BRIAN DAY 1 MORNING

Welcome

Overview

Questions

- Who are we and how do we approach teaching?
- What should you expect from this workshop?

Objectives

- Introduce yourself to your fellow workshop participants.
- Describe what will and will not be covered in this workshop.
- Understand that the Carpentries are communities of volunteers who develop lessons and teach workshops on basic computing and data skills for researchers.

Welcome to Country

Play curtin welcome to country

Code of Conduct

Hello everyone, and welcome to the the Carpentries instructor training. We're very pleased to have you with us.

To make clear what is expected, everyone participating in Carpentries activities is required to conform to our Code of Conduct. This Code of Conduct applies to all spaces managed by the Carpentries including, but not limited to workshops, email lists, online forums and on GitHub. Please review the Code of Conduct to familiarise yourself with it.

- Use welcoming and inclusive language
- Be respectful of different viewpoints and experiences
- Gracefully accept constructive criticism
- Focus on what is best for the community

- Show courtesy and respect towards other community members

Introductions

Hi, I'm Doctor Brian Ballsun-Stanton, Solutions Architect (Digital Humanities) for Macquarie University. My PhD is in the Philosophy of Data, but my day job revolves around the technical implementations of data: Ethics applications, webpages, python and bash scripting, research design, and grant writing. I run a class on the digital humanities for Arts students and use Software Carpentry lessons in my teaching.

I've run five Software Carpentry workshops since 2017, covering SQL, Bash, Git, Python, R, and Webscraping for over 150 learners, and contributed fixes and major lesson updates to the carpentries' curriculum.

I've run checkout teaching demonstrations, am a badged Software Carpentry Instructor and Instructor Trainer, and I am really looking forward to bringing new instructors into the carpentries!

With me today is Masami Yamaguchi...

Talking Orange

In my experience, classes tend to divide themselves up: active participants, quiet but engaged participants, and passive participants. Masami has brought us a Talking Orange – the person holding the talking orange has the floor, if you don't have the orange, listen to the person who does.

We do ask that you don't **throw** the orange, but pass it along gently.

For folk who know each other, we ask that you not sit together. If you know someone to your left or right, wave hello and then trade places with someone else. It's very

Information for Today's Learners

1. Add your name to the Etherpad
2. Icebreaker: Shuffled Array

- Introduce yourselves verbally! In your introduction, (a) explain your work in 3 words and (b) say something suprising about you. Hand the orange to someone who hasn't spoken yet.

Our First Exercise

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. This exercise should take about 2 minutes.

Start at: 9:05

End at: 9:07

Most of you are probably here because you have attended or helped at a Software or Data Carpentry workshop. To make sure everyone has the same context, we'll give a brief overview of the Software and Data Carpentry organizations.

A Brief Overview of the Carpentries

Software Carpentry, Data Carpentry, and Library Carpentry are official Lesson Programs of the Carpentries. Together, they comprise communities of volunteer researchers, educators, and more who develop lessons and teach two day workshops on basic computing and data skills for researchers. Software Carpentry focuses on helping researchers develop foundational computational skills; Data Carpentry focuses on helping researchers work effectively with their data through its lifecycle; and Library Carpentry focuses on teaching data skills to people working in library- and information-related roles.

The main goal of these organizations is not to teach specific skills, per se - although those are covered - but rather, to convey best practices that will enable researchers to be more productive and do better research.

Instructor Training Workshop Overview

This two-day training focuses on developing teaching skills that are broadly useful across all of our lessons. These skills will be developed through discussion, practice, and feedback with fellow learners.

We will **not** be going over Data Carpentry, Library Carpentry, or Software Carpentry workshop content in detail (although you will get familiarity with some of the content through the exercises), This workshop is the first step in becoming a certified Carpentries instructor. The follow-up steps for certification will require that you dig into the workshop content yourself. We'll talk about that more tomorrow afternoon.

To orient yourself, there is a schedule on the workshop webpage which is linked through this workshop's Etherpad.

How Learning Works

One of our main emphases will be discussing the "best practices" of teaching. We will be introducing you to a handful of key educational research findings and demonstrating how they can be used to help people learn better and faster.

Building Teaching Skill

Just like learning a new language, a musical instrument, or a sport, teaching is a skill that requires practice and feedback.
We will have many opportunities to practice and give each other feedback throughout this workshop. We welcome questions and dialogue at any point. We'll be using the Etherpad to help facilitate discussion. Please feel free to ask questions verbally or to put them into the notes or chat box of the Etherpad.

Creating a Positive Learning Environment

One part of making this a productive two days for all of us is a community effort to treat one another with kindness and respect. This training, as in all Carpentries workshops, is subject to the Code of Conduct. We will be able to give our best effort (and have the most fun!) if everyone abides by these guidelines. We will also be discussing and practicing teaching techniques to create a positive and welcoming environment in your classrooms.

Carpentry History and Culture

We will be introducing you to the teaching practices that have been adopted by the Carpentries communities, and our overall philosophy and procedures in order to prepare you to teach Carpentries workshops. The greatest asset of the Carpentries is people like yourselves - people who want to help researchers learn about these ideas and share their own experience and enthusiasm. We hope that this training gives everyone a chance to meet new people and share ideas.

Expectation management and today's schdeule

- Part 1 will run until 11:10 and cover how novices build skill with practice and how expertise shapes how we communicate.
- Morning break at 11:10 to 11:25
- Masami will discuss how cognitive load impacts learning and how we can use feedback to improve our teaching practices
- Lunch will start at 12:30 and run until 1:30

- I will cover motivation and demotivation in the classroom environment, and the dangers of the word “Just” We’ll also explore how important different sorts of praise can be.
- Afternoon break runs from 3:15 to 3:30
- Masami will wrap up the day by exploring how we can improve our teaching, and lead us through one up and one down feedback.
- We should be done by 4:45

Day 2

- Another 9am start, where we explore Live Coding and preparing to teach a workshop
- Break is at 11:10
- Masami will then continue Live Coding practice and discuss how to manage a diverse classroom
- Lunch at 12:40
- I’ll talk about the carpentries and how you folks can check out and complete the process to becoming fully badged instructors
- Afternoon break at 3:10
- Masami will take us to the end by talking about a good workshop introduction and how all of these various topics tie together

We should be done by 4:30

Sticky Notes!

Feedback is really important. I’m a very big believer in red and green sticky notes (which is something we’ll cover during building skill with feedback). In front of everyone should be a red and greenish sticky. When you’re working on an exercise, sticky notes should go down. When you’re done, put up a green sticky. If you have problems, put up a red sticky.

I use these stickies to also get a sense of how the room is flowing. If people just leave their green stickies up, I know that they’ve checked out – and it’s a really useful immediate feedback route during each workshop.

Before you folks head out, (and we’ll remind you then) before each break, we’ll be asking for feedback on how the workshop is going on the sticky notes. But more on that later.

Assessing Trainee Motivation and Prior Knowledge

It’s important to first assess the prior knowledge of the workshop participants because this will influence (to some extent) how you motivate the activities and how you communicate with the attendees.

Background

In the etherpad, put an x next to each line of the survey which represents you. It’s OK to put more than one mark.

Start at: 9:15

End at: 9:17

Now that we have a better idea of everyone’s prior knowledge and familiarity with some of the key Carpentries teaching practices, we’re ready to begin our training. Our goal is that by the end, you will have acquired some new knowledge, confidence, and skills that you can use in your teaching practice in general and in teaching Carpentries workshops specifically.

Building Skill With Practice

Questions

- How do people learn?
- Who is a typical Carpentries learner?
- How can we help novices become competent practitioners?

We'll now get started with a discussion of how learning works. We'll learn some key findings from educational research and identify how these principles are put into practice in Carpentry workshops.

The Carpentries Pedagogical Model

The Carpentries aim to teach computational competence to learners. We take an applied approach, avoiding the theoretical and general in favor of the practical and specific. By showing learners how to solve specific problems with specific tools and providing hands-on practice, we develop learners' confidence and lay the foundation for future learning.

A critical component of this process is that learners are able to practice what they are learning in real time, get feedback on what they are doing, and then apply those lessons learned to the next step in the learning process. Having learners help each other during the workshops also helps to reinforce concepts taught during the workshops.

A Carpentries workshop is an interactive event – for learners and instructors. We give and receive feedback throughout the course of a workshop. We incorporate assessments within the lesson materials and ask for feedback on sticky notes during lunch breaks and at the end of each day. In this way, instructors can adjust the pace and content of a workshop to meet the needs of the learners in that group, as well as collect feedback that will help us improve lesson materials or methodology for future workshops. Instructors and helpers also provide feedback to learners throughout workshops in order to get them unstuck or provide a greater depth of understanding about the materials. Interacting in this way with the learners helps us determine how they are doing and figure out what concepts people are having trouble with or where we might be able to move along more quickly.

The Acquisition of Skill

Our approach is based on the work of researchers like Patricia Benner, who applied the Dreyfus model of skill acquisition in her studies of how nurses progress from

novice to expert (see also books by Benner). This work indicates that through practice and formal instruction, learners acquire skills and advance through distinct stages. In simplified form, the three stages of this model are:
Novice, Competent Practitioner, Expert

- *Novice*: someone who doesn't know what they don't know, i.e., they don't yet know what the key ideas in the domain are or how they relate. One sign that someone is a novice is that their questions "aren't even wrong".

  Example: A *novice* learner in a Carpentries workshop might never have heard of the bash shell, and therefore may have no understanding of how it relates to their file system or other programs on their computer.

  - No expectations of the programs or how to "get to where they're going" and no real ability to research solutions.

- *Competent practitioner*: someone who has enough understanding for everyday purposes. They won't know all the details of how something works and their understanding may not be entirely accurate, but it is sufficient for completing normal tasks with normal effort under normal circumstances.

  Example: A *competent practitioner* in a Carpentries workshop might have used the shell before and understand how to move around directories and use individual programs, but they might not understand how they can fit these programs together to build scripts and automate large tasks.

  - Much Rote knowledge, fragile and unconnected – but comfortable when things are proceeding normally

- *Expert*: someone who can easily handle situations that are out of the ordinary.

  Example: An *expert* in a Carpentries workshop may have experience writing and running shell scripts and, when presented with a problem, immediately sees how these skills can be used to solve the problem.

  - Developed and well articulated mental models which correspond well to the system

Cognitive Development and Mental Models

Effective learning is facilitated by the creation of a mental model. A mental model is a collection of concepts and facts, along with the relationships between those concepts, which a person has about a topic.

We can distinguish between a *novice* and a *competent practitioner* for a given domain based on the complexity of their mental models.

- A *novice* is someone who has not yet built a mental model of the domain. They therefore reason by analogy and guesswork, borrowing bits and pieces of their mental models of other domains which seem superficially similar.
- A *competent practitioner* is someone who has a mental model that's good enough for everyday purposes. This model does not have to be completely accurate in order to be useful: for example, the average driver's mental model of how a car works probably doesn't include most of the complexities that a mechanical engineer would be concerned with.

Mental Models

We will discuss the mental models of experts in more detail in a later lesson.

Your Mental Models

> In the Etherpad, write your primary research domain or area of expertise 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?
>
> This discussion should take about 5 minutes.
>
> Start at: 9:25
>
> End at: 9:30

One key insight from research on cognitive development is that novices, competent practitioners, and experts each need to be taught differently. In particular, presenting novices with a pile of facts early on is counter-productive, because they don't yet have a model or framework to fit those facts into. In fact, presenting too many facts too soon can actually reinforce their incorrect mental model.

Most learners coming to Carpentries lessons are novices, and do not have a strong mental model of the concepts we are teaching. **Thus, our primary goal is *not* to teach the syntax of a particular programming language, but *to help them construct a working mental model*** so that they have something to attach facts to. In other words, our goal is to teach people *how to think* about programming and data management.

The Importance of Going Slowly

> If someone feels it's too slow, they'll be a bit bored. If they feel it's too fast, they'll never come back to programming. — Kunal Marwaha, SWC instructor

The transition from novice to competent practitioner is primarily the construction of a new mental model of this new intellectual domain. The goal of education for novices is, therefore, to help them form the right categories and make connections among concepts. Until they've done that, trying to impart "mere information" just confuses them (and is therefore a waste of time).

As an example of this idea's practical implications, Software Carpentry's lesson on the Unix shell only introduces 22 commands, and we usually spend two and a half hours working through it. Six and a half minutes per command may seem glacially slow, but the lesson's real purpose is to teach learners about paths, history, wildcards, pipes and filters, command-line arguments, redirection, and all the other big ideas on which the shell depends, and without which people cannot understand how to use commands.

We often run out of time three fifths through the shell lesson, even though we have so few commands to teach. That's because building appropriate mental models and having students working along (and making mistakes) slows things down nicely.

That model also includes things like:

- Anything you repeat manually, you'll eventually get wrong (so let the computer repeat things for you by using tab completion and the `history` command).
- Lots of little tools, combined as needed, are more productive than a handful of programs. (This motivates the pipe-and-filter model.)

These two examples illustrate something else as well. Learning consists of more than "just" building mental models and adding information to them; creating linkages between concepts and facts is at least as important. Telling people that they shouldn't repeat things, and that they should try to think in terms of little pieces loosely joined, both set the stage for discussing functions. Explicitly referring back to pipes and filters in the shell when introducing functions helps solidify both ideas.

Meeting Learners Where They Are

> One of the strengths of Carpentry workshops is that we meet learners where they're at. Carpentry instructors strive to help learners progress from whatever starting point they happen to be at, without making anyone feel inferior about their current practices or skillsets. We do this in part by teaching relevant and useful skills building an inclusive learning environment, and continually getting (and paying attention to!) feedback from learners. We'll be talking in more depth about each of these strategies as we go forward in our workshop.

How “Knowledge” Gets in the Way

Mental models are hardly ever built from scratch. Every learner comes to a topic with some amount of information, ideas and opinions about the topic. This is true even in the case where a learner can’t articulate their prior knowledge and beliefs.

In many cases, this prior knowledge is incomplete or inaccurate. Inaccurate beliefs can be termed “misconceptions” and can impede learning by making it more difficult for learners to incorporate new, correct information into their mental models. Correcting learners’ misconceptions is at least as important as presenting them with correct information. Broadly speaking, misconceptions fall into four categories:

- (1) Proposition-level misconceptions. These are "fun facts" that are wrong; knowing whether they are true or false doesn't really affect your life or your vision of yourself. An example is the idea that humans only use 10% of their brains. This is not true.
- (2) Flawed mental models. These are cases where the student has the right metaphor, but the wrong application. An example is how the heart works. Students might know that the heart pumps blood, but not realize that there have to be two separate chambers, one to pump from the lungs to the body (the oxygenated blood), and the other to pump from the body to the lungs (the oxygen-deplete blood).
- (3) Ontological miscategorizations. Ontology is the examination of what is meant by "being," and this kind of misconception is a fundamental misunderstanding about how the world works. It is having the wrong metaphor or lack of a metaphor completely. An example is a circuit with a power source, three bulbs and a switch which closes circuit before the third bulb; not knowing how throwing the switch affects the intensity of the third bulb indicates that you don't know enough about how electricity works to address the question.
- (4) Embedded beliefs. These are misconceptions that are tied to identity, family, culture or community. An example is evolution, where a student accepting this theory of how life became so diverse would have to leave large parts of their social network behind and perhaps fundamentally change what kind of person they imagine themselves to be. It is considered that this kind of misconception is practically impossible to correct.

Since the Carpentries workshops are focused on novices, and the building of strong mental models, we’re most interested in the middle category of misconceptions. While teaching, we want to expose learners’ broken models so that we can help them build better ones.

(More details at: https://github.com/carpentries/instructor-training/issues/584)

Exercise:

Misconceptions we've encountered:

5 minute discussion on some of the misconceptions we've encountered during teaching and how they map to the 4 tier model of: Proposition-level, Flawed mental model, ontological miscategorisation, or embedded belief.

Identifying and Correcting Misconceptions

How do we expose misconceptions, especially as they pertain to broken models? How can we, in-class, know whether the learners already understand this topic (so that the class can move on), and if not, what misconceptions and gaps in their knowledge we should address.
To be effective, instructors need feedback on their learners' progress, and insight into their learners' mental models. This feedback comes through formative assessments.

- *Formative assessment* takes place during teaching and learning. Its main purpose is to provide guidance to the instructor and the learner about what to focus on. Learners don't "pass" or "fail" formative assessments. For example, a music teacher might ask a learner to play a scale very slowly in order to see whether they are breathing correctly, and if not, what they should change.
- In contrast, *summative assessment* is used to judge whether a learner has reached an acceptable level of competence. Learners either "pass" or "fail" a summative assessment. One example is a driving exam, which tells the rest of society whether someone can safely be allowed on the road. Most assessment done in university courses is summative, and is used to assign course grades.

Formative assessments provide feedback to both instructors and learners about learners' level of understanding of the material. For learners, this feedback can help focus their study efforts. For instructors, it allows them to refocus their instruction to respond to challenges that learners are facing.
Formative assessment is most useful when it happens frequently (we'll talk about how frequently later) and when the results are easily interpretable by the learner and instructor.

Repetition vs. Reflective Practice

> The idea that ten thousand hours of practice will make someone an expert in some field is widely known, but reality is much more complex. Practice is not doing the same thing over and over again: practice is doing similar but subtly different things, getting feedback, and then changing behavior in response to that feedback to get cumulatively better. Doing the same thing over and over again is much more likely to solidify bad habits than perfect performance. This

is why we emphasize practice and feedback for learners at our workshops and for trainees in our instructor training program.

Formative Assessments Come in Many Forms

There are many types of formative assessment, and each have their advantages and disadvantages. The most widely used is probably multiple choice questions (MCQs). When designed well, these can do much more than just measure how much someone knows. For example, suppose we are teaching children multi-digit addition.

A well-designed MCQ would be:

```
Q: what is 27 + 15 ?
a) 42
b) 32
c) 312
d) 33
```

The correct answer is 42, but each of the other answers provides valuable insight.

Identify the Misconceptions

Choose one wrong answer and write in the Etherpad what the misconception is associated with that wrong answer. This discussion should take about 10 minutes.

Start at: 9:40

Discuss at 9:45

End at: 9:50

Solution

- If the child answers 32, they are throwing away the carry completely.
- If they answer 312, they know that they can't just discard the carried '1', but doesn't understand that it's actually a ten and needs to be added into the next column. In other words, they are treating each column of numbers as unconnected to its neighbors.
- If they answer 33 then they know they have to carry the 1, but are carrying it back into the same column it came from.

Each of these incorrect answers is a *plausible distractor* with *diagnostic power*. "Plausible" means that it looks like it could be right: instructors will often put supposedly-silly answers like "a fish!" on MCQs, but (a) they don't provide any insight and (b) learners actually don't find them funny. "Diagnostic power" means that each of the distractors helps the instructor figure out what concepts learners are having difficulty with.

Formative assessments are most powerful when an instructor modifies their instruction depending on the results of the assessment. An instructor may learn they need to change their pace or review a particular concept. Knowing how to respond to the results of a formative assessment is a skill that you will develop over time.

Handling Outcomes

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

1. mostly one of the wrong answers?
2. mostly the right answer?
3. an even spread among options?

For one of the above, enter your answer in the Etherpad.

This discussion should take about 10 minutes.

Start at: 9:50 (skip if prior went long)

End at: 10:00

Solution

1. If the majority of the class votes for a single wrong answer, you should go back and work on correcting that particular misconception.
2. If most of the class votes for the right answer, it's probably safe to move on.
3. If answers are pretty evenly split between options, learners are probably guessing randomly and it's a good idea to go back to a point where everyone was on the same page.

Modeling Novice Mental Models

Take 10 minutes to create a multiple choice question related to a topic you intend to teach. Type it into the Etherpad and explain the diagnostic power of each its distractors, i.e., what misconception is each distractor meant to identify?

Start at: 10:05

End at: 10:15

A Note on MCQ Design

- A good MCQ tests for conceptual misunderstanding rather than simple factual knowledge. If you are having a hard time coming up with diagnostic distractors, then either you need to think more about your learners' mental models, or your question simply isn't a good starting point for an MCQ.
- When you are trying to come up with distractors, think about questions that learners asked or problems they had the last time you taught this subject. If you haven't taught it before, think about your own misconceptions or ask colleagues about their experiences.

Designing an MCQ with plausible distractors is useful even if it is never used in class because it forces the instructor to think about the learners' mental models and how they might be broken—in short, to put themselves into the learners' heads and see the topic from their point of view.

There are many types of formative assessments other than MCQs. One (non-exhaustive) list can be found in the Edutopia assessment group.

Formative Assessments Should Be Frequent

Instructors should use a formative assessment ideally every 5 minutes and at least every 10-15 minutes in order to make sure that the class is actually learning. Since the average attention span is usually only this long, formative assessments also help break up instructional time and re-focus attention. Formative assessments can also be used preemptively: if you start a class with a question and everyone can answer it correctly, then you can safely skip the part of the lecture in which you were going to explain something that your learners already know.

How Many?

The Carpentries use formative assessments often. How many have we done since the start of this workshop? Put your guess in the Etherpad along with one example and the purpose that assessment served.

This discussion should take about 5 minutes.

Solution

This will depend on the event they are attending. Most attendees will guess low. The purpose of this exercise is to emphasize the importance of frequent formative assessments and that an individual assessment doesn't have to take a lot of time.

A Corollary Regarding Novices

Learners are commonly *far* too satisfied to not understand key points and remain confused. If learners remain confused, they are far less likely to adopt our “good enough” practices at the conclusion of the workshop. Encourage helpers to support confused learners as far as possible within the flow of the workshop.

Optional exercises

Confronting the Contradiction

Describe a misconception you have encountered in your own learning or teaching and how to get learners to confront it.

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.

Expertise and Instruction

Overview

Start at: 10:20

End at: 11:10

Questions

- What type of instructor is best for novices?
- How are we (as instructors) different from our learners and how does this impact our teaching?

Objectives

- Identify situations where you have an expert blind spot.
- Demonstrate strategies for compensating for your expert blind spot.

- Demonstrate strategies for avoiding demotivating language.

We now discuss what distinguishes expertise from earlier stages of learning, how being an expert can make it more difficult to teach novices, and some tools to help instructors identify and overcome these difficulties.

What Makes an Expert?

An earlier topic described a key difference between novices and competent practitioners. Novices lack a mental model, or have only a very incomplete model with limited utility. Competent practitioners have mental models that work well enough for most situations. How are experts different from both of these groups?

What Is An Expert?

1. 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?
2. 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?

This discussion should take about 5 minutes.

The answer is not that experts know more facts: competent practitioners can memorize a lot of information without any noticeable improvement to their performance. The answer is rather that experts have more connections among pieces of knowledge; more “short-cuts”, if you will.

One way to illustrate this is to model storing knowledge as a graph in which facts are nodes and relationships are arcs. (This is emphatically *not* how our brains work, but it’s a useful metaphor.) The key difference between experts and competent practitioners is that experts have many more connections among concepts. Their mental models are much more densely connected. Therefore experts can jump directly from a problem to its solution because there is a direct link between the two in their mind. Where a competent practitioner would have to reason “A therefore B therefore C therefore D therefore E”, the expert can go from A to E in a single step (“A therefore E”).

Connections and Mental Models

The graph model of knowledge explains why helping learners make connections is as important as introducing them to facts. The more connections a fact has to other facts, the more likely the fact is to be remembered. This builds on our earlier idea of mental models - a mental model is (in part) a set of connections or relationships among facts or concepts.

Limitations of Expertise

Because your learners' mental models will likely be less densely connected than your own, a conclusion that seems obvious to you will not seem that way to your learners. Another feature of expertise that has important consequences for teaching is the ability of experts to make use of *fluid representations*. Two ways of thinking about a problem will seem interchangable to an expert, but will not seem that way to a novice. For example, someone with experience using the bash shell will be able to change back and forth between absolute and relative paths with no difficulty and in fact may not even notice they are doing so. A novice learner, however, would be confused by this unexplained use of two different ways of representing a concept.

More Examples of Fluid Representations

- Programming: Referring to an R object like `abcde` as both "character vectors" and "strings".
- Programming: Switching among `df[,1]`, `df[,'foo']`, and `df$foo` notation when talking about columns in a data.frame.
- Biology: Switching between common species names and Latin names (e.g. "mouse" vs "*Mus musculus*").
- Biology: Using both three letter and one letter amino acid codes interchangeably (e.g. Lys = K = Lysine).
- Chemistry: Switching between "Reference material" and "Standard"
- Mathematics: Thinking of things algebraically vs geometrically.
- Navigation: Switching among different routes between two locations.

Fluid Representations

> In the Etherpad, give at least one example of a fluid representation that you use in your own work. If you can, also give an example of a fluid representation that might occur in a Carpentry lesson.
>
> Building awareness of how you can represent the same concept in multiple different ways will help you avoid doing so without explanation while teaching.
>
> This discussion should take about 5 minutes.

Experts are also better at diagnosing errors than novices or competent practitioners. If faced with an error message while teaching, an expert will often figure out the cause of the error and develop a solution before a novice has even finished reading the error message. Because of this, it is very important while teaching to be explicit about the process you are using to diagnose and correct errors, even if they seem trivial to you, as they often will.

Diagnosis (Optional)

> 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 explanation in the Etherpad.
>
> This discussion should take about 5 minutes.

Experts are frequently so familiar with their subject that they can no longer imagine what it's like to *not* see the world that way. This is called *expert blind spot* and can lead to what's known as the *expertise-reversal effect* - experts are often less good at teaching a subject to novices than people with less expertise who still remember what it's like to have to learn the things. This effect can be overcome with training, but it's part of the reason world-famous researchers are often poor lecturers.

Blind Spots (Optional)

1. Is there anything you're learning how to do right now? Can you identify something that you still need to think about, but your teacher can do without thinking about it?
2. Think about the area of expertise you identified earlier. What could a potential blind spot be?

The challenge of identifying and working around expert blind spots is one reason why we welcome instructors who still identify as "novices"! Someone who is still in the process of learning can be a more effective instructor because they are speaking from their own recent experience.

In these ways and others, the high connectivity of an expert's mental model poses challenges while teaching novices. However, that's not to say that experts can't be good teachers. Experts can be effective as long as they take the time to identify and correct for their own expert blind spots. You can use some of the exercises we've done while preparing to teach to help you overcome these challenges.

Dismissive Language

Experts often betray their blind spot by using the word "just" in explanations, as in, "Oh, it's easy, you just fire up a new virtual machine and then you just install these four patches to Ubuntu and then you just re-write your entire program in a pure functional style—no problem." This gives learners the very clear signal that the instructor thinks their problem is trivial and that they therefore must be stupid.

With practice, we can change the way we speak to avoid this type of demotivating language and replace it with more positive and motivating word choices.

Changing Your Language

What other words or phrases can have the effect of demotivating learners? What alternatives can we use to express this meaning in a positive and motivational way?

In the Etherpad, make a list of demotivating words/phrases and alternatives.

This discussion should take about 5 minutes.

Solution

Courtney Seiter lists 10 words and phrases that can change a conversation: *if*, *could*, *yes*, *together*, *thank you*, *choose to*, *and*, *because*, *willing*, and the person's name. These are motivating words and phrases that can shift mindsets. Jason Fried lists several dirty four-letter words: *need*, *must*, *can't*, *easy*, *just*, *only*, and *fast*, as well as examples of how they are used to demotivate. Statements like:

- "We really need it."
- "If we don't we can't …"
- "Wouldn't it be easy if we just did it like that?"
- "Can you try it real fast?" can be perceived as dismissive or demeaning or worse.

Another language choice that can have very positive effects on learner mindset is to ask "What questions do people have?" rather than "Does anyone have any questions?" Asking "Does anyone have any questions?" can create the impression that you hope people don't have questions, so that you can continue on with the lesson. By asking what questions people have, you are setting up an expectation that people will, indeed, have questions, and that that is normal and expected.

You Are Not Your Learners

One way to overcome these limitations is by understanding the goals and motivations of your learners. We will discuss motivation in more depth in a later lesson but for now, consider some of these ideas about the typical audience for Carpentry workshops.

- Most scientists' primary goal is not to program, but to do scientific research. They may not care how hash tables work, or even that hash tables exist; they just want to know how to process data faster. We therefore have to make sure that everything we teach is useful right away, and conversely that we don't teach anything just because it's "fundamental".
- Believing that something will be hard to learn is a self-fulfilling prophecy. Researchers already believe that computing is hard, a belief that is exacerbated

> by “experts” telling them that things should be easy when, in the researcher’s experience, they’re not. Paradoxically, this is why it’s important not to say that something is easy. If someone who has been told that tries it, and it doesn’t work, they are more likely to become discouraged.

It’s also why installing and configuring software is a much bigger problem for us than experienced programmers like to acknowledge. It isn’t just the time we lose at the start of workshops as we try to get a Unix shell working on Windows, or set up a version control client on some idiosyncratic Linux distribution, or ask people to download and unzip files. It isn’t even the unfairness of asking learners to debug things that depend on precisely the knowledge they have come to learn, but which they don’t yet have. The real problem is that every such failure reinforces the belief that computing is hard, and that they’d have a better chance of making next Thursday’s conference submission deadline if they kept doing things the way they always have. For these reasons, we have adopted a “teach most immediately useful first” approach. We’ll talk much more about this when we discuss motivation.

The Carpentries Is Not Computer Science

> Many of the foundational concepts of computer science, such as computability, are difficult to learn and not immediately useful. This does *not* mean that they aren’t important, or aren’t worth learning, but if our aim is to convince people that they can learn this stuff, and that doing so will help them do more science faster, they are less compelling than things like automating repetitive tasks.

The Importance of Practice (Again)

All of the above points illustrate the importance of using formative assessments frequently. The right formative assessment at the right time will give you valuable information about your learners’ goals and motivations, making it easier for you to target your lesson materials to their needs. This strategy also helps you as an instructor overcome your expert blind spot. It doesn’t matter how easy you think a task is, if your learners aren’t getting it, it’s probably more complicated than you thought.

Key Points

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