the learning objectives are met
 - True we did not talk about them much, will pick that up better for next session.
- UNIX is basically a simple operating system, but you have to be a genius to understand the simplicity. - Dennis Ritchie. You need to do a lot to prove that to be wrong, simply put.
- Allocate some time for learners to type in the answers so that we don't have to listen and type at the same time.

Any other comments?

- It's great to hear other people's experience

- Good to hear :)
- Looking to meet you onsite if there are any events and get more emails from you :)
 - We will keep you informed :)
- Thanks for putting this together, I got a lot of inspiration for my own courses
 - Good to hear :)
- I didn't know where to ask questions. It would work best *for me* to do it in zoom chat as that's the application that I'm watching in..
 - Sorry for that, just to figure out why: Did you join a bit late?
 - No. Perhaps I just missed that or it was too quick for me or something?
 - Ok, sorry. Will try to make this more clear in next session.
 - Perhaps also because there were so many things going on and jumping around. For me, the zoom chat is the constant (and obvious) place to ask questions while watching a shared screen, trying to find the right place in a moving collaborative document is difficult. It tears my attention into too many pieces. :+1:
- Had a lot of content that assumed extant knowledge that I didn't have. This wasn't in the pre-requisites.
 - Do you have examples?
 - Yes! The code refinery templates and even just the way of doing that. I was expecting more on "when designing a lesson you should think about these steps" rather than this is how we do it and this is our tool for doing it. I guess I should have read the pre-workshop content better?
 - Ok, thank you for your feedback. You are right that we actually had some content that for sure would have been easier to follow for people with background knowledge about a certain topic.
 - "If you are in CodeRefinery TTT, you probably know what version control is and why it is important." - extant knowledge, why not just add a sentence or two?
 - Please can you update in preparation for the next workshops so that I can decide whether or not they will be right for me?
 - Yes, we will think more thoroughly about this and send the info in the next email going out Wed/Thu
 - Thank you!
- Lots of the content was relevant to code refinery but not the wider topic (e.g. it was how to design a code refinery lesson and not how to design a lesson as stated in the session title)
 - We thought of the content of showing one way of doing things, ie "what we have learned", if you have suggestions on how we could make that more clear on the event page, please let us know :)
 - "in a code refinery lesson" in the lesson/workshop description?
- When you have pre-written "thank you for your active participation" it feels fake!

- He, that is true, did not think about that. Thank you for the feedback. I would not have added it if Radovan and Richard would not have provided good feedback from circling the rooms.
- I found it vague and disorganised and difficult to follow. It felt like the instructors were presenting the content for the first time and didn't know what to do next.
 - Thank you for your feedback. This is not the first time we have this training, but it is the first time in this constellation/way. The way of teaching may also sometimes make it seem new to instructors.
- I really liked the active exchange with other participants, but I also would have enjoyed a bit more technical content in the presentations. In a sense it felt less like a 'training' and more like a get-together :+1:.
 - Thank you for your feedback. We thought the name "workshop" would combine the training and exchange nature of this event. But maybe it didn't do so enough? Do you have suggestions on how to clarify it on the event page?
 - Well, the 'train the trainer' title created the anticipation of being trained ;) don't get me wrong, I liked the exchange, but I felt that the training aspect fell a little short. imho the workshop would benefit from a little more (instructional) content that is presented in a more structured way by the instructors (as opposed to 'crowd-sourced knowledge' that is collated in a loose collection of questions, notes and links)
 - Other than that, you could add another section on the main page, sth like 'Workshop structure: we aim for a mix of modular presentations and work in small groups. The latter will be done in breakout rooms - be prepared to switch on your camera and have a fruitful exchange with fellow participants!'
 - Thank you for the suggestions, will add something like this :tools:
 - Added : <https://github.com/coderefinery/train-the-trainer/commit/03308595436e636b3bb37ed9d1f0dee39eecd0f>
- My understanding (so far) of what (unlike the competition) CodeRefinery DOES NOT seek to exemplify - https://images.ctfassets.net/rxqefef3t5b/2k45BFtw4sBMIBfjr1OZ90/5c6ecd3628a43b0cdc79815f665e224a/Screenshot_2023-05-02_at_09.35.23.png?fl=progressive&q=80
- My understanding of what (unlike the competition) today's session of CodeRefinery DID exemplify - Definition of 'Workshop' from the Oxford dictionary of the English language "a period of discussion and practical work on a particular subject, in which a group of people share their knowledge and experience"
- In my experience, even for the most seasoned Programmer coding, by its very nature, is nothing trivial, this being a simple straightforward fact just like there is no such thing as an egg which is a cube. I mean there is a reason why not just about everybody is a poet either. But, other minds have pointed this out without having to first explain why the universe is what it is in a way that could be understood without effort ... <https://i.pinimg.com/originals/7b/7e/c4/7b7ec4b91d2cf9d8e1b064678d205467.jpg>

- The inclusive environment feels very welcoming. Keep up the good work!

Thank you all for your feedback! Highly appreciated!

Day 2 : Session 2 (20.08.24)- Tools and techniques adopted in CodeRefinery workshops

:calendar: Schedule

Time (CEST)	Title	EEST (UTC+3)	BST (UTC+1)
8.45 - 9.00	Connecting time	9.45 - 10.00	7.45 - 8.00
9.00 - 9.15	Intro and Icebreaker	10.00 - 10.15	8.00 - 8.15
9.15 - 9.25	About CodeRefinery workshops	10.15 - 10.25	8.15 - 8.25
9.25 - 9.55	Collaborative notes and interaction	10.25 - 10.55	8.25 - 8.55
9.55 - 10.10	Break	10.55 - 11.10	8.55 - 9.10
10.10 - 10.45	Workshop overview, roles, onboarding	11.10 - 11.45	9.10 - 9.45
10.45 - 11.05	Sound	11.45 - 12.05	9.45 - 10.05
11.05 - 11.20	Break	12.05 - 12.20	10.05 - 10.20
11.20 - 11.50	How to prepare a quality screen-share	12.20 - 12.50	10.20 - 10.50
11.50 - 12.00	Outro and feedback	12.50 - 13.00	10.50 - 11.00

:icecream: Icebreaker

Check-in

Which emoji best describes your current state of mind? [Emoji cheat sheet](#)

- :-)
- https://media.istockphoto.com/id/148421550/vector/bored-emoticon.jpg?s=612x612&w=0&k=20&c=uo5KBTqdJZbBQb_Xldhzuovx5t08fHfGR7dw7TCk90I=
- :saluting_face:
- :sleepy:
- :coffee:
- :umbrella:
- :cocktail: :coffee:
- :seedling:
- :tired_face:
- :sun_with_face:
- 🙄 (:woozy_face: not converted?)

- :coffee:
- :sunflower: :book:
- :satisfied:
- :coffee:
- :coffee:
- 🥱:tired_face:
- :bread: - I am baking bread today so I have been kneading dough while listening!

Introduction in breakoutrooms

Name / Affiliation / Location

Teaching

- Do you teach and organize teaching alone or with others? What would you prefer and why?
 - Coming together is a beginning; keeping together is progress; working together is success. – Edward Everett Hale
 - These days always with others. Alone is easier to prepare but almost always is harder during it.
 - Teaching with very diverse partners, it can be challenging to find a common language with people very different than you.
 - I teach alone, in the future there will be more collaboration with multiple team members. Both approaches have their pros and cons.
 - Both has its own distinct advantages
 - Most of my teaching is on my own, as a freelancer its more expensive to work with colleagues than to deliver a course on my own.
 - Teaching together, learns from each other, feedbacks to improve
 - Not formally taught yet, only given presentations actually .
 - collective teaching would benefit the teachers with collaborative curriculum and experiences sharing.
- If applicable, have you seen any challenges when teaching together and how to overcome them?
 - It actually requires preparation,
 - Is it about teaching or about inspiring ? Discover the answer and get inspired ...
 - It's like being in a band: you might be great at improvising and your band-partners might be great at improvising too, but a bit of rehearsing makes even the impro gig much better
 - Need to plan before session, difficult if people don't 'plan' in the same way, e.g. with the same time frame.
 - Heterogeneity in learners make it impossible to get the same result from everyone.
 - In fact, the same applies to teachers :D
 - Teaching together requires to have similar opinions on how to teach. I share vews only with ~40% of my colleagues I think.
- No Teaching experience but it is good to give the outline of the lessons beforehand and

:question: Questions

General / Practicalities

- What is this document for?
 - Asking questions, collecting discussion points, taking collaborative notes (Zoom chat deactivated, can only be used for communication with the hosts)
- Will the material be made available?
 - Materials are and will stay available at <https://coderefinery.github.io/train-the-trainer/>
 - Can I share them with a colleague
 - Yes please :)
 - Can I reuse them in my own teaching?
 - Yes please :)
- Will these sessions be recorded?
 - No, but the materials (see above) should include everything needed for self-learning.
- Can I get a certificate for this workshop?
 - You can get a certificate of attendance after the workshop by sending an e-mail to support@coderefinery.org, telling us a bit about what you have learned and which sessions you attended.
- How often will we be talking in breakout rooms?
 - We'll have about one breakoutroom session per episode
 - BR 1 : The Zoom chat does not work for the breakout room so we should take note of that. No webcam/mic = expect silence.
 - Thanks for notifying! This was not intended. Turned the chat back on.

Episode 1: CodeRefinery

Materials: <https://coderefinery.github.io/train-the-trainer/coderefinery-intro/>

- (questions continue here)
- Is there the plan to build capacity / be paid to deliver courses?
 - We've discussed it but right now we don't have many people who could accept money to do this (and maybe shouldn't, because of their jobs). But, we would encourage others to use our materials as "independents" to deliver paid courses
 - One idea we have had is that paid courses could be delivered through the DRA (link below), who we know. Basically, DRA could find instructors and handle the money transfers (If someone wanted a paid course, we could help find someone to do it through DRA or independently)
 - :smile:
- Have you come across DRA, <https://digital-research.academy/>, Heidi Seibold, Joyce Kao who are doing similar work? I can connect you if you like.

- Yes we have been discussing about possible future collaborations and are already sending possible clients their way (we, as a project cannot take money) :+1:
 - Excellent! :-D :smile:

Episode 2: Collaborative Notes

Materials: <https://coderefinery.github.io/train-the-trainer/collaborative-notes/>

- So anyone can type anonymously?
 - Yes, as long as you do not log in, you are Guest XXX
 - And at least in our hedgedoc case you cannot log in so you will always be anonymous. You can do [name=xxx], to add your name to a comment
- Test of +1
 - +1 :+1: :+1:
 - :-1: a test
- You are hosting your own version of HedgeDoc, and Hackmd.io is a similar (free) option. What are the limitations of the free hackmd.io? Should each center install their own HedgeDoc instance?
 - It **seems** the free version of HackMD has gotten worse lately. It works still well enough with a few people editing at the same time.
- Have there every been problems with the anonymity and CoC?
 - We haven't had any issues so far.
 - With a different anonymous tool in a bachelor course, we had issues with anonymous participants (participants were posting slurs and unsafe links). The course organiser decided to stop allowing anonymous tools.
- Why not google documents?
 - Google documents are actually setting a hard limit for simultaneous writing, no more than 100 concucurrent editors. I am unsure if microsoft onedrive word documents would allow more users writing at the same time.
- What's the most people you have every had on HackMD / Hedgedoc?
 - I believe about 200 active participants?
- What are your thoughts on using the build-in Zoom Q&A?
 - We haven't tried this, in part because the way we do streaming now (you'll see in session 4), participants aren't in Zoom so don't have access to that. The doc-format works pretty well though, when we keep it in this limited "write at end" format.
- How do you see what has changed? e.g. if leave the session at the end, can I come back tomorrow and see X has been added? e.g. a diff for this maybe?
 - In the menu up right there is revision, it does show the document and changes at different time step :+1: Thanks!
 - We recommend people only write at the bottom (and since we answer things live), so as a first level approximation you find where you were and read below that. Think of it more like chat than a full document. (Yes, we do revise questions at the end)
- How about flinga boards or similar tools?
 - My take is that the simplicity of a simple text document wins over free-form in-

person-workshop-like tools.

- There's miro, can be a bit confusing though
- How does this work with small groups, e.g. 10-15 people?
 - Good question: it can be hard if people aren't rigorous about it, people start only using chat/voice. So there seems to be some lower bound in size but we haven't tested this much
 - And if the instructors never screenshare it to show what goes on it, then seems to be less used.
 - And if there aren't other people to read/answer it during the session (perhaps less likely in a small course), it seems to be less used.

Breakoutrooms until xx:55

Exercise/discussion: <https://coderefinery.github.io/train-the-trainer/collaborative-notes/#exercise>

- Breakout room 1:
 - Use of etherpad for collaborative discussion.
 - Use of slido for polling and quizzing.
 - Presemo as a slido alternative (<https://www.aalto.fi/en/services/presemo-a-fully-web-based-classroom-participation-system>)
 - Use of text based approach for questions rather than live can be helpful if there is a language barrier issue.
 - A good teaching session which had many questions coming spontaneously to the teacher is a non-entity of sorts. If the overall flow of feedback / questions increases it could indicate that the session will not prove to be a helpful experience. A signal that generates noise is, well, noise, if calling a spade a spade signals anything. Note that the 'overall' bit and the 'to the teacher' aspect are important. Both the best and the worst teaching session have two things in common: a teacher, and the fact that NO Materials of any kind are used, ie speech is the only channel of communication via a common language. The only difference is that it is not the same teacher, at least not in the same session at the same time.

Another less mentioned, but perhaps more relevant indicator of a great session is the simple fact that at the end of the session EVERYONE feels empowered, NOT tired. This is not an easy to quantify thing, but screams out the quality of the experience loud. More often than not, such sessions differ little from each other, apart from in terms of the theme / topic and the participant mix. It ends up being the 'keep your cake and eat it too,' of sorts, not quite the far less amusing post-bombardment devastation scenario in the wake of a take it or leave it go at anything.

- profanity in collaborative documents where people can add content anonymously could be a problem (happens very rarely, fortunately and somewhat surprisingly)
- Breakout Room 2:

- Good experiences with Zoom polls
- Handling questions in hybrid setups: all participants (on-site and online) ask questions in written form
- Collaborative notes can be challenging when conducting a course alone (without teaching assistants) :+1:
- Providing a method to ask questions anonymously can make asking easier for shy participants
- Questions can give valuable feedback on how the course is going
- Online collaborative whiteboarding platforms such as Miro
 - Large monitors are useful for this not to have to drag a “periscope” around
- Room 4
 - good experience with whatsapp chat for questions. since most learners have the app already and one thing less to install and set up
 - reply to question is a way to handle multiple questions at the same time but it works well for smaller groups
 - experience with collaborative docs in in-person workshops: voice questions might “win” over the document
 - “seeding” questions may help to not have an empty document
 - it can help somebody to start and create a structure which helps others to see and follow
 - we discussed the Carpentries sticky notes system (red sticky note: signal tech issues and use for feedback, green sticky note: signal that exercise worked and for positive feedback)
 - Nice summary of sticky notes system: <https://teachtogether.tech/en/#s:classroom-sticky-notes> :+1:
 - What is the success rate of in-person workshops after COVID? Discussion: Teachers prefer in person and learners prefer remote or recorded to spool back and relearn.
- Room 5
 - Good experience with Collaborative tools as HackMD and HedgeDoc while teaching with no bad feedback
 - When used with smaller groups works best as one-way, like sharing links with students and so on
 - Needs to be done as a team, quite difficult if teaching alone
 - Most useful when online/hybrid
 - Anonymity is important for transparent feedback and no bias in classrooms
 - Not very collaborative but useful to share history or jupyter notebooks live https://github.com/mwakok/software_carpentries?tab=readme-ov-file#set-up-gitautopush
 - Used Flinga before that seems like Figma and also onsite teaching using sticky notes
 - There’s miro.com but can be a bit confusing, more for brainstorming than for teaching
 - Using quizzes apps like Kahoot is beneficial while teaching to have more engaging students
 -

- Room 6

- teaching teenagers (in presence) but using shared pad becomes quickly very chaotic, so only usable for short moments when answering very precise questions
 - Yeah, I can imagine. For CR we try to keep it in this very limited format (all questions bullet points, only add to the end) and I think that helps.
- using shared notepad for small workshop (5-10 people) to explore common thematic (mainly technical) to keep a shared notes and condense afterward (and publish on wiki to archive)
- Difference between teaching in person and online. Some people prefer in-person and sometimes you have no option but to organise it online.
- There are sometimes cultural differences and learners feel shy to ask / question the knowledge of the Teacher, so shared notes are a good way to safely ask questions.

Questions continue:

- Have we demonstrated how we do polls in HedgeDoc? Add your vote by adding an
 - yes: oooo (*in session 1*)
 - no:
 - :smile:
 - not sure: o

Episode 3: One workshop, many perspectives

Materials: <https://coderefinery.github.io/train-the-trainer/overview/> Next CodeRefinery workshop in September: <https://coderefinery.github.io/2024-09-10-workshop/>

Exercise until xx:39

<https://coderefinery.github.io/train-the-trainer/overview/#discussion>

- Room 1

- Send pre-workshop email with pre-requisite instructions on what should be set-up beforehand.
 - But, also mention that if help is needed for set-up, we will be ready before the session officially starts.
- Discussion on whether we push to work in a pre-ready environment (e.g. Binder, Colab) or in their own environment.
- Lots of help needed both before and after.
- Installations: as much as we can provide ready made environments for the learners, maybe installations will be something that the learners will face sooner or later. Maybe one should teach more about installations?
- My only cherished role in a teaching-learning session is the that of one who is living the moment as if there were no tomorrow. What cuts all ways really is that “By failing to prepare, you are preparing to fail.” — Benjamin Franklin. A helpful motto: “When one

teaches, two learn.” — Robert Heinlein. The best experience cannot be given, it can only be allowed to flow its way, so get out of its way.

- Room 2

- Frequently, a single person fills all the roles for a single or multi-day workshop
- Collaboration with different centers/ universities can be challenging
- Preparing participants: it's essential for the instructor to go through the demos and be sure that they will work (no unpleasant surprises live). This can be extended to advance-prep for participants.
- Best experience is to hear (live or in feedback notes) some positive things directed at you :)
- Fully prepared environments (eg self-hosted JupyterHub) can make preparation for participants easy

- Room 3

- Some students have problems with setuping the environment. Maybe having a step by step instructions could help. But some students don't want to “start” the course before hand.
- Preparing the environemnt before hand ensures everyone is onboard from day 1. The same thing happend in CSC training sessions and another session Peng attended.
- Had experience in attending as teacher and student in previous workshops

- Room 4

- knowing the background of the audience is helpful
 - Try and ask in advance of the background of the audience. Do a poll at the start of the workshop. In person one can do it interactive.
- among co-instructors: discuss topics and flow
- if more people are involved, it can help changing roles from time to time to see different perspectives
- preparation is key
- Circulate instructions beforehand
 - Do a poll before session: Have you installed (QGIS) successfully?
 - Push hard to do this in the welcome email
- communicate clearly to the participants
-

- Room 5

- We've all taken most roles
- How to prepare in advance?
 - Meet and discuss once or twice
 - Make a pre-survey to check for interest and background
 - Going through the materials
 - Checking the toolkit (zoom)
 - Agree on the reponsibilities and back-up plans (bit of scenarios planning)
- How to prepare trainings in advance?

- Collect resources and prepare the workshop plan
- Having the material readily available online and sharing it
- Making a pre-survey
- What was the best workshop experience for you as learner, helper or instructor? What made it great?
 - learner
 - Engaging and practical workshop with more exercises and follow-up theoretical part explanations
 - helper
 - When the instructor is well prepared and as helper there's only typos to fix and interesting questions to answer
 - Assist the instructor, coordinate the collaborative notes, and make sure that the time is respected
 - instructor
 - I have a colleague whom I've ran the same workshop a few times already, goes quite smoothly now and we've made a few changes. Hard to reach that point with new colleagues but getting there
 - Students/participants were engaging with the material and many good feedback collected even some of them asked for the next workshop
- Room 6
 - What has made a great experience for you as a learner/helper/instructor?
 - **Common ground between learners & teachers**
 - Motivation
 - Teaching is much more enjoyable when the learners want to be there
 - I spend a lot of time at the start of a workshop making sure learners feel that the content is possible for them to understand and master

Poll: I have joined a CodeRefinery workshop before: (add an for your answer)

- yes (within last 2 years): ooooooo
- yes (longer time ago): oo
- no: ooooooooo
-

(questions would continue below:)

- What is the motivation / advantages of streaming workshop versus using zoom?
 - We'll cover this more in week 4, but roughly: zoom became unmanageable and essentially quiet once things got big. Streaming allowed "one to many" communication with no limit, and collab notes allowed better interaction anyway. By separating the rooms, it was more pleasant to teach. Videos could be released immediately since there was no risk of privacy of participants. There could still be in-person/online small groups who have "watching parties."

- What are the blue red and green font colors here, does it symbolify anything or just a convention HedgeDoc uses?
 - Visual separation of formatting, mostly. (It helps to quickly realize you've missed/mistyped a control Markdown symbol etc.)
 - Yeah, it's basically "code highlighting" for markdown. No special meaning but quite convenient for us.
- Is this next workshop only for team leads or also open for helpers and coordinators?
 - You mean the next session of this workshop or the upcoming "CodeRefinery workshop"(this one: <https://coderefinery.github.io/2024-09-10-workshop/>)?
 - The upcoming workshop coordinators are already working, but can be joined if you are interested
 - As helper roles we currently only have "collaborative notes helper", which is all about helping to answer questions in the collaborative notes
 - We do have some co-teaching slots open :)
 - If interested: join the discussion in our chat: <https://coderefinery.zulipchat.com/#narrow/stream/316508-coderefinery-tools-workshop/topic/Fall.202024.20CodeRefinery.20workshop>
- Can a person have multiple roles in a workshop as a helper and a teacher?
 - Yes, many teachers also contribute to the rest of the workshop by answering questions in collaborative notes. :+1:
 - Some teachers also host a local classroom in week one and do their teaching in week two of the workshop
 - Roles that do not go well together are broadcaster/director/host and instructor (goes well until there is a problem; co-teaching solves that partly)

Episode 4: Sound

<https://coderefinery.github.io/train-the-trainer/sound/>

- To me, Richard is a bit quieter than the other two :+1: :+1: :+1:
 - to elaborate: I think that the audio coming from him is missing some mids
 - Do you have any links about this kind of evaluation and adjustment?
- Do you suggest / recommend a specific headset(s) with mounted microphone?
 - in our experience headsets with a microphone will almost always be better than a separate microphone sitting on the desk or in the monitor
 -
- Speakers have different tones of voice and speaking styles, is there a recommendation or experience from CR workshops for how this is adjusted and catered to environments for listeners?
 - Yeah, people's voices are different and we haven't gone as far as voice training or improving.
- How to change the volume for speakers? Tried Zoom settings and system settings and didn't help.
 - From linux I can use different pulsemixers `pavucontrol`, `pulsemixer` to set the gain to above 100%. I'm not sure the equivalent on other OSs but

Exercise until xx:10 <https://coderefinery.github.io/train-the-trainer/sound/#exercises>

- room 1
 - Practically, no typical headset is designed or can capture the technical purity of human voice within its natural environment. But one needs to understand the polar pattern of the (specific) microphone or how much of the signal will be picked up by the microphone from different directions, a sine qua non when it comes to mitigating unwanted sound sources to bleed into the signal.

For best results, keep the right direction & distance to your (directional) mic. Usually, that's between 15-30 cm to get a natural sound. A microphone records everything it can, also silence, so check the headphone output BEFORE adjusting the gain, in some specific cases aim for a dry sound, ie the influence of the room or how audible the room is on the recording ! Finally, adjust the mix between dry and room sound, or change the frequency response in tandem using a compressor to reduce the dynamic range of a signal for a more consistent listening experience = the quietest parts of your signal will appear louder by reducing the loudest parts.

It is all about one thing and one thing only - love your voice, let people fall in love with the acoustic quality of your charm. So remember: GIGO, it matters!

- Room 3
 - Had different headsets and we could see the difference (bluetooth low quality)
 - How to change the volume?
 - It's surprisingly hard to get more control than the basic slider you see in the apps!
You need to look.
- room 4
 - headset is probably better listening experience but they can be uncomfortable to wear during long meeting days :+1:
 - Yeah. Luckily only instructors
 - Therefore breaks are non-negotiable
 - loud keyboard can be an issue
 - if one has a silent room that does not echo, it can be ok to teach without headset
 - on Linux I am using `pulsemixer` to adjust levels and `pavucontrol` to change outputs/sources
 - mic modes on mac
 - check default microphone for zoom.
- room 5
 - Most tricks help tuning yourself but hard to do it as a group. Perhaps doing a short recording.
 - Yeah. Some independent recording/playing around, but also asking for advice. I would do small group/one-on-one things.
 - Background noises cancellation tools or wise choice of quiet environments
 - Audio check with colleagues or through the app with the audience saying numbers for

example

- Zoom or Teams have sound checks but work different in the app than the browser. They also work different across Operating systems
- Some sound issues might be related to network speed or stability, also to RAM available. Worthwhile to try and control those as well
- Make it welcoming for the audience to give you feedback on the audio and other settings, give reminders

Episode 5: How to prepare a quality screen-share

Materials: <https://coderefinery.github.io/train-the-trainer/screenshare/>

- If your Zoom does not support “share portion of screen”, sharing a tab and resizing it to portrait for yourself is a possible workaround (this at least works when joining the call from Chromium browser)
 - Another workaround is to use an editor/IDE that displays multiple panels in one window and can show a web page, code files and terminal session +10
- Is ‘share a portion of your screen’ where you draw a box in Zoom to share a section?
 - yes :+1:
- When using zsh, one can run something like this in a terminal

```
tail -n $NLASTCOMMANDS -f ~/.zsh_history | sed 's/^...../LAST COMMAND:/g'
```

(I place this in a window under the main terminal, I use i3)

- I am a bit confused on what platform you are putting this code?
 - Radovan is using Linux with (I think) the “i3 window manager”. It can be quite involved and there is lots of personal customization here.
- When teaching with a custom prompt, is it worth starting with the default (that your learners will see) and explicitly mentioning “I change these things to make it clearer when teaching”
 - One should always comment on what may be different. Thing is “default” may look different for many people :+1:
 - “I am *not* going to teach you how to customise your prompt, but here’s an online tutorial, be prepared to lose several days of your life...” :laughing:
- Now I can’t see the last line in the terminal
 - Yep, that’s a problem. It can be good to keep a buffer on the bottom of the screen.
 - If you move your mouse outside the Zoom window, the controls should go away. But still a good note.

- My mouse was on another screen.
- Ok, thanks.
- I think one can also make the zoom bars not auto-hide (then it won't display under), but that is annoying also since it wastes space.
- Does anybody still use shellshare? Does it work?
 - We in CR haven't used it much but we have seen it.
- Does the stacking of the screens work on windows?
 - Don't know unfortunately

Exercises (done together) <https://coderefinery.github.io/train-the-trainer/screenshare/#exercises>

Write your cool tips and ideas below. Stay after xx:00, to

- To be safe, I have created on my computer an account that I use only for teaching, so I do not need to do any manual set up
- share history or jupyter notebooks live (5-10 secs refresh rate) into a git repo. Share the link with the learners so they can catch up https://github.com/mwakok/software_carpentries?tab=readme-ov-file#set-up-gitautopush

Wrapup

- Next session Tuesday August 27 9 CEST: About teaching & cool things we all would like to share
- All workshop materials will stay available (and be put on Zenodo right after the workshop, currently other episodes are still in development)
- CodeRefinery community: <https://coderefinery.zulipchat.com/>
- Cool gems session: If you want to share something, please edit your registration and fill it in there (link to modify in registration confirmation); or e-mail to support@coderefinery.org
- If you want to co-teach with us in the upcoming (or any future) CodeRefinery workshop; or just want to learn more about this opportunity: Let us know in chat, send an e-mail to support@coderefinery.org, or join any of the upcoming Monday planning meetings at 14:00 (Europe/Stockholm time)

Feedback about today's session

What one thing was the most valuable/useful thing you learned today? Or generally one thing you liked about this session:

- Sound and screen share optimization as well as relevant tools and good practices for

online instruction. “While you may be unable to control your users’ mindset, you can anticipate it, and make sure that your user experience caters to it.” - Jonathan Cherki

- Nice opportunity to (re)think about my online teaching setup :+1: :+1: :+1:
- Other ways to display command history :+1:
- Did not know that bluetooth gear had so much latency! :+1:
- A good reminder about headsets with microphones
- Nice and friendly teaching. Thank you!
- Screen share with command history and terminal customisation
- Really useful tips on sound and screen sharing :+1:
- Nice experience share
- very useful and practical tips!
- Easy to follow along and filled with useful tips on sound, video etc.
-

What one thing would you improve about this session?

- Just digging to find something: I have read some of the things you have talked about here already in some blog posts, so perhaps these sessions are not as shokingly full of new content as I imagined (but actually sharing experiences is something I can't do by reading a blog post)
 - Perhaps: could you link some blog posts or other sources for what you have discussed so far? If it makes sense. Not sure.
- Talking about interface optimization approaches, which applies to sound as much as to visuals or any other aspect either for that matter, it might offer a perspective to bear in mind the concept of introducing asymptotic complexity increment. The latter would mean the goal is measuring how fast the effort of engaging, on one or more of the three parallel levels of interaction, must change to reach higher levels of partial certainty on the correctness of the implementation of the method in question, in this case the pedagogical paradigm. For the former, lower is better I guess, or in this case, in the intended sense, slower is faster. Let us consider an abstract example: People liked using the search engine Google (a Google session to put it in a fancy way) in the early days because it was as simple as it could get, a literally featureless entry point with no bells and whistles. That was the level of its visible complexity, or rather Google set a bar as low as it one could have, and then any higher point of VISIBLE complexity that was to be introduced, such a move would only be on the path of an asymptotic curve curving towards the set bar, never reaching it, as far as the entry point is concerned. We still see the same level of complexity when we fire up google and we never feel distracted to start with, except for the rather forgivable nuisance of the GDPR conformity pop-up notice which appears only AFTER the first move has been made, and thankfully it works plain and simple from there on (unless one starts feeling lucky, pun intended). Now, think about Youtube ... the user experience complexity is a totally different matter, the same company today (unfortunately) yet one now has a steadily worsening engagement experience. Good learning sessions tend to be more of the Google search engine experience and a far cry from the current Youtube trend I would opine. Or, to sum it up and keep it sweet and simple, it helps to identify the limits of human-human and human-technology interaction,

within the framework of any applicable technology-technology interaction and joy is the result of respecting those limits. Wishful thinking ?

- Bring the cat back!
 - I wish it came, but it's been resting all morning! I don't bother it or stage appearances...
- Maybe having a digest/ summary of previous iterations of this session might be interesting. I feel a lot of experiences have been exchanged and a lot of techniques/ tools have been mentioned. Spending some time in compiling this into a selected list and adding it as content to the course itself might be worthwhile.
 - Good idea! We will write a blog post about this workshop and will include a summary of all discussions there :)
- Possibly have some command line examples from Windows rather than (just) *nix? I will look at the screenshots after the session. :+1: :+1:
 - Also, a thought if you do do breakout sessions with this work, do one for Unix, one for Windows, one for Mac as an option.
- I could not follow the part Radovan showed codes (Radovan is using Linux with (I think) the "i3 window manager".)
- ..

Any other comments?

- Looking forward to the topics of following sessions.
- Looking forward to the next sessions!
- Thank you!
- Relax, nothing else matters, live a life and have fun ! If you obey all the rules you may end up missing all the fun. 'The best way to predict your future is to create it.' - Abraham Lincoln :+1:

Day 3 : session 3 (27.08.24) - "About teaching and cool things we all would like to share"

:calendar: Schedule

Time (CEST)	Title	EEST (UTC+3)	BST (UTC+1)
8.45 - 9.00	Connecting time	9.45 - 10.00	7.45 - 8.00
9.00 - 9.15	Intro and Icebreaker	10.00 - 10.15	8.00 - 8.15
9.15 - 9.45	Computational thinking	10.15 - 10.45	8.15 - 8.45
9.45 - 10.00	Break	10.25 - 11.00	8.25 - 9.00
10.00 - 10.30	Teaching philosophies	11.00 - 11.30	9.00 - 9.30
10.30 - 11.00	Co-teaching	11.30 - 12.00	9.30 - 10.00
11.00 - 11.15	Break	12.00 - 12.15	10.00 - 10.15
11.15 - 11.50	Sharing teaching gems (voluntary)	12.15 - 12.50	10.15 - 10.50

Time (CEST)	Title	EEST (UTC+3)	BST (UTC+1)
11.50 - 12.00	Outro and feedback	12.50 - 13.00	10.50 - 11.00

:icecream: Icebreaker

Check-in

Which emoji best describes your current state of mind? [Emoji cheat sheet](#)

- ...
- :sleeping:
- :coffee:+2
- :robot:
- Still little sleepy :)
- 🤪 (:woozy_face: still not converted 🧑)
- Our planet could be a cube. It has six sides to it. It must be a cube. A cube it is, this Blue Planet ... just wondering ...
- 🦵
- :sweat_smile:
- computer says no :-) some network trouble
- running late, and participating from a :train2:

Introduction in breakoutrooms

Name / Affiliation / Location

Best classroom experiences

As a learner or teacher; what was the best workshop/lesson/seminar you have attended?
What made the experience great?

- Room 1
 - First summer HPC streamed workshop (as a teacher). Somehow the first felt the best
 - CSC HPC summer workshop
 - story telling teaching approach
- Room 2
 - NVIDIA DLI Course on Deep Learning
 - very enthusiastic instructor
 - intuitive and very accessible way of showing technical details and fostering deeper understanding
 - a lot of hands-on material and smaller coding exercises
 - Anything as long as its interactive and somehow enjoyable from both teacher and learner sides
- Room 4:

- EPCC MPI Distributed training workshop.
 - Lots of hand on coding / experience gained.
 - Very interactive instructor who also discussed ideas for more personalised cases of distributed computing.
- Coderefinery RSE course
 - relevance of topic to my current work.
 - Dynamic / friendly / positive teaching style.
 - Modular Code Development; very instructive
- Some CSC courses / LinkedIn learning courses
 - up to date materials and topics.
 - Good way to merge theory with practicalities without making learner feel bored (Example: Jonathan Reichental)
 - Challenging information to learn more and research about the topic
 - Trainers are the experts in the field

Episode on teaching gems

If you want to share or demo something about tools and techniques you have found helpful in your teaching, please add your name/initials and rough topic below for smooth transition.

Note: It does not have to be your tool or your idea, this episode is meant to share experiences and discover new things. You also do not need any slides or materials, a story will do fine :) Length of the stories will be depending on interest, but be prepared to fit it into 3 min (shortest possible).

- RD: teaching clock
- RB: [Containerized teaching setup](#)
 - Isolated home dir is a good idea!
- AVM: [Screenkey for showing keyboard shortcuts](#)

:question: Questions

General / Practicalities

- What is this document for?
 - Asking questions, collecting discussion points, taking collaborative notes (Zoom chat deactivated, can only be used for communication with the hosts)
- Will the material be made available?
 - Materials are and will stay available at <https://coderefinery.github.io/train-the-trainer/>
 - Can I share them with a colleague
 - Yes please :)
 - Can I resuse them in my own teaching?
 - Yes please :)
- Will these sessions be recorded?

- No, but the materials (see above) should include everything needed for self-learning.
 - Can I get a certificate for this workshop?
 - You can get a certificate of attendance after the workshop by sending an e-mail to support@coderefinery.org, telling us a bit about what you have learned and which sessions you attended.
 - How often will we be talking in breakout rooms?
 - We'll have about one breakoutroom session per episode
-

Episode 1 : Computational thinking

Materials: <https://coderefinery.github.io/train-the-trainer/computational-thinking/> (Link to slides in the middle of material)

Questions to audience

How might breaking down a complex problem into smaller parts change your approach to problem-solving in your current projects?

- Avoids cognitive overload
- I used to give an example about this in workshops of cutting long tree into pieces from top to avoid that it falls and destroy a house bloc.
- Easier to start on something small
- Each individual part is familiar and can take existing solutions.
- Only focus on the new things
- One can lose oversight of where is supposed to be getting to due to problems with summation of biases introduced during the solving of the parts.
- .

Can you think of a research project where identifying patterns in your data led to new insights or breakthroughs?

- Multi-targeted drug design where you need to identify both chemical and biological pathways/patterns
- Distilling complex physics problems into simpler 1D statistics is very often done.
- I used to do bioinformatics, pretty much everything in biology is about patterns in strings of characters (DNA, proteins). Finding those patterns and using them is the way to go
- Pattern recognition DOES NOT lead to new insight, it is new insight which allows for the recognition of a 'pattern', the latter being an allocation of possible bias.
-
- ..

What challenges do you face when trying to simplify complex concepts in your field, and how do you decide which details to focus on?

- Absence of a terminology / jargon to describe a new idea. :+1:
- Similar to above, when teaching often students know what they want to achieve, but don't have the terminology to express it, so working through what they want is helpful, after teaching them some terminology.
- Whatever is most useful to the learner first?
- I work with researchers in different fields. I don't always fully understand their domain knowledge but I can take the basics or generalities and get code that works for them
- A very practical problem is trying to resist going for a coffee in the midst of taking a shot at a complex concept, in the hope that the true details are to be found in the coffee.
- not all information is easy to find or a lot of conflicting information or even sometimes information explosion

How do you determine the priority of tasks when designing algorithms for your academic projects, and what criteria do you use to ensure that the most critical tasks are addressed first?

- Working chronologically when going through a problem - start at the beginning.
- What gives the more insight with the least effort can be a nice start
- Talk to colleagues and Subject matter experts to understand various perspective to decide and prioritize.
- Logic will get you from A to B. Imagination will take you everywhere. - Albert Einstein.
- ..

Questions from the audience

- About the 4 core concepts, is this based on the model of how human brain works and perceives vision / abstract ideas?
- I don't know if it's based on how the brain works, but it's related to creating a framework for solving programming programs that has been extrapolated to solve any other problem as well.
- Devil's advocate question: cooking (by recipe) actually employs all four principles outlined here and has done so since before the advent of computers. Why then is this concept presented as "computational"?
 - Random aside, this is an interesting connection to our "cooking metaphor to HPC"
- Devil's answer to the Devil's advocate: Great question, cooking, as far as the act happens in the Devil's kitchen, does not employ numerical algorithms. The Devil introduced the computer precisely to make man lose appetite for that what is cooked in the Devil's kitchen to be served to all. Only then can the Devil eat the cake and keep it too ...
 - Well, an algorithm without numbers is still an algorithm, i.e. a set of rules :)
- Are there other names for this technique? I may have missed something crucial, but it strikes me as 'logical thinking' in a way.
 - Yes, you could also just see it as problem solving. Most people know how to do it and if you've worked with programming, you've probably applied it. It can be the case that it's very obvious to some, but not to others.
- Any chance we can get a link to the slides?

- They are in the course material: <https://coderefinery.github.io/train-the-trainer/computational-thinking/> :thumbs_up:

Exercise questions to discuss

1. Conceptually would the strategies provided by computational thinking help you in completing tasks more efficiently?

- Room 1:
 - divide and conquer technique
 - We discussed what happens if you *don't* use these, since they are so natural
- Room 2:
 - in some cases yes, if the task is not too small
 - not sure if more efficiently but it does help at least getting started
 - Computational thinking is an extension of what others have called solutionism: the belief that any given problem can be solved by the application of computation.
Whatever the practical or social problem we face, there is an app for it. - James Bridle
- Room 3:
 - Helps in starting the task.
 - For many it is the normal way to do problem solving. Also in Agile & Scrum methodologies.
- Room 4:
 - Yes, it helped me during my scientific research to manage multiple projects and research publications simultaneously.
 -

2. Do you think that the effectiveness of computational thinking depends on the person's personality type?

- Room 1:
 - yes, we agree it's quite different. But good vision and motivation this can be effective
- Room 2:
 - yes, laying out all the steps necessary to achieve an overall goal can be very daunting; some people are not good at/ comfortable with following fine-grained instructions
 - once the problem is broken down, different personalities might choose different ways to tackle the parts
 - The degree to which something, that is a result of what one means by 'Computational Thinking,' is successful in producing a desired result, that is success, cannot be a function of personality.
- Room 3:
 - Yes. Different people have different ways to communicate.
 - Online tools vs pen and paper
- Room 4:
 - We are all agree that it depends on the personality type but it could be learned and trained

3. Would this type of thinking be an option for you to integrate into your current workflow?

- Room 1:
 - Yes; we didn't know the name explicitly; but we use it
 - It's quite important for
- Room 2:
 - I'm already doing it implicitly :smile: :+1: :+1:
 - Writing a program is like writing a song, its all about the tune, lyrics are merely the necessary nuisance a coder needs to put up with. The latter can, if so written, result in a lot of noise.
- Room 3:
 - Easy if you are working alone. But different team may have their own ways of working (inertia). One way would be then to gradually show the rest of the team its effectiveness.
 - <https://teampologies.com/key-concepts>
- Room 4:
 - We are using it in our daily life so it's already an option for us, only the terminologies are a bit new.

Episode 2 : Teaching philosophies

Materials: <https://coderefinery.github.io/train-the-trainer/teaching-philosophies/>

Questions to discuss:

- Share your approach to teaching and your teaching philosophy with your group.
- Please share your tricks and solutions in the live document for others.

Additional ice-breaker questions:

- What is your motivation for taking this training?
- How structured or informal are your own teaching needs?
- What difference do you notice between the teaching what we (also Carpentries) do and traditional academic teaching?
- What other skills need to be taught, but academic teaching isn't the right setting?

Breakout Room exercise

- Room1
 - Motivation
 - For best practices in teaching, learning from each other, collaboration
 - Learning applied tech is like an apprenticeship: there is theory, but real learning happens by co-working with others.
 - Worse is better: teach what can be used more quickly and it can be improved later
 -

- I think there are two bits of teaching - teach the basics and terminology so people know what they want to know, and then teaching how to find out how to do things like Richard said - how to find answers yourself.
- Using Jupyter notebooks - good for learning a technique, but not for learning the basics.
- Possibly command line usage has a branding problem - “programming other programs”?
- the skills we learn in CR training are expected to stay , it is not exam oriented
- Room 2
 - State objectives clearly and repeat them a few times
 - Try to keep it informal, interactive, flexible
 - Helps when having practical tasks in mind for the materials taught
 - Leave some reasoning to the learner, not always giving full answers
 - How to deal with frustration though: try some personal approaches, more clues...
 - The mediocre teacher tells. The good teacher explains. The superior teacher demonstrates. The great teacher inspires. — William Arthur Ward
 - Teaching kids to count is fine, but teaching them what counts is best. –Bob Talbert
- Room 4
 - Question 1: What is your motivation for taking this training?
 - Currently do instruction, wanting to improve as an instructor and helper when conducting software carpentries.
 - Planning to start some side hustles providing online courses and why not joining the CR in the near future
 - Question 2: How structured or informal are your own teaching needs?
 - More recently, follow software carpentries, but look into what aspects can be tailored to keep to different time slots.
 - I used to be more informal in the past, repetitive practice sessions. Would like to be more structured given the teaching tools.
 - I've done life-coaching and some teaching but I feel i'm both of them
 - Question 3: What difference do you notice between the teaching what we (also Carpentries) do and traditional academic teaching?
 - Smaller size, and more targetted.
 - Collaborative tools and targeted audience working with small groups
 - Academic probably focuses on an audience with a more focused/narrow background, while this style encompasses a wider audience.
 - Academic teaching keeps theory and practical lab sessions very separate, whereas carpentries combine both.
 - Question 4: What other skills need to be taught, but academic teaching isn't the right setting?
 - Public speaking, collaborative meeting skills
 - Industrial cases and related work skills

Episode 3: Co-teaching

Materials: <https://coderefinery.github.io/train-the-trainer/co-teaching/>

- In general: Co-teaching is a beast with at least two personalities each of which
 - demand higher preparation requirements
 - face remarkable complexity in terms of coteacher coordination
 - in the virtual component face shortcomings related to the technical equipment and the failure of the human factor
- Co-teaching requires the framework of co-operative learning. Team teaching does not by default translate into parallel learning, and vice versa.
- Re: CodeRefinery in-person beginnings – I agree with the “didn’t work so well in-person”, I’m not sure if it’s that we weren’t as ready for it, or that the different characteristics made it not work so much
- How do the downsides compare to a single instructor monologuing?
 - That’s a great question! The main difference is that a single instructor can teach himself to be good (i.e. not just boring monologuing) **alone**, while for teams of 2+ to work well, **everyone** needs to be baseline-competent. So the effort to avoid downsides gets reduced if (and only if) all team members share it.

Exercise/Discussion: <https://coderefinery.github.io/train-the-trainer/co-teaching/#exercise>

Questions:

- Have you already tried this or similar model in your teaching?
- Does it seem natural to apply this model in your subject area (tell what it is)? How could it be adapted to fit best?
- Have you tried or seen a different model for two instructors to present? Please share with us how it works.
- room 1
 - We have tendend to do presenter/interviewer more
 - challenges are co-teaching with different personalities
- room 2
 - Main challenge: finding enough personell :smile:
 - It’s not always possible/feasible to have colleagues co-teaching
 - A combination of both, but presenter and interviewer could be difficult to plan an to follow
 - Guide and demo-giver with technical / non-technical roles could be nice
 - How to plan the guide and demo-giver
 - First guide, then demo
 - Demo, then guide describes
 - Alternative model: main presenter + ‘technical expert’
 - main presenter introduces concepts from a birdseye view and passes on to the technical expert for additional details, experiences and answers to questions from the audience
 - We frequently do a classical division: on person in presenting, one or more are

answering questions in the chat; there is rarely any interaction between them however

- As has been pointed out by Paolo Freire, in general, 'education is suffering from narration sickness' ... 'The outstanding characteristic of this narrative education, then, is the sonority of words, not their transforming power'. Perhaps one approach could be to rephrase the question after having the answer: It is not apriori about the model implemented, it is about whether that what was implemented was judged aposteriori to have succeeded in becoming, or at least could pretend to have done so, the intended context matched model implementation it probably was computed to be.
- room 4
 - the co-teaching approach does not seem to be widely practiced
 - it can be interesting to see "mistakes" which are less likely to happen in solo-teaching?
 - co-teaching can be nicer for audience but maybe more stress/anxiety for instructors to prepare? more stress in terms of not knowing all the questions in advance. less stress worrying about forgetting something.

Teaching gems

(scroll up to find the green box "Episode on teaching gems")

What's your favorite way of teaching? Any nice tools you use?

Has anyone found a nice online tool to draw?

- I love this tool suite <https://excalideck.com/>
 - For drawing <https://excalidraw.com/>
- I have seen Miro ...
- I think the Code Refineries use something for their graphics that looks a bit 'cartoony' - but I cannot remember what it is called!
 - Drawn on remarkable and then coloured and tidied up in Inkscape. :+1:
- I am a big fan of remarkable. I think sharing option is fairly good. Unfortunately bit expensive tool.
 - true, for me it was worth it already for the pdf annotation for research papers, cannot read on computer screen
- <https://webwhiteboard.com/>
- <https://www.youtube.com/watch?v=4-l8MY5kYGc>
- Figma and Kahoot are useful as well

Screenkey: [Screenkey for showing keyboard shortcuts](#)

- Is there a Windows version people can recommend?
 - I found this list of alternative for windows users like me: <https://alternativeto.net/software/screenkey/> :+1:
- I sometimes use the on-screen keyboard already provided by the OS :+1:

Wrapup

- Next session Tuesday September 3rd 9 CEST: About teaching & cool things we all would like to share
- All workshop materials will stay available (and be put on Zenodo right after the workshop, currently other episodes are still in development)
- CodeRefinery community: <https://coderefinery.zulipchat.com/>
- If you want to co-teach with us in the upcoming (or any future) CodeRefinery workshop; or just want to learn more about this opportunity: Let us know in chat, send an e-mail to support@coderefinery.org, or join any of the upcoming Monday planning meetings at 14 CEST
- Preparations for next week will be sent out by e-mail

Feedback about todays session

What one thing was the most valuable/useful thing you learned today? Or generally one thing you liked about this session:

- The co-teaching lesson and experiences from others
- Discussions were quite good, had a nice group
- Breakout rooms and collaborative tools
- Computational thinking and co-teaching are interesting ideas! I have seen it before, but never knew or even recognized it was a method.
- Thanks for group discussions working with my chat only interaction! :smile:
- ..

What one thing would you improve about this session?

- Minor point - it would be useful to have the teaching games box at the bottom of the document rather than the top - tricky to scroll between! In the end I opened two windows which was easier.
 - Thanks, yes agree. Also had that problem, but did not want to move it in between.
- The meaning of teaching “philosophy” wasn’t very clear, felt more like teaching “styles”
 - Thank you, we will take that into consideration
- Some prepared slides to present for an intro about the topics
 - You mean in addition to the materials? To use in the beginning of each session?
- The *instructor views* segment could be named as additional material, if it was not meant to be discussed within the course.
 - Thank you, we will take that into consideration

Any other comments?

- Keeping it short and sweet: If at first you don’t succeed in teaching, try to explain it like you’re talking to a room full of cats. Caveat: In a room full of computational thinkers

maybe if you tell students that the brain is a computational device that is forbidden, they might get excited and start using it.

Day4: Session 4 (3.09.24) - Streaming and video editing

:calendar: Schedule

Time (CEST)	Title	EEST (UTC+3)	BST (UTC+1)
8.45 - 9.00	Connecting time	9.45 - 10.00	7.45 - 8.00
9.00 - 9.15	Intro and Icebreaker	10.00 - 10.15	8.00 - 8.15
9.15 - 10.00	Why we stream & Behind the scenes	10.15 - 11.00	8.15 - 9.00
10.00 - 10.15	Break	11.00 - 11.15	9.00 - 9.15
10.15 - 10.35	Video Editing	11.15 - 11.35	9.15 - 9.35
10.35 - 10.55	Exercise: Video Editing	11.35 - 11.55	9.35 - 9.55
10.55 - 11.10	Break	11.55 - 12.10	9.55 - 10.10
11.10 - 11.30	Open Broadcaster Software(OBS) introduction	12.10 - 12.30	10.10 - 10.30
11.30 - 11:50	OBS setup & what next	12.30 - 12.50	10.30 - 10.50
11.50 - 12.00	Outro and feedback	12.50 - 13.00	10.50 - 11.00

:icecream: Icebreaker

Check-in

Which emoji best describes your current state of mind? [Emoji cheat sheet](#)

- :coffee: +
- :runner:
- :tired_face: Tired!
- :nerd_face:



- :cloud: :tea:

Introduction in breakoutrooms

- Name / Affiliation / Location

What's the most number of people you have taught to?

- 20 at a time, with 1 asleep, 2 fidgeting with some silly device and the rest were... well... staring at their good luck... straight ahead it was of course, right before them.
- ~35
- ~80 in a lecture in a previous (lecturer) role, 26 in a training course setting
- ~30
- ~50 in training spaces and more than 100 for public audience talk
- 35 (software carpentry)
- CodeRefinery answer: The biggest ones are maybe 200-300 people. "small" for a stream is ~100 people.

What's the most number of people you have taught with?

- 5 or so
- 2 or 3 in total
- 3, as in three, none of them smiling. I wish I could figure out why... it was such fun really...
- 7
- 2 or 3
- 3 (helpers, not really co-teaching)
- CodeRefinery answer: our biggest workshops are something like 10-20 people helping out in many different roles.

How can you divide teaching into separate independent tasks?

- By content blocks, having roles like main and assistant...
- Having responsibility for different sections of the course.
- Wow, now that's what we call a question... bravo... indeed, how does one separate sleeping and snoring into separate independent tasks...
- Course, exercises, resources, tools and forms.
- Different people teach different topics
- CodeRefinery answer: the big logical blocks are instructors, in-person and breakout room helpers, and Notes-questions answers.

What's the most interesting or useful thing you've learned from an online workshop, and how have you applied it/planning to apply in your life or work

- Having breaks every hour or so
- New approach to screensharing, using the 'portrait' approach
- Manage breathing: reduce stress and use silence to let the audience grasp what you're saying
- Well, the most interesting or useful thing I've learned from online workshops is how not to conduct a workshop, or at least some elements of the same, and I have applied this

understanding by engaging in the global effort to discover more about how not to conduct a workshop, or at least some elements of the same, with the motto: That's one small step for a man, one giant leap for mankind..

:question: Questions

Why we stream

- If during streaming there is no interaction between the teacher and the audience, why don't we just record the lectures and stream them? So one can do a better job, perhaps?
 - Streaming it making it a "thing" that's a group experience. It feels good to be a part of it, so we hope that encourages people to actually attend. All the mass Q&A is fun, and the audience helps to make it easier to present. Still, this is a good question.
 - I really appreciate it being "a thing". If it was just recorded, I would say I would watch it tomorrow, and then I never would.
 - But then you can just fake it, and still have all the breakout rooms and collaborative document and so on. I appreciate the value of having a proper "event".
 - I think I follow you, but if you pre-recorded the talking bits, then the presentors couldn't change what they do based on the feedback in the shared document.
 - true, but I assumed that the teachers were too busy to actually do that. The more (less)division of labor between the teachers and the helpers (or collaborative document editors), the more (less) sense it makes to pre-record - IMHO
- What interaction options do 'streamers' use, e.g. on Twitch/YouTube (not Code Refinery)?
 - I'm besides chat and things that happen in chat, I'm not sure. The Notes-doc is definitely unique to us.
- Streaming is a kind of exercise for the movie star type. If you cannot be one, act like one, pretend, fool the listener into thinking you are one, take their brains and emotions under control. And if they don't leave the session with tears of love for learning running down their chubby cheeks you know who to kick for doing a job just that bit worse than it should have been necessary.
- Have you kept track of streaming stats? like how many people logged in and stayed throughout, how many interacted?
 - Yes please, it would be good to see the stats
 - Here is stats [repo](#)
 - Thanks, I get a 404 error, is it private?
 - No, I don't think so. <https://github.com/coderefinery/workshop-stats>
 - Still get 404. Does it work for anyone else?
 - Yeah I think it's private. We need to fix this...
- How big is small (10-15), medium (~30-40)?
 - I would say below 50 is small.
- Have you considered scaling to more events instead of to a larger audience? I mean, this seems the opposite of scaling, but this kind of content would be more useful the more often is it taught ("I need to learn GIT/Python/* in December!")
 - We would, but right now we simply don't have enough time and people to do more

small events. But yes, “scheduling conflict” is a big problem - the best we can do right now is written material+videos to follow up on (and hope it’s interesting enough)

- [Python for Scientific Computing 5-7/November/2024](#)

- This was just an example. I mean, if I am a PHD student that needs to start working on a topic on Month X which happens to be inconveniently placed in the year, I will not benefit as much from having a workshop at Month X+3
 - The materials are open and the videos are publically available on YouTube channel
 - What’s the ratio of people watching videos afterwards vs people attending the event? That ratio can give an idea of the popularity of the two possible approaches.

- The more you scale up the less interactive the course / workshop will need to be. It then becomes a question of priorities, and if teaching is a lower priority then learning than I guess there really should be no prizes for guessing which approach should reign the airwaves.

- I guess outside Europe / Africa, time zones could be a problem?

- yes, that is true

-

- Sorry, I might have missed this: have you been rehearsing and/or doing dry runs?

- With instructors I like to try to do a dry run. I often don’t do a live broadcast dry run these days since I’m confident enough, but when you are just starting, it’s a good idea.

- Do you always use Zoom for the ‘presenter end’ or have you tried other things? Teams? Jitsi? etc.?

- we use mostly zoom

- It was first pioneered using Jitsi. You can probably use others, too.

- A comment: for deRSE24 we have been using the [GWDG streaming service](#) and OBS, it worked quite well (with a 20s delay or so)

- Cool! Yes, there is plenty of livestreaming options. Twitch+our OBS config gives us 2-3s of latency (which is low enough for live Notes interaction). You might be able to tune this if needed.

Behind the stream

- Does OBS take a lot of RAM or CPU? what are the specs on your machine?

- Richard have a relatively powerful computer with 8 AMD CPUs.

- 4 CPUs are relatively enough for a good OBS setup, most important is a stable internet connection (ideally wired).

- An extra monitor is recommended for the setup

- See the hardware notes recommendation [here](#)

- Do you do a screenshare from Zoom -> OBS or OBS -> Zoom?

- Instructors share to Zoom, OBS captures Zoom, OBS sends to the world.

- When I am using OBS sfor a single-person presentation recording, I go the other way: OBS captures my desktop/camera, OBS generates the preview window, and then Zoom captures and broadcasts the preview window.

Video editing

Poll

- Have you tried video editing? Add an as answer below
 - yes: ooooo
 - no :
 - Video editing makes all the difference. I have never known a way to avoid this sacred of steps, so to speak. In short, no recording can ever be production ready by default in the books of the wiser among the streamers. In fact, during the editing one can considerably refine and improve the learning experience in great ways. But then one would need to invest both time and effort.
- If Which are the tools you use?
 - Windows Movie Maker / ClipChamp
 - (raw stream from Zoom)
 - I think it was shotcut - I did only basic trimming
 - Kdenlive
 - Clipchamp, OpenShot, Sony Vegas
 - also ffmpeg from command line to cut beginning and end

Questions

- What is Whisper?
 - Open-source model from OpenAI that converts speech to text transcripts.
 - this one? <https://openai.com/index/whisper/>
 - curious if anyone has tried it with languages other than english
 - I heard that one of my spanish colleague try it with another software. I don't remember the tool. But with AI bloom there are many licensed tools that can do it with different language
- What do you use to crop the videos?
 - I see now, ffmpeg
- how to merge multiple parts in one? I asked because you edited only the icebreaker part?
 - you can manage it in input part, see ffmpeg-editlist readme.
- Do you generate custom thumbnails for the video?
 - no, because of the time, but its good to have. If we had a volunteer to manage that, then we should!
- Does anyone recommend any other (GUI) tools that work well? ffmpeg is very new to me!
 - if you want a basic trimming and editing , you can use Quicktime player/imovies on mac
 - Clipchamp as the Microsoft video editor
 - Thanks :smile:
- How do subtitles get uploaded to YouTube?
 - You upload the video and subtitles as part of upload.
 - :+1:
- How does codewhisperer compare against youtube's automatically generated subtitles?

- by experience, whisper was better couple of years ago. We haven't compared it recently
- I guess an advantage is that you can edit whisper subtitles in case something is odd or whatever, probs cannot do that in YouTube

Open Broadcaster Software (OBS) introduction & setup

- Are the profiles public?
 - Yes, you can see it [here](#)
- Does it work in Linux+Wayland?
 - I think everything we do with OBS would.

Wrapup

- All workshop materials will stay available (and be put on Zenodo right after the workshop)
- Upcoming Workshops
 - [CodeRefinery workshop September 10-12, 17-19, 2024](#)
 - [Build Systems Course and Hackathon](#)
 - [Python for Scientific Computing](#)
- CodeRefinery community: <https://coderefinery.zulipchat.com/>
- If you want to co-teach with us in the upcoming (or any future) CodeRefinery workshop; or just want to learn more about this opportunity: Let us know in chat, send an e-mail to support@coderefinery.org
- A summary email will be sent out to all participants

Feedback about today's session

What one thing was the most valuable/useful thing you learned today? Or generally one thing you liked about this session:

- ffmpeg seems great, will give a try for fun!
- great to see everything in practice!
- Very insightful to see what happens behind the scenes.

What one thing would you improve about this session?

- Some more interactivity would've been nice but since most the group preferred demo instead of exercise perhaps it's only a personal opinion
- It would have been good to share the OBS profiles repo beforehand in the setup email - so I could have had it to hand. (<https://github.com/coderefinery/obs-config>)
 - +1 good idea, we should do this next time
- Perhaps a simpler starter OBS setup?
 - Good idea... I should have, but basically ran out of time to prepare.

Any other comments?

- Cool to see the switch from RSE to trainer to AV guy, impressive!