Welcome

1. Who are Data and Software Carpentry and how do we approach teaching?
2. What should you expect from this workshop?

Welcome

- Code of Conduct (SLIDE)
- Introduce yourself and your co-instructor
- Intro exercise - quick verbal intro, work in three words, something you're proud of
- Exercise & intro to etherpad - name, best class you ever took, why it was so good
- encourage note-taking
    - Carpentry History & Culture - more about the projects, community, and culture
    - Exercise: assessing motivation and prior knowledge

Welcome

- Overview of SWC & DC
  - SWC: helps researchers develop foundational comp skills
  - DC: helps researchers work effectively with their data

Welcome

- Overview of curriculum:
  - based on current state of research into effective teaching and learning techniques
  - How Learning Works - theory around how people learn new things
  - Building Teaching Skill - some practices to harness the knowledge learned above
  - Creating a Positive Learning Environment - environment influences learning
    Carpentry History & Culture - more about the projects, community, and culture
- Exercise: assessing motivation and prior knowledge

How Learning Works: The Importance of Practice

1. How do people learn?
2. Who is a typical Carpentry learner?
3. How can we help novices become competent practitioners?

The Importance of Practice

- Carpentries take an applied approach to teaching
  - avoid theory in favour of practise
- practise *while* learning
- regular feedback for learners and instructors
  - helps us to adjust pacing and content

The Importance of Practice

- approach is based on research of Patricia Brenner (SLIDE)
  - applied Dreyfuss model of skill acquisition to study of nurses
- Novice -> Competent Practitioner -> Expert

Mental Models

- progress between these stages facilitated by development of **mental model**
  - collection of concepts and facts and the relationships between them
  - example... resident of the USA, cricket captain, chef..?

Mental Models

- can distinguish between stages above by complexity of mental model
  - novice: no mental model, reason by analogy and guesswork. don't know enough to even ask the right questions
  - competent practitioner: model good enough for everyday, most cases
  - more about experts later
- Exercise: your mental models

Mental Models

- people at each of these stages need to be taught differently
  - novices need a framework to fit facts into before they're presented with them
  - avoid reinforcing an incorrect mental model - they will fit facts into whatever framework they do have
- our primary goal is **not** to teach the syntax of a particular programming language
  - 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.

Go Slowly

- go slowly (SLIDE)
  - emphasis is on correctly categorise concepts/ideas and form connections between them
  - e.g. Shell lesson: only introduces 22 commands in ~2.5 hours
  - value is in teaching paths, history, wildcards, pipes, arguments, redirection, etc
  - allows them to reduce manual entry (tab completion) and understand that you can combine simple things into complicated processes

Misconceptions

- mental models are hardly ever built from scratch
- prior knowledge can be accurate or inaccurate (misconceptions), relevant or irrelevant
- **misconceptions** - broadly split into three types
    - factual errors
    - broken models (e.g. motion and acceleration must be in same direction) - fix by reasoning through examples
    - fundamental beliefs - often deeply connected to social identity and hard to change
- our workshops focus on correcting the middle category

Formative Assessment

- how to identify misconceptions?
- two types of assessment: formative and summative (SLIDE)
    - formative: takes place during learning
    - summative: judge competence after teaching
- we use many forms of formative assessment in our workshops
    - most effective when used frequently
    - encourages reflective practice instead of simple repetition

MCQs

- MCQs are one type of formative assessment (SLIDE) - more types introduced later
- **put e.g. in etherpad and on screen, talk through it, exercise**
- every wrong answer should be a plausible distractor with diagnostic power

Formative Assessment

- formative assessments are most powerful when instructor modifies instruction in reaction to result
- **exercises**
- **how long do you think the average attention span is?**
  - (SLIDE) every 5-15 mins is optimal for assessment - matches this average

How Learning Works: Expertise and Instruction

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

Expertise & Instruction

- what makes an expert?
  - **exercise**
- experts don't (necessarily) know more facts (SLIDE)
  - instead they make more connections, can proceed quickly from one step to next (SLIDE)
- when it comes to teaching, there are downsides to being an expert

Fluid Representations

- experts make use of fluid representations
  - e.g. switch effortlessly between relative, absolute paths and those stored as variables
  - other example 'character vectors' and 'strings' in R, switching between mouse and *Mus musculus*, driving different routes between locations

Error Handling

- how experts deal with errors is also different from how a novice/CP does the same
  - crucial to talk through the process as you fix errors when teaching
- **exercise: diagnosis**

Expert Blind Spot

- expert blind spot: forgetting/inability to imagine life *not* knowing something
  - "expert-reversal effect" - experts make worse teachers
  - **exercise: Blind spots**
- we welcome instructors who still identify as novices/CPs. experts just as welcome but will need to work harder to overcome blind spots

Dismissive Language

- “just"
  - e.g. Docker, HPC cluster, adapt script to accept multiple command line arguments
- **exercise**
- "what questions do you have?": setting expectations

Dismissive Language

- “just"
  - e.g. Docker, HPC cluster, adapt script to accept multiple command line arguments
- **exercise**
- "what questions do you have?": setting expectations

You are not your learners

- motivations, priorities may differ
- things you find interesting may be boring to them
- **when someone tells you that they find something hard, believe them**

Coffee!

(SLIDE)

How Learning Works: Working Memory and Cognitive Load

1. What is cognitive load and how does it affect learning?
2. How can we design instruction to work with, rather than against, memory constraints?

Cognitive Load

- final topic in educational psychology is idea of cognitive load

Types of Memory

- long-term/persistent memory
    - essentially unbounded but slow access
- short-term/working memory
    - what you use to actively think about things
    - really fast
    - limited space
    - original research suggested this was 7±2 things
    - this is why phone numbers are ~this length
    - more recent research suggests it's even less than this
- important when teaching to be aware of this limit and design/pace material accordingly
    - exercise

“Chunking”

- "chunking" allows us to remember more things (SLIDES)
  - e.g. familiar patterns of letters (words) and shapes (e.g. dots on a die)
- this is why concentrating on connections between concepts is important in a workshop

Concept Maps

- formative assessment is an opportunity for learners to transfer information from short-term to long-term memory
- concept maps help us to plan our teaching to successfully manage cognitive load
    - e.g. for loop in Python
- SLIDES

Concept Maps

- don't worry if you do this and your map is too large
  - helps to judge where to insert new assessments and how to split up sections
- exercise: concept mapping

Guided Practice

- all of this is about providing guided practice
  - setup structure for learners to test their understanding and receive regular feedback
  - doesn't require learners to simultaneously master a domain’s factual content and its search and problem-solving strategies
- one researcher splits cognitive load into three categories:
  - **intrinsic**: what they have to keep in mind in order to carry out a learning task
  - **germane**: the (desirable) mental effort required to create linkages between new information and old
  - **extraneous**: everything else that distracts or gets in the way
  - proponents of CL theory suggest that eliminating extraneous load accelerates learning
- point to faded examples in etherpad, but move on

Building Teaching Skill: Getting Feedback

1. How can I get feedback from learners?
2. How can I use this feedback to improve my teaching?

Surveys

- to help us adjust our instruction and assess the impact of our teaching, we gather feedback at various points
- provide pre- and post-workshop assessments to learners

Minute Cards

- prompt for specific types of feedback, e.g.
  - "what was most important thing you learned?"
  - "what was one this that you found confusing?"
- summarise feedback you got and talk about what you're going to do about it

One Up, One Down

Give Us Feedback

- last exercise before lunch