BRIAN DAY 1 AFTERNOON

Motivation and Demotivation

Overview

Teaching: 20 min
Exercises: 45 min
Questions

- Why is motivation important?
- How can we create a motivating environment for learners?

Objectives

- Identify authentic tasks and explain why teaching using them is important.
- Develop strategies to avoid demotivating learners.
- Recognize and overcome imposter syndrome in yourself and your learners.

In order for learners to step out into new and unfamiliar terrain, they will need encouragement. This section discusses typical ways that learners are motivated (and can be demotivated!) and provides practice opportunities for you to become confident in motivating your learners.

Creating A Positive Learning Environment

Creating a positive learning environment is an important first step to setting the stage for learner success. As instructors, it is crucial to establish the workshop setting as a safe space for learning. Establishing a safe space for learning is a combination of many factors:

- *Presenting the instructor as a learner*. Admitting that you don't know everything is part of showing that it is acceptable to make mistakes, and encouraging a growth mindset in learners (we'll talk much more about growth mindset in a later lesson). Using participatory live coding, our chosen method for teaching concepts, is very useful for this reason. It is common to make errors while coding. When done well, these errors can be very instructive for novice learners.
- *Establishing norms for interaction*. This can be done by having, discussing, and enforcing a Code of Conduct or by having rules of interaction such as ensuring

turn taking in discussions, possibly by passing around a talking stick, or by encouraging quieter people to contribute.

- *Encouraging learners to learn from each other*. Acknowledge that some of the material can be difficult and that they will learn more working together. Asking more advanced learners to help beginner learners is a good way to maximize learning for both.
- *Acknowledging when learners are confused.* Understanding why learners are confused provides useful feedback for instructors. We use formative assessments to pinpoint learners' misunderstandings. Acknowledging that their misunderstandings are valid is also key to encouraging a growth mindset.

Teach Most Useful First

People learn best when they care about a topic and believe they can master it. This presents us with a problem because most scientists don't want to program: they want to do science. In addition, their early experiences with computers are often demoralizing, and believing that something will be hard to learn is a self-fulfilling prophecy.

Imagine a graph whose axes are labelled "mean time to master" and "usefulness once mastered". Everything that's quick to master, and immediately useful should be taught first; things in the opposite corner that are hard to learn and have little near-term application don't belong in our workshops.

What to Teach

Actual Time

> Any useful estimate of how long something takes to master must take into account how frequent failures are and how much time is lost to them. For example, editing a text file seems like a simple task, but most graphical editors save things to the user's desktop or home directory. If people need to run shell commands on the files they've edited, a substantial fraction won't be able to navigate to the right directory without help. If this seems like a small problem to you, please revisit the discussion of expert blind spot.

We have therefore adopted a "teach most immediately useful first" approach. We try to have learners do something that *they* think is useful in their daily work within 15 minutes of starting each lesson. This not only motivates them, it also helps build their confidence in us, so that if it takes longer to get to something they find useful in a later topic, they'll persist with the lesson.

Authentic Tasks: Think, Pair, Share

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.

This exercise and discussion should take about 10 minutes.

Other Motivational Strategies

In addition to teaching things that will make our learners' lives easier and focusing on authentic tasks, there are a number of other strategies we can use to motivate learners.

Strategies for Motivating Learners

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 briefly describe in the Etherpad how we can apply these strategies in our workshops.

- Strategies to Establish Value

1. Connect the material to learners' interests.
2. Provide authentic, real-world tasks.
3. Show relevance to learners' current academic lives.
4. Demonstrate the relevance of higher-level skills to learners' future professional lives.
5. Identify and reward what you value.
6. Show your own passion and enthusiasm for the discipline.

- Strategies to Build Positive Expectations

1. Ensure alignment of objectives, assessments, and instructional strategies.
2. Identify an appropriate level of challenge.
3. Create assignments that provide an appropriate level of challenge.
4. Provide early success opportunities.
5. Articulate your expectations.
6. Provide rubrics.
7. Provide targeted feedback.
8. Be fair.

9. Educate learners about the ways we explain success and failure.
10. Describe effective study strategies.

- Strategies for Self-Efficacy

1. Provide learners with options and the ability to make choices.
2. Give learners an opportunity to reflect.

This exercise and discussion should take about 5 minutes.

Provide an Example

Insert a personal story here about how you establish value in the classroom.

Or, insert an example story about establishing value, which goes like this:

"In the SWC Unix "Finding Things" episode, a haiku is used to to teach grep. This is a great didactic tool, but it can be hard for learners to see how it applies to research. After the didactic example, I connect my bioinformatics users with domain-specific examples by showing a list of one-line unix commands consisting of `grep`, `sort`, `head`, and `uniq` to explore biological sequence data. This emphasizes how they can apply what they learned with haikus to real bioinformatics research problems."

Brainstorming Motivational Strategies

Think back to a computational (or other) course you took in the past, and identify one thing the instructor did that motivated you. *Pair* up with your neighbor and discuss what motivated you. *Share* the motivational story in the Etherpad.

This exercise should take about 5 minutes.

Not Just Learners

What's missing from this list is strategies to motivate the *instructor*. As we said in the introduction, learners respond to an instructor's enthusiasm, and instructors need to care about a topic in order to keep teaching it, particularly when they are volunteers.

Why Do You Teach?

We all have a different motivation for teaching, and that is a really good thing! The Carpentries want instructors with diverse backgrounds because you each bring something unique to our community.

What motivates you to teach? Write a short explanation of what motivates you to teach. Save this as part of your teaching philosophy for future reference.

This exercise should take about 5 minutes.

How Not to Demotivate Your Learners

One of our biggest challenges as instructors when teaching a workshop is to not demotivate participants through our words or actions. None of us goes into a workshop with the intention of creating a hostile environment or making the learners hate the tools we're teaching, but we can all accidentally do just that if we don't pay attention to what we say and how we interact with our learners. We'll discuss some common demotivators and help you develop strategies for avoiding them.

Things You Shouldn't Do in a Workshop

- Tell learners they are rubbish because they use Excel and/or Word, don't modularize their code, etc.
- Say negative things about Windows and praise Linux, e.g., say that the former is for amateurs.
- Criticize GUI applications (and by implication their users) and describe command-line tools as the One True Way.
- Talk contemptuously or with scorn about any tool. Regardless of its shortcomings, many of your learners may be using that tool. Convincing someone to change their practices is much harder when they think you disdain them.
- Dive into complex or detailed technical discussion with the one or two people in the audience who clearly don't actually need to be there.
- Pretend to know more than you do. People will actually trust you more if you are frank about the limitations of your knowledge, and will be more likely to ask questions and seek help.
- Use the J word ("just") or other demotivating words we talked about in a previous lesson. These signal to the learner that the instructor thinks their problem is trivial and by extension that they therefore must be stupid for not being able to figure it out.
- Take over the learner's keyboard. It is rarely a good idea to type anything for your learners. Doing so can be demotivating for the learner (as it implies you don't think they can do it themselves or that you don't want to wait for them). It also wastes a valuable opportunity for your learner to develop muscle memory and other skills that are essential for independent work.
- Feign surprise. Saying things like "I can't believe you don't know X" or "You've never heard of Y?" signals to the learner that they do not have some

required pre-knowledge of the material you are teaching, that they don't belong at the workshop, and it may prevent them from asking questions in the future. (For more on this see the Recurse Center's Social Rules).

It can be difficult to avoid these demotivators entirely. Some people are so used to making fun of Windows-users with their friends, or talking about how terrible Excel is that they initially fail to realize they're doing it while teaching. Teaching yourself to avoid these types of comments takes practice, but is well worth the effort. No one likes to be made fun of, and talking badly about people who use GUIs, or who aren't writing their thesis in LaTeX is unlikely to make them excited about learning your favorite scripting language.

Brainstorming Demotivational Experiences

> *Think* back to a time when you were demotivated as a student (or when you demotivated a student). *Pair* up with your neighbor and discuss what could have been done differently in the situation to make it not demotivating. *Share* your story in the Etherpad.
>
> This exercise should take about 5 minutes.

Psychological Demotivators

As instructors, we can learn to avoid talking disparagingly about our learners' choice of text editors and levels of technical knowledge. This helps a lot in avoiding demotivating learners. There are other factors, however, that contribute to demotivation, some of which are either systemic, or built into our psychological makeup as human beings. We can still have a positive impact in fighting these demotivators by thinking carefully about the language that we use and how we interact with our learners.

Stereotype Threat

One major psychological demotivator is *stereotype threat*. Reminding people of negative stereotypes, even in subtle ways, can make them anxious about the risk of confirming those stereotypes, in turn reducing their performance. This is called *stereotype threat*, and the clearest examples in computing are gender-related. Depending on whose numbers you trust, only 12-18% of programmers are women, and those figures have actually been getting worse over the last 20 years. There are many reasons for this (see Margolis and Fisher's *Unlocking the Clubhouse* and Margolis's *Stuck in the Shallow End*). Steele's *Whistling Vivaldi* summarizes what we know about stereotype threat in general and presents some strategies for mitigating it in the classroom.

While there's lots of evidence that unwelcoming climates demotivate members of under-represented groups, it's not clear that stereotype threat is the underlying mechanism. Part of the problem is that the term has been used in many ways. Another

is that there are questions about the replicability of key studies. What *is* clear is that we need to avoid thinking in terms of a deficit model (i.e., we need to change the members of under-represented groups because they have some deficit, such as lack of prior experience) and instead use a systems approach (i.e., we need to change the system because it produces these disparities).

Never Learn Alone

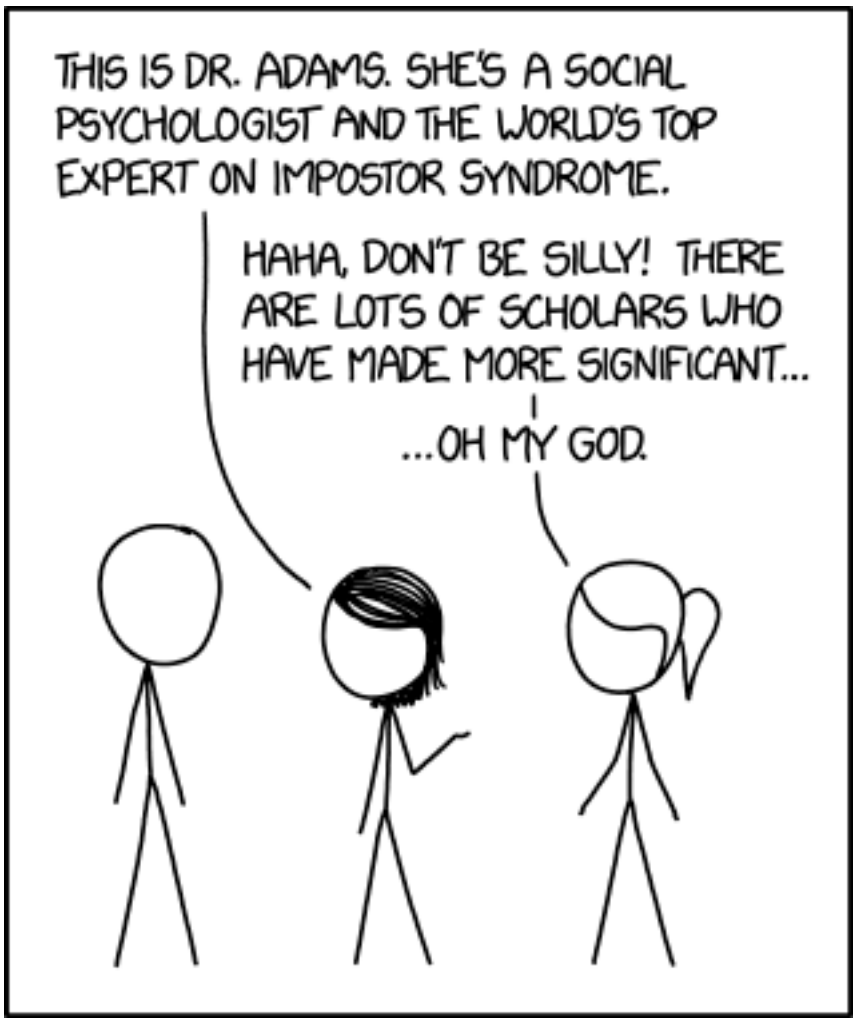
> One way to support at-risk learners of all kinds is to ask people to sign up for workshops in small teams rather than as individuals when possible. If an entire lab group comes, or if attendees are drawn from the same (or closely-related) disciplines, everyone in the room will know in advance that they will be with at least a few people they trust, which increases the chances of them actually coming. Furthermore, if people attend a workshop with their labmates, it's more likely they will work together to implement what they've learned after the workshop has ended.

The Importance of Having Rules

> To make clear what is expected, and all participants in our workshops are required to conform to the Code of Conduct. Its details are important, but the most important thing about it is that it exists: knowing that we have rules tells people a great deal about our values and about what kind of learning experience they can expect.

Impostor Syndrome

Another major psychological demotivator is *impostor syndrome*. Imposter syndrome is the belief that one is not good enough for a job or position, and that one's achievements are due to luck rather than talent or skill. This is also accompanied by the fear of being "found out".



https://xkcd.com/1954/

Academic work is frequently undertaken alone or in small groups but the results are shared and criticized publicly. In addition, we rarely see the struggles of others, only their finished work, which can feed the belief that everyone else finds it easy. Women and minority groups, who already feel additional pressure to prove themselves in some settings, may be particularly affected.

Two ways of dealing with your own impostor syndrome are:

1. Ask for feedback from someone you respect and trust. Ask them for their honest thoughts on your strengths and achievements, and commit to believing them.
2. Look for role models. Who do you know who presents as confident and capable? Think about how they conduct themselves. What lessons can you learn from them? What habits can you borrow? (Remember, they quite possibly also feel as if they are making it up as they go.)

As an instructor, you can help people with their impostor syndrome by sharing stories of mistakes that you have made or things you struggled to learn. This reassures the class that it's OK to find topics difficult. Being open with the group makes it easier to build trust and make learners confident to ask questions. (Live coding is great for this: typos let the class see you're not superhuman.)

You can also emphasize that you want questions: you are not succeeding as a teacher if no one can follow your class, so you're asking learners to help you learn and improve. Remember, it's much more important to *be* smart than to *look* smart.

The Ada Initiative has some excellent resources for teaching about and dealing with imposter syndrome.

Overcoming Imposter Syndrome (Optional)

> Think of a time when learning something was difficult for you, or you made a mistake that seemed silly or embarrassing. Is that task still hard for you? In the Etherpad, describe how you might use this as a motivational story to help your learners overcome their own imposter syndrome.
>
> This exercise should take about 5 minutes.

Accessibility Issues

Not providing equal access to lessons and exercises is about as demotivating as it gets. If you look at our old lesson on Python, for example, you'll find that the text beside the slides includes all of the narration—but none of the Python source code. Someone using a screen reader would therefore be able to hear what was being said about the program, but wouldn't know what the program actually was.
While it may not be possible to accommodate everyone's needs, it *is* possible to get a good working structure in place without any knowledge of what specific disabilities people might have. Having at least some accommodations prepared also makes it clear that hosts and instructors care enough to have thought about problems in advance, and that any additional concerns are likely to be addressed.

It Helps Everyone

> Curb cuts (the small sloped ramps joining a sidewalk to the street) were originally created to make it easier for wheelchair users to move around, but proved to be equally helpful to people with strollers and grocery carts. Similarly, steps taken to make lessons more accessible to people with various disabilities also help everyone else. Proper captioning of images, for example, benefits people with no or limited vision by giving screen readers something to say: but it also makes the images more findable by exposing their content to search engines.

The first step is to know what you need to do. There are a number of good resources for learning about accessibility.

Learning about Accessibility

> The UK Home Office has put together a set of posters of dos and don'ts for making visual and web-based materials more accessible for different populations. Take a look at one of these posters and put one thing you've learned in the Etherpad.
>
> Note: There are translations available in a number of languages, including Dutch, French, Spanish, Swedish, Portuguese, German, and Chinese.
>
> This exercise should take about 5 minutes.

The W3C Accessibility Initiative's checklist for presentations is a good starting point focused primarily on assisting the visually impaired, while Liz Henry's blog post about accessibility at conferences has a good checklist for people with mobility issues, and this interview with Chad Taylor is a good introduction to issues faced by those with no or limited hearing.
The second is to know how well you're doing. For example, sites like WebAIM allow you to check how accessible your online materials are to users with no or limited vision.

The third, and most important, is to *involve people with disabilities in decision-making*. Our mailing lists are a good place to ask for advice, and updates to our accessibility checklist are always welcome.

What Happens When Accessibility is an Issue? (Optional)

> Think of a time when you've been affected by, or noticed someone else being affected by issues with accessibility. This may have been at a conference you attended where the elevator was out of service, or maybe a class you were taking relied on audio delivery of content. Describe what happened, how it impacted your (or someone else's) ability to be involved and what could have been done to provide better accessibility in this case.
>
> This exercise should take about 5 minutes.

Every Little Bit Counts

> Looking at people who work with disability and accessibility, it's easy to be overwhelmed by all the different ways we could make instruction more accessible. *Don't panic.* Instead:

- *Don't do everything at once.* We don't ask learners in our workshops to adopt all our best practices or tools in one go, but instead to work things

in gradually at whatever rate they can manage. Similarly, try to build in accessibility habits when preparing for workshops by adding something new each time.

- *Do the easy things first.* There are plenty of ways to make workshops more accessible that are both easy and don't create extra cognitive load for anyone: font choices, general text size, checking in advance that your room is accessible via an elevator or ramp, etc.

Accessibility Testing

Find the nearest public transportation drop-off point to your building and walk from there to your office and then to the nearest washroom, making notes about things you think would be difficult for a wheelchair user. Now borrow a wheelchair and repeat the journey. How complete was your list of challenges? And did you notice that the first sentence in this challenge assumed you could walk?

Inclusivity

Inclusivity is a policy of including people who might otherwise be excluded or marginalized. In computing, it means making a positive effort to be more welcoming to women, people of color, people with various sexual orientations, the elderly, people with physical disabilities, the formerly incarcerated, the economically disadvantaged, and everyone else who doesn't fit Silicon Valley's white/Asian male demographic. Lee's paper "What can I do today to create a more inclusive community in CS?" is a brief, practical guide with references to the research literature. These help learners who belong to one or more marginalized or excluded groups, but help motivate everyone else as well.
Many of these can be applied to our workshops, such as:

- asking learners to email you before the workshop to explain how they believe the training could help them achieve their goals;
- reviewing notes to make sure they are free from gendered pronouns, that they include culturally diverse names, etc.;
- emphasizing that what matters is the rate at which they are learning, not the level of knowledge they had when they started;
- encouraging pair programming; and
- actively mitigating behavior that some learners may find intimidating, e.g., use of jargon or "questions" that are actually asked to display knowledge.

Key Points

- A positive learning environment helps people concentrate on learning.

- People learn best when they see the utility in what they're learning, so teach what's most immediately useful first.
- Imposter syndrome is a powerful force, but can be overcome.
- Accessibility benefits everyone.

Mindset

Overview

Teaching: 20 min
Exercises: 20 min
Questions

- How does mindset influence learning?
- How should we praise our learners?
- How should we talk about errors?
- What are successful habits of lifelong learners?

Objectives

- Explain the difference between fixed and growth mindset and its implications for classroom performance.
- Develop strategies for giving effort-based and improvement-based praise.
- Respond positively to learner errors.
- Model habits of lifelong learners.

The Importance of Mindset

Do you believe that you're just not good at math? Or maybe you're not artistic? Not good at sports? If any of these sound like you, you probably have a fixed mindset when it comes to mathematical, artistic and/or athletic ability.
Carol Dweck, now a professor of Psychology at Stanford University, has conducted decades of research focused on understanding people's attitudes towards ability, our beliefs about whether ability is ingrained or developed, and how those beliefs affect our perseverance in learning. She's discovered that, in any given area, people tend to have either a fixed mindset - believing that someone's ability in that area is intrinsic, you're either born with it or you're not - or a growth mindset - believing that, with enough effort, almost anyone can learn almost anything. (She's written a very

accessible popularized summary of her findings. There's also a short summary available.)

People with a fixed mindset tend to avoid effort and to give up when faced with failure or challenge. They also tend to take fewer risks, be less likely to take an opportunity to learn something new and to feel best about themselves when they perform flawlessly. In contrast, people with a growth mindset tend to actively seek challenges and to work harder when faced with failure. They regularly take risks and take advantage of opportunities for learning new skills. In fact, initial difficulties in a topic make them more interested and motivated to learn!

People who believe that ability is intrinsic view every situation as a test. A failure indicates that they don't have the natural ability they need to succeed. Thus, students with a fixed-mindset will *decrease* their study time after receiving a poor exam score, because they see this score as a judgement that they aren't good at that subject and shouldn't waste their time trying. In the same situation, a growth-mindset student will increase the amount of time that they study, seeing the poor score as an indication that they need to try harder to improve in that area.

Their attitude towards failure leads fixed-mindset people to avoid situations where they will be required to perform. They will not volunteer to answer questions in class or to demonstrate a sports technique to their team. This tendency prevents fixed-mindset people from using opportunities to practice and receive feedback, which, as we've discussed, is essential to learning.

Does Mindset matter?

> Think: What kind of mindset do you have about different areas? Is there anything you believe you are "not naturally talented" at? Mindset often varies in different areas – someone might have a fixed mindset with respect to artistic ability, but a growth mindset with respect to computing skill. Then, think about your learners. How might a learner's mindset about computational skill influence their learning in a workshop setting?
>
> Pair: Discuss your thoughts about the influence of mindset in a workshop. Try to come up with a few different ways or situations in which mindset might be relevant.
>
> Share: A few thoughts in the Etherpad (or go around the room and discuss)
>
> This exercise should take about 5 minutes.

If you recognize aspects of the fixed mindset in yourself, don't panic! Like everything else, a growth mindset can be learned. We're going to talk about some

strategies that will help you promote a growth mindset in your learners and yourself.

Praise Influences Mindset

The way in which we praise learners has an important impact on their mindset development. Carol Dweck's research investigated three types of praise: performance-based, effort-based and improvement-based, and looked at how these types of praise influence learner's mindsets.
We're most used to hearing **performance-based praise**. This sort of praise focuses on outcomes and implies that the outcome (for example, getting a correct answer) is the most important part of the task. It is often (but not always) combined with language that explicitly or implicitly supports the "intrinsic" model of competence. For example, "You did a great job! You must be very smart."

Effort-based praise focuses on the learner's hard work. For example, "You did a good job, you must have worked very hard." Exposing learners frequently to this type of praise helps them to transition from a fixed mindset to a growth mindset by changing their perception of effort. For fixed-mindset people, if something doesn't come easy to you, it means that you don't have natural ability and shouldn't bother. People with a growth mindset are much more likely to persevere in the face of challenge and accomplish their goals.

Improvement-based praise is related to praise based on effort, but explicitly draws a learner's attention to the progress they've made. For example, "You're doing so much better at this than last time, you must have worked very hard." This helps reinforce a growth-mindset by highlighting our ability to improve with effort.

Choosing our Praises

> 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!

Solution

- Effort-based.
- Improvement-based.

- Performance-based.
- Improvement-based.

Errors are Essential to Learning

> The typos are the pedagogy.
> — Emily Jane McTavish

One of the barriers to learning is avoidance of making errors. Errors are associated with negative emotions, which leads to learners being fearful of making them. *Error framing* is the process of presenting errors as an integral part of the learning process and using them as teaching opportunities. Error framing encourages learners to understand that making errors provides valuable learning opportunities instead of having negative consequences. For example, the Language Acquisition Made Practical (LAMP) system for learning language encourages learners to develop phrases and try them in a variety of social situations with native speakers. By being willing to make mistakes, LAMP learners receive useful feedback from native speakers in real-world social situations. Error framing has shown to be useful in learning complex tasks such as programming. In one study, Steele-Johnson showed that error framing showed positive effects on metacognition and self-efficacy.
How can we incorporate error framing into our lessons? We need to reduce the negative emotions associated with errors, instead casting errors as a "natural part of learning". Sharing experiences of learning from errors can encourage learners to adopt a growth-mindset. Associating improvement-based praise with errors can also encourage positive emotions associated with making errors in learning. Finally, the use of participatory live coding as the focus for teaching allows for instructors to model how errors can be positively framed and the process for overcoming them.

Helping Learners Learn From Mistakes

> 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.
>
> This exercise and discussion should take about 5 minutes.

Perseverance Predicts Success

Angela Duckworth, a Psychology professor from University of Pennsylvania, suggests that something called *grit* is an essential trait in learning. Grit can be defined as perseverance and passion in the face of difficulty towards a defined long term goal. Learners with grit are willing to fail in their pursuit if they feel that it gets them closer to their goal. Learning and recovering from failure is an essential skill in many fields. Learners with grit have shown to have a higher degree of success later on in life. However, grit is not an innate trait; there is much evidence that suggests that grit is a trait that can be learned and instilled in learners.
There are many ways we can help our learners attain grit.

- *Modeling resilience by sharing experiences of struggle*. Sharing your own experiences of having difficulty learning certain computing concepts may help learners understand that the process of learning can be difficult. Sharing failures that were learning experiences may aid in modeling resilience.
- *Promoting Perseverance*. Part of grit is being able to push through even when things are difficult. Improvement-based praise is an important motivating tool to promote perseverance.
- *Sharing Passion*. Talking about why you persist in working in your field despite the difficulties and challenges is another way to instill grit.
- *Emphasizing Long-Term Goals*. Showing the effects of learning the concepts in terms of learners' individual long-term goals can instill passion and drive for learning despite the difficulties of the topic.

How Are You Gritty? (Optional)

A previous exercise asked you to think of a time when learning something was difficult for you, or you made a mistake that seemed silly or embarrassing.

How did you motivate yourself to continue learning? How did it feel to persist in the face of challenge? How do you feel now about your capabilities in this area?

In the Etherpad, describe how you could use this story to illustrate the importance of grit for your learners.

This exercise should take about 5 minutes.

Habits of Lifelong Learners

Surviving the job market nowadays requires the ability to be a lifelong learner, in order to learn and adapt to new skills that may be required by employers. Many of these positions can be described as *Knowledge Work*, work that requires "non-

routine" problem solving. Knowledge workers need to be capable of what Cal Newport calls "deep work": defined as the "ability to focus intensely on cognitively demanding tasks".
Lifelong learning arises from those who have the growth mindset. Both grit and curiosity have been identified as success factors in lifelong learning. Often, the process of learning is a difficult journey. By modeling "gritty" behavior and how it has helped us achieve long-term goals, we can foster a "learn-it-all" rather than a "know-it-all" attitude in learners.

Emphasizing the role of help-seeking behavior in the lifelong learning process is important, as it is a critical skill of lifelong learners. Encouraging learners to seek help can be difficult when societal norms view seeking help as "stigmatizing, self-threatening behavior". Because of this, we need to reframe help-seeking behavior as a positive behavior to cope with difficulties. Part of this reframing can be modeled by instructors emphasizing that they are also lifelong learners and by admitting the limits of their knowledge. Emphasizing that one's cohort is a powerful source of help is important to establishing a positive learning community.

Key Points

- Growth mindset and grit promote learning by making effort a positive thing.
- Presenting errors as essential to the learning process helps learners learn from their mistakes.
- Successful lifelong learners aren't embarrassed to ask for help.

Afternoon Break