Welcome to The Carpentries Etherpad!

This pad is synchronized as you type, so that everyone viewing this page sees the same text. This allows you to collaborate seamlessly on documents.

Use of this service is restricted to members of The Carpentries community; this is not for general purpose use (for that, try https://etherpad.wikimedia.org).

Users are expected to follow our code of conduct: https://docs.carpentries.org/topic_folders/policies/code-of-conduct.html

---

The Carpentries Instructor Training

03-06 May 2021

Trainers:

- Anne Fouilloux (she/her) / University of Oslo / annefou@geo.uio.no / Twitter: @AnneFouilloux Github: @annefou
- Toby Hodges (he/him/his) / The Carpentries / tobyhodges@carpentries.org / Twitter: @tbyhdgs GitHub: @tobyhodges

Sign in: Name (Pronouns, optional), Institution, Email & Twitter (optional)
Please sign in so we can record your attendance.
Put yes/no at the end of the line to indicate if you are happy for Al/The Carpentries to contact you to follow up about your experience with the Carpentries. (no response = "no")

- David Chamont (he/him/his), CNRS / IN2P3 / IJCLab, chamont@in2p3.fr
- Nathan (Nate) Quarderer (he/him/they) / CIRES; Earth Lab; CU Boulder / nathan.quarderer@colorado.edu / @nquarderer/ Yes please feel free to reach out to me
- Emmy Tsang (she/her) / TU Delft / emmyftsang@gmail.com / Twitter @emmy_ft github @emmyft - yes
- Lance Simpson(he/his)/The University of Alabama, lsimpson@ua.edu/lancemsimpson@gmail.com/yes, please connect if you would like
- Luca Ferranti (he/his), University of Vaasa, luca.ferranti@uwasa.fi yes
- Nonkululeko Nhlapho(she/her),University of the Witwatersrand/nonkululeko.nhlapho2805@gmail.com,yes
- Nate Jones | University of Alabama (Roll Tide!) | cnjones7@ua.edu | @FloodHydrology | Happy to connect
- Karin Knudson (she/her), Tufts University, karin.knudson@tufts.edu, @karinknudson, yes
- Sam Bail (she/her), independent, samanthapbail@gmail.com, @spbail
- Danielle Sieh (she/her), The Carpentries, danielle@carpentries.org / Yes
- Talisha Sutton-Kennedy (She/Her), The Carpentries, talishask@carpentries.org | Yes, open to connect

- Yara Abu Awad (She/Her)/ Concordia University / yara.abuawad@concordia.ca | Yes
- Robert McKenzie (he/his) / Caribbean Development Bank / mckenzr@caribank.org / Yes
- Jay DesLauriers (he/his) / University of Westminster / j.deslauriers@westminster.ac.uk
- Al Obayuwana (he/his)/ The Carpentries/ al.obayuwana@carpentries.org
- Andrew Brown (he/him/his) Queen's University Belfast / andrew.brown@qub.ac.uk / Yes
- Jess Cohen-Tanugi (she/her/hers) / Harvard University / jessica_cohen-tanugi@harvard.edu / YES!
- Addi Malviya-Thakur (she/her/hers) / Oak Ridge National Laboratory / https://www.ornl.gov/staff-profile/aditi-malviya / malviyaa@ornl.gov - work and research related / aditimalviyathakur@gmail.com - general/ Yes
- Samarpita Roy (She/her)/Singapore Centre for Environmental Life Sciences/ samarpita.roy@u.nus.edu/Twitter: bacteRoy/ Yes
- Tayaza Fadason (he/him/his) / Liggins Institute, The University of Auckland, New Zealand / t.fadason@auckland.ac.nz / Yes

Please fill out the pre-training survey at
https://www.surveymonkey.com/r/instructor_training_pre_survey?workshop_id=instructor-training

You can keep track of the time in your current timezone at https://timeanddate.com/worldclock.

---

0. Resources
Please add links to any relevant resources here:

Notes by Day:
Day 1: approx. line 70
Day 2: approx. line 490
Day 3: approx. line 1050
Day 4: approx. line 1580

--Day 1--
I. Welcome

Code of Conduct:
https://docs.carpentries.org/topic_folders/policies/code-of-conduct.html

Using Zoom
1. Keep your microphone muted when you are not talking
2. Please keep your video on if you are comfortable doing so, and if your internet connection allows it. If you find that you are having connection difficulties, it can help to turn off your video feed.
3. When you would like to say something, ask a question, you can:

- click the "Raise Hand" button under "Reactions"
- type "hand" or "/hand" into the Zoom chat
- or type your question directly into the Zoom chat
- (Please do not raise your hand in the video feed, because the Trainer cannot always see everyone's picture and we don't want to miss you.)

Introductions
1. who you are and where you are, one sentence about your work, and the last time you taught something to someone

Our First Exercise (2 min)
In the Etherpad, write down your name, the best class you ever took (or one class from your top ten, if you can't decide), and what made it so great.
Luca Ferranti -- computational physics during my 3rd bachelor year, because that's how I got started with scientific computing and open source.
David Chamont -- any online lesson which I can follow at my rate, zapping what I do not care for, and concentrating on what I care for, and with an environment where I can immediatly practice, with no need for any kind of installation :)
Karin Knudson -- high school earth science class (in my top 10!) because it literally changed the way I saw the world around me -- e.g. I could see and ask questions about different and new things just looking at a landscape. Also just a really welcoming, fun class community.

Robert McKenzie | Digital Ecosystem Infrastructure | Practical, lessons learnt could be applied more than any other. university course.

Nate Quarderer - Critical Discourse Analysis - way outside my comfort zone at the time & led to publication

Nate Jones -- Wetland Soils -- Active learning [i.e., we got to dig in wetlands every class period!]
Jay DesLauriers -- Internet Security, because the Lecturer was so incredibly passionate about the material
Talisha SK - Undergrad Spectrometry course - it was really fun and challenging.
Addi Malviya-Thakur: The best class was Aesthetic computing in grad school. It was very hands on and visual, with the TA(s) as well as Professor involved deeply with the class for every hour during the session.

Lance Simpson, EN 102 (from undergrad), the course instructor had us consider monsters and introduced us to the act of "monstering" or creating monsters-- my instructor was excited and challenged us to consider us to more intentionally examine our own place and views in the world.
Emmy Tsang -- neural networks lectures from Bachelor degree, mindblowing because it made me questioned everything I thought I knew previously about the topic
Yara - EPI 201 - Changed the way I think about Causal Inference - really well taught and organized class

Danielle, Child's Psychology in college, I had an amazing teacher and I loved what I was learning
Jess Cohen-Tanugi, The Apocalypse in American Culture and Society AND Post War French Social Theory (I think?); opened my mind to some theoretical concepts on how we think, fueled a passion for linguistics and study of rhetoric, and changed the way I looked at communication.

Sam - learned a lot while working with oncologists at my former company, eg during lunch&learn sessions. Learning directly from experts with decades of experience in the field and having them talk about their own experience working with patients made the complex topic "make sense".

Al Obayuwana top 10 class was 6th grade math class, best teacher most engaing, I love math because of Mr. Manion

Nonkululeko NHlapho-The best class I took was a python class,It was my 1st day and I did not know anything but my teacher was gently and took me through the whole thing ,I'd say that basically when I fell in Love with python and it still interest me to this day and I also want to know more other programming languages.

Andrew Brown: A-level music (A-level is end of high-school qualification in the UK). Teacher was so passionate and brought the subject to life, and would follow little rabbit trails of enquiry when they came up, which encouraged us to get curious.

Samarpita Roy: Strategies and Techniques to analyze Microbial Populations (STAMPS) at MBL. The teachers were enthusiastic, well read and broke down complex topics in a manner that allowed it to stick with me for more than 2 years now!

Tayaza Fadason: High School Physics. Teacher knows his stuff, is passionate about the subject, and can describe complex concepts using everyday objects and language.

A Brief Overview of the Carpentries
https://carpentries.org/workshops/

- A program that combines three communities (Software Carpentry, Data Carpentry, and Library Carentry), together we form commutities of practice who teach data and computational skills. The main goal is not to teach specific skills but to convey best practices that will enable them to become more productive and do research better.

Instructor Training Workshop Overview

- How learning works - what the research says about how people learn and what we can do with that to make our teaching better
- Building teaching skill
- Creating a positive learning environment
- Carpentry history and culture

Background (3 min)
Have you ever participated in a Software, Data or Library Carpentry Workshop?
answer with x

- Yes, I have taken a workshop. xxxx
- Yes, I have been a workshop helper. xxx
- Yes, I organized a workshop.
- No, but I am familiar with what is taught at a workshop. xxxxxxxxxxx
- No, and I am not familiar with what is taught at a workshop.xxxx

Which of these most accurately describes your teaching experience?
answer with x

- I have been a graduate or undergraduate teaching assistant for a university/college course. xxxxxxx
- I have not had any teaching experience in the past.

- I have taught a seminar, workshop, or other short or informal course.xxxxxxxxxx
- I have been the instructor-of-record for my own university/college course / continuing education. x xxxxxx
- I have taught at the primary education level.xxxxx
- I have taught informally through outreach programs, hackathons, laboratory demonstrations, and similar activities.xxxxxxx

Key Points:

- The Carpentries are communities of practice. We strive to provide a welcoming environment for all learners and take our Code of Conduct seriously.
- This episode sets the stage for the entire workshop. The introductions and exercises help everyone begin to develop a relationship and trust.
- This workshop will cover general teaching pedagogy and how it applies specifically to the Carpentries.
- Learner motivation and prior knowledge vary widely, but can be assessed with a quick multiple choice question.

II. Building Skill with Practice
https://carpentries.github.io/instructor-training/02-practice-learning/index.html

The Carpentries Pedagogical Model

- Applied knowledge rather than theories
- Develop learners' problem solving skills, directing them to places to find answers rather than comprehensive knowledge of everything, also because workshops are only 2 days
- Lots of exercises, interactive discussions with learners rather than presentations with slides: this is based on research in learning psychology and pedagogy

Acquisition of Skill
https://carpentries.github.io/instructor-training/fig/skill-level.svg

- Novice
- Competent practitioner
- Expert

nice resource for mental models : https://fs.blog/tgmm/ The books are excellent

Question: People at different levels require different instruction. Is that something that the curriculum covers or that instructors need to bring to the table? Lesson materials are aimed at novice learners. Learning new things about a topic are all about buildling on knowledge that you already have. So we introduce lots of opportunities for checking understanding and correcting any misunderstanding as early as possible.

Break until 13:20

Cognitive Development and Mental Models (5 min)
In the Etherpad, write your area of expertise or primary research domain and some aspects of the mental model you use to frame and understand your work. What concepts/facts are included? What types of

relationships are included?

- David Ch C++ : choices must be made between several programming paradigms (imperative, object oriented, generic, functional...), between things done at compile-time versus run-time, hand-coded or based on some internal or external libraries, the solution will be typically designed as a set of classes with public interfaces and private implementations. Relationships are static, between those classes, dynamic whe one write a use-cases. One can also express some dataflow if data is the dominant aspect... well this question is too complex !
- Nate Qu: Science education research - we use these mental models extensiveley (though we call them concept maps in my group); having students create these models at the start of a unit, be it about Newton's laws of motion or relationships within an ecosystem, and having them revisit those models throughout the unit; it is informative to watch how these models grow, and how nodes and the connections within and among those nodes become strengthened over time.
- Emmy open community design - concepts in this area are taken from parallel ones in user experience design (user personas), service design (service maps), communications (messaging and segmentation), open source software development (asynchronous collaboration), social innovation (impact valuation & sustainability)... depending on project/community maturity, this requires different sequences of these research and techniques being applied (not sure if I understood "relationships" well), and depends on the stakeholders too
- Lance-Areas of experitise: information literacy--concepts include understanding your own research process, how information is organized in your academic discipline--I work with students in engineering to consider what their deliverables for projects are, to use computational thinking principles to create search terms and develop search strings, to better understand how controlled vocabulary works (connecting back to how information is organized), to understand how to evaluate information and sources.
- Luca computational methods for positioning, the "concepts" are 1) mathematics (numerical analysis), 2) coding skills and 3) domain expertise (satellite poisitioning e.g.). 1 and 2 are strictly interconnected, if you only master 1 your algorithm will be unusable, if you only master 2, your code will do the wrong thing, although in a very efficient way. 1 and 2 form together a bigger skill that is applied to 3. 1 and 2 together form a tool to form a tool to solve the problem that is formalised in 3.
- Nonkululeko:Mathematical statistcs-Python and R ,I am not really an expert I'd say I am a competent practitioner.I use R mostly in sta where given data I use R to manipulate the data but Python is alos very useful in these
- Nate Jo Hydrology (i.e., how does water run down hill!). The 'box and arrow' systems diagram fits our science quite nicely -- we often think about areas of water storage (e.g., a lake) and how different storage zones connect (i.e., rivers or groundwater flowpaths). Although, I guess you could say these physcially based facts are one "piece" or "category" of information. We also need to know facts/concepts from the geospatial sciences, ecology, and engineering. Water fluxes, and its impact on ecosystems and infrastructure, often connect these concepts.
- Karin computational neuroscience and machine learning- statistical or machine learning models and where they can be used appropriately ways to extend/modify them, relationships between models; tools for model checking and evaluating results; probability and mathematical fundamentals; type of data being considered; biological relevance of the data and what biological question we are trying to understand -- helps inform modeling choices and criteria for model selection/evaluation;
- Sam computer science/Python coding/data analysis and data engineering - I talk a lot about pipelines and data "flowing" from one place/system/team to another and being transformed at each stage, so it's a lot of graphs with nodes that have inputs and (transformed) outputs
- Danielle Health and Wellness - exercise - types of exercises - form - nutrition - mental health

- Talisha Genetics at least how we teach it at the elementary level– Genes are segments of DNA -> Transcribed to mRNA -> Translated to Protein; Dominant and Recessive Alleles -> genotype -> phenotype
- Yara Area of expertise: Causal Inference | Concepts / facts: temporality, types of bias, g methods, definitions of measures of association and risk, DAGs, probability notation, statistical methods | Types of relationships that are included: which method to use to address which type of bias, how to show a type of bias via a DAG, transforming a DAG to probability notation, transforming probability notation to a data analysis, interpreting data analysis to make inferences.
- Robert
- Jay Domain: Orchestration in Cloud Computing | Concepts: fundamentals of virtualisation; understanding of key technologies/tools; importance/approaches to abstraction; coding skills | Relationships: apply coding skills to link different key technologies; abstract technical details of complex tools; understanding of tools relies on a solid base in the fundamentals of virtualisation
- Al Community development, fund/friend-raising, :community impct model, storytelling, capacity building, organising, budget, impact, non profit policies
- Andrew: Laser matter interaction, Mental model: First principles thinking/thought experiment. Concepts: Quantum mechanics, partial differential equations, computational techniques, parallel computation. Relationships: tend to be one-direction- using tool X to solve the main problem (laser matter interation).
- Jess : My core work is generally helping folks get to the right visualization and visualization tool for their project. I take into account data quality, level of expertise at technical work, time available, funding status. Relationship between wide vs long data and which tools prefer each, relationship between visualization task (e.g. showcase outlier, show trend, have users identify slight differences) and chart type, relationship between person's time + expertise + interest and which tool will work for them.
- Addi: area of expertise: interdisciplinary: Data science, computational science, numerical methods, how do we realize science and scientific experiments via software; Relationships: Which numerical models express which physical processes, how do these processes be implemented computationally ? model: the Scientific method with software engineering principles
- Samarpita Expertise: wastewater treatment 1) Physicochemical processes for water purification 2) Biological treatment - understand microbial communities and the relationship between different microbes. Concepts - Inorganic or organic inflow rates, carbon cycle, nitrogen cycle Relationship: Function of microbes to wastewater effluent quality.
- Tayaza: Genetics of heritable diseases. Gene function, 3D structure of the genome, biological pathways, genetic variants. The 3D genome brings different parts of chromosomes together to enable genetic variants affect gene expression. The impacted genes in turn impact biological pathways, which lead to disease.
- Robert: Expertise - Network Infrastructure, network model 1) Applciation Layer - software 2) Transport Layer - software and network meet 3) Network Layer - addressing and routing 4) Data Link Layer - hardware control 5) Physical layer - message conversion. All work together, to send and recieve messages.

The Importance of Going Slowly
https://carpentries.github.io/instructor-training/fig/mental_models.svg

Mental model:

- important to know if someone is a novice / competent practitioner / expert -> change the way you

approach teaching.

- How to help build student's mental model: Go slowly, introduce things piece by piece. Best to go slowly so you don't lose anyone. It's ok if some are bored.
- ID and address misconceptions that learners arrive with -> misconceptions lead to broken mental model. If you break a mental model early on, very hard for a student to build a correct mental model later. Start with a solid foundation.
- Teaching code to novices within 2 days: how to think about and approach programming so they can build on their mental model.

How "Knowledge" Gets in the Way

Misconceptions

- Factual errors (e.g. Vancouver is the capital of Canada, e.g. to correct w/ info from authoritative source). Simple factual error is easy to correct.
- Broken models e.g. motion and acceleration always in same direction (most effort in carpentries aimed at this)*. Fix by having exercises -> allow them to find misunderstandings when they try to apply
- Fundamental beliefs eg. humans have an effect on climate change (probably best not to try to change these in a workshop)

* importance of skills for self-correcting broken models. E.G spending time explaining how to read error messages.

Identifying and Correcting Misconceptions

- Formative assessment - cooking example is when the chef tastes the dish and there is still time to make changes or updates; in a workshop this might look like ask for questions and then give an exercise to check understanding, there is still time for learning after the assessment.Spend a lot of time on formative assessment. Used to help people learn throught course.
- Summative assessmentwhen cook takes food right out, guest get first taste. Used to assess someone's understanding at the end of course. Difference between quizzes throughout class to facilitate learning and final exam which assesses understanding and knowledge gained. Pass or fail. All opportunity for learning has ended.

Repetition vs.Reflective practice

Formative Assessments Come in Many Forms

Identify the Misconceptions (10 min)
Choose one of the wrong answers to the question below and write in the Etherpad what the misconception is associated with that wrong answer.
Q: what is 27 + 15 ?
a) 42
b) 32
c) 312

d) 33

- David Ch - d) 27 + 1 + 5 ?!
- Nate Qu - 32; fogetting to carry the one
- Emmy - c) Adds first digits and second digits seperately, then just joins them together
- Lance-C-Adds the first digits from each number, then the second digit, and combines the consecutively. 2+1=3, then 7+5=12.
- Luca b) forgets carry
- Nonkululeko-32)add the first 2 numbers of each digit(1+2) for left and the second 2 numbers ofeach digit(7+5=12) but either forgot to write 1 or chose to skip it.312) again adds numbers of each digit left and goes on with 2 number and write them as they are (2+1=3,7+5=12 the conclude the answer is 312).33) add numbers of each digit from left but kind of round off to 2 digits hence 33(wrong rounding anyway)
- Nate Jo B -- did not add 10
- Karin in b) adds ones and tens but does not carry the one
- Sam c - does not understand that 10 carryover is added, not "inserted"
- Danielle - B. Only added the ones, did not add the tens. This shows that the learner may not understand the concept of adding two digit numbers , that you have to add both places
- Talisha - C, added ones place to get 12 and then first digits (or tens) to get 3, then put 3 in front of answer but did not combine these partial answers appropriately (or vice versa). Showing that learner may not realize that the 3 is a tens position or actually represents 30, also possibly other digit place misconceptions
- Yara. (b). forgot to carry the 10 from the 5+7 and add to the 20.
- Robert
- Jay d) carried the one into the ones column instead of into the tens column
- Al C: does not understand how to carry properly from ones to tens
- Andrew (A) assumes working in decimal number system... c: treats each "column" of addition independently
- Jess B forgot to carry the 1; C added first digits and second digits directly and put the two numbers next to each other; and agree with Jay's answer for d (read it before I could figure it out)
- Addi: b) adds the last digit but does not carry over, they do not understand how the carry works on additions d) add the last digit, but adds teh acrry to the last digit; They do not understand how carry works c) thye added the digits individually, they don not understand the concept of double digit numbers
- Robert: "D" added 5 and 1 together and then added 27 instead of adding 15, not sure when to add numbers together
- Samarpita b) Forgot to carry the unit's digit to the 10's c) Doesnt know how to carry forward d) Adds the unit's carry forward to the sum of the units digit instead of carrying it forward to the ten's digit
- Tayaza: b) Didn't carry over 1 to the tens column.

Handling Outcomes (10 min)

Formative assessments allow us as instructors to adapt our instruction to our audience. What should we do as instructors if the class votes for:

1. mostly one of the wrong answers? - Ask someone to volunteer and say why they picked the answer, and someone who picked the right answer to give some rationale for the answer they picked and then have a discussion after those responses. This is a great option because as the teacher we may have already tried to explain the concept so this gives a voice so that someone else may be able to examplain it and

they may use different language
All wrong answers should address some misunderstanding that you can correct.
2. mostly the right answer? briefly summarize and share logic that brought folks to the right answer. - This is a good approach to take. In carpentries workshops we have helpers that can help out folks who are helping trouble. This allows us to move forward if the majority of the class is ready to move forward and the helper can step in to help the person(s) who need it., brief summary can include common misconceptions and
- The ability to do this has been limited by the pandemic moving workshops online: take lots of short breaks to allow helping those who are struggling
3. an even spread among options?

Important to have frequent formative assessment every time you teach a new concept. Frequent breaks are also important.
Consider attention span and provide opportunities to absorb materal (?) Cognitive load: not introduce too many concepts at once and allow learning.

Key Points:

- Our goal when teaching novices is to help them construct useful mental models.
- This requires practice and feedback.
- Formative assessments provide practice for learners and feedback to learners and instructors.

III. Expertise and Instruction
https://carpentries.github.io/instructor-training/03-expertise/index.html

What Makes an Expert?

What Is an Expert? (5 min)
Name someone that you think is an expert (doesn't matter what they're an expert in). As an expert, what makes them special or different from other people? What is something that you're an expert in? How does your experience when you're acting as an expert differ from when you're not an expert?

- David Ch : understanding of the underlying reasons for some features, the history of it, how it evolves and why, the meaning behind => ability to face new contexts, new situations, new problems, ability to choose the best path among several ones available.
- Nate Qu: Dr. Anthony Fauci (physician); leading expert in the US dealing with the caronavirus; i trust what he has to say on this issue without much hesitence; I'm an expert in knowing when my dog Bruce needs to be fed or go outside; I don't have to look things up or feel like I'm unsure about something; done it enough times and seen the evidence
- Emmy our project's communication advisor - she's very good at ensuring that all the materials have a strong message for the correct target audience and I'm always amazed how she see all the places this could be applied, from a tweet to an infographic; I'm not an expert in communication, so I ask her to brainstorm/soundboard/give feedback on most of my work; I know open science and community methodologies well- i can see how people in the community can be connected and am confident to make/build those connections
- Lance: Dr. Tony Lyza (meteorology, specifically related to the study of tornadoes/tornado genesis) Regularly studies recent memory severe weather events, applies knowledge, shares through academic scholarship, but also in trying to inform the public (two very different means of sharing knolwedge). I really appreciate Dr. Lyza's realtime assessment of severe weather events. I

have quite a few things that I am proficient in, but I'm always hesitant with the word expert. I really love building search strings, and working with students in building search strings--it's fun to me to transition from simply developing a search string for researchers to use to teaching explaining each step for the decisions that I made in building the string. When I don't feel like an expert, I look for connections, but am a bit more hesitant to volunteer information. I try to find any connections between what I may already know, particularly if this a framework that I know and can apply.

- Luca: my first boss, I just have to confusily sketch my problem and could debug my code without even looking at it (my mistakes were pretty basic as I was beginner). I know a little of many things and a little more of some things but I don't really know much about anything :D . I feel that the more confident I am on some topic, the more confident I am in talking about it. So if the topic is something I know well, I am more comfortable in disagreeing with people. If I feel like a beginner, I accept other opinions despite I feel I would disagree.
- Robert : Mia Mottley; communicator, able to share ideas across disperate sections of the population and create coalitions to push forward a progressive agenda.
- Nonkululeko:My Statistics Lecture Dr Michael Rogans ,He can answer any stat related question you bring to him and he is really passionate about but also what I like is that he is still open to learning new things and improve his knowledge even though he is an expert.I don't really describe myself an expert in anything really maybe I would say I am really good in working with R and manipulating data but I still believe I have to learn more about it.
- Nate Jo My grandmother | Southern Cooking | Decades of experience cooking + access to 'old knowledge' passed down through oral traditions. (The assumption here is that knowledge is not as accessible today.) | Not sure I'd say I'm an expert in anything yet...however I have started to build mental models that allow me to aquire, organize, and aply information in unqiue ways in the hydro sciences.
- Karin A mentor teacher at the school I used to teach at (high school) - part of what made them an expert as teaching was how much they could SEE/observe in their classroom (or watching someone else's) - they could take in a ton of complex information about how students were working, problem solving, and processing material, and react quickly to that information. They had a whole host of strategies that they could use to adapt to how students were using, and could flexibly and quickly make an effective choice on the spot. Also just a huge amount of empathy and openness to what students were experiencing and how they were learning. I'm an expert in certain aspects of statistics and machine learning, which helps me be flexible in choosing a model/approach and adapting it for the situation.
- Sam: My former engineering manager - expert in software testing - has a good mental model and vocabulary of different components of testing, knowledge about different approaches and differences/pros and cons between them. You can throw any problem at them without much explanation of the context and they'll have an idea of what you're talking about. I'm an expert in data analysis and data engineering - I manage to navigate questions or problems without having to look up too many things, whereas as a non-expert I often have to ask a lot of "what is this" "what does this do" type of questions to navigate issues.
- Danielle My Dad. He is a computer engineer and he has been doing it for a long time. He has gotten to grow with the technological advances made since he started his career almost 40 years ago. He is able to figure out problems relatively easily. He is able to mentor others in his area of expertise. When I am doing something I am an expert at I am comfortable, I don't have to over think what I am doing, I can give extra information without having to look it up.
- Talisha - My friend, he is an infrastructure engineer and familiar with all things platform development. When I am doing something that I am an expert in I find that I am able to operate fluidly and move through challenges efficiently. When I am doing something that I am not an

expert in I move very slowly and ask a lot of questions, I find myself speding time trying to make connections between the pieces of imformation I am learning.

- Yara: Miguel Hernan: deep understanging of the material, excellent communication and teaching skills, a lot of confidence. My expertise is in Epidemiology. Difference in behavior when acting as an expert vs when not: more confident, more comfortable, able to explain a concept from many different perspectives, can provide many practical examples. Other: able to improvise, flexibility, thinking ahead. Reflected in conceptual model: i..e jumping from A to F. Disadvantages when teaching something you're an expert thing: connections seem obvious to you but not to students so you ASSUME they know / understand something when they don't. Conclusions not as obvious to novice and it can be hard to understand why this is the case. Fluid representation: expert's ability to think about a problem in multiple different ways
- i.e. refer to objects as character vectors / strings and knowing they're the same thing.
- Robert
- Jay David Attenborough: being an expert makes him trustworthy / Me (expert in frying eggs) - running on auto-pilot while frying eggs (can easily multi-task) vs. concentrating very hard when poaching eggs (can't maintain a conversation)
- Al Ali/Tyson: expert boxer. I am an expert teacher in Tang Soo Do, 7th degree Black belt studied karate for 15+, teaching students and running camps etc... There is a piece of control and volunerability that you are very comfortable sharing when you are an expert.
- Andrew My former PhD supervisor: has an enormous store of knowledge and experience, so that he can forecast potential problems or solutions three steps ahead of anyone else. Me: expert in playing songs by ear, can do it intutitvely without lots of mental effort vs learning from sheet music where it takes a lot of effort.
- Jess: Alberto Cairo. He's a visualization guru who is expert enough not just to quote examples, but to take examples of a could-be-improved data visualization in live chat and sketch in Illustrator on the spot, discussing the ways in which choosing the chart (even chart type) in this way would improve this aspect, transforming a graph live on the spot with commentary on why he makes the transformations he does.
- Addi: Jack Dongara, Parallel computing, considered a top aiuthority in scientific parallel computing, numerical method. Several awards in the HPC field. I am an expert in engineering research software. The experience differs from when I am not, is the confidence and understanding of a problem and beign able to confidently say if the problem can be solved and an estimate. When I am not an expert, it makes me hesistant to commit to timelines or solution correctness without following a thorough estimate or validation process.
- Samarpita: My supervisor/Prof| Understand the subject matter deeply enough not only for teaching but also to identify why or how a learner is thinking about some aspect of the topic incorrectly and then correcting that misconception|
- Tayaza: Miles Davis--knows how to improvise on the spot. Genome regulation: Confident and take a "for granted" attitude. I tend to think very linearly in subjects I am not an expert in.

Diagnosis (5 min)

What is an error message that you encounter frequently in your work? (These are often syntax errors.) Take a few minutes to plan out how you would explain that error message to your learners. Write the error and your explaination below.

- David Ch
    - "test.exe": command not found (while you see the file in your shell current directory)
    - the shell search the executables files in the directories listed in PATH, and this shell variable doe not necessarily contains ".", the current path. You may type "./test.exe", or

extend PATH.

- Nate Qu- 'Object not found' error in R. Typically i've either misseplled the object I'm trying to work with or simply forgotten to add the object using read.csv or something simple like that. Now that I've worked with R for a few years this is something that I can quickly recognize and correct but early on, the first few times this would happen I would be quick to hop onto Google or Stackexchange or some online community to see if I could find a solution to my error.
- Emmy Time zone conversion errors when setting up a meeting. Explanation: this error causes your attendees will not turn up at the time you'd like them to/you thought you'd agree on, leading to a lot of confusion. Time zones are difficult: there are many of them, their relationships change over the year (daylight saving :S), they are often named differently. It can be avoided if you use a webapp to double check time zones, or some calendars have built in functions to specify the event's time zone and do the conversion for you.
- Lance-Returning zero search results when executing a search string in a database, or sevearl hundred thousands results when i was only anticiapting a few. Many times, this happens because I have either made the search too specific, and I will talk learners through looking back at the search string to consider which term or operator may be the limiting factor. It's possible also that I have left out a parenthese, or that I've built my search in a way that does not mathematically make sense. I teach my students that this is an iterative process, and that error message/zero search results is a great way to hone the search string, and not to panic when they receive it. If the search string is short, or only contains a few words or phrases, I encourage them to consider: are the terms or phrases they have selected the way that others in the literature would have used. This give a great opportunity to introduce the thesaurus, or to explain how controlled vocabularly works.
- Luca: While programming in Julia: "No method matching f(::SomeType, ::SomeOtherType)". The error means that you are trying to call a method associated to a function with the wrong number and/or the wrong type of parameters. While it may seems trivial, sometimes its pretty subdule if you are e.g. using parametric types and get confused with type system.
- Nonkululeko:I normally forget to put a colon when writing code using loops or creating a function and then I get an error ,I do this more frequently and it is very weird
- Nate Jo -- Error message when doing raster algebra (e.g.,, adding to raster layers together). This is one of the most common errros when using GIS (command line or otherwise), and the error is typically due to mismatch in spatial extent, raster resolution, or coordinate reference system. If using R or Python, its simple enough to print the this information and check.
- Karin "Missing 1 required positional argument" or similar in Python, when I don't specify the arguments to a function correctly -- need to check what arguments the function takes and make sure I'm speficying them correctly. Quick to fix, but definitely comes up a bunch especially if I'm working with functions form an unfamiliar package!
- Sam: I'm almost guaranteed to forget how to try/except an error in Python - try/catch is ingrained in me having written Java for years. I'd explain it by looking at the stack trace that says "invalid syntax" and points at a specific line and character - look at line/character and if needed, look up syntax documentation for that specific command that's being used.
- Danielle When I am entering a workshop into AMY, if I don't fill out all of the blanks correctly I will get an error telling me so, and it will not let me move on without fixing the errors. If I were explaining this to someone, I would go through the different blanks that need filled in and explain what needs to be filled in at each blank and why. Have them fill them in and try submitting again to see if it works.
- Talisha - As a workshop administrator on error that we may see is "Slug Invalid". This happens when the slug (or unique identifier) for a workshop website does not meet AMYs validation. There are many ways to resolve this and so the steps I take depend on multiple factors. If the

workshop has already happened or will be starting within a week, we enter the recommended slug in AMY and then need to make some corrections to do other steps manually. - If the workshop is more than 1 week away, I may check witht he instructor/host to see if they can update the repository.

- Yara 'There were 14 warnings (use warnings() to see them)'. I usually type warnings(), check what they are and act accordingly (I usually ignore them because it's just an automated message that means my analysis is running the way I want it to run but for some reason R doesn't like it).
- Robert - "Disconnected from application, contact your network administrator", when connecting to Remote Desktop Apps, if you experience a disruption in your connection to the Internet or a drop in the download speed, you will get this message. Do not be alarmed, do not call the Network Administrator, check your Internet connection, clear your cache and if all else fails, check if the VM running the service is up.
- Jay "ubuntu@192.168.1.1: Permission denied (publickey)." when trying to SSH to a machine. Either I don't have my default key configured correctly, didn't specify a specific key with `-i` or haven't added my key to `.ssh/authorized_keys`. Understand the error (key problem) and try to work from the simplest mistake (didn't point to the correct/default key) before attempting the slightly more complex "fix the authorized_keys file"
- Al attachment error, when I forget to attach a doc and it reminds me because the word attach(ed) (ment)
- Andrew (Have the advantage of having just compiled an "error message cheat sheet" for students):

"Lastly, a common place you see this error is for code like this
y = ((4*x) / (3+z)
print (y)
which will give an error message
Traceback (most recent call last):
File "python", line 5
print (y)
^
SyntaxError: invalid syntax
Here, there is clearly no problem with the print command. The problem is on the previous line, where we have left one of our brackets open. If you don't close the brackets python reads onto the next line expecting that you're still typing in part of the equation, and thus when it hits the print it thinks there's a problem. It doesn't have to be print, by the way. If you get a SyntaxError and there is seemingly no problem on the line that's flagged: check the previous line for unclosed brackets."

- Jess: When Tableau disobeys me and won't let me pick the chart type I want. I know it's because of an intricate interaction between discrete and continuous data and what data is automatically aggregated versus not, but it's subtle. I also now know enough to know when I'm encountering a bug in the program and should restart, e.g. when filtering isn't working correctly but I KNOW I did everything right. It's lovely to have that confidence.
- Addi: Missing dependencies or wrong dependencies causing a build fail. I would say that each one of us have a different version of dependencies libabries installed when we start; just because it fails after you pulled it down, doesnt necessarily mean something is wrong with the way you did it. I encouraghe people to build it up front before they make any changes and also try to make the dependecy versions explicit in the build documentation. i would then ask members to first check that they have the right versions they need and that every error is not necessarily because they committed somethign wrong; i would explain that a build can fail for any reason, including lack of resources on their machine.

- Samarpita Incorrect password for logging into HPC. I will explain to learners first that when typing the HPC passwords there are no '*' characters unlike other password in web application. So watching what they type on the keyboard is improtant and if theres an error message, double check the password and key it in again.
- Tayaza:Forgeting to include brackets in the print command after moving from python 2.7 to 3.x.

More things to be aware of as an expert teacher
Expert Awareness Gap: can't remember trouble that we had learning so it's easy to miss things that are hard for beginners
Dismissive language: experts will show how much they know about something by using the word 'just'. Gives learners a clear signal of how easy we find this thing they're struggling to learn so it is demotivating. Will make them want to give up if they're struggling with something considered 'trivial' or 'simple'.

Remember that what learners find challenging might not be what you found challenging.
Many attendees will not have the same motivation to learn as you did.

Remember that a software carpentries workshop is not a computer science course. More focused on practical tasks.

Key Points:

- Experts face challenges when teaching novices due to expert awareness gap.
- Expert awareness gap: knowing something so well that it seems easy when it's not.
- With practice, we can learn to overcome our expert awareness gap.

IV. Memory and Cognitive Load
https://carpentries.github.io/instructor-training/05-memory/index.html

Types of Memory
When learning move from short-term -> long-term memory. Need to support learners in doing this & need to give them time
Only 5-7 items can be held in short-term memory.

Test Your Working Memory (5 min)
This website https://miku.github.io/activememory/ implements a short test of working memory. In this test, you will see about twenty words, each for a short amount of time. Try to memorize as many as you can.

What was your score? Write down your score to refence later.
3, I think I may reach 4 if I close my eyes after the 4th one...
5 -> 6 (:( ) sentences helped a little, I think trying to do chunking distracted me more, I started thinking of categories and forgot previous words
5, 7
10 // 11
9
5

9, 10 (two attempts) // (9, 10) -- same
5 (best score out of 3 attempts)
2, 8 (two attempts) (2nd attempt, 7 and 10 for each)
11/8. Then 8/13. Chunking works really well only if you get a string of words to make a nice sentence!
8/9
1st: (6) | 2nd: (8, 7) -- too fast to really apply chunking though
5,10//7,12
6, 7, 6, 9 - some chunking worked at the begining, and then I tried to remember the last couple of words by starting to fill them out first
5-5
6 - still 6 - not working for me :/
11 - I wanted to try chunking but it was going so fast I didn't feel that I could even use that technique 9. I tried again this time stopping after a number of words that I could repeat quickly (6) and then instead of closing my eyes I just looked while repeating the 6 words and happened to rember a few more. There were some words that related to my work so those were easier to remember.
6-> 9 stories are hard in a short time span

Strategies For Memory Management

Chunking - split information into small groups that are connected to each other

Test Your Working Memory (5 min)
This website https://miku.github.io/activememory/ implements a short test of working memory. In this test, you will see about twenty words, each for a short amount of time. Try to memorize as many as you can.

Active Learning Through Formative Assessment - applying learning / practice allows learner to transfer info to long-term memory

Limit concepts

(rule of thumb: maybe 9 at most between formative assessments)
Example of how many concepts can be found in one line of code:
https://twitter.com/elliewix/status/981285432922202113

Concept Maps as Instructional Planning Tools
Bubbles containing concepts +
Connectors with verbs about how concepts connect (important to be able to explain why / how concepts are related)

Thoughts: is it useful to ask learners to draw their own concept maps ? Is this tapping into a higher level in Bloom's taxonomy? Incomplete concepts maps can be used as an exercise.

Tool: excalidraw.com

Question: Is it advisable to walk learners through the concept map, or are concept maps more useful as a planning tool? - In the carpentries we do not show very often in the lesson the concept map. Concept maps have been used as a tool to ask learners to make a concept map at the beginning and at the end to

see improvements in learning.

Exercise idea: give incomplete concept maps and ask learners to fill in the gaps (perhaps giving them the correct wods to fill in)

Example concept maps:
https://carpentries.github.io/instructor-training/fig/array-math.png
https://carpentries.github.io/instructor-training/fig/conditionals.png
https://carpentries.github.io/instructor-training/fig/create-destroy.png
https://carpentries.github.io/instructor-training/fig/dict-set.png
https://carpentries.github.io/instructor-training/fig/io.png
https://carpentries.github.io/instructor-training/fig/git_concept_map.png
https://carpentries.github.io/instructor-training/fig/lists-loops.png

Concept Mapping (10 min)
Create a hand-drawn concept map for a part of a Carpentries lesson you would teach in five minutes (ie. the amount of material you would teach before doing a formative assessment). You can use the same subject about which you created a multiple choice question, or a different subject. Trade with a partner, and critique each other's maps. Are there any concepts missing in your partner's map that you would include? Are there more than a handful of concepts in your map? If so, how would you re-divide those concepts to avoid overwhelming your learners' working memory?
Take 10 minutes to draw the concept maps and share with your neighbor. Write "*done*" in the Etherpad chat once you have finished.
A web tool for concept mapping: https://excalidraw.com/
Side note: A lot of librarians I know use concept maps to help students with the research process: https://libguides.metrostate.edu/librarytutorial/brainstorming

Why Guided-Practice is Important

Faded Examples

Faded examples work because they introduce the problem-solving strategy piece by piece. At each step, learners have one new problem to tackle. This is less intimidating than a blank screen or a blank sheet of paper. It also encourages learners to think about the similarities and differences between various approaches, which helps create the linkages in the mental model that instructors want them to form.
For example, someone teaching Python might start by explaining this:

```
# total_length(["red", "green", "blue"]) => 12
def total_length(words):
    total = 0
    for word in words:
        total += len(word)
    return total
```

then ask learners to fill in the blanks in:

```
# word_lengths(["red", "green", "blue"]) => [3, 5, 4]
def word_lengths(words):
    lengths = ____
```

```
    for word in words:
        lengths ____
    return lengths
```

The next problem might be:

```
# concatenate_all(["red", "green", "blue"]) => "redgreenblue"
def concatenate_all(words):
    result = ____
    for ____ in ____:
        ____
    return result
```

and learners would finally be asked to tackle:

```
# acronymize(["red", "green", "blue"]) => "RGB"
def acronymize(words):
    ____
```

Create a Faded Example from a Lesson (10 min)
The following exercise should be done in groups of 2-3.

1. Pick a block of code from an existing Carpentries lesson, or from another lesson you have taught recently.
2. Replace 2-3 pieces of the code with a blank.
3. Write a question to test the student's ability to correctly fill in that blank.
4. Paste your faded example in the Etherpad.

This book chapter http://teachtogether.tech/en/index.html#s:exercises from Teaching Tech Together by Greg Wilson contains many examples of different exercise formats. This is a really valuable resource when designing a new lesson. It includes some guidance on when the different exercises would be most useful.

V. Building Skill with Feedback
https://carpentries.github.io/instructor-training/06-feedback/index.html

Surveys
For links to our surveys see: https://carpentries.github.io/instructor-training/06-feedback/#surveys

Minute cards

One-up, one-Down
- Don't respond to feedback - responding will affect the following feedback/comments; just thank them and move on

--Day 2--

Give Us Feedback (5 minutes)
Use this web form [https://forms.gle/6umQuXT5AdiNrQqM8] to tell us one thing you learned this morning that you found useful, and one question you have about the material or something we could improve for next time. Do *not* put your name on the notes: this is meant to be anonymous feedback.

Start with the most important topics first, then filter through the remaining concepts based on how long versus how useful. This means that sometimes best practices may be left out of the lesson, for example teaching coding without teaching standards for how code should be formatted/indented/etc.

VI. Motivation and Demotivation
https://carpentries.github.io/instructor-training/08-motivation/index.html

Creating A Positive Learning Environment

- Presenting the instructor as a learner. Live coding is helpful as it shows that it's ok to make mistakes. Teaches students how to troubleshoot.
- Establishing norms for interaction. i.e. code of conduct. Formalize how to talk, how to ask questions -> all interactions. This is helpful for those who don't feel confident.
- Encouraging students to learn from each other. this can be more difficult when teaching online, because it is harder for people to "lean over" and help out/discuss with the person next to them. Toby has had some success with creating smaller groups in breakout rooms to work on exercises and help each other out, but has found that it works better for some groups than others e.g. based on their inclination to unmute their microphone and talk to the other group members. Yara has recently received feedback that learners do not like it when they are placed into small groups but that may be improved by setting some expectations for how these groups will work together, establishing a Code of Conduct etc. Andrew also heard of some similar experience, where laying out motivation for doing these small group breakouts and some guidance on how to make the most of them seemed to make a difference.

Explaining motivation for breakout rooms -> explain that students are there to help each other. Possibly having an ice breaker at the beginning.

- Acknowledging when students are confused. Encourage feedback using an ice breaker / low stakes exercise / asking for reactions on zoom (yes/no/thumbs up / faster / slower).

Teach Most Useful First
https://carpentries.github.io/instructor-training/fig/what-to-teach.png

Authentic tasks: tasks from day to day work help the learners understand and relate to the importance of learning the skills you are teaching. Feel more immediately useful, and easier to see how they could apply it to their own work, when compared to a more contrived, "toy" example that seems less immediately applicable.

Authentic Tasks: Think, Pair, Share (10 min)
Think about some task you did this week that uses one or more of the skills we teach,
(e.g. wrote a function, bulk downloaded data, built a plot in R, forked a repo) and explain how you would use it (or a simplified version of it) as an exercise or example in class.
Pair up with your neighbor and decide where this exercise fits on a graph of “short/long time to master” and “low/high usefulness”.

In the class Etherpad, Share the task and where it fits on the graph.
As a group, we will discuss how these relate back to our “teach most immediately useful first” approach.

--> breakout room until XX:51

Breakout Room 1 (Nate^2) - Using ggplot to generate plots in RStudio. We saw this as being pretty useful and would put over to the left, saving some room prior to introducing ggplot to walk through the environment setup and data cleaning pipeline. Discussed exercises that illustrated usefulness of ggplot style plotting.

Break out Room 3 Al working on auto custom emails for members. Robert-moving data from one server to a new server, wrote some simple programming for that.

Breakout Room 4:
- Using R, instead of Excel for financial analysis. Important skill to learn and simple to understand (in this case, basic matrix operations).
- Troubleshooting a "xx rows removed because of infinite values" when ploting graphs with ggplot with a masters student. Involves multiple concepts e.g. understanding logs, googling possible causes of infinite values, and filtering a tibble based on conditions. A little involved but important to master.

Break out Room 5: We discussed teachign bash concepts and SQL query from databases.
Short time to master/high usefulness: counts the words in the file - WC. , how to run select queries on a table
Long time to master/less usefullness: Changign File permissions , how to run DDL queries for database ans table creation.
Question from us: How should we handle it if a learning asks question regarding the long time to master/less usefull for the resy of teh cohort question, should we go into those details.

Break out Room 6:
- rebasing pull requests
- creating dataframes in python:

-We decided that both tasks, on a scale from 1-3, would be 3 (high) on usefulness and 2 (medium) on ease of mastering.

Break-out room 7 (Sam + Emmy)
- Run tests in a code repo following instructions, modify code/feature, run tests again to see if there are failures, modify tests and run/iterate to get them to pass - very useful (you'll probably affect tests if you're working on a codebase), not very hard to master
- git rebase to delete a commit (long time to fully master, not very useful (hopefully i never have to delete commits again))

Break-out Room (Robert and Al)
- creating a data pipeline (userfulness 3; ease of understanding 1)
- creating an automated quarterly e-mail (useflness 3; ease of understanding 3)

Break-out Room 8 (Yara & Jay):
Scale: 1: not useful and 3: most useful | 1: easy to understand, 3: hard to understand
- creating variables/constants in Python (usefulness 3; ease of understanding 1)
- creating simulated data to generate random exam questions (usefulness 1, ease of understanding 3)

Break-out Room 9
- on a remote cloud machine, checkout a given branch from some colleague/student/teacher
- we discussed how working with "real" (large) code helps to motivate the need for version control, whereas seeing only toy examples (few lines), version control seems like overkill.
- highly useful, moderately difficult
- question to carpentries trainers : should we be authentic, and address both the "cloud use" and "git checkout" topics together, or either focus only on one aspect, and be less "authentic" ? ;)

Breakout Room 10
-Cleaning data using dplyr in R in general, specifically making sure R doesn't strip leading zeros in geographic data (hard-ish to learn but extremely useful)
-Resolving a merge conflict in git - high usefulness and high ease of mastering

Breakout Room (Talisha & Danielle)
- Creating a spreadsheet that summarizes data pulled from data exports (using HLookup or Vlookup, SUM) (very useful, not too much time to master)
- using template to bulk upload into database. Useful and quick to learn

Strategies for Motivating Learners (5 min)
How Learning Works by Susan Ambrose, et al, contains this list of evidence-based methods to motivate learners.
In groups of two or three, pick three of these points and describe in one sentence in the Etherpad how can we apply these strategies in our workshops.

- Strategies to Establish Value
    - Connect the material to students' interests.
    - Provide authentic, real-world tasks.
    - Show relevance to students' current academic lives.
    - Demonstrate the relevance of higher-level skills to students' future professional lives.
    - Identify and reward what you value.
    - Show your own passion and enthusiasm for the discipline.
- Strategies to Build Positive Expectations
    - Ensure alignment of objectives, assessments, and instructional strategies.
    - Identify an appropriate level of challenge.
    - Create assignments that provide an appropriate level of challenge.
    - Provide early success opportunities.
    - Articulate your expectations.
    - Provide rubrics.
    - Provide targeted feedback.
    - Be fair.
    - Educate students about the ways we explain success and failure.
    - Describe effective study strategies.
- Strategies for Self-Efficacy
    - Provide students with options and the ability to make choices.
    - Give students an opportunity to reflect.

- Sam Establish value: Have students share situations in which the skill they're learning would have come in handy - this usually works really well in e.g. bash/terminal workshops where they automate tasks
- Danielle Identify and reward what you value- Look for the thing that the person did correctly, or well, and focus on that. You did a great job doing ....

Show your passion and enthusiasm for the discipline - make a big deal of doing things correctly

- Talisha - Provide learners with options and the ability to make choices. An example of this would be providing an exercise that has multiple options and students choose 1 to complete. Another example is providing multiple scaffolding exercises and students pick one+ to complete. This is a great way to motivate because students have the choice to select the option that peaks their interest and/or is at their skill level. This option is also great for scaffolding or meeting students where they are because the exercises can range in difficulty.
- Yara . "Give students an opportunity to reflect". Like the 5 minutes we had this morning, I reviewed yesterday's notes which was nice.
- Rob*ert To build positive expectations we can provide clear rubrics while promoting fairness.*
- Jay *Show your own passion and enthusiasm for the discipline.* -- provide personal anecdotes, stories of challenges faced during our own learning, personal projects,
- Al Provide authentic real world task, use current events from communities present or global news
- Andrew **Show relevance to students' current academic lives**: most learners have to write report or thesis: demonstrate the value of version control by showing typical "bad" practice (one document, copied into several versions), vs good practice (easy recovery of previous drafts etc.)
- David Ch: **Show your own passion and enthusiasm for the discipline:** Give some hot news about the discipline, explain why it is worth to invest on it, with personal stories.
- Nate Qu : **Provide early success opportunities**: scaffold the lesson so that students have the opportunity to succeed on a task that is at their level of confidence or comfort (so starting with strings, lists or variables in Python before moving into more challening topics like loops or conditional statements)
- Emmy Educate students about the ways we explain success and failure - we make it clear that instructors are also learners, show preworkshop survey results to demonstrate the diverstiy in skill level in the participating group, and avoid the use of dismissive language
- Lance Provide early success opportunities. Providing students quick ways to execute principles taught in their own terminal. An example in any program is the "Hello ,World!" program. In Python, having students create and then print variables. Students can see they are successful early on, and feel more confident in proceeding with more difficult tasks later. You can refer back to these early success moments in later, more advanced parts of the lesson to draw relationships between the successful tasks students executed earlier, and the more advanced work they are doing now.
- Luca : **Show relevance to students' current academic lives**. Show e.g. from github some open source projects of research softwares which are in several "hot" research areas. Motivate that having the necessarily software skills open doors to collaborate and enhance those projects that can be used in their own research.
- Nonkululeko Provide targeted feedback.After doing each assessment with the learners you check if they anderstand the work correctly ,if not give feedback of where they be misunderstanding the particular topic and show the correct way to do it.
- Nate Jo Show your passion for material. I normally workshops [or inidividual units] by talking abou thow useful these skills are to me AND how I they've improved my own research. For example, I highlight recent papers where we analyze streamflow data from accross thousands of

USGS gages. I'm not a computer scientists -- but new data tools are accessible and make that type of analysis possilbe for practioners

- Karin Show your own passion and enthusiasm for the discipline: room for instructors to share a little about their background, express warmth and enthusiasm in their tone and response to student questions and feedback; to me this point also relates to the carpentries principle of instructors presenting themselves as learners, can share the relevance of what students are learning to their own work and speak in a meaningful way about why the material is interesting and important
- Jess Connect the material to students' interests: If you have a group of students from a particular sub-group, like the economics department, change the data to something from their field. Or choose data from your particular university or geographical area to increase interest.
- Addi :

connect the material to the learner's interest by picking a dataset from theri domain. Learning how to query this database would then provide immediate value as well as establish relevance. for the 2nd bullet, after teaching the select problem, come up with a set of relevant execercises that answers important questions pertaining to teh field of ineterst and workshop knowledge level. For the 3rd bullet point, provide a few different option to excercise the taught concepts and have a open ended question as to what would work better for their environment for these kind of operations.

- Samarpita Building on what students already know and showing a connection to the new topic being taught. Keep getting feedback from the student on the new concept through formative questions.
- Tayaza : Build examples based on expectations of students and show the professional opportunities to be achieved by describing effective strategies.

How Not to Demotivate Your Learners
Don't shame current data analysis practices (excel, GUI etc.)
Avoid the word "just"
Don't take over the learner's keyboard
Don't feign surprise. ex, "What?! You have never heard of x??"
Avoid unrelatable examples / examples require specific knowledge of an area i.e. sports

Psychological Demotivators
Helps to work with someone who you trust and can give you feedback re the below.

- Stereotype Threat: Don't pinpoint minorities ("Great to see so many women in the class today!")
- Impostor Syndrome: The belief that you are not "good enough."
- Accessibility Issues
- Lack of Inclusivity

The UK Home Office (https://hodigital.blog.gov.uk/category/accessibility/) has put together a set of posters of dos and do nots for making visual and web-based materials more accessible for different populations. Take a look at one of these posters (https://github.com/UKHomeOffice/posters/blob/master/accessibility/dos-donts/posters_en-UK/accessibility-posters-set.pdf) and put one thing you have learned in the Etherpad.
Toby also recommends this blog post about writing alternative (alt) text for data visualisations: https://medium.com/nightingale/writing-alt-text-for-data-visualization-2a218ef43f81

- this is something our lessons are not very good at (yet!) but we are working on it. when you come to make a lesson contribution during your checkout for certification, I woul *love* it if you took that chance to add/improve the alt-text on one of the images in one of our lessons!

Question: Re imposter syndrome, is it worth addressing explicitly? I.E announce the fact that people might be feeling like imposters, and normalise that feeling, even by expressing your own imposter syndrome?

- Toby: I don't think you need to address it explicitly (though I do not know if it would hurt to do so) however, it is a very good thing to refer to when you introduce yourself e.g. at the start of a lesson, or when recounting an anecdote while you teach. E.g. saying something like "I remember when I first started learning this, I felt like I was never going to understand it. And then my supervisor asked me to attend an event where I surrounded by people I was sure knew more about the topic and was much better at programming than I was. Then, one attendee asked me a question about my poster and I explained the method we used, ..." One of the only things that helped with my own imposter syndrome was listening to other people talk openly about theirs. I was lucky to have some great colleagues, who are outwardly confident and objectively successful/good at what they do, talk about their own fear that soon the streak of "good luck" they had been enjoying would come to an end, or someone would ask them a question at a conference that they wouldn't be able to answer, etc. Listening to those kinds of experiences, and surrounding yourself with people you can trust to give you honest feedback, are two great ways to combat your own imposter syndrome. In a workshop, we can try to create this kind of environment for the learners, by being honest about our own experiences and giving plenty of opportunities for learners to measure their progress through formative assessment.
- I think a great issue about imposter syndrome is that by definition it is very hard for people to realize they have it. Because if someone thinks their work is not that good, it is very hard unless impossible to enter the optics "I cannot help but consider my work inferior, but that's just because I have imposter syndrome and even though I cannot see the goodness in my work it is actually good. <- (Toby) yes! this is so true. I think what worked for me was thinking about the statistical improbability that I was the "special" one who really *is* the imposter, among this large number of other people who clearly (to me) suffer from imposter syndrome.
- Thanks for bringing this up. What are your thoughts when cohosting meetings with other members who may be or are (visibly) more confident. I feel like showing uncertainity when pairing up with some of my peers makes my imposter syndrome even worse later when i reflect on how we each handled the meeting. <- (Toby) this can get complicated by power dynamics that make it challenging/risky to be honest in conversations. I certainly don't recommend taking the above approach at all times but, when you are the instructor in a workshop (as long as you can trust your co-instructors), you are in a position of authority relative to your learners. This is a good opportunity to practice this kind of approach. But I must check my privilege as a white cis-gender male and recognise that this kind of vulnerability is safer for me than for many others. -> Thank you, appreciate your thoughts. This topic is close to my heart, it delfintely helps to be aware and mindful of it.

I really like this perspective though, that puts the onus on systems that make certain marginalized groups feel like impostors (i.e. it's not all in our/their heads):
https://www.instagram.com/p/CNITS5tjjgV/?igshid=2el86jojv7nr
For those who don't have instagram: 'it seems like what we really need is to interrogate and name how oppressive systems impact us and how they often convince us to internalize the way others treat us, rather than identifying the pattern and calling" it impostor syndrome
Another issue with impostore syndrome (especially for phd students) is hearing more skilled researchers constantly criticizing the system (review proces is random/broken, conferences are not meritocratic). While it is very likely that the system is not perfect, hearing experienced people constantly and openly criticizing it strengthens in new researchers the idea that "my publications are getting through just

because the system is broken" which on its side strengthens the impostore syndrome.

VII. Mindset
https://carpentries.github.io/instructor-training/09-mindset/index.html

Some learners have the belief that they will never learn to program because they don't have this ability (it is innate / you have to be a natural).
At the end of the workshop, we want learners to have confidence that they can continue learning.
How can we help learners overcome this belief?

Praise Influences Mindset
- praise the effort -> improvement / effort-based praise not the performance
"You have understood X, you have made a lot of progress, you are moving in the right direction.."

Choosing our Praises (5 min)
Since we're so used to being praised for our performance, it can be challenging to change the way we praise our learners.
Which of these are examples of performance-based, effort-based, or improvement-based praise?

- I like the way you tried a couple of different strategies to solve that problem.
- You're getting really good at that. Keep up the hard work!
- You're really talented.
- That was a hard problem. You didn't get the right answer, but look at how much you learned trying to solve it!

- Sam effort, improvement, performance, improvement
- Danielle 1. effort, 2. improvement, 3. performance, 4. improvement
- Talisha 1. effort, 2. improvement, 3. neither, I would categorize this as praising natural talent rather than the outcome which reinforces a fixed mindset, 4. improvement
- Yara . 1. Effort , 2. Improvement & Effort , 3. Performance (implies natural talent - should be avoided), 4. Improvement
- Robert 1 effort-based, 2. improvement-based, 3.performance-based,4. effort-based
- Jay i) effort ii) improvement iii) performance iv) effort/improvement
- Al 1) effort2)improvement3)performance/improvement4)effort/improvement
- Andrew 1. Effort, 2, improvement, 3, performance, 4, improvement
- David Ch 1. Effort, 2. Improvement, 3. Performance, 4. effort and a little improvement
- Nate Qu 1.) effort 2.) improvement 3.) performance 4.) improvement/effort
- Emmy (1) effort (2) effort/improvement/performance (3) none? it seems to say their performance/ability is innate (4) improvement
- Lance 1-effort, 2-performance, 3-performance (trying to decide if contextually, this may also be improvement), 4-improvement
- Luca 1 = effort based, 2 = improvement based. 3 = none of the above, 4 = effort and improvement
- Nonkululeko 1)effort,2)Improvement,4)improvement,3)Perfomance based
- Nate Jo 1 Effort | 2 Improvement | 3 Performance | 4 Improvement
- Karin 1. Effor 2. Seems like a combo of all 3? 4- improvement
- Jess 1. Effort 2. Improvement 3. Performance, but not growth mindset nurturing 4. Improvement
- Addi 1. Effort 2: Improvement 3: None (actually this one sounds more like opinion than feedback) 4. effort and Improvement
- Samarpita 1 - Effort, 2- Improvement, 3- Performance, 4- Improvement

- Tayaza 1: effort-based; 2: improvement-based; 3: performance-based; 4: improvement-based

Errors are Essential to Learning

It's okay to make mistakes. You will make mistakes and learning from your mistakes is what is important

Helping Learners Learn From Mistakes (5 min)
A learner at your workshop asks for your help with an exercise and shows you their attempt at solving it. You see they've made an error that shows they misunderstand something fundamental about the lesson (for example, in the shell lesson, they forgot to put a space between ls and the name of the directory they are looking at). What would you say to the learner?
In the Etherpad, describe the error your learner has made and how you would respond.

- Sam Eg if they got the order wrong for a SQL query, I'd ask them to go back to the part of the lesson that shows how a query is constructed from building blocks (start with a simple select, then add other components) so they could recap/understand in which order the components are added. My goal would be for them to realize what they got wrong *and* where/how to find help instead of me telling them.
- Danielle - Using the cooking example: I would tell the learner that the whole egg can't go into the bowl, you wouldn't be able to mix it in. Then walk them through, first gently crack the egg against the counter, split the egg in half over the bowl, let the inside (yolks and egg white) go into the bowl to mix with the rest of the ingredients.
- Talisha - Using another cooking example, lets say the learner was having difficulty with the recipe beccause the amounts for ingredients are in a separate location than the instructions. I would take a moment to explain the layout of a recipe and work through a few examples with the learner and then let them move forward independently.
- Yara I had a student convert a numeric column into factor but she saved it as a vector object outside the data frame. I praised her for doing the conversion correctly but I showed her that no column was created in the data frame and I showed her how to write that code and then we checked the data frame again. We discussed why it was important to save this new variable in the data frame instead of as a separate vector.
- Robert - The student had a syntax error, he had mistyped the name of a command, I would show them the cause, and ask them if they were comfortable making the correction. If the student does correct the mistake we would continue, otherwise we would review the incorrect command and related syntax.
- Jay -- explain how the shell understands what was typed by looking at the error message (ie. thinks `ls/home/ubuntu/` is a path to a file/directory). explain the `ls` command on its own. allow the student to make a link between the error message output and using ls on its own
- Al
- Andrew I would ask what the particular piece of code was trying to achieve, to ensure they understood the task. This might also shed light on the problem. "I'm trying to list the files". Then we could break the command into clear instructions "tell the computer you want a list", then "what do you want to list". If the misunderstanding was ~~just~~ a simple factual misunderstanding (didn't know we needed spaces), then that can just be stated.
- David Ch - Syntax error - Make it try a working example from the lecture, and modify it step by step until it is close to the failing code, and the syntax difference is clearly visible.
- Nate Qu - I would have the student share their screen if we were working online and they were comfortable sharing and we might begin to debug their code line by line. Having the student read

back to me what they thought each line was doing for their code. If we got to a place where there was a clear, fundamental misunderstanding with their code, we could spend time going back over those basic steps at a slower pace, or providing additional examples of that concept (working with strings or lists in Python for example) and then revisit their broken code to see if those remedial steps were helpful. If not we would continue iterating this process (explanation and examples) until the student's issue was resolved. If this still proved to be unhelpful I might try to recruit another student who *did* have a clear undstanding and bring them in for some peer review if needed.

- Emmy - I would try to not assume why they made the mistake - there would be an error message (maybe), and I would ask what they think went wrong and walk me through that thinking process. Then pick the step that was wrong, verbally correct it (if they didn't, say e.g. "you need a space for it to be understood correctly") and emphasise that I have made the same mistakes many times before (if that's the case). Then ask if they have further questions and try to address those (or say we'll address them later) if time permits
- Lance Still a bit new to the Unix shell, but I would likely explain the student the way that Unix understands arguments positionally, and how spaces play into the argument. There are parts of the what the student typed that are correct, and I would identify those, but explain how to present this in a way that the terminal can identify as a valid command. In reading the responses of my colleagues here, I really appreciate asking the learner to take a look first to explain what they had typed, and what they think may be the issue.
- Luca : I would ask the student to explain me their proces on how they got to their conclusion, like what their assumption were, what were their choices based on. If I spot the mistake in their thinking process, I would offer a counter example that shows the error, like "you assumed this works like that, but what about this situation, what does it happen? Then the student would realize the error and have this "a-haa" moment which I think greatly helps long term learning. I guess this might have the downside that the student feels like they are being tested.
- Nonkululeko:Your command is correct you have just a minor error of not including the space between ls and your directory ,try to solve that and run your command again.Always remember to separate these with a space bar
- Nate Jo - I normally ask student to start talking through the process. What are the steps they need to complete? If this dosn't promt them to fix the problem, I normally let them complete, and then back pedal to the step they ommitted or got wrong.
- Karin I'd probably first see if they could spot the error on their own, like: "Ooh, interesting! What do you think might be going on here?" They might be able to fix it right away or with a moment of thought. If it seems like they're more stuck, I could say something like "okay, cool, it looks like you're missing a space between ls and the name. The spacing is going to matter in our shell scripting, so it's something we'll always have to be careful about." And "Here, ls is the command, and the space separates it from the name that it's getting applied to."
- Jess I'd say "Oh, nice job. One thing that's a little confusing when we first start out is that there's a command, which is the first thing to write, and you can think of it as a strange word that does something magical. Then the next word is what you want to do that magical thing on. But since they're separate words, you need a space between them. In computer science, we call the magic word the command and the things you want it to act on the argument, which in this case is a folder on your computer, which is called a directory." (I say it with a clear sense of humor so people don't think I'm treating them like a child, I promise).
- Addi : Hey great job understanding the ls command, all you are missing there is a space. Ls is a command which is an instruction where as directory is a parameter or argument thats is an input to the command. When issuing this instruction, the syntax requirs us to include a space between the command and its argument.

- Samarpita I would ask the student to explain their thought process for a particular chunk of code and then compliment how they have applied concepts correctly. Then move onto to explain any misconception/error they might still be making i.e. the space between command and the argument. FInally, end it by telling the studnet how far they have come in this particular concept/understanding
- Tayaza: I'd first ask what they think about the error, then point out that they have grasped the command except for the space. Then I'll generalise the lesson for future cases (i.e. there are always spaces between commands and arguments)

When a learner makes a mistake:
1- Identify why they made the mistake and what they missed in the lesson / if training material was unclear
2- Praise the student's effort
3- Give an example related to their background
4- Ask them to retrace over steps

Perseverance Predicts Success
- emphasize that learning will continue after the workshop
- acknowledge that learning is hard
- encourage students to keep trying
- acknowledge that within two days they can only learn so much and that's ok

VIII. Teaching is a Skill
https://carpentries.github.io/instructor-training/11-practice-teaching/index.html

Lesson Study: Applying a Growth Mindset to Teaching

Having a second instructor teach with you allows for them to also give you feedback

Giving Feedback (10 min)
We'll start by observing some examples of Carpentries-style workshop teaching and providing some feedback.
Watch this example teaching video (https://www.youtube.com/watch?v=-ApVt04rB4U) as a group and then give feedback on it. Put your feedback in the Etherpad. Organize your feedback along two axes: positive vs. negative and content (what was said) vs. presentation (how it was said).

Content +
Content -
Presentation +
Presentation -

- Sam
    - Content+ Lots of examples illustrate the idea of functions well
    - Content - Unclear what the goal of the lesson is, could motivate, give outline and limit concepts taught, structure more clearly
    - Presentation + Live coding is engaging
    - Presentation - Uses lots of jargon/new terms without explaining, and they're likely not relevant to explain the basics of functions, I'd drop them

- Danielle
    - Content +- did ask for questions about what was taught at the end
    - Content -assumed that people would know what they were talking about and doing without explaining what it meant or why. "I am just going to do this" and fixed it without explaining.
    - Presentation + - showing what they are doing on a screen as they do it
    - Presentation -used phrases containing the word "simple", as in, "this is really simple stuff" often
- Talisha
    - Content + : Your explanation of X was really good, keep doing this.
    - Content - : Next time, try to be more clear in the rationale for using the X function and the expected outcome
    - Presentation + : It was great how you demonstrated typing out the code, making mistakes and correcting them on screen, keep doing this.
    - Presentation - : Next time, try avoiding de-motivational language "this is really simple stuff even excel users can understand this"
- Yara Content:
        - Keep doing this ????
        - Work on this next time: Explain what each line of code is doing and slow down
    - Performance:
        - Keep doing this. Acknowledging when you make a mistake / live coding & make a mistake when coding to show that it's ok
        - Work on this next time: tone when addressing students. Better lighting. Make text bigger so it's more visible.
- Robert
    - Content +: Explnation of function is accesible for novices.
    - Content - New concepts are introducded too quickly.
    - Presentation + Acknowledged your mistake.
    - Presentation - Warmup and introduction needs work, try to make everyone comfortable, maybe an icebreaker.
- Jay
    - Content + ..."any questions?"
    - Content - ...code is not related to any concrete (real-world) example
    - Presentation + ...it was nice to come back from the break with a re-cap (content of re-cap is not great though...)
    - Presentation - ...speaking too quickly with too much technical jargon
- Al
- Andrew
    - Content+: functions used were reasonably simple, few lines.
    - Content-: invoked too many concepts (polymorphism, objects, types, high-level function etc). Most of these don't need to be mentioned to convey the key idea.
    - Performance+: made a mistake showing that errors are part of process, but...
    - Performance- "don't worry about that, I'm just fixing it up" doesn't explain the process of fixing the error, or make clear that errors are expected. Screen not visible. Used dismissive, demotivating language ("even excel users etc.").
- David Ch
    - Content + I do not know, display too small
    - Content - I do not know, display too small

        - Presentation + live coding
        - Presentation - display too small ! look at his phone, say "simple", etc.
- Nate Qu
    - Content +
    - Content -
    - Presentation + asked if there were questions but waited until the very end after he'd introduced a bunch of new concepts; does quick review of what was happeing before the break
    - Presentation - 'even Excel users can do this'; talks about how easy this all is [really simple stuff]; barks at students to sit down at the start of class; font too small on display screen
- Emmy
    - Content + :
        - The explanation on using "def" to define the function is clear at the beginning
    - Content - :
        - I got lost at "This is what you'd expect, trust me" - the "jump" in logic in the explanation is bigger than what I can follow
    - Presentation + :
        - He made mistakes, which can be nice for learners to see (that also experts make mistake), if only he had taken the time to explain the error and his fix...
        - good amount of hand gestures? (although that's very subjective)
    - Presentation - :
        - "this is really simple stuff", and a lot of "just"
        - the font size on screen is too small to see from afar or even in the first row?)
- Lance :
    - Content +
    - Content -
    - Presentation +
    - Presentation -
        - Use of the word "just" multiple times to be dismissive. "Easy enough for an Excel user." Continually saying, "This is simple stuff." Language is alienating.
- Luca
    - content +: introduces the topic correctly (I think, I could not see what he was typing)
    - content -: too many details (invocation vs definition, function polymorphism).
    - presentation +: reconnects to what had been done before at the beginning of the lecture.
    - presentation -: agressively shouts down student at the beginning. Dismissive language "this is very easy, this is trivial". Terminal font too small, I could not see anything. Uses specific jargon which can be difficult for beginners "this is obviously polymorphic" a beginner probably does not know what a polymorphic function is.
- Nonkululeko

Content +:"He explained on using def to define function"
Content -:No structure
Presentation + :"He apologized for making a mistake"
Presentation - :"this is really simple stuff"

- Nate Jo

Content +

- Functional programming...

Content -

- -No structure to the presentaiton. What are the main concepts, how do they build on each other, and how do they connect
- -Did not allow students to practice concepts
- -Did not provide summary of concepts as wrap up

Presentation +

- Live coding is a great idea!

Presentation -

- -Was confrontational to begin lecture
- -Difficult to see his ppt
- -Delivery not smooth
- -skipped steps and did not explain

- Karin
- Content + Examples seemed varied had some flow from one to another?
- Content - Think about keeping things simple and be careful referencing topics or using language that a learner might not know about yet. For example, mentioned that you're "instantiating a function object" or describe a function as polymorphic, remember that these are unlikely to be meaningful statements to someone who's just learning to define their first function in Python
- Presentation + Took the time to make concrete examples and type them out for everyone to see. It was helpful how you left those up on the screen so that people could see several different examples together.
- Presentation - Think about tone when you give students instructions, and go for warmth over showing your frustration. I'd urge you to give them the benefit of the doubt because that will set a tone of respect and welcome. "Could you sit down? Yeah, now please" (Under breath) "Jeez" was pretty jarring
- Jess
    - Content +
        - I mean, the video is called "bad teaching video"
    - Content -
        - Doesn't explain functions, just says, "Don't worry about it"
        - Doesn't provide useful examples that connect to anything real
    - Presentation +
        - Livecodes, at least
    - Presentation -
        - He's irritable with the audience,
        - "even Excel users understand this" actively fosters imposter syndrome
        - Checks phone
        - Doesn't explain what's happening, just says "it does what you'd expect"
- Addi
    - Content: Keep doing- included live code, included example, Work on this for next time: Simpler concepts first before going to advanced content
    - Presentation: Keep doing- Usage of live monitor, talking through the steps being written Work ont his for next time: The instruction is too fast for any non experts to follow, same for example, usage of non-motivating language like just, too simple, comment about excel

users, these should be avoided, the process isnte xplained in detail as to why the content behaves that way, that can be improved, seemed a little frazzled at the start of the meeting; could work on directing attention in a more positive way

- Samarpita

Content + The instructor is teaching a concept that is important and will be frequently used by coders as they gain more experience.
Content - Might help to not make students feel concious while they are settling in, maybe say 'lets take a quick minute to settle down'.
Presentation + Live coding for class group is good
Presentation - While live coding, students might pay more attention if the teacher types out slowly and the font size can be increased to improve legibility.

- Tayaza
    - Content +
    - Content -: Does not explain new concepts
    - Presentation +: Live coding helps with understanding concepts
    - Presentation -: A little rushed; looking at phones; a little flippant with words like "just", "simple"; not using the mouse or pointer to direct attention; a bit impolite and impatient at the beginning.

Feedback on Yourself (25 min)

1. Join your breakout room with two other participants.
2. One of you should visit https://pad.carpentries.org/ and click "New Pad" to create an etherpad for taking notes in your group. Share this link with the others in your group so you can all take notes together.
3. Individually, spend 5 minutes preparing to teach a 90-second segment of the lesson episode you chose before the start of the training course.
4. Get together with your group and have each person teach their segment to the others. one person should be responsible for the timekeeping each time, and should keep the lessons strictly to 90 seconds. Use visual aids if available (Note: You do not have to live demonstrate during the lesson. There will be a chance to practice that tomorrow.) Keep a strict time limit of 90 seconds per person.
5. After each person finishes, the other group members should give them feedback. Then rotate the roles so that the person who just taught becomes the timekeeper, and the person who was not keeping time takes their turn teaching.
6. After everyone has given and received feedback, return to the main group and put everyone’s feedback about you into the main Etherpad.

- Sam
    - Content + Example well balanced : complex enough, not too complex. Really liked the pairing of examples with select * and select a particular column - just a nice bite-sized step to show how select can do more than one thing
    - Content - I was wondering what "plot_id" was as a column... I liked having the other columns because they made more sense to me in examples (yes! agree, it's a confusing example!)
    - Presentation + Example well balanced : complex enough, not too complex (yes, agree!)
    - Presentation - Having trouble coming up with a good response here -- I thought the presentation was right on target! I guess in a live coding demo you'd also have the

opportunity to look at the output together, which would be valuable

- Danielle
    - Content +easy to follow, info was presented well
    - Content -
    - Presentation + shared screen well and showed examples
    - Presentation -

    - Content +: it's nice you mention the benefits of keeping track of your analysis
    - Content -:
    - Presentation +: nicely paced!
    - Presentation -: it sounded a bit like you were reading from a script - but will surely improve with practice!
- Talisha
    - content +: good amount of content, gave motivation of why
    - content -: connection to previour work
    - presentation +: it was very clear, appropriate speed
    - presentation -:
- Yara
    - Content +: Great introduction, introduces basics; good for novices and those newly familiar with R; live demo
    - Content -: should've started with motivation for learning R
    - Presentation +:live demonstration ; good introduction; good pace
    - Presentation -: make fonts larger
- Robert
    - Content + The BMI example is awesome
    - Content - Missed the jump from variables to reproduceability
    - Presentation + I really like the pace!
    - Presentation - ....
- Jay
    - Content + Relatable introduction and example; important concept for new learners
    - Content - different numbers for the example
    - Presentation + Easy to follow, calm demeanour of trainer
    - Presentation -
- Al
- Andrew
    - Content +
        - -Reminding students of previous skills aquired
        - -Providing motivation for skills
        - -Live coding demo
    - Content -
        - -Finish lecture with summary (but we only had 90 seconds!)
        - -Provide specific example of new skill (again, only 90 seconds!)
    - Presentation +
        - -simple slides
    - Presentation -
        - -highlight how to open bash script
- David Ch

        - Content: - example too complex, + do not forget to say a fuction can be several lines long...
        - Presentation : +well motivated, -"local variable" not explained
    - Nate Qu
        - Content +: Reviewed what was covered yesterday, overview of what was happening today also good
        - Content -: maybe ask if someone had questions from the previous lecture; maybe too much content for the time slot?
        - Presentation +: Text nice and big :); clear and easy-going pace
        - Presentation -: went a little too quickly over everything to be covered today, I couldn't follow. Maybe just present 2 /3 main points so students can follow along.
    - Emmy
        - Content +shared options, explained all four parts; explained what a function is
        - Content -
        - Presentation +gave examples; showed a visual example of function
        - Presentation -
    - Lance
        - Content +Good use of pre-loaded code + very clear explanation of what it does, line by line, with broader context on why you're doing it that way. Great energy and thank you explaining what numpy is and why you import it as np
        - Content -
        - Presentation + Great introduction on how to upload data in python in order to apply all the knowledge we have done to analyse data thus far
        - Presentation -The only thing I would suggest is when you're getting started, it might be good to check in with students that they can see your font at that size, or if you should make it bigger.
    - Luca
        - content +: Great job starting out connecting with the previous lesson, going back to the lesson overview diagram. Included live coding and the lesson content. Terminal size
        - content -:
        - presentation +: Good speed, information was very clear
        - presentation -:
    - Nonkululeko :

Content + Nice intro to the shell!
-I like the way you immediately mentioned the differences between OSes, because that sets the students up for success.
-Provide good examples of what to expect to see what you first open the shell
-Explained how to get started with the directory.
Content -

- -Didn't get into to technical of an explanation of what Bash is (wasn't needed for this lesson), making this really accessible for novice users.
- -Explained how to get started with the directory.
- Presentation + -I also like the way you pointed out the dollar sign as the thing after which you will type, because it might be confusing for first time users.

Presentation --You spoke very clearly, but maybe a tiny bit fast at times.The only thing that might be confusing (depending on what came before this) is to say what a working directory to people who haven't heard the term before - but then, I cut you off, you were probably getting to that.

- Nate Jo

Content + :

- - Gave very relatable examples (Google Earth Etc).
- - Visual aids were very helpful.

Content - :

- - Some aspects of content may be too detailed and distract from main point (e.g. explaining the 30X30m resolution of land-use map, Vector graphics: if not important for this lesson say so, and point to later lessons where this may be taught.)

Performance + :

- - Really nice explanation of different raster data examples, clear presentation, nice visual aids.

Performance -:

- - Didn't give definition of GIS, RGB acronyms.
- Karin Content + Showing before and after of the hidden .git directory was helpful for understanding what init does.Content - presentation was short, so nothing to improve about content (maybe splitting up git init and git status Presentation + Living coding helpfulPresentation - Foreshadowing about branches was both good and bad - maybe not relevant here. Separate out explaining git status from git init
- Jess
    - Content + Accessible to first-time users, set students up for success
    - Content -
    - Presentation + started with activity, which provided inspiration
    - Presentation -
- Addi :
    - content +: Outlining what SQL was in the beggining was very helpful to provide perspective on what we were about to learn. Demonstrating an example and walking through the line of code provided concrete example
    - content -:
    - presentation +: Clear, good speed, having the online editor available to provide an example
    - presentation -:
- Samarpita
- Content + nice introduction!
- Content - "simply type `ls`"
- Presentation + great pace - perfect for understanding basic concepts
- Tayaza
    - Content+: love your analogy that a function is a verb (I will steal it) and your explanation of variables, ; putting into terms that students can understand
    - Content -:
    - Presentation +: calm, friendly and approachable; good to build off the lesson from yesterday
    - Presentation -: text could be bigger; make sure you're in the spot in your document when you start so you don't have to move around up/down to find the spot you're looking for (this is tough given the limited time!!)

Feedback Is Hard

Most effective when you give people guidance on what you expect from feedback.
Be specific on what could be improved.
Balance positive and negative feedback (i.e. compliment sandwiches).
Try to separate out what was taught vs how it was taught.
How to set ground rules: initiate feedback by approaching people and asking for specific feedback and how things could be improved / changed, encourage positive feedback as well (what am I doing well?).
If you tend to fixate on negative feedback, can ask a co-instructor to read the feedback for you and give you a summary i.e. act as a feedback interpreter.
Remember to be kind to yourself. Feedback is not always thoughtful.

Using Feedback (5 min)
Look back at the feedback you received on your teaching in an earlier exercise. How do you feel about this feedback? Is it fair and reasonable? Do you agree with it?
Identify at least one specific change you will make to your teaching based on this feedback. Describe your change in the Etherpad.

- Sam - select an example that's obvious and doesn't raise any questions as to what it means
- Danielle - Yes, I think that the feedback is fair and reasonable. Looking at the feedback, one specific change I will make is to become more familiar with the information that I am teaching so that I am comfortable enough with it that I don't have to reference my notes as often and sound like I am reading rather than teaching, or talking.
- Talisha - Yes, I really appreciate the suggestion to connect information to previous learning or a real life example. This is something that I would add to the beginning of each lesson and/or after returning from a break. One of the notes that was also shared was that summarizing what was previously learned would also be helpful if someone missed a portion of the previous lesson, to provide them with context. So this is also helpful for that as well.
- Yara - I thought the feedback was fair and reasonable. Next time, I will consider the size of text when presenting and the order of presenting things i.e. start with motivation.
- Robert - I thought it was fair and constructive criticism. I agree, there should be more cohesiveness between subtopics and the transition should be clear. Based on this feedback, I will work on the examples I use and make sure they can work together to illustrate the topic.
- Jay - Fair & reasonable & agree. This was a really useful exercise. I hadn't considered the output of my example (in more general terms, I hadn't "tested" my example either on myself or on someone else). In the future I'll get some tests in my lesson planning CI/CD platform
- Al
- Andrew- The first thing I will do in future is assess the domain knowledge of the learners, then tailor explanations to them, using more familiar, specific examples (rather than general, unfamiliar examples.)
- David Ch - My partners were very good at the exercise. In the future : be sure any new notion/concept/keyword is explained
- Nate Qu - I havent taught in almost a year so this was a bit of a challenge but i appreciate the feedback. I know that 90 seconds is an unrealistic amount of time to really get a handle on one's teaching so i take the feedback with that context in mind; that said i do probably try to cover too much too quickly and that is something that i need to be conscious of going into this summer's workshops; last year we got feedback telling us that we introduced way too many tools in too short of an amount of time so this summer we are just introducing one new tool each day during the 1st week;
- Emmy - really nice to have the reassurance from the "audience", and the feedback is reasonable and fair :) It's helpful because I wasn't confident if my example was worth putting in, but the feedback mentioned it as a positive, so I'll keep coming up with and work on improving those

- Lance: Feedback was very fair, and I was grateful for the kudos and affirmation as I was a bit nervous. The recommended improvement related to presentation and accessibility of materials as the font size was a bit too small in the screen share to be readable by learners in the workshop. For the future, I will check with the classroom to ensure that students can easily view the screen, and will also provide a link the chat to the Colab project document so that learners who are accessing content with a screenreader can also have full access to the content.
- Luca : the feedback I received was reasonable and given in a very constructive way. One thing I will work on in my teaching style is involving students more and make sure to be more interactive. Also try to give more "engaging" context such as a starting with concrete aplications.
- Nonkululeko:The feedback was fair and I agree with everythin they said,I should definetly work on my pace and be and do some depth explanation instead of just moving by
- Nate Jo -(1) Focus on high level concepts and build in details if I have time (2) Slow down and define acronyms!
- Karin Yes, the feedback was thoughtful and felt like it was based on careful observation of my 90s lesson. One change I'll make is to think more carefully about really separating out the bite sized portion of, say, a command that students are learning for the first time, and avoid or be cautious about foreshadowing more advanced ideas that I may be tempted to mention before students are ready.
- Jess Yes, the feedback was fair. Next time, I will paste a link in chat and give more time to allow people to explore the page I was showing, making it a real activity instead of a kind-of activity.
- Addi : Great Feedback, there was somethign about comments that I exp.ained only at the end of the lesson,got great feedback that it was unclear what that was in the beginning and got clarified inthe ends. This was a great point, next time, iw ill amke sure to address everythign on the screen up front so learners can focus and follow from the beginning
- Samarpita Feedback was encouraging. I will make a clearer concept map from the next time to ensure that I am not jumping from one concept to another but moving through smoothly for the learners.
- Tayaza: Yes, feedback was thoughtful. I'd ensure I do the necessary setup (e.g. font size and location of material) before begininning a lesson.

Key Points

- Like all other skills, good teaching requires practice and feedback.
- Lesson study is essential to transferring skills among teachers.
- Feedback is most effective when those involved share ground rules and expectations.

IX. Wrap-Up and Homework for Tomorrow
https://carpentries.github.io/instructor-training/12-homework/index.html

To prepare for tomorrow, please:

1. Prepare for the live coding exercises. If you haven’t already, pick an episode from an existing Software, Data or Library Carpentry lesson and read through it carefully. Tomorrow, you will use this to practice live coding for 3 minutes in groups of three. Your group members will comment on the delivery and content. Recommended episodes are listed here: https://carpentries.github.io/instructor-training/12-homework/.

Feedback (5 min)
Use this web form [https://forms.gle/6umQuXT5AdiNrQqM8] to tell us one thing you learned this

morning that you found useful, and one question you have about the material or something we could improve for next time. Do *not* put your name on the notes: this is meant to be anonymous feedback.

Reflecting on the Day (Optional Homework)
1. As with computational skills, people have a tendency to think of teaching as something you are "just good at" or not. However, teaching is a skill, and expertise develops with attentive practice. Examine the descriptions of "novice," "competent practitioner," and "expert." Where do you think you fall with regard to teaching? What have you learned about teaching? What are you aware of that you still need to learn?
2. Think over everything we covered in the last two days. On a piece of paper, write down something that captures what you want to remember about the material. The Trainers won't look at this - it's just for you. If you don't know where to start, consider the following list for a starting point:

- draw a concept map, connecting the material
- draw pictures or a comic depicting one of the day's concepts
- write an outline of the topics we covered
- write a paragraph or "journal" entry about your experience of the training today
- write down one thing that struck you the most

--Day 3--

X. Welcome Back

Questions from the feedback yesterday:
How can we better identify negative feedback to give to another instructor?

- https://carpentries.github.io/instructor-training/demos_rubric/index.html

What does one do if inadvertently demotivated a student?

- What suggestions do you have?
- (reflection from day-2 and further discussions)
- apologize and do not draw unnecessary attention
- make yourself reachable/accessible (breaks, etc.) and explain they will have more opportunity to try.
- Explicitly acknowledge what you did wrong / what you could have done better
- ID how you could have done it differently
- Take time to address definitions of technical terms
- Be intentional about avoiding the same mistake again
- Re-motivate student -> opportunity to make use of helpers and co-instructors to discuss future steps with them to address it during break
- i.e. ask helper to keep an eye on someone without making it obvious that they're getting special attention

And an area where we can improve:

- in the 2x2 matrix the terms "content -" or "presentation -" should be improved. I realized during the peer feedback exercise that saying out loud "content -" sounds very harsh, as it should not be intended as a minus, but more like an opportunity to improve.

Improve template for feedback (avoid +/-)

Key points from days 1 & 2:

- Instructors guide learners to construct the proper big picture (accurate mental model) of the topic rather than focus on details.
- Instructors rely on frequent feedback from learners to monitor their own presentation of the material.
- Instructors introduce a few concepts at a time to avoid cognitive overload.
- The best way to motivate learners? Show them how to do something they can immediately put to use and be enthusiastic about it.
- Teaching is a learned skill.

XI. Live Coding is a Skill
https://carpentries.github.io/instructor-training/14-live/index.html

Up and Down (5 min listing + 5 min discussion)
List some advantages and challenges of participatory live coding from both a learner's and an instructor's point of view.

- Robert
    - advantage - able to share the same experience, debug encountered issues together, can lead to quicker skill acquisition
    - challenge - unforseen issues slow down coding session
- Jay
    - Learner (adv) - follow along by coding in real-time, appreciate how everyone makes mistakes
    - Learner (chall) - may not make own mistakes since just copying from a screen
    - Instructor (adv) - can show rather than tell
    - Instructor (chall) - typos can be frustrating/flustering
- Al Learn+learn from miatakes in real time, learning -may be shy or nervous, instructor+ can see how learners respond real time, shows they are human and make /work through challenges real time, Instructor -mistakes can confuse learners
- Andrew
    - Advantages for learner
    - * Immediate practical experience
    - * can experiment and take notes live in line with the code
    - Challenges for Learner
    - * Following along in real time takes a lot of focus (probably helps with retention though)
    - * If tasks are complex or instructor moves too quickly, might miss the motivation or theory behind the tasks
    - Advantages for Instructor
    - * Immediately identify points of misunderstanding in the learners
    - * "Doing it live" leads naturally to errors, which can be turned to an opportunity for learning
    - Challenges for Instructor

    - * can't just wing it!
    - * requires a lot of energy to explain the steps (can't rely on pre-prepared material)
- Jess Learner: adv. sees the instructor reasoning through the process of coding, and possibly making mistakes. Learner/Instructor adv.: it slows the instructor down, making it better for the learner to absorb. Instructor adv.: helps keep a good pace and allows for an opportunity to really explain things step by step, and if there is a mistep, describe what led to the mistep and why it wasn't right. Instructor challenge: it's stressful to type live, especially if things go wrong.
- Addi : pros: Makes the thinking process accessible to the learner vs the final result only; showcases that everyone goes through dry runs and erros and it is normal part of teh process; Cons: If not designed with the audience in mind, content can be overhwlming especially if learner has had less experience than others in the class on particular conceps/technologies. Also makes it hard to follow for certain skills where you may not have the resources available for everyone, example: issuing basic commands to a big HPC cluster training.
- Samarpita Learner's perspective: (adv) if the setup on the instructors skill matches that of the learner, then it sends out the mesage to the learner that we will take this code one step at a time. This might be helppful for a beginner. (dis) If the learner is far away from the screen mirroring the instructor's computer, the font might not be legible to them. Instructors will usually have a good typing speed and going super fast cn be demotivating for the learner.
- Instructor's perspective: (dis) Looking back and forth between the computer screen and projected code may leave the learners feeling neglected
- David Ch
    - Benefit for teachers : do not forget some "obvious" steps which are not so obvious to anybody, direct proof of what they want to demonstrate
    - Benefit for learners : practice make them better appropriate the concept (active learning)
    - Challenge : find some simple enough example
- Nate Qu
    - Learner: + gets to see an 'expert' practice coding in realtime; - may feel reluctant to raise hand or ask for help
    - Instructor: + opportunity to make mistakes in front of learners lets them know that you (instructor) are human and can be more interactive than simply watching a video or talking about how to code; - may not be able to see how students are responding (body language, etc.) especially if being done online or virtually
- Emmy
    - advantages: for learners - it makes it more "real" to see that the code executes and produce expected results (not "magic"), and also to see instructors' thoughts processes and that they make mistakes too; for instructors - i guess it's easier for staying on the same page as the learner
    - challenges: for learners - they would also have to type along, and one can quite easily get lost (i iimagine) if the instructor is too fast; for instructors: can be stressful as you try to type code and talk at the same time, or when you make a mistake
- Tayaza
    - Advantages for learners: sees the "how" of the lesson; sees that it's okay to make mistakes
    - Advantages for instructors: gets to become better at the skill of live coding; slows the pace of the presentation to allow ideas to "sink in"
    - Challenges for learners: for some learners, it may be too much to try to get their heads around new concepts and coding at the same time.
    - Challenges for instructors: may be too focused on the coding and lose gauging engagement of learners (in face-to-face meetings, at least)

- Lance
    - Adanvantages:
        - Learner: get to see the process of building lines of code. Not simply copying and pasting large chunks of code. Get to see the instructor make typing errors, or possibly work through mistakes in code. It's possible they may see an error message happen to an instructor before or not long after it happens for them, and they have a framework for how to handle this based on what the instructor does.
        - Instructor: Forces the instructor to slow down a bit, and to be more methodical and intentional with content delivery. Helps in ensuring that learners across skill levels will have a better chance to keep up.
    - Challenges:
        - Learner:
        - Instructor: can be a bit unnerving to feel some pressure as mistakes both on your part and on the part of the software. I've struggled with this, especially in demonstrating coding during virtual trainings, as I am tempted to remove as many variables for the technology to fail as I can. However, this isn't great for my learners as they aren't getting to see me go through the process.
- Luca
    - Learner + : learns by doing, can experiment straight away and ask questions if encounters problems
    - Learner -: attention split, it has to focus both on understanding the concepts and typing correctly at the same time
    - Instructor +: Live coding is a lot of fun :D, helps the instructor setting the pace better, because if they just scrolled through the codes they could easily go too fast, having to type is a natural way to slow down
    - Instructor -: May feel more stressing, actually more work to do (you have to both prepare the material and make sure you can reproduce it on the fly)
- Nonkululeko

Learner's perspective:Benefits-Everything is clear and close on PC unlike in a physical class where the projector may be far and not necessarily see clearly

Instructor perspective:Disadvantage-No prior preperation,everything is done now and a person can be nervous ,Need to explain each step and explain right now if there are any errors which can be a chalenge if it is an error you are not familiar with

- Nate Jo

- Advantages:
- It communicates/provides a concrete, real world example
- It provides learners to practice in a structured environment
- Promotes active learning [if opportunities are provided for inclass or takehome activities to follow up]
- Disadvantages:
- -Easy to overly focus on operational aspects lesson. [Great for short workshops!!! For longer courses though, live coding is not always the best medium for communicating foundational knowledge.]
- -Often times live coding is "pre-baked', so learners don't experience the 'troubleshooting' aspects of hashing code!

- Karin + Slows the instructor down a little bit and so can give space to really discuss each command
- + can use to give learner time to follow along
- + learner can see every step of what they will do
- + can model making and fixing mistakes
- - can be harder for the instructor to notice other things while live coding -- e.g. student questions/reactions (especially over zoom when it's hard already)
- - takes practice to live code well, and it can be frustrating for learners to watch live coding if it's really choppy or full of mistakes
- - need to think harder about when/where to stop and give the big picture
- - if the live coding is really fast, it can seem intimidating
- - if you scroll past code that a learner was looking at, may need to go back, or have a separate resource where learners can see everything at once
- Sam:
    - Instructor + More engaging even as instructor than just talking, concrete hands-on examples make it easy to explain something
    - Instructor - Tech might not work, live-debugging even small typos is challenging, switching (visually or mentally) between script and live coding is difficult, helping multiple learners and getting the pace right is extremely challenging
    - Learner + More engaging than just looking at content, very satisfying to produce output immediately (almost instant gratification!), can see "how the sausage is made", can see approaches to live debugging (and see that "experts" make mistakes too), a-ha effect when something is run and shows results
    - Learner - Can be confusing to follow along due to context switching between watching and doing, difficult to focus or catch up when getting lost or stuck, can be frustrating, some learners might process more slowly
- Danielle - from the learners perspective, having a visual can be beneficial. The learner can also see when the instructor makes mistakes, which might help them see that it is okay to make mistakes. Also gives a kind of guidline for what is to be expected. Similar to setting the tone for an online workshop. For the instructor, sometimes it's easier to explain something when you have multiple ways of doing so. A verbal and visual explanation.
- Talisha
    - Learner practicing and trying to apply what is being learned in the moment. This gives an opportunity for the learner to discover mistakes or error that might happen while they are still with those who can provide support and feedback. Leave the training with some product. Creates an interactive experience. Possibly also slows down instruction at least a little to give time for coding and resolving errors.
    - Installation of programs, software/installation errors can create challenges for coding. If learner runs into many challenges may be frustrated. If any part of the workshop is missed, learner will not be at the same point as others (in not having the code that is being worked on)
- Yara . Adv for learner: Can see errors being addressed live and that it's ok to make mistakes. Adv for instructor: more fluid, can respond to student questions in the moment and address confusion immediately.

Instructor Training Bonus Modules: https://carpentries.github.io/instructor-training-bonus-modules/

Compare and Contrast (15 min)

Watch the two live coding videos as a group and then summarize your feedback on both in the Etherpad. Use the 2x2 rubric for feedback we discussed earlier.
In the videos, the bash shell *for* loop is taught, and it is assumed learners are familiar with how to use a variable, the *head* command and the content of the basilisk.dat unicorn.dat files.
(Tip: press the "CC" button to turn on closed captioning)

poor live coding: https://www.youtube.com/watch?v=bXxBeNkKmJE&feature=youtu.be

Content

Keep Doing This
Repeat the exercise once to show that the variable name is arbitrary; also allows for folks to see it again and absorb
The explanation of the lack of $ indicating prompt to keep typing is quite clear
Example seemed to show what he wanted to show -- that the variable name was arbitrary
gave a short context introduction at the beginning
Explained some background, that was helpful

Work On This For Next Time
Explain what each line in the loop is doing. +1 and introduce each command while typing+1+1
didn't explain the concepts that were introduced. For example did not talk (or if he did it was so fast that I missed) about string interpolation using $.
Explain why he got the error so that learners know what the error means and why it appeared+1
Think about starting with some motivation, if teaching loops for the first time -- why would we want to use a loop?+1+1
very high level content, may not be very aligne with novice learners+1
when using "animal" as variable name, do not type twice ".dat" and move it in the for loop !

Presentation

Keep Doing This
made a mistake - said "oops" corrected it and moved on+1+1
terminal and font size was readable +1+1
repetition+1
friendly tone
appropriate speed when talking
Approachable demeanor, using live coding instead of slides
Good explanation of the bash prompt changing and why

Work On This For Next Time
Explain what is being entered and why +1 +1
Notification popping up on desktop that can distract learners+1+1
Pause and make eye contact with participants.
The explanation was really fast between typing+1
Turn off noticifations on phone so as not to distract learners
Talked too fast in some spots
Declutter learning environment. Turn off notifications, turn off phone remove other windows from view +1
Ask for questions +1

Check to see if the learners have completed their attempt at the code+1
Need to look up from the screen to see if students have questions/ how they are responding
Could the mistake have been used as a learning opportunity?
No checks with the audience
the second version of the script was slightly different (didn't use the n flag, used semicon and colon) but did not talk about this
The shell window wasn't full screen - one can get distracted by the window behind...
Walk around the class to see if learners are keeping up, that ways instructor gets a break from coding too+1
too fast and doesnt walk through any intermediate steps ; also stays seated +1
Sitting down feels low energy and doesn't allow pointing at stuff, which would make it a lot easier to follow
explain how you solved your mistake+1
use the pointer to show what you talk about (when not typing)
---

good live coding: https://www.youtube.com/watch?v=SkPmwe_WjeY&feature=youtu.be

Content

Keep Doing This
Explaining why you got an error and how to fix it+1+1+1+1
Showing how the loop can be written on one line instad of multiple+1
Explaining how to tell when shell is waiting for more code+1
explains each part of the command he types+1
Able to clear error and explain why it happened
Lot of helpful approaches and tips and tricks based on experiences along with the main content
Replaces one variable name ("filename") with another ("x") to demonstrate that you can give any name to a variable (but also explains that it makes sense to use "filename" in this case)
Stopped to explain what code actually does
Gives a simpler starting point- saying you can do this another way, but realising there are steps that may be forgotten if you start out trying to do it the other way
explain how you solved your mistake

Work On This For Next Time
The purpose of the semicolon is not really clear - although i guess that may also distract from the main content of the lesson+1
Depending on the background of students, consider making time to remind them what head -n 3 is doing (so they don't get stuck on this when they're supposed to be seeing the broader structure of the for loop)+1

- give a real world example of when you would need to write a bash for loop

when using "animal" as variable name, do not type twice ".dat" and move it in the for loop !
Minor comment, but I find the names confusing (what's a basilisk? Also animal implies something about the content rather than it's a *file*)

Presentation

Keep Doing This
Standing up (rather than sitting)+1+1
Using a white background with dark font is easier to read than dark background with white font+1+1
Explaining in a easy to follow manner.+1

Pointing at the screen +1+1+1+1
Explaining what is being learned and the goal or expected outcome
Reading out what's being typed as it is typed+1+1+1+1+1
Repetition is nice, super helpful when learning new information
very clear and detailed explanation+1
"Trust me I've been there" in reference to mistakes that are common to make with these commands
Discussed code while typing
Saying out loud what's being typed makes it a lot easier to follow and "chunks" the content+1
Walkign along the screen to be present in teh context, goin through concepts step by step, friendly demeanor,a pproachable
Eye contact+1 (looking at the students so could see their reactions)
Explained error message clearly, and how to fix it, and used it as a chance to explain the other way of writing for loops.

Work On This For Next Time
Coding background could use more contrast to increase its visibility
could be more interactive, like asking students if they have questions, ask if they see the error he just made
room lighting and ambience
Could use the repeat demo of "you can use any variable" as an opportunity to ask learners what they think will happen (even rhetorically)
pause more, check in and verify learners understand
Moving pretty quickly through the lesson--need more time for learners to try commands on their own

Top Ten Tips for Participatory Live Coding in a Workshop

Stand up and move around the room if possible.
Go slowly (and "type out loud").
Mirror your learner's environment.
Use your screen wisely.
Use illustrations.
Turn off notifications.
Stick to the lesson material.
Leave no learner behind.
Embrace mistakes.
Have fun!
A paper where you can read more about these tips: https://doi.org/10.1371/journal.pcbi.1008090
The fortnightly *Carpentries Clippings* newsletter includes more teaching tips from our Instructor community. See the archive and subscribe here: https://carpentries.org/newsletter/

Practice Teaching (25 min)
Teach 3 minutes of your chosen lesson episode using live coding to one or two fellow trainees, then swap and watch while the other person(s) live codes for you. (For this exercise, your peers will not "code-along", but will instead observe and give feedback.)

Explain in advance to your fellow trainee(s) what you will be teaching and what the learners you teach it

to are expected to be familiar with.

Do not record this exercise. Give each other feedback using the 2x2 rubric we discussed previously and enter the feedback you received in the Etherpad.

This exercise should take about 25 minutes.

- Robert
- Al
- Andrew
- Jess
- Addi
- Samarpita:
    - Content+: Good intro, great setup for a novice learner; good to use the visualization of the dataframe in introduction; Great explanation of what a dataframe is
    - Content improve Wasn't clear if the learners were using the jupyter GUI or something else (honestly, not sure if this is important); maybe think about making the fonts larger in your RStudio; maybe restructure content so you actually get to livecode within the 3 minutes.
    - Presentation+ Presentation was clear and legible, ; good to walk through what a ‘tibble’ is as that’s not totally obvious for novices
    - Presentation improve: possibly smoother transition between browser and RStudio;
- David Ch
    - Content- : too long, It must reshape to fit in 3 minutes.
- Nate Qu
    - Content+:
    - Content improve
    - Presentation+
    - Presentation improve
- Emmy
- Tayaza

- Content+ Clear learning objectives; simple, beginner level is appropriate for this audience
- Content Improve: Don't know if the learners were acquainted with the way R GUI is set up.;
- might use first few seconds as a reminder about what was done during the last lesson
- Presentation + Live coding was well done; good work speaking as you type (typing out loud)
- Presentation improve: might clean up the R studio environment so only the panels needed to complete lesson are visible to students

- Lance
    - Content:
        - Keep doing this:
            - Getting oriented
            - explained some commands and what they do
            - Good explanation of what you're doing as you type, saying what you're typing
        - Something to try next time:
            - changing your terminal name! Be careful at the difference between files and directories

- Presentation:
    - Keep doing this:

                - great energy
                - Introduction
                    - talking about what is being typed
                    - using inclusive language enter/return
            - Something to try next time:
                - explanation of why entering code or code structure
                - You could slow down a little bit (though it's hard with 3 minutes)
- Luca
    - content +
        - Really like the use of the use of the illustration
        - Nice analogy of staging area = post office
        - Explaned everything thoroughly
    - content -
        - Does the vim shortcut work on windows and mac?
    - presentation +
        - Good pace, Confident.
        - Typed out loud really well.
    - presentation -
        - Text too small both for illustration and shell
- Nonkululeko -Need to improve on the pace not too fast,next time use bullet point slides not fully written and reading from it

content+:Great explanation and live coding ,explained why I got an error ,Good introduction to Unix Bash and how to use it in different operating systems

- Nate Jo:
    - Content+: clear background and goals
    - Content improve:
    - Presentation+: Nice step-by-step explanation of commands as they are used.
    - Presentation improve: Maybe explain what the RUN button does?
- Karin
- Content+ Clear learning objectives
- Content Improve:
- Presentation + energy
- Presentation improve: Shorten my terminal prompt so that it's not distracting, can slow down explanations
- Sam
    - Content
        - Keep doing
        - Introduced skill at the beginning, and then provided example
        - Many repetitions of the same topic/command with slight variations
        - Opportunities
        - Provide real world examples of how/why you would use query
    - Presentation
        - Keep doing
        - Clearly explained as you typed
        - saying what you're typing - you are amazing at this!

        - Opportunities
        - In introduction of skill, describe range of tasks you are going to display in example
- Danielle - explain the difference between R and Rstudio.
- Talisha
- Yara

content +: appropriate amount of content,
content -: not sure whether variables detour was necessary, explain that run works only on line selected / line that cursor is on

presentation +: very clear, appropriate pace, explained all shortcuts.
presentation -: type live as much as possible

Key Points

- Live coding forces the instructor to slow down.
- Coding-along gives learners continuous practice and feedback.
- Mistakes made during participatory live coding are valuable learning opportunities.

XII. Preparing to Teach
https://carpentries.github.io/instructor-training/15-lesson-study/index.html

Learner Profiles

João is an agricultural engineer doing his masters in soil physics. His programming experience is a first year programming course using C. He was never able to use this low-level programming in his activities, and never programmed after the first year.

His work consists of evaluating physical properties of soil samples from different conditions. Some of the soil properties are measured by an automated device that sends logs in a text format to his machine. João has to open each file in Excel, crop the first and last quarters of data values, and calculate an average.

Software Carpentry will show João how to write shell scripts to count the lines and crop the right range for each file, and how to use R to read these files and calculate the required statistics. It will also show him how to put his programs and files under version control so that he can re-run analyses and figure out which results may have been affected by changes.

See examples for Software Carpentry: https://software-carpentry.org/audience/
and for our lesson on Jekyll/GitHub Pages:
https://carpentries-incubator.github.io/jekyll-pages-novice/#target-audience

Reverse Instructional Design (and Preparation!)

1. Identify target audience
2. Define objectives and outcomes
3. Choose/prioritise exercises and other formative assessments that will tell you when those objectives have been reached

4. Plan how you will guide the group from each assessment to the next

If you are interested in the reverse design process for creating new lessons, check out:

- The Carpentries Curriculum Development Handbook: https://cdh.carpentries.org
- The Carpentries Lesson Development Study Groups material: https://carpentries-incubator.github.io/study-groups/
- The chapter, A Lesson Design Process, from Teaching Tech Together
    - english: http://teachtogether.tech/en/index.html#s:process
    - spanish: http://teachtogether.tech/es/index.html#s:process

Key Points

- To teach effectively, you have to know *who* you are teaching.
- Good learning objectives communicate the intended effect of a lesson on its learners.
- A good exercise provides useful guidance to instructors about next steps needed in teaching.

Collaborative live coding resources:
A project to follow if you teach with/use JupyterLab: https://github.com/jupyterlab/rtc
Google Colaboratory allows sharing so that multiple users can work on the same notebook:
https://colab.research.google.com/notebooks/intro.ipynb

XIII. More Practice Live Coding
https://carpentries.github.io/instructor-training/17-performance/index.html

Round Two (25 min)
First, have a look at the rubric that is given to trainers as a suggested framework for evaluating the online teaching demonstration sessions that are part of instructor checkout: https://carpentries.github.io/instructor-training/demos_rubric/. Does this rubric make sense? Take a moment to think about things you'd like to do differently with your next live coding practice. This is also a good time to ask questions about teaching demonstration.

Next, get back into the same groups you did your live coding with. Take turns re-teaching your chosen live coding session, making sure to incorporate changes based on the feedback you received and any new ideas based on reading the rubric. Give feedback to each other using the rubric this time.

When you are finished, add some thoughts on this process to the Etherpad: What did you change? Did it work better or worse with the change? How might you do it if you were to teach it again?

https://cuckoo.team/ttt-2021-05-05

Write a few highlights about your demo. You can use the template below (not mandatory!).

Content

Keep Doing This

Try This Next Time

Performance

Keep Doing This

Try This Next Time

Add a + here if you need five more minutes:

- Robert Able to improve transitions between subtopics and clearly show download process to student
- Jay Felt alot better about this one :) Slowed down, used better examples, explained technical terms
- Al + think additional time is needed, maybe set aside 7-10 minutes to get into lesson more
- Andrew Didn't get to finish the first time so got some more feedback second time round. I can make more of the motivation for what we were doing, to help get learners enthusiastic about the topic. I should also think about addressing problems with learners following along.
- Jess
    - Explain error, why it showing up and specifying that users may not see it+1
    - Outlining the structure of code
    - Explaining what is being tyes and why
    - Commenting in code
- Addi : I think there was improvement wrt what would be helpfulto know if we expect learners to do live coding. Also got great feedback about how to make sure learners are following along and not running into errors. Did better; but also good feedback about setup of the leson; screen resolution/brightness; Making sure to checkin with learners before moving on to the next comcept.
- Samarpita: Great walkthrough of code output, handled poor internet connection well. ; cool under pressure when the connection got bad - didn't panic or get nervous; assuming the connection is working you might want to get right into the live coding and cut back on the description/background piece; otherwise great job!!
- David Ch: need to better control time
- Nate Qu: Clean code formating, detailed explanation of steps ; I felt more pressed for time in the 2nd round and noticed I was skipping some of the elements when I was typing outloud (punctuation, quotes, etc.); didn't tell learners to hit run at top of jupy notebook to see their output
- Emmy : explained some of the terminology that was unclear before / used alternatives; more guidance on how to use the copied URL; I said many more "just" as I got distracted; need to remember to check if learners are following OK
- Tayaza: Content: Followed the Carpentries curriculum well and explained each step really well. ; good work on the details as you talk while typing (period, underscore, parenth. etc.); good explainer on tidyverse; might want to walk through what the panels in RStudio are rather than say upper right as participants might have environment/panels set up differently that you do
- Presentation: Good typing speed for learners to keep up. Maybe explaining the R interface will help learners.
- Lance: I felt more nervous about the second time, but my reviewers felt that I had improved. It

was really helpful to do the same lesson twice--I had a few things that I needed to do better from the first round that I felt I was able to accomplish with the second. For my lesson, I shared an introduction to file directories in the Unix Shell, and we talked as a group for some possible visualizations to include for the next time to more clearly illustrate the levels in file directories. The new rubric was helpful for me, as it gave some clear directives on best and not so great practices.

- Luca ; felt more confident, used bigger font and pictures, improved font text.
- Nonkululeko :Improvement in pace and Better introduction into Unix Bash and why we use it .Not showing Long written notes which makes it better to follow.
- Nate Jo I worked on better introducing the topic and providing foundation for skillzzzz presented
- Karin Worked on slowing down my explanations relative to last time - great to get to practice this! And nice to practice it on just a bite-sized 3min chunk.
- Sam - feedback was very helpful, prepared a better example/motivation and explained a component I glossed over the first time round. Being familiar with the content also helped.
- Danielle - I feel like I did worse this time!! But with practice and getting to know the material better I think I will improve with time.
- Talisha : I worked on improving the introduction and flow. I think it went better because I had more time to prepare and think about what would be talked about before talking about it. I really didn't know what I was teaching about but had some time to look more and so I think having an understanding made it better the next time.
- Yara : Did better this time. Took time to write out code and explain every part of the function, Explained the motivation behind using this function, encouraged students to ask questions (reminded them how they can ask questions).The rubric was helpful to identify what areas to focus on. I found this second practice helpful.

XIV. Managing a Diverse Classroom
https://carpentries.github.io/instructor-training/18-management/index.html

What Are the Challenges? (5 min)
What are some of the challenges you might expect when teaching learners with a broad range of expertise? Discuss with a partner and put your thoughts in the Etherpad.

Room 1 -

- How do we manage the eperson who brings their personal problem and continues to push it forward?
    - Possible solutions are helper stepping in for a side conversation, asking to talk more at the break, or noting the content will not be convered in the worksop and why
- When someone isn't understanding what is being typed and they ask for each line to be explained:
    - taking some time to explain to them especially during the break because we don't want them to get left behind

Room 2 -

- When one workshop participant isn't clear on even, say, what spreadsheets are
- Advanced folks like to lead you down rabbit holes -- ask specific questions
- Suggestion for categorizing classes into intermediate, advanced etc and provide helpful criteria for learners to be able to assess that

Room 3 -

- Suggestion to "employ" the apparent experts to assist the novices (i.e. encourage peer learning and assistance). At the very least, invite them to share their experience of a particular topic to engage them and gain new perspectives.

Room 4 -

- finding the right pace, or cadence
- reading the room to see who is understanding and who isn't from visual cues (can be easy as intructor to just notice the people who are understanding well and then move too fast!)
- encouraging people to speak up if they are lost
- good idea is to put people into groups mixing up someone who is more advanced with someone who is a novice so that they can learn from one another

Room 5 -

- Gauging the level of individual learners is challenging as an instructor ("everyone is nodding")
- Handling advanced learners with in-depth questions is difficult, can refer to break time to cover more advanced questions (like Toby does ;) - can also be used to catch up learners that are struggling
- Have clear categories and rubrics for beginner/intermediate/advanced courses - "if you have skill X Y Z you're in this category" - better than self-assessment just based on the title

Room 6 -

- advertising to ensure we attract novices: state clearly what one can expect to learn, and prerequisite - if you already know all the points in the outcome, then you really shouldn't attend (;
- expert learners could correct instructors
- disegnaged advanced learners
- extra helpers to help more novice learners
- it's hard to balance and find the optimal pace and content if you have a big diversity in skill level
- (thanks for taking these notes Emmy!) (welcome! Please feel free to edit/amend, not sure if i captured everything :))

Room 7 -

- Its difficult to establish a common luangage with people from diverse discplines/experiences
- Difficult to teach to different levels
- When dealing with learners with different levels of expertise it is difficult to find examples and models to teach the same concept that would appeal to learners on different sides of the expertise spectrum.

- Jargon busting: https://librarycarpentry.org/lc-overview/03-jargon-busting/index.html

https://cuckoo.team/ttt-2021-05-05

Code of Conduct Violations
https://docs.carpentries.org/topic_folders/policies/index_coc.html

Always report violations of the code of conduct/ at the very least discuss with code of conduct committee.
Code of Conduct Committee's transparency reports which include a summary of incidents reported and handled:
https://github.com/carpentries/executive-council-info/tree/master/code-of-conduct-transparency-reports

Homework
To prepare for tomorrow, please:

1. Look through these checklists to learn what hosts and instructors need to do to prepare for a workshop (https://docs.carpentries.org/topic_folders/hosts_instructors/hosts_instructors_checklist.html), and read over the differnce between a centrally-organized and self organized workshop at the bottom of this page (https://carpentries.orgn/workshops/).

Feedback (5 min)
Use this web form [https://forms.gle/6umQuXT5AdiNrQqM8] to tell us one thing you learned this morning that you found useful, and one question you have about the material or something we could improve for next time. Do *not* put your name on the notes: this is meant to be anonymous feedback.

-- Day 4 --

Homework: feedback for us to provide some more advanced warning about homework tasks involved in the training, to allow participants to set time aside for completing these.

XV. Checkout Process
https://carpentries.github.io/instructor-training/20-checkout/index.html

Yesterday we asked you to read some resources about the logistics of teaching and running Carpentries workshops. Please add your questions about logistics and preparation to the Etherpad. We will answer these questions in the Etherpad during your work time and will return to this list later today.

This activity should take about 5 minutes.

- Robert
- Jay Can we/should we be on the Carpentries Slack?
    - Yes! See the "Get Connected" section below.
- Al
- Andrew Is it expected/required that we should assist with a workshop run by a more experienced instructor, before attempting to run one ourselves? I don't imagine I'd be comfortable jumping straight in anyway, but just wondered if there was some rule about it.
    - It is definitely a gret idea to run a workshop with a more experienced instructor. For Centrally-Organised workshops we always place an instructor who has taught two or less workshops with an instructor that has taught more than three. For Self-Organised this is where slack and community discussions are great. You can meet other instructors with more experience.
- Jess
- Addi How do we set uo bullet point 3- the demo with an instructor. Is it any caroentry lesson of our choice? Is one lesson one subullet that has a page dedicated under a topic or is it one topic?
    - the community discussion that we will participate in today, would that qualify step 2 or do we need to participate ina nother one besides this one
    - what is considered substantive feedback?
    - In bullet point 1 , do we complete all the sub bullets or one of them?

- Samarpita 1) How many helpers are recommeded for a workshop size of 15 - 20 students? Does it change with the course being taught?
    - For in-person teaching, I like to have at least one helper for every eight learners. For online, I aim for one helper for every five. Our recommendations for online workshops also include guidance on specific roles to give to some of your helpers: https://carpentries.org/online-workshop-recommendations/
- David Ch About the centrally vs self-organized workshops, don't you plan some evolution and/or official adjustments for theONLINE flavor of them ?
    - See https://carpentries.org/online-workshop-recommendations/ & https://docs.carpentries.org/topic_folders/hosts_instructors/resources_for_online_workshops.html & https://carpentries.github.io/instructor-training-bonus-modules/
- Nate Qu How has the role of the 'Helpers' changed now that most of these workshops have transitioned to online? I can picture them hovering around the back of the room in a face-to-face setting and going around from participant to participant answering questions as the 'Trainer' leads a session from the front but I'm curious how we position and make best use of the helpers if we're doing this all viturally? Any feedback would be much appreciated.
    - Helper roles are discussed in the recommendations for online workshops. It has definitely changed! https://docs.carpentries.org/topic_folders/hosts_instructors/resources_for_online_workshops.html
    - Helpers can be great for answering questions in the chat, or notifying the instructors of questions in the chat. Sometimes it is hard to see the chat when you are screen sharing. I also know that helpers have been utilised for people who are stuck by going into a breakout room and helping the person.
- Emmy What should helpers know before helping in a workshop - especially this (joining as a helper) comes across as potentially one of the first touchpoints for folks into the Carpentries training programmes, how can hosts/instructors help manage expectations and concerns (e.g. about prerequisites in taught material/area, fear coming from unfamiliarity with format, etc)?
    - See the info about helper roles in the recommendations for online workshops: https://carpentries.org/online-workshop-recommendations/
    - and also our Helper checklist: https://docs.carpentries.org/topic_folders/hosts_instructors/hosts_instructors_checklist.html#helper-checklist
- Tayaza: What's the ideal number of learners for an online training?
    - Generally, it depends a bit on the number of helpers you have (fewer helpers=fewer learners). But we recommend having a class size of 20 learners or fewer.
- Lance -for the most part, I felt comfortable with the lists-I see my questions repeated by a few other colleagues here.
- Luca I noticed that institutions are asked to refund for instuctors traveling. Does that mean that instructors should only apply to volunteer to workshops nearby (say same continent) unless explicitly asked by organisers if they don't find instructors nearby? I am asking because I would assume that organisations requesting a workshop would prefer instructors nearby to save money. (well that's less relevant now as everything is online) Moreover, are instructors who browse a list of workshops and choose where to go or are organisers who contact instructors and ask to come teaching?
    - https://carpentries.topicbox.com/groups/instructors is the list where teaching opportunities will be shared for centrally-organised workshops.
    - We do try to place the closest instructors to a region at a workshop first. There will be lots of opportunities sent out by email through the instructor emailing list. SOmetimes also on

the instructors slack.

- It was mentioned somewhere that we should teach a workshop within one year from getting the certification. Shall we also notify someone after having taught the first workshop?
    - Part of the process of a workshop being organised is that the workshop webpage URL is shared with The Carpentries team and logged in the database. The metadata from that webpage (date, instructors, location, helpers, etc) is harvested from the site and stored in the database too, so The Carpentries will know when you have taught a workshop.
- Nonkululeko
- Nate Jo What are the boundaries for running workshops within an organization? Specifically, within the university setting, can I require my students to take a workshop as part of a unversity course that is being taken credit? [I think this may be a bit sticky because they are paying for course credit from the Unviersity.]
    - I am genuinely not sure about the answer to this one! maybe one of the other team members will know. But, if not, this is an ideal question for the Slack/mailing list....
    - I would say ask the University in this case. I know that there have been people who have attended workshops as a pre-requisit.
- Karin What are the norms and expectations around participants paying for a workshop. I saw "decide if you want to charge a workshop fee" in the checklist for the centrally organized workshop, and that this could offset the fee paid to the Carpentries for the central organization. Do self-organized workshops also sometimes have a participant fee, or are they open? Also, do the instructors/organizers get to choose any combination of carpentries lessons/units to teach within a workshop, or is there more specific structure?+1
    - Hope we covered this in the sessions and notes below. Please reach out if you are still unsure.
- Sam I'd love just a brief voiceover of the centrally organized vs self-organized workshops and some examples if possible, or maybe phrase it from the perspective of an instructor (what types of things will I be doing/can I expect)+1
    - Hope we covered this in the sessions and notes below. Please reach out if you are still unsure.
- Danielle - how soon should I fill out a workshop request form before I want to hold a workshop?
    - It depends on the type of workshop
    - Centrally-Organised workshops need to be requested a minimum of 2-3 months before the requested start date(s) in order to provide enough time to locate and connect with instructors
    - Self-Organised workshops - Please notify us as soon/early as possible
        - For genomics workshops we can provide AWS Instances if needed. These take some time to setup so we ask that you notify us of the workshop well in advance (1 month +)
        - As long as we are notified in advance, we try to send out the link for instructors to view the survey results 2-3 weeks before the workshop. I would say let us know about the workshop even still at least 1 month or more in advance.
- Talisha
- Yara - Will we get the opportunity to attend a workshop we want to teach?
    - To have an opportunity to observe before you teach the workshop yourself? I would recommend reaching out to the hosts/instructors of an upcoming workshop to ask if you can attend as a helper (people are almost always delighted to get more helpers!). You can find upcoming workshops at the bottom of The Carpentries website https://carpentries.org/ If I am misunderstanding the question, and you were asking whether you would find opportunities to teach at workshops, we mentioned the Instructors mailing list

https://carpentries.topicbox.com/groups/instructors earlier which is where teaching opportunities are shared for centrally-organised workshops.

One of the later exercises will involve setting up a workshop webpage with GitHub Pages. We would like to group you according to the relative levels of experience you have with this kind of task. Please help us by answering the two questions below.
1. How often do you use GitHub? (Daily/Weekly/Monthly/Yearly/Never)
2. Have you ever modified a website built with GitHub Pages before? (Yes/Not sure/No)

- Robert
    - 1. Monthly
    - 2. Yes
- Jay
    - 1. Weekly
    - 2. Yes
- Al
    - 1. weekly but novice
    - 2.no
- Andrew
    - 1. Weekly
    - 2. Yes
- Jess
    - 1. Yearly
    - 2. Yes
- Addi
    - 1. Monthly
    - 2. No
- Samarpita
    - 1. Weekly
    - 2. No
- David Ch
    - 1. Monthly
    - 2. Yes
- Nate Qu
    - 1. Monthly
    - 2. No
- Emmy
    - 1. Weekly
    - 2. Yes
- Tayaza
    - 1. Weekly
    - 2. No
- Lance
    - 1. Never
    - 2. No
- Luca
    - 1. daily
    - 2. yes
- Nonkululeko

        - 1. Never
        - 2. No
- Nate Jo
    - 1. Monthly
    - 2. Yes
- Karin
    - 1. Weekly
    - 2. Yes
- Sam
    - 1. Daily
    - 2. Yes (but a while ago)
- Danielle
    - 1. Yearly (very rare, not extensively)
    - 2. No
- Talisha
    - 1. Daily
    - 2. Yes
- Yara
    - 1. Never
    - 2. No

Checkout process

Application Form
- https://amy.carpentries.org/forms/request_training/

Instructor Checkout

1. Lesson contribution: Make a contribution to any eligible repository: https://carpentries.github.io/instructor-training/checkout/index.html#eligible-repositories

- find a list of issues with the "help wanted" label from these repositories on this page: https://carpentries.org/help-wanted-issues/
- Some information to help with the Lesson Contribution:
- Posting your contribution on the lesson Repo
    - Locate and navigate to the lesson page you'd like to contribute to
    - At the top or bottom of the page you should see a link that says 'Improve this page' or 'file an issue on GitHub'.
    - Clicking that link will take you into the GitHub repo for the lesson where you can navigate to the issues tab and create a new issue to share your comments.
    - After making the contribution please send the link to checkout@carpentries.org
- Sending your lesson contribution via email
    - Locate and navigate to the lesson page you'd like to contribute to
    - Email your contribution to checkout@carpentries.org, please include in your email the url to the lesson page
- If you have contributed to a lesson in the incubator:
    - Email your github ID and a link to the lesson repository to checkout@carpentries.org
    - If you have questions about lesson development in The Carpentries, please email tobyhodges@carpentries.org or schedule a call at https://calendly.com/tobyhodges/

2. Community Discussion: Take part in a community discussion

- this event is 1 hour in length,
- sign up at https://pad.carpentries.org/community-discussions
- After the discussion session the host will report attendance to checkout

3. Teaching Demo: 5 minute Live demo, also one hour, up to five participants doing 5 min demo + 5 min to get feedback, sign up at https://pad.carpentries.org/teaching-demos

- For the demo, you should choose a lesson and prepare to be able to teach that whole lesson. The Trainer running the session will choose a starting point for you from within your chosen lesson. We have a list of good and bad starting points for lessons, so we will not ask you to start from anywhere in the lesson that requires a large amount of setup/has a lot of dependencies from earlier in the lesson (a long git history, a dataframe that has gone through a lot of previous processing steps, etc)
- After the demo the host will report attendance to checkout

A note from Talisha: connection info is always at the top of the Etherpads for the community discussions and teaching demos. Also recommend to create a calendar event/reminder when you sign up, so that you don't forget/miss it.

Checking Out Review with Questions & Answers (5 min)
In small groups, read and discuss one of the three checkout procedures. Make notes in the etherpad and when you're done, report back to the full group about the requirements for that stage of the process. What questions do you still have about the checkout process?
1. are there any lessons for teaching demo that do not involve live coding?

- yes, e.g. the OpenRefine lessons. but they still include teaching by live demo, which is a crucial part of the teaching style we want to teach here and assess in these teaching demos

2. What is "adequate participation" in the community discussion?

- ask a question, comment during a discussion, generally unmute and contribute to the conversation. Something more than "lurking" on the call with your video and audio switched off.

3. We need to email checkout@carpentries.org for the lesson contribution. Do we need to do this for the teaching demo and/or community discussion too?

- No, the teaching demo/discussion host will take care of this.

4. What happens after the steps are completed? Do I need to email checkout@carpentries.org to tell them that I am done?

- No, we will know this automatically, Talisha will email to acknowledge that the steps have been completed (congratulations!), and you will receive another email ~1 week later with your Instructor Certification. Your certification will be for one of the three Lesson Programs (Software, Library, or Data Carpentry). This will be based on the lesson you chose for your teaching demo. But Talisha can adjust this according to your preference if you email checkout@carpentries.org. Regardless of which Lesson Program is stated on your certificate, it is valid for teaching *all* types of workshop.

Schedule a Discussion or Demo (5 min)
Visit the discussion Etherpad http://pad.carpentries.org/community-discussions to sign up for a session. If the session you would like to attend is full, contact the discussion host and co-host to ask if you can

attend.

If you'd prefer to do your teaching demonstration before your discussion, visit the demo Etherpad and sign up there https://pad.carpentries.org/teaching-demos

done: ++++++++++++++++++++

Break --> XX: 23

A Brief History

Timeline: https://carpentries.github.io/instructor-training/fig/SWCDChistory.png

Software, Data and Library Carpentries

SImilarities & differences between the three Lesson Programs: https://carpentries.github.io/instructor-training/fig/carpentries-venn-diagram_20200904.svg

What is a Carpentries Workshop?

All Carpentries lessons are licensed "CC-BY:" Creative Commons Attribution license. This means they can be freely re-used, modified, and apadted as long as you credit the original source. (Check the License page of any lesson to see the details.)
But a workshop can only be called a Software/Library/Data Carpentry workshop if it (https://carpentries.org/workshops/#workshop-core):

- covers the core topics
- is taught by at least one certified Instructor
- uses the standardised pre- and post-workshop surveys (links provided by The Carpentries team)

Data Carpentry workshops should contain the following:

- A lesson on data organisation
- Any three other lessons from in the same domain in the Data Carpentry curriculum

Library Carpentry workshops should contain the following:

- Any three lessons from the Library Carpentry curriculum

Software Carpentry workshops should contain Software Carpentry lessons on:

- The Unix Shell
- Version Control with Git
- A programming language (R or Python)

https://datacarpentry.org/lessons/

https://librarycarpentry.org/lessons/#core-curriculum
https://software-carpentry.org/lessons/

Questions:
1. If you want to run a Data Carpentry workshop, you pick one curriculum and teach all the lessons? Or can you mix and match?

- As long as you teach the lesson on Data Organisation, you can then choose any three other lessons from the same curriculum (i.e. Ecology, Geospatial, Genomics, or Social Sciences).

2. Can a workshop be offered over multiple shorter sessions instead of consecutive days?

- Yes, scheduling is flexible for Self-Organised workshops. Scheduling is slightly more rigid for Centrally-Organised but can still be spread out over multiple consecutive days.

Self-organised workshops & Centrally-organised workshops
Self-organised workshops: you (or the host) organises everything, sends the workshop webpage URL (see the next section...) to team@carpentries.org. Workshop will then be listed on our websites, included in our records, and pre- and post-workshop surveys links will be provided.
Centrally-organised workshops: the workshop host pays a fee to The Carpentries, who then find Instructors, set up workshop webpage, handle registrations, etc. The host should provide the venue and take care of advertising the workshop.

Instructor sign-ups for centrally-organised workshops: sometimes the Carpentries Team will send out a call for instructors (to the "instructors" mailing list, link coming soon below), giving details of venue/host, dates, skills/experience required, type of workshop, expected audience etc. Certified instructors can then sign up to teach at that workshop.

Practice With Carpentries Infrastructure (25 min)
Go to the workshop template repository (https://github.com/carpentries/workshop-template) and, as a team, follow the directions to create a workshop website using your local location and today's date. You should make one site for your entire group, listing your group members as Instructors for the workshop. The person who imports the repository template and edits the files should share their screen while they go through the process. You should discuss which group member will do this part e.g. it might be most beneficial for one group member to use this as a chance to get more experience with GitHub, while other, more experienced member(s) of your group guide them through the steps.
Replace the placeholder URL below with the link for your workshop website.
Note: Sometimes web browsers will cache the workshop webpage, so when you make changes in Github, they don't show up on the workshop webpage immediately. Two ways to avoid this are to use a "private" or "incognito" mode in your web browser or by following the instructions at https://en.wikipedia.org/wiki/Wikipedia:Bypass_your_cache to bypass your browser cache.

Group 1: https://jaydesl.github.io/2021-05-06-group1-online/

Group 2:https://lancemsimpson.github.io/2021-05-06-carpentries-trainer-online/

Group 3: https://jessctanugi.github.io/2021-05-06-Breakout-Room-3/

Group 4: https://yararaa.github.io/2021-05-03-montreal-online/

Group 5: https://emmyft.github.io/2021-05-06-Amsterdam/index.html

Get Connected

There are many ways to get connected with the Carpentries community:

- Our websites are:
    - Software Carpentry https://software-carpentry.org
        - Blog https://software-carpentry.org/blog/
    - Data Carpentry http://www.datacarpentry.org
        - Blog http://www.datacarpentry.org/blog/
    - Library Carpentry https://librarycarpentry.org
        - Blog https://librarycarpentry.org/blog/
    - The Carpentries http://carpentries.org/
        - Blog http://carpentries.org/blog
        - Get involved (community overview) https://carpentries.org/community/
- Our lessons are hosted on GitHub; contributions to them and discussion of changes happens via GitHub pull requests and issues, and the lessons are published using GitHub Pages. More details are given below:
    - Data Carpentry on GitHub https://github.com/datacarpentry
    - Software Carpentry on GitHub https://github.com/swcarpentry
    - Library Carpentry on GitHub https://github.com/LibraryCarpentry
- The Carpentries share public discussion lists that host everything from lively discussion on teaching practices to job postings and general announcements:
    - https://carpentries.topicbox.com
    - **The main "all-community" discussion/announcement list: https://carpentries.topicbox.com/groups/discuss**
    - **The "instructors" list where the team posts calls to sign up to teach at centrally organised workshops: https://carpentries.topicbox.com/groups/instructors**
    - **There are also many lists for regional communities. The names of these groups typically begin with "local-" e.g.**
        - local-uk
        - local-us-midwest
        - local-africa
        - local-aunz
        - local-nordic
        - etc
    - See https://carpentries.topicbox.com/groups for the full list of groups to join, and contact team@carpentries.org if you would like to discuss setting up a TopicBox list for a new group
- We publish a joint newsletter. https://carpentries.org/newsletter/
- Host monthly community calls and weekly instructor discussion sessions:
    - Check out our community calendar https://carpentries.org/community/#community-events
- You can also find us on
    - Twitter:
        - Software Carpentry on Twitter https://twitter.com/swcarpentry
        - Data Carpentry on Twitter https://twitter.com/datacarpentry
        - Library Carpentry on Twitter https://twitter.com/LibCarpentry
        - Carpentries on Twitter https://twitter.com/thecarpentries
    - Slack https://swc-slack-invite.herokuapp.com
        - Many public channels for different topics, an active channel for posting job

opportunities, etc
- Gitter https://gitter.im/LibraryCarpentry/Lobby (Library Carpentry)
- Facebook https://www.facebook.com/carpentries/

XVII. Workshop Introductions
https://carpentries.github.io/instructor-training/23-introductions/index.html

Setting the Workshop Environment

Your Academic Past (5 min)
Think back to courses or workshops you really liked or didn't like.

- How did those courses start on the first day?
- Were you confident in the instructors abilities?
- Did you feel like they were enthusiastic about the course and invested in you?
- Was it clear what you were going to be learning?
- Were you excited about the material?
- Did you leave that first day thinking the instructor was uninterested, that you weren't the students they wanted to be teaching or you had no idea what the course was supposed to be about?

- Sam: Workshops in the past that I didn't enjoy or struggled to keep up with didn't set the scene or give an overview of the different sections, what would be hands-on and what wouldn't, but just dove right into the topic. Not knowing about the material, what the pre-requisites were, or what the goal was made it hard to keep up, even if the instructor was clearly knowledgeable and enthusiastic. (Interestingly, I remember several of these where it wasn't clear what we were actually building...)
- Jay: Memorable courses/workshops start with a 'bang!' - something interactive, something funny, something weird, something interesting. First impressions are key. I can't remember ever having the 'uninterested instructor' but I have certainly felt that some instructors lack enthusiasm.
- Addi: Courses that I enjoy most have the instructor either doing live coding or all of us doing the coding with them. The most memorable one was a course where we had access to AWS instances and could finish from start to end a whole concept implemented within the day. Instructor was very positive and approached every topic with no assumptions in mind. The course that I did not enjoy was an introductory numerical math course which wasnt introductory at all. Instructor stared with an equation that he assumed everyone knew and talked about it the first half withour any feedback from the audience. Then we went to the next class of equations and so on. There was no high level overview or terms/definitions used. Also there were no helpers in the class and the instructor could stay past and answer any questions. It was not a positive experience.
- Nonkululeko:
- Talisha: Courses and workshops that I've enjoyed have done a wonderful job of setting clear expectations and guidelines, as well as creating a welcoming environment, from the start. I'm not sure if it the instructors abilities that I am confident in but in these cases I did believe that the instructor would be honest when something was outside of their skillset. and they often provided the information they had and opened the floor for those who knew more. Yes, the instructors were vey excited about the information they were teaching. Even in cases where I have not been excited about the material, the instructors enthusiam seemed to help me be more comfortable and excited about what I was learning. Not at all.

- Samarpita: Workshop I didnt like: Started by a very short intro by the instructor but no outline of learning objectives or the course we would be taking. The instructor seemed well versed with the material but had some difficulties in explaining that to novice learners. I realize now that the course content didn't seemlessly fit together which made it seem like in class we were jumping from one topic to another.
- Nate Qu: Intensive bootcamp-style workshops for two-year college and highschool physics faculty; really just jumped right into the training and kept us engaged late into the day; i had confidence in the instructors and was excited about the material; i knew that I'd be able to go back to my classroom the next week and immediately implement these new methods and resources
- Lance: Courses/workshops I enjoyed in the past usually started out with some sort of exercise that centered us as a group, provided introductions, and made me feel welcomed as a learner. They usually followed a lot of recommended practices for teaching adult learners. This usually included valuing any lived experiences that learners brought into the course or workshop. I've always appreciated when workshop hosts work to create a sense of community early on--early excercises that create low barrier success points for group members, and the idea that we are all able to contribute to community of learning.
- Luca: best course I had was computational physics. The instructor was a young very motivated phd student who clearly put an effort in designing the course and creating a positive environment. He emphasized group discussion, had students discuss their solution among each others, showed by doing what was the correct way and always answered questions in slack. One thing I particularly remember was that once he had miscalibrated the difficulty of week exercises, as most students kept asking questions till late evening. He actually remained in slack answering students' questions and helping those till 1:00 AM of the night. Most classes I didn't enjoy were because the lecturer conveyed the atmosphere, "I don't like teaching, I teach because university forces me".
- Al: The ones that I have enjoyed most the teachers were passionate, caring and personal. The ones that have been less ideal, I did get the feel that instructors were not enthusiastic or very interested seemed more like a task not a passion or career. Even if there are hiccups the attitude is 80%+ for me.
- David Ch:
    - Starting with a "not too serious" quizz is a nice opening, which nicely introduce what you will be talking about, and hopefully demonstrate to the audience there are things they already know about, and things they will learn during the training.
    - Terrible and frequent : I lack the few first installation steps for the training practice, and I am lost for all the rest of the training...
- Jess: Workshop I didn't like: started with abstract concepts and it wasn't clear how they would relate to doing something in the real world. I was skeptical of the instructor's expertise. Workshops I like the most are hands-on with lots of creating things, be it code, animations, etc.
- Karin: Included some discussion of how communication in the workshop should be -- welcoming questions, explaining that we'd be collaborating and why that was valuable, etc. Then, the instructor needs to follow up by actually responding to questions in a welcoming way, setting a positive tone, etc. I also really like having a clear agenda when I'm attending a workshop -- even if we end up deviating from it-- makes me feel like the instructor knows what they are doing and has a plan, helps me know what I will get out of the workshop. Appreciate when the instructor seems happy to be there and shares about themselves and their connection to the material in an enthusiastic way. I like when there's a chance early on to connect to other attendees, even if it's just a moment to say high to a neighbor. I also kind of love it when there is a code of conduct that is mentioned. It might seem overly formal to some people, but to me it conveys that the instructor is serious about everyone treating each other with respect, and that immediately makes me more comfortable.

- Andrew: Workshop started with "attendee speed-dating": we got to meet a few other attendees for two minutes each with a suggested prompt question to answer ("what are you reading/watching atm"). Set the stage for a really collaborative, peer-assisted learning environment. Have been to others where the code-of-conduct was the first thing covered, which seemed strange at the time ("get to the content!") but again, made it clear that interaction was inevitable and encouraged.
- Robert:
- Yara:
- Emmy: One I really liked: started with instructors saying sharing how they personally got involved in the org/course, their relevant experience after and somepersonal things about how they felt when they were in the learners' positions- I was able to relate. There was a clear learners' path that we followed over weeks. One I didn't like: i believed in the instructors skills because they were established in the field, but they were going really fast and it was quite clear that some of the techniques that were taught were outdated/they no longer do the hands-on work (the stuff they were supposed to teach).
- Tayaza: The courses I loved were delivered by enthusiastic instructors who knew how to relate the contents of the materials to pratical use cases.
- Nate Jo:
    - When logistics permit, I always think its great when learners/participants are asked to introduce themselves, what they want to get out of the workshoph, and then some funny/quirky fact.
    - ^^^ This is even better when the instructor is good about referencign what people said throughout the course. For example, I was in a geospatial course and the instructor said, "and Nate, this will be relevant to your interest" when we started talking about a subject that was relevant to my research.
- Danielle: I recently attended a series of workshops that I really enjoyed. Each time we came into the Zoom room, the host would have music playing and encouraging attendees to introduce themselves in the chat with their name and location and maybe a small tidbit, like the weather in your area that day. It was a great way to set the tone and excitement and comfotability in the workshop. A workshop that I attended recently that I did not enjoy, there wasn't really a way to get questions answered. Or when they were answered it was way after the fact. I got really behind and frustrated.

What's in an Introduction? (10 min)
Get into small groups (3-4 people) and discuss the questions below for 5 minutes. Take notes on your answers.

- What do you hope to accomplish in a workshop introduction?
- What information do you need to include in an introduction to accomplish these goals?

Group 1:

- Accomplish: Set expectations, make people feel comfortable and at ease, awake interest in topic, give learners a "map" and overview of what the sections/lessions are
- Information: let them know what they're going to learn, give motivation why topic is important, real-life example as to why something is important, describe learning outcomes and "at the end of this you'll be able to", provide outline of workshop, set the tone and make clear whether it's structured/professionally run or more casual

Group 2:

- Great motivating introduction
- excitment to come across
    - Set this to create that "ambience" in the room.
- make it a safe space
- talk about background, go through goals and objectives for the day
    - parking lot questions
- Setting the tone and expectation for the workshops
- Show attendees how to ask questions
- Introducing the Code of Conduct early as a means of creating and maintaining safe space
    - Getting early agreements from participants with checkmarks/thumbs up/affirmation
- Creating a welcoming physical or online space (possibly music playing, early introductions)
- Creating space for attendees to ask questions anonymously or to DM instructors

Group 3:
-Need a brief overview (especially when topic is very abstract) - want to give high-level motivation, why is this useful, what is the big picture. Don't immediately dive into the mechanics/details, but think about setting the stage for the workshop and topic.
-Having that kind of background about the lesson primes you for learning
-Learning objectives -- show a map of what we'll do, what are you going to get out of this
-Create rapport/establish a relationship. Have some introductions -- at least learn names; if you can learn more about students (e.g. professional interests), you can use that to contextualize info throughout the workshop for them.
-Make it into a conversation not a monologue!
-Create a sense of safety -- students can feel comfortable asking questions, being new with the topic. Sense that the group is in it together and the instructor is there to meet them where they are at and support their learning.

Group 4:
- Expectation management (both learners and instructors)
- Praise and reassure novices
- Encouraging engagement!
- Start with some ice-breaking activity (quizz), or asking them one concept that they love/dislike in the topic

Group 5:
- set clear expectations for what learners will learn
- set the tone, e.g. all questions are welcome

- - also set expectations for the activities upcoming, like pair programming, breakout rooms, just to prepare learners so they can prepare mentally & technically

- motivate! as folks may not know what they can benefit from
- make sure everyone has the required set-up and get a feel of why people are attending, what are they expecting

- - doing this ahead of time help, e.g. with a preworkshop survey, and video tutorials in an email on how to set-up

Goals For the Introduction
After the introduction learners should:

- believe in your competence to teach the workshop
- be able to predict the type of instruction
- know what will be taught
- know what will be required of them

The instructor should:

- have an understanding of who is taking the workshop and what their expectations are

Components of the Introduction
To meet these objectives an introduction should:

1. Set positive first impressions
    1. incl how you dress <- easier to dress more formally and then relax from there
    2. Think of learning environment: e.g. zoom background
    3. How to greet participants: are you frantically preparing or saying hello?
2. Introduce yourself effectively (and have other workshop leaders do the same)
    1. share experience that makes you an 'expert' and able to teach
    2. share challenges you've had in the past with this material (be seen as a role model)
    3. motivation for teaching a carpentries workshop
    4. think about what NOT to say - avoid demotivating
3. Clarify learning objectives and expectations
    1. describe pre-requisites (after checking pre-workshop survey)
    2. highlight main aspects of schedule (breaks, sections)
    3. learning objectives
    4. teaching strategies
    5. roles that different people will take on (you, helpers)
    6. explain code of conduct and how to report violations
    7. make sure learners know how to ask for help / and that they should ask for help
    8. ask for feedback
4. Help learners learn about each other (i.e. ice breaker)
    1. https://carpentries.github.io/instructor-training/icebreakers/
5. Set the tone for the workshop (encourage participation)
6. Collect baseline data on learners' knowledge and motivation
7. Whet learners' appetite for workshop content
8. Inform Learners of Logistics (break times, accessibility cc, bathroom locations)

Practice Your Introduction (10 min)
Imagine you have completed instructor training and you are about to teach a full lesson around the material you have been practicing teaching yesterday.

1. Write out some notes, covering some of the topics described above:
    2. Introduce yourself effectively
    3. Clarify learning objectives and expectations
    4. Set the tone for the workshop

- Robert Introduce helpers, other instructors and myself. Talk about the experience of the team members with the subject material. We would then then ask learners to introduce themselves and integrate the intros with a brief ice breaker. Once this is complete we can go through the learning objectives for the session and answer any questions that follow. We will reinforce the safety of the learning space by reviewing the code of conduct before we begin the lesson.
- Jay greet participants by name as they arrive? introduce instructors, helpers, self with my involvement/experience in Carpentries/programming. Round of learner introductions. Establish learning outcomes, code of conduct, acknowledge workshop format (live coding, breakouts, discussions). Demonstrate patience, respect, courtesy, enthusiasm. Encourage questions, mistakes, errors and 'I don't knows'
- Al arrive early, maybe have music playing, introduce self and others, let excitment shine through, describe objectives, agree on ground rules, review COC, discuss "Parking lot", housekeeping-breaks, bathrooms if in person etc...share motivational story, intro of class with icebreaker question
- Andrew
- Jess
- Addi : When I introduce myself, start with specific background that relates to the learners in the workshop,
    - Talk about learning goals and background
    - Talk about workshop code of conduct as well as various workshop guidelines to ask questions, take a break etc
    - Make sure to encourage everyone to introduce themselves as well; focus on what they hope to learn from teh workshop rather than their background as not all learner may want to talk about that.
- Samarpita I would start by setting expectation for the course and asking students to give a quick introduction. Next I would ask them what is the one thing they are looking forward to learning in this course and the one thing that they want to improve in this course. Move onto introducing the helpers for the course and asking students whi wer eunable to set up the programs required for the course to approach the helpers before we start the session. I would move on to setting some learning goals and describing the schedule that will be followed over the coruse of the workshop.
- David Ch
    - I am computer scientist, working for physicists, specialized in C++/Python
    - The training should help novices to progress, and I hope for some feedback
    - Quizz !
- Nate Qu
    - Brief welcome and introductions from the leadership team & helpers
        - Names, backgrounds, roles within organization, struggles we've faced with Python
    - Give students 30 seconds or so to introduce themselves to the entire group (smaller group introductions within each cohort or institution will come later)
        - Name (nicknames)
        - Where from? What school they're attending?
        - What program they're in (degree or certificate they're seeking)
        - Something interesting or funny about themselves
    - Menti survey asking participants to share their levels of experience with python / anxiety about taking the workshop (to be done so responses will display anonymously)
    - High-level overview of the agenda for that day
    - Set the environment/atmosphere so that everyone is made to feel welcome and that this is an inclusive space where questions are welcome and it's okay to make mistakes
    - Walk through the code of conduct and expectations

        - Introduce tools that we'll use for communication (Spatial Chat, Slack, Discourse, Google Docs, Jupyter Notebooks/Google COLAB)
            - Note you won't want to go through all these tools on the 1st day, rather introduce 1 new tool each day
        - End with time dedicated to letting participants complete the pre-program survey
- Emmy Brief welcome before official start time (make sure to not spend too much time w/ people i might've already met/knew). Introduce the code of conduct. Introduce co-instructor and helpers. Share relevant experience to content taught (e.g. academic background, coding experience), share personal motivation for learning to code/work with data reproducibly - then ask co-instructor to do the same. Ask learners to also share (briefly) their names and motivations with each other. Emphasise that everyone has a different starting point and prior experience, all questions are welcome. State learning objectives and rough schedule (esp if multiday), what they should have learnt by the end of the day. Share exercise formats (e.g. breakout rooms, demos) and tools & workflows (e.g. how we use etherpads, how to ask questions). Check if everyone has had followed the emailed instructions to set-up OK, and be sure to have helpers help those who are struggling with this. Start!
- Tayaza
- Lance ---took notes in my own notebook.
- Luca introduce myself: saying my name and affiliation, that I code everyday and have had the luck to have good teachers who instructed me to do the proper way and as I truly believe that helped in my work, I want to pay back and help others harness the power of good coding practices. Say something about reproducibility being a fundamental ethical principle of research (if I know audience is composed by mainly researchers) and that good coding practices can ensure that these ethical principles are satisfied. Also good coding and reproducible software increase visibility and networking opportunity. Give a short introduciton of the topics of the workshop, e.g. using the homepage as supporting material, for each lesson one sentence to describe what it is and one sentence to describe why it is useful. Remind about code of conduct, that if they feel uneasy or have problem can report to us (instructors), other necessarily housekeeping announcements (when breaks are, etc.).
- Nonkululeko
- Nate Jo
- Karin
- Sam tell people where I'm from (people always wonder about my accent), where I'm based, what my work background is and what kind of typical work I do (which qualifies me for teaching the topic). Tell learners why I find the topic exciting/useful and give a concrete example. Ask learners (via the chat or in person) what they're hoping to learn. Explain "at the end of this course, you'll be able to do xyz". Explain the mode of the workshop, whether they will follow along live, whether there will be breaks for hands-on exercises, or it's mostly just me teaching - highlight if there's different notation to indicate what to do.
- Danielle -
    - Introduce yourself effectively- Include name, location, role in the workshop, background with The Carpentries.
    - Clarify learning objectives and expectations- objective is to effecively become familiar with the given topic, and be given the tools to start utilising it, and through practice become a competent practitioner. Expectations should include the code of conduct, ways for communicating questions and thoughts, other instructors/helpers roles, letting people know that it is okay to have their video off but it would be helpful to have it on so you can gauge attendee's understanding.
    - Set the tone for the workshop- smile, talk to your attendees as they enter the room, play

music

- Talisha
    - Introduce self (Name, based/location, work, and background)
- Yara

Key Points

- A planned introduction is a helpful tool in setting the workshop environment.
- Introductions should include both practical information and start building relationships.

XVIII. Putting It Together
https://carpentries.github.io/instructor-training/24-practices/index.html

Picking up the Pieces (optional)
Make a list of all the concepts you've encountered in this training. Your list can include everything from educational/teaching theories to specific in-classroom practices.

Organize Your Knowledge (Optional)
Put the pieces together by creating a visual organization of information.
We suggest doing this in two ways:

1. If you are comfortable with / like concept maps, trying integrating all the topics above into a single concept map.
2. Use the provided handout to organize topics. Here are two examples:
    - Handout One (https://carpentries.github.io/instructor-training/files/handouts/Wrap-Up-doc.pdf, with example content: https://carpentries.github.io/instructor-training/files/handouts/Wrap-Up-doc-example.pdf)
    - Handout Two (https://carpentries.github.io/instructor-training/files/handouts/Carpentries_teaching_practices.pdf)

Work on this on your own. There's no "right answer" – this is about you building up a mental model, moving from "novice" to "competent practitioner".

Parting Thoughts (Optional)
If you didn't think about these issues when organizing your topics in the previous exercise, now consider:

- What is your mental model of teaching?
- Can you identify why each topic above applies to teaching for the Carpentries?

Key Points

- Having a plan makes it easier for you to remember to implement the important teaching practices you've learned.

XIX. Wrapping Up

https://carpentries.github.io/instructor-training/25-wrap-up/index.html

Feedback (5 min)
Use this web form https://forms.gle/6umQuXT5AdiNrQqM8 to tell us one thing you learned this morning that you found useful, and one question you have about the material or something we could improve for next time. Do *not* put your name on the notes: this is meant to be anonymous feedback.

Post Workshop Surveys (5 min)
Assessment is very important to us! Please complete this five-minute post-workshop survey.
https://www.surveymonkey.com/r/post-instructor-training

Key Points

- Feedback applies to all kinds of learning, including learning how to teach.