Welcome to Software Carpentry Open Online Instructor Training!

2017-10-30 and 2017-10-31

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

Users are expected to follow our **Code of Conduct**:

http://software-carpentry.org/conduct.html

Zoom meeting room:
https://carpentries.zoom.us/j/8243150376

Sign in: Name, Institution, Email, Twitter (optional), github (optional)

Please sign in so we can record your attendance.

Your instructors for today:

- Anita Schürch, UMC Utrecht, the Netherlands, a.c.schurch@umcutrecht.nl, @AnitaSchurch, @aschuerch
- Mateusz Kuzak, Dutch Techcentre for Life Sciences mateusz.kuzak@gmail.com, @mkuzak, @matkuzak

Participants

- Yo Yehudi, yochannah@gmail.com, t: @yoyehudi, g: @yochannah
- Vasu Venkateshwaran, W.L. Gore and Associates, vasu.chemical@gmail.com
- Diogo Aguiam, Instituto Superior Técnico, Universidade de Lisboa, Portugal diogoaguiam@gmail.com, @diogoaguiam, @daguiam
- Bagus Tris Atmaja, JAIST, btatmaja@gmail.com @btatmaja @bagustris
- Andrew Sanchez, Pathogen Microbiome Institute, inbox.asanchez@gmail.com, t: @informandrew, g: andrewsanchez
- Alice Minotto, Earlham Institute, alice.minotto@earlham.ac.uk, @rosysnake, @aliceminotto
- Désirée Treichler, University of Oslo, Norway, desiree.treichler@geo.uio.no, github: @desireetreichler
- David De Sancho, NanoGUNE, daviddesancho@gmail.com, t: @daviddesancho
- Ahmed Moustafa, American University in Cairo, amoustafa@gmail.com, t: @ahmedmoustafa
-

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.

Alice: neurobiology at Uni, because we were given lot of freedom in choosing the topics and our main assignment was of our own choice so we were actually interested
Andrew: a class on the music of Bach I took last year. Why? The teacher was very passionate, he was an expert, and he was not overly focused on grades/performance. He literally graded students based on enthusiasm and passion.

Yo: My first JavaScript class. I'd always wanted to make computers go but hadn't managed to figure it out on my own. Suddenly I had the power!
David: Masterworks in Spanish Literature. Basically an off-topic course I took while I was Biochemistry undergrad. A scape from transcription factors and regulatory pathways taught by a professor who was a sort of Robin-Williams-in-Death-Poet's-Society with a Quixotesque touch to him, precisely at the time of your life when you need literature most.
Diogo: Chemistry class in first year university when I didn't understand anything, the professor was pretty bad, but then it clicked after my father, a Chem Engineer, explained it to me. Every following class was great afterwards because I had the basics well understood
Ahmed: High school physics because it put math into nature and action.
Vasu: High school mathematics class. The instructor was brilliant and taught the class without a blackboard.
Désirée: A block course in glaciology (glacier science) in Svalbard, that's an archipelago at ca. 78 degrees N north of Norway. It was so good because it was very intense and hands-on - partly because the glaciers are right next door, but to a large degree because the community feeling made it impossible not to participate (the university centre there is in a very small village, and the setting was a little bit like a scout's camp).
Bagus: MIR (music information retrieval) class by Thomas Lidy (TU Wien) because it provides a step-by-step to retrieve information from music signal along with its explanation and physical meaning.

Background

1. Have you ever participated in a Software Carpentry or Data Carpentry Workshop?
    A. Yes, I have taken a workshop.
    B. Yes, I have been a workshop helper.
    C. Yes, I organized a workshop.
    D. No, but I am familiar with what is taught at a workshop.
    E. No, and I am not familiar with what is taught at a workshop.

2. Which of these most accurately describes your teaching experience?

A. I have been a graduate or undergraduate teaching assistant for a university/college course.
B. I have not had any teaching experience in the past.
C. I have taught a seminar, workshop, or other short or informal course.
D. I have been the instructor-of-record for my own university/college course.

E. I have taught at the K-12 level. (K–12 comprises the sum of primary and secondary education in India, the United States)
F. I have taught informally through outreach programs, hackathons, laboratory demonstrations, and similar activities.

3. Which of these questions assesses flaws in a student's mental model of a domain? You don't need to provide answers for these questions.

A. I'm not sure what a mental model is.
B. "In Python, what is the expected output for the following statement: 1 + '2'" (a) '12' (b) TypeError (c) '3' (d) 3
C. "Rate your experience with the R programming language." (a) never used it (b) beginner (c) intermediate (d) expert
D. "What does the Unix command 'cut' do?" (a) Extracts sections from each line of input. (b) Sorts fields of a line (c) Searches the input file for lines containing a match to a pattern (d) Removes a given input from a line

Put an asterisk behind your answer

1.

A. *
B. * *

C.*
D.** ***
E.

2.
A. **
B.*
C.******
D.*
E.
F.****

3.
A.***
B. *****
C.
D.*

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?

Andrew: My primary research domain is in developing software for genomics research. I use mental models from both biology and programming to understand my work. This

includes facts of microbiology, genomics, databases, research techniques, python data models and tools, and linux. Most important relationships are those with users of my software, primarily my supervisor and others in our lab.

Mateusz:
http://dataphys.org/list/wp-content/uploads/2014/12/Watson-Crick-DNA-model.jpg

Diogo: I work in Reflectometry to measure plasma densities in Nuclear Fusion reactors. This is similar to a RADAR system. I send microwaves to the plasma to measure distances and then reconstruct the shape of the plasma. I use models of electronics, signal processing and the physics behind microwave propagation in plasmas.

Alice: I administer the cloud instance of the project, this means having an idea of the network and which virtual machines need to communicate with each other and who has to be able to log in and work there.

Yo: I create software for biologists, which has both APIs and UIs to allow them to access data. But when I'm explaining this to someone who doesn't have computational and/or biological knowledge, they may not understand what I'm trying to say if I phrase it like this. Instead, I'll try to guess what their mental model might be, and phrase is more carefully - what I do is I take data from experiments and make websites (not UIs) where scientists can search the experiment data, as well as creating ways for programmers to write code to access the data (not APIs).

Ahmed: I teach R (for students with no programming background). A mental model that I try to explain is looping. The included concepts are the need for iteration by looping and that at each iteration of the loop, the code is executed once.

Bagus:I teach Unix command line. When I explaining about directory, I make an analogy to cabinet or locker, a place to save something. Something, for example, is file. By using this analogy it will easier to explain command like mkdir, cp, mv and rm.

Vasu: Physics and Applied Mathematics background. I develop mathematical descriptions of processes using physical laws. Typically most models are constructed by writing down equations that describe "evolution" and "restrictions" in a particular system. E.g. mass has to be conserved, material properties changes in some characteristic way.

David: I work in Protein Biophysics and my primary interest is protein folding. Proteins are molecular entities that are able to fold into three dimensional structures. In order to understand this process, we make experiments and simulations. With these we measure the thermodynamics and kinetics of the system, which are connected using fundamental relationships: the partition function and the master equation.

Désirée: Spectral remote sensing uses parts of the light spectrum that are different from the red, green and blue light waves we see - for example, different kinds of infrared. Explaining and visualising this can be very challenging for novices (e.g. a false colour composite, where

we use the screen's red, green and blue colours to display completely different parts of the light spectre). Helping the students to form their model requires that they understand the spectrum of light, and a useful way of visualising multispectral data is a cube, where the x and y axis are the spatial domain (like a grayscale image), and the z axis composes the different channels - this gives a 3d matrix.
https://image.slidesharecdn.com/hyperspectralimaging-150220022926-conversion-gate02/95/hyperspectral-imaging-8-638.jpg?cb=1424399572

Identify the Misconceptions (3 min)
Choose one wrong answer 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: (a) The answer is giving the result of the sum, while the question is asking what is "27 + 15", which in fact is a sum of two integers.
Désirée: b) - forgot to update the tens digit, only added 5 to 27
Ahmed: c) misunderstanding about the positions of the numbers (ones vs. tens). b) Forgetting to carry the one ten (from 5+7) to the tens position. d) Does not know how to add 5 to 7?
Alice: b) forgot to add the tens that come from 5+7 = 12 back to the 2+1 sum in the first position
Yo: B: the student erroneously added the first two digits and the last two digits, but forgot that when adding the final two digits results in a sum over ten, that new ten needs to be added to the total as well.
Diogo: b) Didn't add the +1 to the second digit
Andrew: c) placed a 1 in the tens place instead up increasing the number that was already there.
Bagus: b) forget to add 1 from the right side to the left

Vasu: a) Correct model, b) & c) Misconception on carrying over d) Fundamental problem with number system

Modeling Novice Mental Models (10 min) until 11:18
Create a multiple choice question related to a topic you intend to teach.
Type the question and the a,b,c,d, options into the Etherpad
Explain the diagnostic power of one distractor, i.e., what misconception is each distractor meant to identify?

Désirée: Students need to use geospatial software (geomatica) and data on a UNIX server (the K drive), both of which they can't access from elsewhere easily. To work efficiently, they need to know what is where and have a mental model of the network.
Question: How can you access your remote sensing project from home?

A. I start geomatica from the window start button and work the same way as at the University.
B. I map the network drive K: as shown in the lab, then work as from the University
C. I start geomatica from the window start button and work on the project saved on the memory stick.
D. I use a remote desktop connection to the University's windows server, there I have both the software and data.
E. I connect to the UNIX environment via the shell, there I have both the software and data.

A and B won't work, as the network drive can't be accessed from outside the network - even if they have installed the software locally. C, D and E would work - and D/E are of course smarter.

Yo: Question: In Javascript, which of these statements is false?

1. `0 == "0"` ← bad answer, designed to check if students understand that == doesn't test for full equality. A string is not the same as an integer!
2. `0 === 0` ← bad answer - this evaluates true, is checking that students understand the **===** operator correctly. It checks for strict equality.
3. `0 === "0"` ← correct answer. 0 the integer and "0" the string are different, and === will get angry.
4. `0 == false` ← this is another bad answer. Checks that the students understand javascript truthy values (0 is non-truthy) and that they understand the difference between == and ===. == is non-strict equality comparison, and will return true if both values are both falsey or both truthy, even if they aren't identical.

Alice: To register an application the user needs to write a small wrapper script. It's easier if all the output file are in the same folder "output" at the end. If the outputs are filtered_data.csv 0.log 0.err and 0.out, what should you write as last command (given the folder exist)?

a) Mv 0.log 0.err 0.out filtered_dat.csv output/
b) Cp 0.log 0.err 0.out filtered_dat.csv output/ (used copy command instead of move)
c) Mv output/ 0.log 0.err 0.out filtered_data.csv (mistake in the order of the arguments)
d) Mkdir output (didn't consider the folder exist? Also didn't move the files)

David: What is a pandas dataframe?

(a) It is an object -> the answer is unhelpful, as everything in python is an object
(b) It is a python data structure with columns -> True answer!
(c) It is the python equivalent of a spreadsheet -> the student is confounding what he uses the dataframe for or its representation for its actual definition
(d) It is a digital picture of bears native to central China -> the student really does not know what we are talking about.

Diogo: Question: How is value of *var* and *var2* at the end of the program:

```
var = 1
var2 = 0
def change_var(var):
    var = var + 1
    return var
var2 = change_var(var)
```

a) var = 1, var2 = 2 - correct
b) var = 2, var2 = 0 - incorrect, The var inside *change_var* does not change
c) var = 1, var2 = 1 - incorrect var2 takes the value of var+1
d)

Andrew:
Q: How do you look at a single line that contains text “x” in file Y.txt?
A. cat Y.txt | grep x
B. grep x Y.txt
C. Use `less Y.txt` then search for x by typing /x
D. Open Y.txt in a text editor then search for x ← Student may not understand the importance of gaining fluency in basic unix commands for productivity.

Bagus:
How to remove unempty directory using unix command?
a) rm `dirname` ← novice will assume rm can remove dir
b) rmdir `dirname` ← some novices know about rmdir command but the don’t know it only works for empty directory
c) rm -i `dirname` ← novice is assuming by using `interactive` flag they can remove unempty directory
d) rm -fr `dirname`

Vasu: In python, what does 4 + 3 % 5 evaluate to
a) 4 (may not know what mod operation is)
b) 7 (has the correct model)

c) 0 (I did not know other misconceptions the student would make, so …)
d) 2 (Order of precedence misconception)

Ahmed: In the following code, what will be printed on the screen

```
sum = 0
for (i in 1:5) { sum = sum + i  ; print sum}
```

a) 0 – The loop does not change the value of the sum
b) 1,2,3,4,5 – The assignment just puts the value of i in the sum
c) 15 – The correct answer

What do you do when most people answer… (until 11:34)

a. most right
    i. Proceed with the lesson. All is well! :)
    ii. If you don't know their answers/explanations: a quick summary on why right is right (and maybe what could have been wrongly perceived)
    iii. Explain why the other answers are wrong
    iv. Ask the wrong students, why they reached the wrong answered.
    v. Brief summary, ask whether there are questions and continue.

    Exper answer are wrong
    Explain briefly why is that correct

b. most wrong
    i. Figure out which bits students don't understand and try to explain, maybe in a different way
    ii. Reconsider the model itself.
    iii. Unless it was a trick question on purpose, go back to the explanation as it was probably not clear
    iv. Reiterate key points from the lesson - Walk through step by step the question and how you solve it - Ask another question after to see if people have grasped the concepts
    v. Acknowledge defeat due to instructor's fault and repeat
c. equal distribution
    i. Consider getting students who understand the concept to tutor the ones who don't!
    ii. I agree with the above. Getting other students who answered correctly to explain how the arrived at the correct answer can give students who are struggling a different perspective than was given by the teacher.

iii. Get the students who understand help those who don't in 5 extra minutes
iv. See the commonality of the wrong answers.

Break. meet at 11:50

What differentiates the expert from the novice or competent practitioner?

Experts have more background knowledge and memory "shortcuts" that allow them to think and link much faster than novices, that (still) need to make every step in a cognitive process

David: Self awareness
Y: Experts think things are easy when they don't remember how long it took them to learn
Diogo: Experts understand everything underneath. They have the whole mental model in RAM, while the competent practitioner may still need to google stuff.
An expert may point to the best/more efficient way to do something, but that may also be conceptually hard to get for a novice, and it would be best for him to be explained an easier way.
Andrew: Experts may not be aware of all of the prerequisites/dependencies that their understanding relies on.

Bagus:Expert usually knows earlier than competent practitioner and novice. He knows how to master knowledge.

Vasu: Has abstracted core ideas beyond a rigid set of rules. Know what is possible.

Exercise: *Dismissive* Language

"All you have to do is"
"It's easy"
"It won't take long" ← when it does, you feel bad
Andrew: "simply," "as you know," "obviously,"
"It is just ...."
"As you probably already know...."
"It's obvious that..." "clearly (not-at-all-clear consequence)"
"But you already know this"
Abbreviations: HPC, API
"I thought I had already explained blah, blah, blah..."

types of memory:

Andrew: semantic, spatial, visual, episodic
Short and long term
Short and long term memory :)
Offloaded memory: I don't have to remember because it's in my notes

Muscle memory, things you know without even thinking

short term memory exercise:
This website

Ahmed: 5 words, 2 start, 1 middle, 2 end.
Désirée: 2 words, 1 start, one middle - got completely thrown out by you talking (;
Diogo: 7 correct words, 3 beginning, 3 middle, 1 end
Yo: 5 words: 1 start, 1 middle, 3 end.
Bagus: 6 correct words: 2 start, 2 middle, 2 end
Alice: 9 words, 2 start, 3 middle, 4 end
David: 3 words, all end (!)
Vasu: 7 words correct, 2 beginning, 2 middle, 3 end
Andrew: 1, 2, 3, 6 total

Concept maps:

- Array Math
- Conditionals
- Creating and Destroying Files
- Sets and Dictionaries in Python
- Input and Output
- Lists and Loops
-

Faded Examples

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

```
    return total

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

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

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

Surveys:

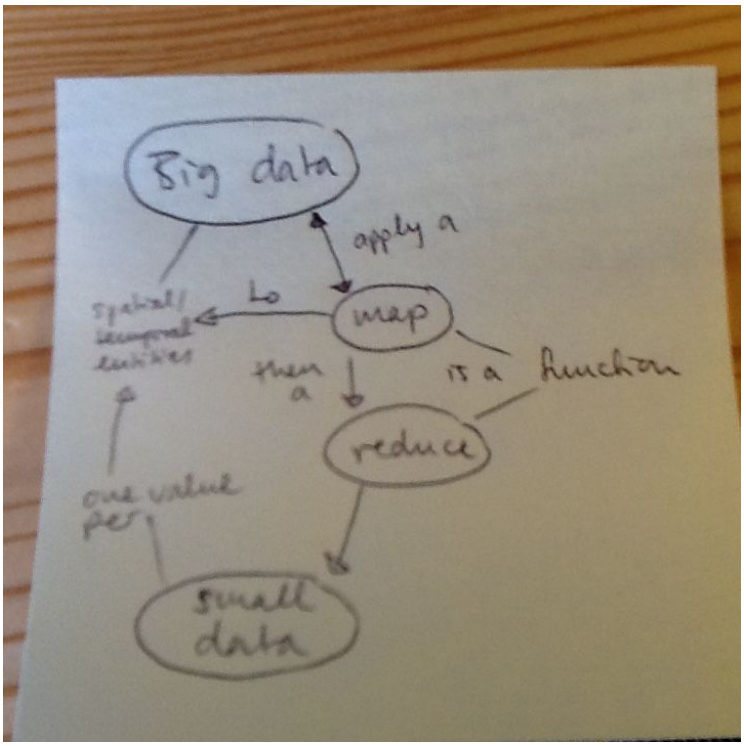
- Software Carpentry pre-workshop survey
- Software Carpentry post-workshop survey
- Data Carpentry pre-workshop survey
- Data Carpentry post-workshop survey

Give us feedback:
https://goo.gl/forms/3jW9bHIxApq1nWNP2

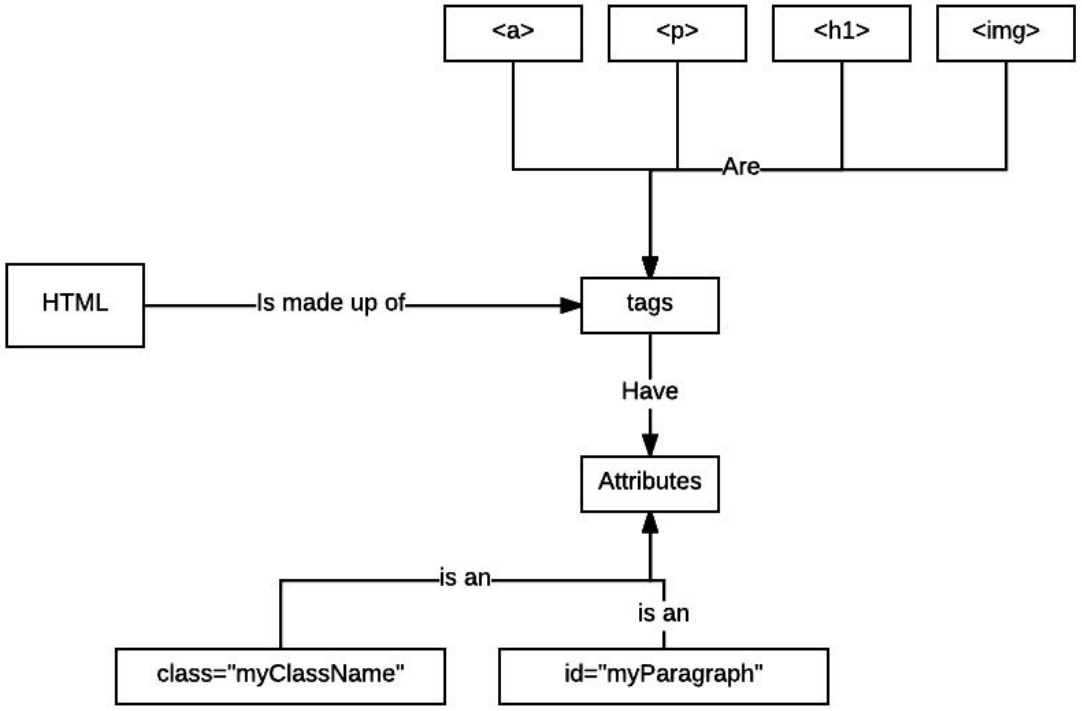
Please paste your concept maps here:
Create a hand-drawn concept map for a part of a Carpentry lesson you would teach in five minutes (ie. the amount of material you would teach before doing a formative assessment). You can use the same subject about which you created a multiple choice question, or a 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?

Back at 13.45

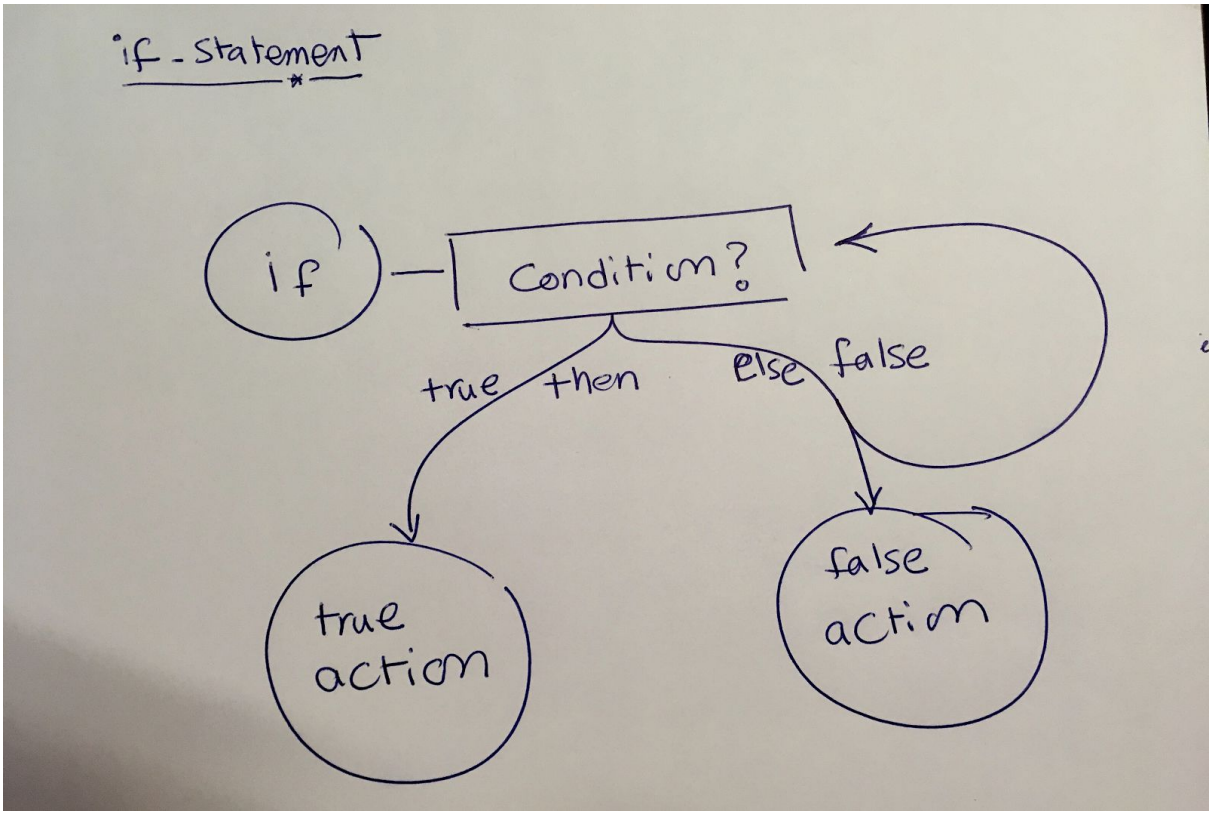
Désirée: map-reduce



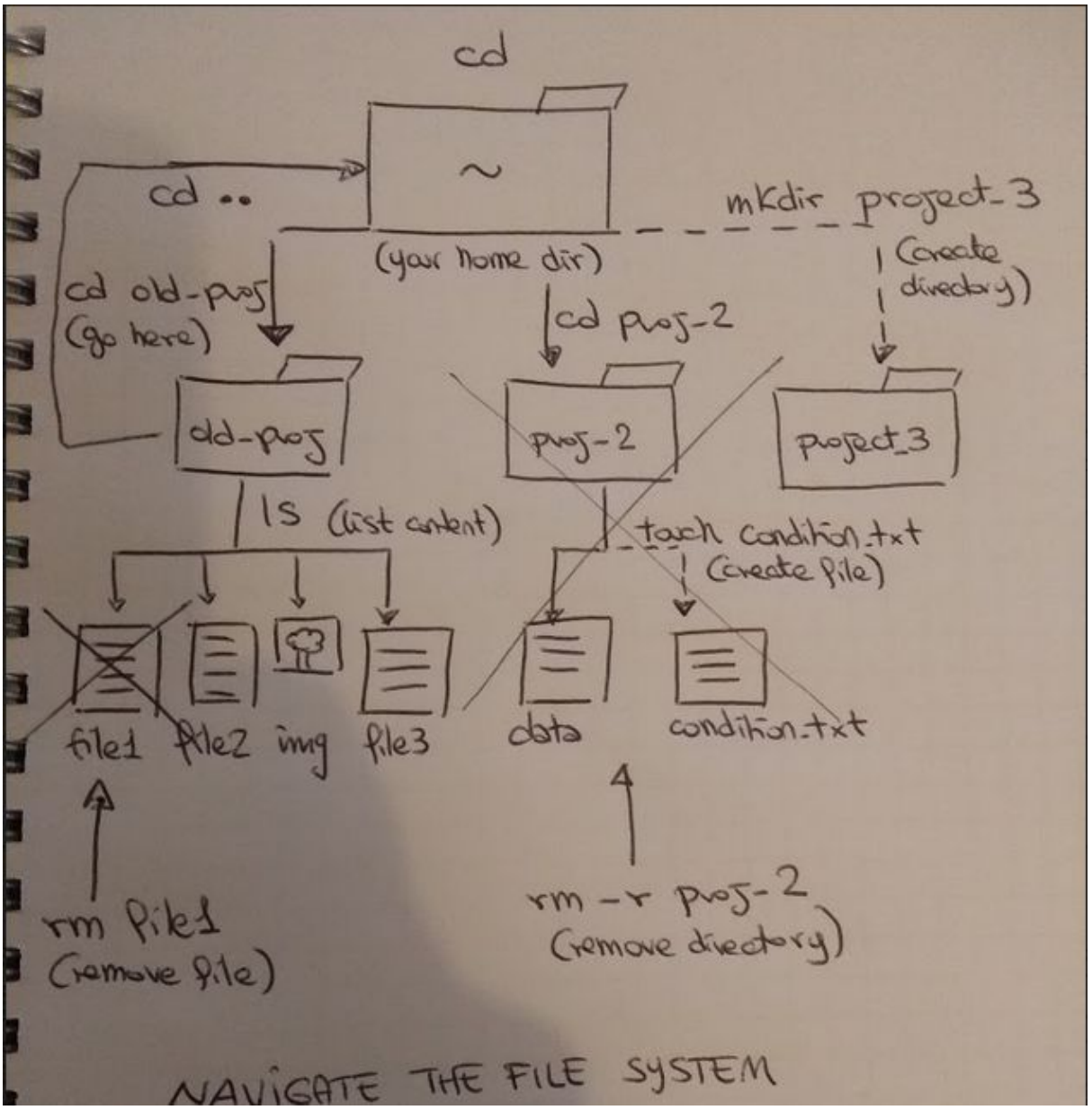
Yo:



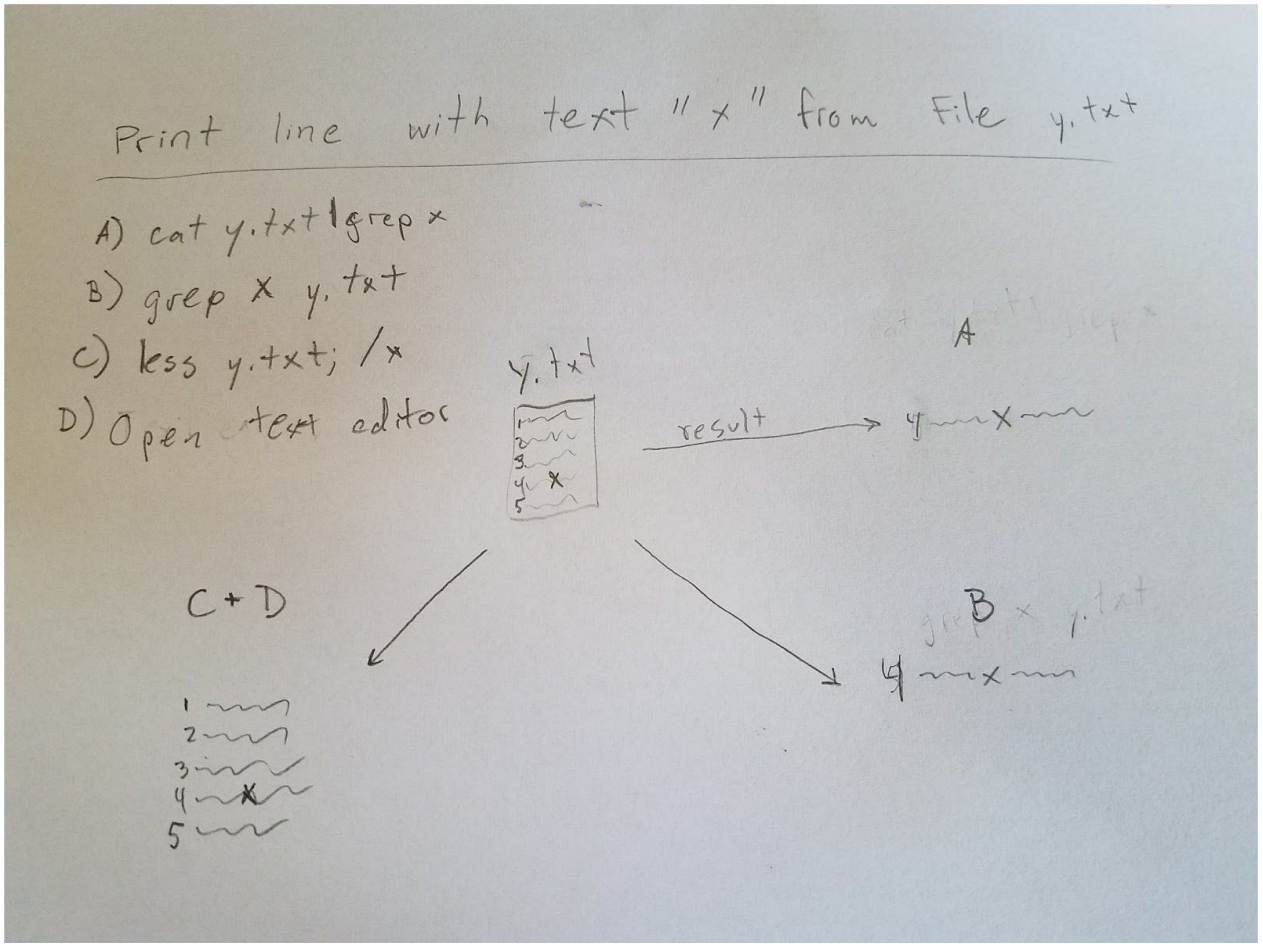
Ahmed:


if - Statement
if
Condition?
true
then
else
false
true action
false action

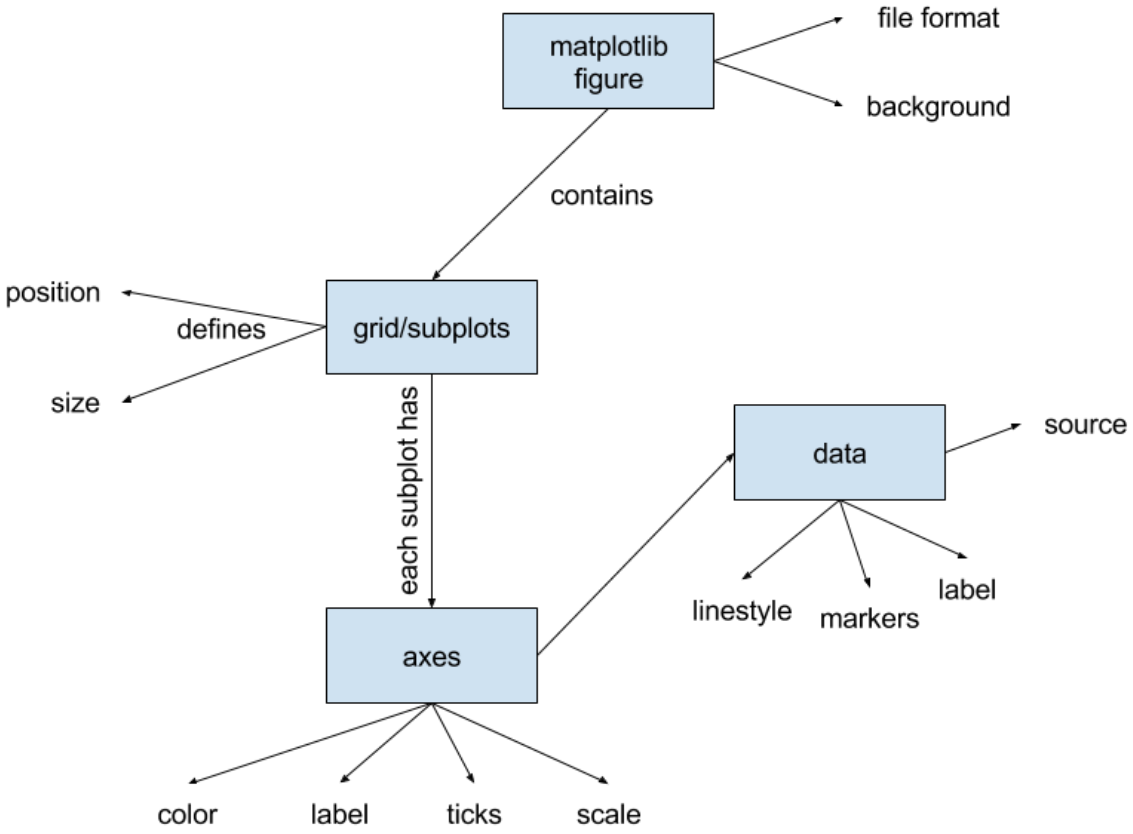
Alice:


cd
~
cd ..
mkdir project_3
(your home dir)
(create directory)
cd old-proj
(go here)
cd proj-2
old-proj
proj-2
project_3
ls (list content)
touch condition.txt
(create file)
file1
file2
img
file3
data
condition.txt
rm file1
(remove file)
rm -r proj-2
(remove directory)
NAVIGATE THE FILE SYSTEM

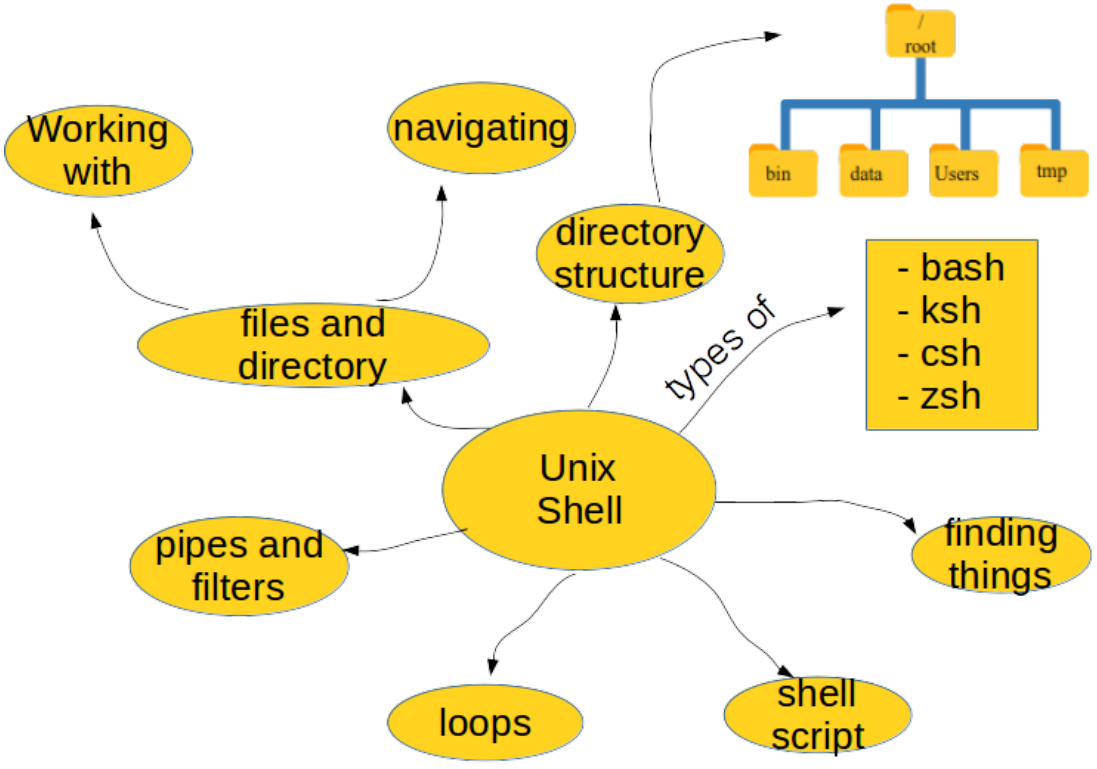
Andrew


Print line with text "x" from File y.txt
A) cat y.txt | grep x
B) grep x y.txt
C) less y.txt; /x
D) Open text editor
y.txt
result
A
C + D
B

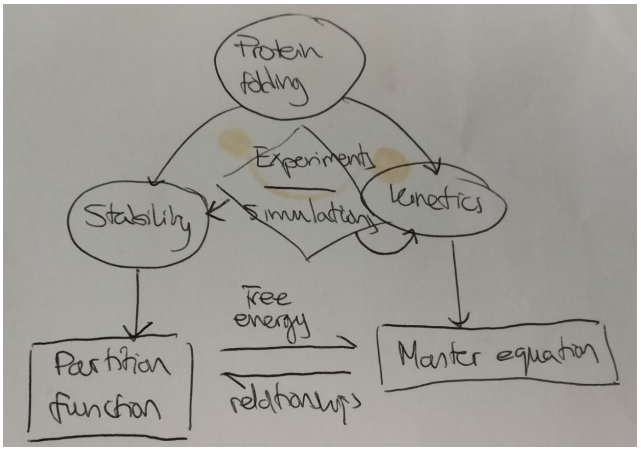
Vasu:


matplotlib figure
file format
background
contains
position
defines
size
grid/subplots
each subplot has
data
source
linestyle
markers
label
axes
color
label
ticks
scale

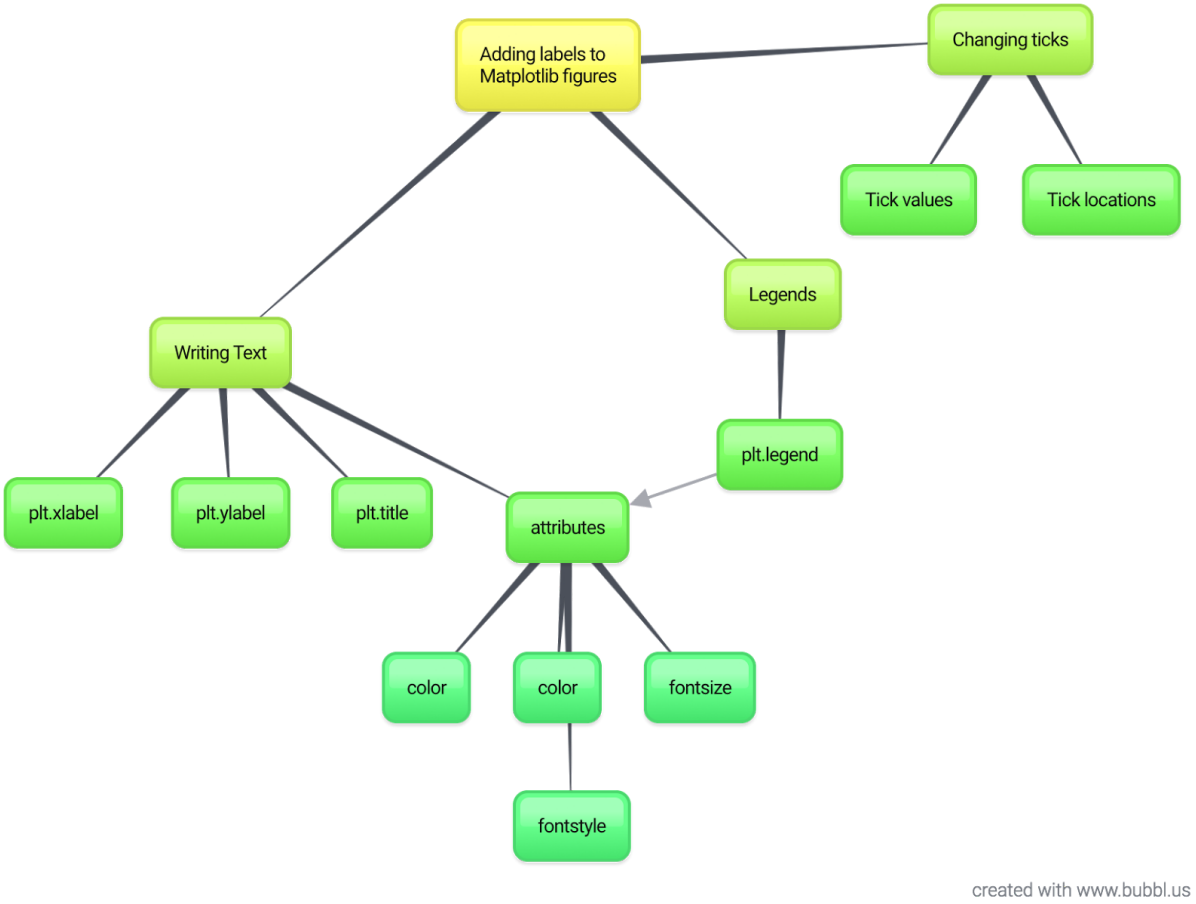
Bagus:


Working with
navigating
/ root
bin
data
Users
tmp
directory structure
files and directory
types of
- bash
- ksh
- csh
- zsh
Unix Shell
pipes and filters
finding things
loops
shell script

David:



Diogo:

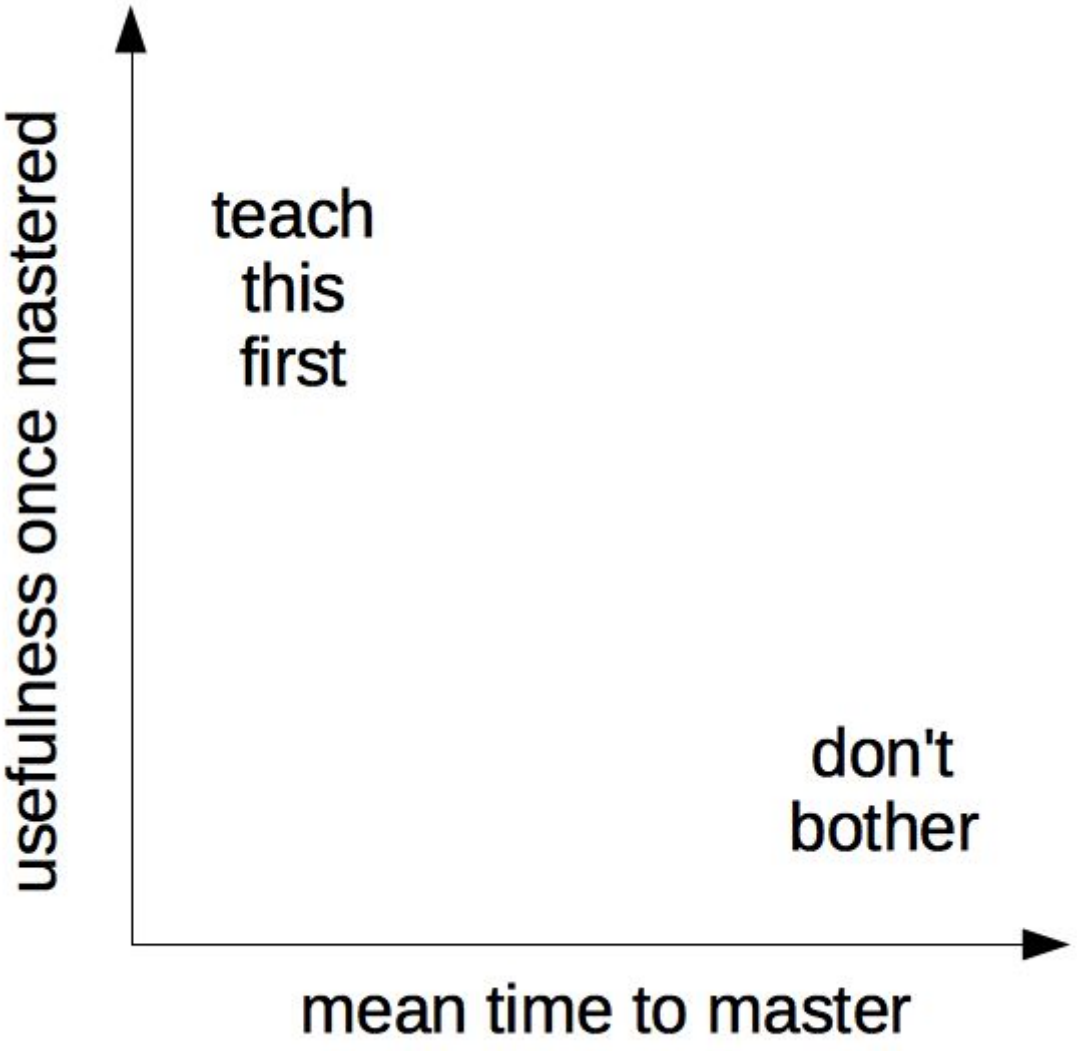


Motivation and demotivation

Enthusiasm

- Presenting the instructor as a learner
- Establishing norms for interactions

- Encourage learners to learn from each other
- Acknowledge when learners are confused



Teach Most Useful First

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, did some stats 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 2x2 grid of "short/long time to master" and "low/high usefulness". In the class Etherpad, **share** the task and where it fits on the 2x2 grid. As a group, we will discuss how these relate back to our "teach most immediately useful first" approach
Back at 14.18

Andrew: Importing and using python modules - what, how, why, etc. It fits in the top left of the grid, I believe.
Yo: Git basics: *incredibly useful* with only the basics - top left. (Vasu shared example where a colleague lost all their data/work, but a bit of it was backed up on git. After that, *everything* went on git)
Diogo: Starting with data: Very useful to use Pandas/Python but can become complicated. Place it to the top but in the middle, covering a wide part of the *time to master* axis. Also worked alot with git to backup my thesis/figure scripts, also top middle.
Alice: write bash test scripts to check functionality of the main script - don't bother as it's a lot of work for a novice to understand it and code it, and they could more easily and quickly check by hand if it works as expected.
Ahmed: Used dplyr in R to summarize and normalize counts in a table. dplyr sounds like a teach-it-first tool/concept because it can provide many useful functionalities once the learner learns the concept and how to use it.
David: Creating a package, very simple technically, although quite involved to explain.
Vasu: Created a function that will generate a skeleton(figure, axes, gridspec, subplots) for a 2D matplotlib figure. *short/moderate time to master *extremely useful *Have used it repeatedly
Bagus: I plan to teach unix shell with the following 2x2 grid → top left: whoami, pwd, ls, cd; top right: cp, mv, rm, mkdir; bottom left: pipes, filter; bottom right: loops; script
Désirée: To speed up my code, I hard-coded a mapping function that only reads bits of a (huge) image cube into memory at a time. Very easy and immediately useful. Writing a generalised way of processing this would be very time-consuming

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 Google Doc 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.

Until 14:28

David: *Value* -> use examples from the student's field of expertise; show snapshots from news articles that show how valued computational data analysis tools are for industry.
Vasu: Challenges and Early success - Pick a problem or task that the students can relate to or regularly do and use that as an example in the workshop
Bagus: Building positive expectation for learner is needed to strengthen motivation by explaining the benefit if they can master the teaching materials.
Désirée: Relate to professional life: an example from my own teaching - students got to choose the project topics themselves and we explained that this could be a typical situation/project they can bump into in their professional life. 100x higher student motivation! - but this is harder to achieve for SWC/DC students, as they often come from different fields
Andrew: Use relevant, real world examples to immediately demonstrate gains in productivity.
Yo: real-world value. Back to Git: remind people that they can have their thesis easily and securely stored elsewhere on other computers :)
Diogo: Using examples that are familiar to the students so this way it is easier for them to understand new techniques on top of what is relevant to them. They also need time to reflect on what they learn instead of going over multiple different examples and then revise them on the class notes.
Alice: value: provide an example of when i needed to loop over a number of data files to analyze and doing that by hand was very tedious and error-prone; positive-expectations: ealry success opportunity maybe asking about something they need to do and they think it can be done with what was explained
Ahmed: [Provide authentic, real-world tasks] For life science (bioinformatics) students, providing an example from actual published dataset (real-world) can be very interesting to the students, where they can relate to and find it useful while learning how to analyze the

dataset. [Provide early success opportunities] probably the learners will be excited to see the outcome of their learning activities in solving real-world problem.

Why Do You Teach? (5 min)
We all have a different motivation for teaching, and that is a really good thing! SWC wants 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.
Until 14:40

Brainstorming Demotivational Experiences (5 min)
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.

Andrew: Being in classrooms where students have a wide range of skill levels, i.e. absolutely no programming experience - computer scientists, can be very demotivating because the teacher will try to meet everyone's needs and it ends up causing the lessons to move forward very slowly.
Yo: My boss once carelessly lost a lot of work I had done :(I would have liked him to pay more attention when deleting things, but I also recognise he is human.
Diogo: To avoid demotivating students: Break the consecutive Instructor-> Learner flow and give students an exercise to do, or ask for student participation, or even longer coffee breaks to refresh the ideas. It's not demotivated in the class, it's getting lost in the continuous speech and then having trouble getting back on track. This happens to me often.
Alice:
Ahmed: Giving the students a very challenging/hard exam or task.
David: Having a suggestion immediately disregarded by advisor. Excessively challenging task that one cannot even approach. Unapproachable advisor after being left alone with a new project.
Vasu: I have not been demotivated as a student. I was demotivated as an instructor for a class when I realized most of the students were not paying attention and their only reason for being there was in hope that I was going to give the answers to the homework problems or hint at what was going to be on an exam.
Bagus: Make effective time table to prevent learner demotivated, for example: limiting teacher explanation maximum 20 minutes, then giving assessment as well as providing some breaks to keep student fresh (e.q breaks every three assessments).
Désirée: SWC installation/debugging breaks that take very long: hard for the students where everything works not to lose focus and start doing something else. Solution: Offer installation help before the course, if large debugging problems come up, rather announce an official break and when the course will resume.
Until 14:57

Psychological Demotivators:

- **Stereotype Threat**
- **Impostor Syndrome**
- **Accessibility Issues**
  https://github.com/UKHomeOffice/posters/tree/master/accessibility/posters_en-UK

What's Your Mindset? (5 min)
You can learn about your own mindset by taking the official mindset self-assessment (https://mindsetonline.com/testyourmindset/step1.php). Keep in mind that your mindset may differ in different areas. For example, you might have a fixed mindset with respect to artistic ability, but a growth mindset with respect to computing skill.
If you're comfortable sharing, put your "score" in the Google Docs. How does it compare with others?
Until 15.20

Yo: You agreed with 0 of Fixed Mindset statements and 8 of Growth Mindset Statements. You have mostly a Growth Mindset.
Diogo: You agreed with 0 of Fixed Mindset statements and 8 of Growth Mindset Statements. You have mostly a Growth Mindset.
Did anyone get confused with the questions? Don't they all seem the same??
Bagus: You agreed with 1 of Fixed Mindset statements and 8 of Growth Mindset Statements.
Désirée: 2 fixed, 8 growth
David:

You agreed with 4 of Fixed Mindset statements and 4 of Growth Mindset Statements. You have a mixture of the Fixed and Growth Mindsets. The book will show you how to avoid the pitfalls of the Fixed Mindset, how to develop more of a Growth Mindset, and how to apply your Growth Mindset most effectively.

Ahmed: 8 of Fixed Mindset statements and 1 of Growth Mindset Statements

Vasu: You agreed with 0 of Fixed Mindset statements and 7 of Growth Mindset Statements.
Alice: 7 fixed, 1 growth

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 **p**erformance-based, **e**ffort-based, or **i**mprovement-based praise?

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

Break until 15:50

If you **have not** participated or seen the Software or Data Carpentry workshop, please watch those two videos before tomorrow.
https://vimeo.com/139316669
https://vimeo.com/139181120

Feedback Rubric

https://www.youtube.com/watch?v=-ApVt04rB4U

	Positive	Negative
Content		
Presentation		

Andrew:
Content negative - refers to concepts without explaining them, i.e. assumes background knowledge
Content positive - using python (stole from David :P)
Presentation Negative - uses words like "really simply," "even…," "of course," "as you'd expect"
Presentation positive - asks for questions at the end

Yo:

	Positive	Negative

Content	Apologised when he made a mistake	"This is really simple stuff. Even excel users could understand this." :(
Presentation	He was friendly and casual which made the course delivery seem comfortable.	I couldn't read the screen :(

Diogo: Content: + solves issues, - too much technical jargon "polymorphic",
Presentation: + live coding, admits errors, - condescending to the learners
Alice: content +:
Content -: i couldn't follow because i can't read and he talks very fats, i just know he's talking about functions
presentation +: he looks very involved in the explanation and point to stuff on the screen
Presentation -: he talks too fast, and the text is way too small, also he's rude to the students!
Ahmed: Distracted instructor or at least not focused/prepared, checking cell phone, throwing jargons, assuming prior knowledge, very small text on the screen, instead of talking to the audience, the instructor is talking to the screen or to the computer, the instructor sometimes sounds like talking to himself.

David:

	Positive	Negative
Content	He's teaching Python	Very difficult to grasp what he's talking about, using complicated concepts ("polymorphic", "object")
Presentation	He recognises that he's making mistakes "I am going to fix this up", asks for questions from audience	He invokes belief from his audience ("trust me"), handwaving uselessly, messy, checks phone, does not look at his audience, swearing

Vasu: Content - Postives-Reminds where they stopped. Negatives - Too much jargon, no concrete examples, berates excel users. Presentation - Postives - Asks for questions, Negatives - Too fast, too much movement, Uses the "J" word and "easy" too much, distracted with his cellphone.
Bagus: content:+ Just works vs - to small font, bad word (he said "trust me", but not in joke manner), presentation:+ He looks expert vs - too fast, interrupting teaching by looking smartphone
Désirée: Good: summary after break, engaged in subject, fixes and admits a mistake. Not so good: seems annoyed (start), easily distracted (phone) and superiour to students (this is

easy), and probably the students couldn't see the font on the screen any better than me. Can't judge the content after watching once only.

16:10

Feedback on Yourself (25 min)

1. Split into groups of three.
1. Individually, spend 5 minutes preparing to teach a 90-second segment of the Carpentry lesson episode you chose before the start of the training course.
1. Get together with your group and have each person teach their segment to the group. Do not use live coding. We'll be practicing with live coding later. Keep a strict time limit of 90 seconds per person.
1. After the first person finishes, rotate roles and then rotate roles again.
1. After everyone in the group of three has finished teaching, watch the videos as a group. Everyone gives feedback on all three performances, i.e., people give feedback on themselves as well as on others.
1. After everyone has given feedback on all of the performances, return to the main group and put everyone's feedback about you into the Etherpad.

Alice:
Ahmed: +ve: introduced/defined the topic and its importance/relevance. -ve: used jargons, not necessarily already defined.
David:
Vasu:

	Positive	Negative
Content	Clear presentation	Did not introduced jargon "index"
Presentation	Speaking confidently	Too monotonic

Andrew: speak more clearly. Be more articulate and precise.
Yo: Tried to explain jargonistic words, provided real-world examples beyond coding of why git might be useful (e.g. backing up a thesis and versioning it). Possibly should have provided more code examples!
Diogo: Tried to explain for loops in python by explaining they are used to make the computer repeat things for us. I did a bit of coding, mistakenly, and was suggested to make an introduction to range before using it. Also try to make a better analogy to explain loops
Bagus: show the how-to not the why only.

Désirée: Basic SQL queries (data carpentry): pos: good introduction that required no prior knowledge. Explained query structure: distinction of capitals/small letters and semicolon. Neg: used some jargon terms (syntax)

Back at 16:40

Exercise: Using Feedback

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 Google Doc.
This exercise should take about 5 minutes.

Andrew: The feedback was fair and on point. Sometimes I stumble over my words and I know this about myself so it's definitely something I need to be aware of when teaching. I think I can improve by making sure I am prepared before teaching with materials to help guide me while teaching, i.e. slides, outlines, etc.
Yo: Feedback was fair and nice! I wasn't sure if I defined all of the jargon words before using them, so I'll try to make sure that I am careful about not using undefined words.
Diogo: Feedback was fair and I agree with it. I'll make a better analogy to the methods to make them more approachable to the learners.
Alice: fair feedback, i noticed myself I sounded confusing in a part of the explanation, so i think telling the lecture by myself a couple of time beforehand would make me see which parts don't sound as linear as possible, and maybe remove some details to improve clarity.
Ahmed: Fair and constructive feedback. Unless already defined or explained, I will avoid using jargons without clarification.
David: Perfectly fair feedback, constructive although difficult to deal with. I don't know how to stop mumbling and using words like "like" all the time!!!
Vasu: Fair and useful feedback. I will modulate my voice and emphasize or highlight important concepts in the future.
Bagus: I am happy to receive this feedback. The feedback was fair and can be used to improve my teaching. I can add what is missing in my teaching and remove what is not necessary. I am agree with this.
Désirée: fair and reasonable feedback. Matches my feeling of the teaching exercise. Improvement: make a list of jargon words for myself and be sure to introduce them properly/avoid them if they are not necessary for lays.

IX. Wrap-Up and Homework for Tomorrow

To prepare for tomorrow, please:

1. Read about the two types of Carpentry workshops: self-organized (http://www.datacarpentry.org/self-organized-workshops/) and centrally-organized (http://www.datacarpentry.org/workshops-host/) and the checklists these pages link to.
1. Prepare for the live coding exercises. If you haven't already, pick an episode from an existing Software or Data Carpentry lesson and read through it carefully. Tomorrow, you will use this to practice live coding for 5 minutes in groups of three. Your group members will comment on the delivery and content. Recommend episodes are listed here:

- Data Carpentry
    - Faceting and Clustering in OpenRefine
    - Basic Queries in SQL
    - Starting with Data in R
    - Starting with Data in Python
- Software Carpentry

- Working with Files and Directories in the Unix Shell
- Tracking Changes in Git
- Selecting Data in SQL
- Repeating Actions with Loops in Python
- Exploring Data Frames in R

Post -it feedback:
https://goo.gl/forms/RFPxeJeZPh9pJ6tG2

https://drawings.jvns.ca/drawings/surprise.png

high impact papers about mindset research:
http://www.pnas.org/content/113/31/8664.full
http://psycnet.apa.org/record/2012-20864-001

##

DAY 2

##

Key Points Day 1

- 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.

Questions (5-10 min) until 09:17

Yesterday we asked you to read some resources about the logistics of teaching and running Carpentry workshops. Please add your questions about logistics and preparation to the Google Doc. We will answer some of these questions now and make sure to answer the rest by the end of the day.

Yo: courses seem quite expensive! Do we get lots of takers?. Also, comment rather than question: All the templates and checklists look great - it looks really easy to set up a course with all that guidance.

Diogo: The fees for the courses are just for certification? Are we assigned to all courses or do we specify which ones we want to give? Are there specific instructions on how to give a specific course?

Andrew: I would like to know more about the fees. Where does the money go? Particularly interested in where the money for self-organized workshops goes.
David: For co-instructors, what is the typical way of recruiting them? Do you seriously need to check computing skills by looking at CVs? With respect to lessons (maybe off-topic), are new data carpentries (one on physical sciences, for example) out of the question?
Vasu:What is the typical number of courses you teach in a given area in a year?
Bagus: How the carpentry make money beside the fee from participant? There are a lot of activities in carpentry and I think funding from workshop fee is not enough.
Alice: does everything has to be kind of the same at each carpentry? (e.g. let's say some instructors have a different method than post it for feedback, could they use it?)
Ahmed: 1) Rather than totally waiving the cost, is there sort of a reduced cost for some countries (like Egypt)? 2) What is the likelihood of a new lesson to be proposed and approved to be offered? 3) Does a SWC workshop always require more than one instructor? 4) If there is a need to set up the physical environment (e.g., cloud VMs), who would take care of something like that (in terms of fees)?
Désirée: I wasn't even aware of the course fees. I think this can be hampering the establishing of small software carpentry branches in non-university environments (I know of several governmental departments in rich Norway that would really, urgently NEED to introduce some programming in their completely outdated way of working, and the old guys sitting at the top are the last ones speaking any money for such unnecessary new ideas).

A learner profile for Software Carpentry might be:

background:

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.

the challenge:

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.

how the workshop will help them:

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.

Learner Profiles (10 min) until 9:40

Read Software Carpentry's learner profiles (https://software-carpentry.org/audience/ and then write one that describes a fictional colleague of your own. Who are they, what problems do they face, and how will this training help them? Be as specific as possible.

Enter your learner profile into the Google Doc.

Andrew:

background:

Nell is an environmental biologist specializing in field work in the Southwestern US. She works with a crew who collects and analyzes data relating to bird and insect populations. Technology has a very limited role in both Nells personal and professional life. She spends long tedious hours behind her computer, using Microsoft Excel to process and analyze her data.

the challenge:

Nell and her crew spend long tedious hours behind her computer, using Microsoft Excel to process and analyze their data. Because of their lack of computing skills, they dedicate much of their off season to this work - tasks which they accomplish manually, but which can be done much more quickly with basic programming skills

how the workshop will help them:

Software Carpentry will show Nell how to automate the process of wrangling her data into a structured format that will allow for efficient, accurate, and reproducible analyses. Using python will free Nell from reliance on inefficient and unreliable proprietary software.

Yo:

Background:

Sonya is doing her PhD in Biology and spends most of her time in the wet lab. She has never taken a programming course. When she's out of the office she likes to go running with her dog.

Challenge:

Most of the results Sonya has are stored in excel spreadsheets, where she can sort data, clean it up, and create graphs. She's beginning to get to a stage where she has a file so large it makes her computer choke up whenever she opens it, and trying to generate graphs takes ages. She has heard that coding might help her out but she's not sure where to start.

How the workshop will help them:

Software carpentry will show Sonya how to run scripts using R or Python, and how to use basic visualisation packages to produce her graphs. She's not aware yet, but tools like git will allow her to back up her work and version it as well.

Bagus:

Background:

Karina is graduate student in physics. He is strong on physics experiment but average programming skill. His physics experiment collect physical variables every day from collision of laser on diamond. He has five variables that changes over time and he would like to see the trend of variables by adjusting some parameters on his experiment.

Challenge:

All of Karina's experiment result can be shown in real time (logged), but currently he has to analyze manually for single experiment for single day. He want to make visualization automatically from the log of the result. For this reason, he attend the carpentry workshop to gain real-time automatic data processing.

How the workshop will help them;
SW carpentry will teach him about how to acquire data line by line, import the certain value on certain line and then plot it real time on some time interval. This skill will enable Karina to make automatic real-time visualization so she can determine what factor influence his experiment.

Désirée:
Background: Merin Medicus is a PhD student in medical science. She works with large patient data sets provided by different hospitals and health institutions and tries to find how certain diseases or body defects in ageing patients are related to their habits, such as smoking, sports, or medicament use. She has no programming background, but uses statistics software quite comfortably. Her data comes in large Excel files.
Challenge: Handling and integrating the large excel files from different sources is cumbersome, slow and tedious. She loses a lot of time in managing her data and sending it forth and back between statistics software and excel/text files. She would like to have a more consistent environment to work in, to have a better overview.
How the workshop will help: Merin will learn how to create a simple database and writing SQL queries, so that she can keep her large data from different sources in a consistent and transparent way. Learning R could help her to integrate her data directly with a statistical tool.
At least one of the instructors will be female, not to scare off the predominantly female potential participants in this field. (;

Ahmed:
Background: Jasmine is a MS student Biotechnology. She holds BS in Pharmacy. Her background is primarily in molecular biology and biochemistry. She has never programmed before.
The Challenge: Jasmine's MS work includes generation of gut microbiome data from healthy individuals and IBD patients. She would like to be able to process and analyze her data to see if there is a difference between the microbiome composition between IBD and healthy individuals.
How the workshop will help: The "Programming with R" workshop will introduce Jasmine to the basics of the data handling and processing using R. She will learn about the different data types and how to manipulate these data types in R. She will also be introduced to the basic of flow control of her analysis.

David:
Background
Santiago is a tenure track fundamental physicist that travels to a synchrotron several times a year. After returning he has tons of data to analyse, consisting of equivalent datasets that he looks at one-by-one using Origin. He has occasionally received scripts in the form of IPython notebooks that he runs blindly. But then he goes back to the good, old plotting and analysis with endless mouse clicks of his favourite graphical scientific plotting tool.
The challenge

Santiago has to do some small modifications on the scripts to receive, and also feels a need of speeding up his analysis instead of doing his data munging one by one.

How the workshop will help

With Data Carpentry Santiago will see the light of how Python can make him fly. He will learn how to modify and expand the data analysis scripts he's received from his distant colleague, producing unprecedented insight. Additionally, he will save great amounts of time building automated workflows. Finally, after acquiring all-new skills in plotting with Matplotlib he will stop having to use proprietary software for plotting. This will result in some money-saving for his limited budget. This will allow for him to travel once more every year for doing synchrotron measurements, providing opportunity for more experiments, more data and more knowledge.

Vasu:

Background

Minnie Mouse is a materials scientist working on understanding stress response of polymeric materials. Her programming experience is a first year programming course in college using C++ and a limited exposure to MATLAB in other engineering classes. She has not done programming since her time in college.

the challenge:

Her work consists of evaluating physical properties of polymeric materials, especially response of the material to different stresses. Minnie is extremely good with building things and she has built a custom device that measures these properties using off-the-shelf electronic components like the arduino and Phidgets. Once data has been collected she has to manually process it in using tools like JMP or Excel and send feedback to the device on which test to run next. Minnie wants to automate this using Python and shell scripting

how the workshop will help them:

Software Carpentry will show Minnie how to read/write files, do simple mathematical operations on arrays, and generate basic plots in python. It will also show her how to write shell scripts to automate the running of the test and analysis. In will show her how to use git to maintain a versioned list of scripts that implement certain standardized test protocols

Diogo: **Background**: Maria is a Physics researcher working with spectroscopy. She has experience in Fortran and IDL but now mostly commercial products to analyze the spectroscopy data.

The **challenge**: When providing figures for publication she has to export from the software the figure, edit it in another program such as paint or, and save it again. If she wants to repeat the plot for a new figure, she has to manually do everything again. In addition her institute is trying to reduce software licensing costs and using free open source alternatives could be a solution.

How the workshop may help: SWC will help Maria to write Python programs to analyse spectroscopy data. By using Python Maria will also be able to create and export plots, making her Figures reproducible. Since Python is free, there are no up front costs for the institute.

Alice: Sheena is an experienced postdoc researcher in a group that research wheat variants. Shes has always worked in wet lab or in a field before, but for this position she needs to do

some bioinformatics analyses. She feels a bit discouraged to start as “she’s not that young anymore”.
Sheena needs to organize her file system creating an organized set of directories and subdirectories for each condition in study (hundreds of directories!), where to store images and other data. She started doing so manually, but it’s taking her ages. She will also need to run the very same analyses on specific subset of data. Also, she actually prefer to work in the field and would like to spend in the office at little time as possible.
SWC will help Sheena showing her how she can automate these kind of tasks, running just a couple of commands, or saving them in shell script she can reuse in the future.

Working With Learning Objectives

Evaluate SWC and DC Learning Objectives (10 min) until 10:02
Take a minute to select one learning objective from your chosen Carpentry lessons, then complete the following steps to evaluate it.

- Identify the learning objective verb. How specifically does this verb describe the desired learner outcome?
- In your opinion, does the lesson do an effective job of meeting the stated objective?
- Does the lesson meet any objectives that are not stated in the objectives section?

Andrew: http://www.datacarpentry.org/python-ecology-lesson/01-starting-with-data/
Object: Starting with data in python. This describes the learning outcome by making it clear that the objective is simply to get started, rather than diving deep into data analysis. I think the lesson is mostly effective, but seems to skip over some important concepts, e.g. you have to first install Pandas because it’s not part of the standard library. I did not notice any objectives that were met yet not stated in the objectives section.
Yo:
Git novice: http://swcarpentry.github.io/git-novice/04-changes/

- Go through the modify-add-commit cycle for one or more files.
    - Lesson achieves this
- Explain where information is stored at each stage of Git commit workflow.
    - Lesson shows clearly the differences between unstaged, staged, and pushed to a repo
- Distinguish between descriptive and non-descriptive commit messages.
    - Lesson clearly explains the difference, also emphasising not to be *too* descriptive / overwordy
- One additional objective not listed in the lesson: Explain the use and meaning of basic git terminology, including: commit, diff, status, add, and log.

Bagus: Unix shell: working with file and directories. a). LO: 1. Create dir in given hierarchy → basic task to make dir, 2. Create files → basic task in using computer, 3. Display list file → basic computing task, very important. 4. Delete file/dir → basic computing skill; b) Very effective lesson: dense, important and useful. The lesson shows what command to achieve

the learning objective (LO); c) Yes. it shows different command that giving the same result, the shortcut and the editor.

Désirée: SQL basic queries:
http://www.datacarpentry.org/sql-ecology-lesson/01-sql-basic-queries/
Objectives: Write and *build* queries. /***Filter* data given various criteria.** / *Sort* the results of a query.
The objectives are short, crisp and precise.
Filter: The lesson teaches filtering well - variable selection, limiting to certain values, and sorting/limiting the listed results are explained.
Not mentioned learning objectives: **Understand query structure and good practices**. The lesson explains how queries are built up, and why it is important to stick to practices like capitalisation and commenting.

Ahmed:
Lesson: Exploring Data Frames
http://swcarpentry.github.io/r-novice-gapminder/05-data-structures-part2/

Objectives:
Add and remove rows and columns.
Remove rows with NA values.
Append two data frames
Articulate what a factor is and how to convert between factor and character.
Find basic properties of a data frames including size, class or type of the columns, names, and first few rows.

Yes, the objectives will show the learner how to manipulate a data frame in R.
Yes, the objectives are very specific.
Yes, the lesson shows the necessary functions to handle a data frame. The lesson is very focused and write to the point with clear descriptive comments.
Yes, the lesson also covers read and loading data files, which can be considered as an objective but not stated among the objectives of the lesson.

David: Data Carpentry - Starting with data

- *Navigate* the workshop directory and *download* a dataset.
- *Explain* what a library is and what libraries are used for.
- *Describe* what the Python Data Analysis Library (Pandas) is.
- *Load* the Python Data Analysis Library (Pandas).
- *Use* `read_csv` to *read* tabular data into Python.
- *Describe* what a DataFrame is in Python.
- *Access* and *summarize* data stored in a DataFrame.
- *Define* indexing as it relates to data structures.
- *Perform* basic mathematical operations and summary statistics on data in a Pandas DataFrame.
- *Create* simple plots.

Objectives are clear and descriptive.
Bonus objective: get the learner used to the Jupyter notebook environment.

Vasu:
Write (correctly) a for loop. Yes, the lesson meets the objective of demonstrating the correct syntax for writing for loops in Python. Yes, it also demonstrates the use of some built-in functions in python.
Diogo: https://swcarpentry.github.io/python-novice-inflammation/02-loop/

Objective: Learn how to use for loops in python to repeat actions.

The lesson does a good job in teaching for loop basics in Python. It does not assume any prior knowledge of loops using other languages, which is a good thing since they are a bit different in Python.

Also, the lesson gives some insights on how data, such as strings, is structured in memory and provides an interesting more complicated application using python, such as calculating polynomials.

Alice: Objective="*create* files in that hierarchy using an editor or by copying and renaming existing files". The lesson achieves this, the main verb is probably a bit general. There are some very useful explanation in the lecture that may be put as an objective "*understand* how to name files and directories" (not an action, but useful in the list if the learner goes through it and tick the points)

http://jlifeoc.com/wp-content/uploads/2016/02/0316bloomstaxonomykid.jpg

Why Live Coding?

Up and Down (5 min)
List some advantages and challenges of live coding from both a learner's and an instructor's point of view in the Google Doc.

Go to **www.menti.com** and use the code **39 46 95**

Sources for online polling:
Mentimeter.com
Socrative.com
Google Forms

The Bad and the Good (15 min) until 10:35
Watch this video of live coding done poorly
https://www.youtube.com/watch?v=bXxBeNkKmJE&feature=youtu.be
and this video of live coding done right
https://www.youtube.com/watch?v=SkPmwe_WjeY&feature=youtu.be
and then summarize your feedback on both in the Google Doc. 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.
Turning on closed captioning by pressing the cc button will improve the accessibility of these videos.

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

	Positive	Negative
Content		
Presentation		

David:

Done right

	Positive	Negative
Content	Explains what is going on in screen, makes mistakes	Still no full screen, does not explain wildcards, jargon
Presentation	Makes direct contact, constant interaction with what is going on in the screen	Goes up and down, does not tell whether he'll be doing it or whether also the students need to code

Done wrong

	Positive	Negative
Content	Colour code helps	Pop-ups in screen, not full-screen
Presentation	Voice is very clear	Never looks up at people, sitting

Vasu:

Video 1

	Positive	Negative
Content	Specific content. Example meets learning objective	Repetition not effective
Presentation	Speaks clearly	Does not explain errors

Video 2

	Positive	Negative
Content	Specific content. Example meets learning objective	Without terminal colors its easy to miss something
Presentation	Speaks as he is coding and explains errors	Does not explain the wildcard syntax fully

Diogo:

Poor video	Positive	Negative
Content	specific content	too long commands

Presentation	explains what hes doing, corrects	Types too fast, too fast,

Good video	Positive	Negative
Content	Good examples	“You’ll do this wrong, trust me”
Presentation	Standing, Stops to explain	

Alice:

Video 1

	Positive	Negative
Content	Good explanation of the prompt change and variable name changes	Red post it has been ignored?
Presentation	Good writing size	Very plain voice, the browser window behind the terminal is distracting

Video 2

	Positive	Negative
Content	As above, but better timing of the explanation	Pretty fast explanation of why the error is shown twice
Presentation	Explanation given before demonstration, instructor points at things on the screen, the size of the window make so that people in the back will see it	He doesn’t ask people if they are following or have questions

Andrew:

Bad	Positive	Negative

Content	Topic was useful and interesting	Lack of verbal guidance in real time
Presentation	He seemed friendly	Distracting noises, desktop background, uses a customized shell

Good

	Positive	Negative
Content	Uses a plain console with not fancy customization	Maybe it would be better to use real world example (sorry, unicorns).
Presentation	Stops to explain things in real time	Uses hands to point out things on screen

Yo:

Poor:

	Positive	Negative
Content	repeated the same task more than once to show variable naming.	typing lots of lines fast without pausing to explain
Presentation	colourful screen	hard to see what is being typed.

Good:

	Positive	Negative
Content	Explains what he's typing out loud line by line.	Made a mistake. Whoopsie!
Presentation	Stops to explain in realtime, step by step.	"Trust me"

Bagus:

	Positive	**Negative**
Content	Show the different variable name that give the same	The dark background sometimes hard to see

	result	
Presentation	Ask the participant if there any problem	Cascade table can make some participant deoriented/unfocus..??

Désirée:

	Positive	Negative
Content	● Difference between fixed code words and variable names ● Using mistake to teach/repeat extra material (arrow up, one-line code)	● Unsatisfactory explanation of structure and chosen variable names ● Mistake corrected miraculously
Presentation	● Saying the typed words aloud while coding ● Good structure that tries to avoid split attention effect (explanation before/after)	● Dark screen, small font, confusing colours ● Not noticing a red sticky note ● Sitting and no interaction with screen seems much less engaged/interactive

Ahmed:
Bad:

	Positive	**Negative**
Content	● Teaching looping in bash	
Presentation	●	● Sitting instructor ● Not interacting with the audience ● Very script-like performance ● Too small text on the screen with probably uncomfortable or non-standard colors. ● Distracting (probably personal)

	background. ● Seems like the instructor checked the phone (not sure) ● Fast instructor

Good:

	Positive	**Negative**
Content	● Teaching looping in bash ● Used the error to further explain bash tricks.	
Presentation	● Standing instructor ● Facing the students ● Clear text ● Professional desktop ● Explaining the command before typing and the outcome after executing ● Professional background	

Brief List of Tips for Live Coding in a Workshop

1. Stand up and move around the room if possible.
2. Go slowly while you say commands as you type them and either point to or highlight (with your mouse) the outputs on the screen.
3. Mirror your learners' environment by turning off fancy environments and by not using keyboard shortcuts.
4. Use a big font and maximize the window with good contrast so that all learners can see.
5. Use illustrations to help learners understand and organize the material.
6. Turn off notifications on your laptop and phone.
7. Teach from the community-developed lesson material.
8. Leave no learner behind by using sticky notes and helpers.
9. Embrace mistakes and use these opportunities to demonstrate the art of troubleshooting.
10. Have fun, relax, teach, learn, and share!

Practice Teaching (20 min) until 11:05
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. 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. Give each other feedback using the 2x2 rubric we discussed previously and enter the feedback you received in the Google Doc. To make this exercise as similar to the workshop experience as possible, ask your fellow trainees to code along with you, as if they were learners at your workshop.

Alice: +: personality, easy to listen to, make learners comfortable
David:
Vasu: Positive - Clear and slow enough presentation to follow along. Negative - Did not discuss python syntax idiosyncrasies
Yo:
Bagus:

	Positive	**Negative**
Content	Comparison of gui and cli	Make mistake while coding
Presentation	Show some different alternatives to reach the result	Clear the terminal frequently so the participant can show the previous command

Andrew:
Désirée: basic SQL queries

+mentioning capitalisation, +explain select, from, +clear and easy to follow
-emptying query and results window before starting, -changes on screen were not visible in the start

David:

	Positive	Negative
Content		
Presentation	showed a notebook with gaps which may be helpful, clear intro to import statement	Saying "like" all the time, showed a notebook with gaps which may be confusing

Ahmed:

	Positive	Negative
Content	● Explained what to expect from the lesson. ● Sitting up and introducing the lesson	● Should have explained the “practical” need for the objective.
Presentation	● Sounded confident with the material ●	● Should have explained how to execute a cell in Jupyter when I actually did so.

Diogo:

	Positive	Negative
Content	Good examples	Pseudo code before code, not keeping variables in cells, Should explain what is the variable and the values
Presentation	Using colored syntax	Should go slower to give type for learners to follow

We will have a break until 11:25

Key Points live coding

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

Round Two (35 min) until 12:05

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. Give feedback to each other and add it to the Google Doc using our 2x2 rubric. Also add some feedback to the Google Doc on the following questions: 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?

Andrew:
Changed - They told me it was not very different from the first time...
Better/worse - I did not explain some technical terms. Using advanced features in my environment might be confusing.
Next time - Don't forget to explain use of technical terms such as DataFrame. Don't use a special set up, just use what the students will be using (Jupyter notebook in this case.)
Yo:

	Positive	Negative
Content	Explained basic commands clearly, such as ls and cd	
Presentation	Speed was better than last time	Too fast, messy terminal window (`ls` in a home directory that's messy!), don't use a terminal you were using before for a different task!

Changed: started from a 'clean' terminal directory, slowed down and explained things a bit more. Double checked people were ok with the speed partway through. Went much better

Bagus:

	Positive	Negative
Content	Big and clear font	Clearing screen
Presentation	Show the last command before continue the lesson	Could be improved by explaining while typing

Désirée: SQL basic queries

- *Changed:* emptied the results and query windows. Nice side effect: the dummy query appearing at startup was very useful to explain the query structure. Also put even more focus on elements that I was told were helpful (structure, code words, capitalisation) to make this more clear.
- *Worked better:* better flow - mostly because the practice helped!
- *Next time:* prepare a better print-out of the outline/keywords of the content I want to teach. Dry-run.

	Positive	Negative
Content	• Mention semicolon • Dummy query in the beginning	•

Presentation		

Ahmed: +ve explained the objectives and why we would need to do what were we going to learn. Did not assume prior knowledge about a Jupyter Notebook and touched based on how to use it to run code. Made an error and asked the audience why that error might have been caused. I gave further explanation without assuming prior knowledge. Yes, it worked better with the change. Next time, probably I should add an introduction/title to my Notebook.

David: Changed: I went more slowly but thinking twice what I said. Outcome: With less hesitation and less "likes" I even covered more material than before. I still don't exactly know how literally we have to follow the lecture. Next time: Practice practice practice.

Vasu: Clarified idiosyncratic syntax in python. Better with change. I think I will do better each iteration. Need to be cognizant of some terminology differences between different countries.

Diogo:

	Positive	Negative
Content	Clearer with pseudo code on top in markdown	
Presentation	Example code on top as reference Separate typing and talking, its ok to be quiet. Repeat what you wrote after typing the code and before executing	Should still go slower.

Improved: yes
Changed: Added an introduction to the notebook, moved pseudo code there. Spoke more slowly and with a better organized structure.
To do: separate typing and speaking and repeat the explanation after typing, so learners have the chance to keep up.

Alice:

	Positive	Negative
Content	Pointing out rm is permanent, difference between nano and word	Explain better ^ symbol in nano
Presentation	Explanation while typing and velocity	Need to change the terminal contrast and increase font size a bit

Changed: tried to use less words
To do: in a real SWC prepare a clean environment/terminal

Post-it feedback https://goo.gl/forms/xK1sLV53ZIlmgxbC3

Back after lunch at 13:00

Introductions:

Components of the Introduction
- Orchestrate Positive First Impressions
- Introduce Yourself Effectively
    - *What characteristics do you want to convey about yourself?*
    - *What will you need to say to convey those characteristics?*
    - *Why are you teaching a Software or Data Carpentry workshop?*

What should you be careful not to say?

Andrew:

What characteristics do you want to convey about yourself?
I develop software for genomics research at the Pathogen Microbiome Institute. I am a self taught, student intern with no formal training in computer science. The tools I am most familiar with are python and bash.

What will you need to say to convey those characteristics?
I can tell the story about how I became interested in programming and how/why I learned. I started out as a research assistant at the Center for Ecosystem Science and was assigned to an image processing job.

Why are you teaching a Software or Data Carpentry workshop?
To contribute to the advancement of science and to become part of a global community with values that I share.

Advice for success:
Please don't hesitate to ask questions. No question is too simple. The person who asks the simplest question wins a prize. Take advantage of the assistant instructors as well as the post it notes.

Bagus:
- Hi everyone. Welcome to the workshop that will teach you to master what the most advanced skill in 21st century. Make yourself relax, enjoy learning and obtain as much as you can. I would like to short explain about the workshop structure you have description about what we're doing right now. First, please be on time because we wont make messy schedule. I have my schedule and you also. So, make our attendance here to give us most benefits. Today we will learn Unix shell that is divided into 8 parts: intro, file and directories navigation, working with file and directories, pipes and filter, loops, finding and shell script. On each part, make sure you reach the learning objective.

    - To convey characteristics as learner, lifelong learner, I teach here also to learn and share what I've learned
    - I share my experience on first workshop, how to gain knowledge and skill till to be instructor
    - To learn and share what I have learned. To level up.
- Not to say: the difficult thing, bad experience while learning, but give the solution to prevent the bad things and how to handle it

Désirée:
I want to convey that we are in the same boat: I am not primarily a programmer, but an "applier" with programming skills. Also, that I enjoy teaching and sharing: I have been in a similar situation as the learners many times - and I would like them to get started as easily as possible by sharing my knowledge.
I don't want to convey that I am an arrogant, perfect expert that knows everything already and better than them.
Communicate the workshop structure: SQL basics
This 1-day workshop is split into two sessions, before and after lunch, with a coffee break around the middle of each session. We will have a one-hour lunch break around 12 and wrap-up at 4pm - please let us know if you have obligations or need to leave earlier, and we will try to adjust the program accordingly!

The workshop is meant to be very practical and hands-on: throughout the workshop, I will be coding live, and on top of that you will have a lot of time to do exercises to really train and apply what you have learnt. In the morning, we'll start with getting familiar with the data and creating an actual database. We'll also start writing our first simple queries - before the coffee break, we'll mainly learn the general structure of queries and the most important commands to select and filter data from a table. After the coffee break, I will introduce more complex queries… etc.

Ahmed:
- The characteristics are experience in the domain of the lesson and experience in teaching the lesson.
- I will convey those characteristics by providing an overview about my background (academic and/or professional) and highlighting the relevance to the lesson. I will indicate how many times I have taught that lesson before and to what groups of audience. I teach SWC workshop because SWC provides an excellent for teaching and introducing data science to a broad spectrum of audience.
- I will avoid mentioning any irrelevant, especially personal, characteristics about myself.
- **Share some advice for success** – I would suggest that you attempt all the examples that will go through by yourself and if you have any questions or errors please do not hesitate to let us know and we will work with you.

David:

- Orchestrate Positive First Impressions: Welcome everybody to this SWC workshop. I see that there is a very interesting combination of people with very different backgrounds among you. We are going to cover a very broad range of skills that you may have different degrees of knowledge with. But I am confident will have a very positive impact in your daily work doing data analysis. And also in case you are thinking of heading off from Academic research these skills are in very high demand outside Academia. So what you are going to learn in the next couple of days can really be transformative.
- Introduce Yourself Effectively
    - *What characteristics do you want to convey about yourself?* Reasonable level of expertise, but not off-putting. Approachable.
    - *What will you need to say to convey those characteristics?* I am a biophysicist but in fact I spend the day in front of the computer. But I am not a Computer Scientist so started in the same exact place as you did. Please ask at any point.
    - *Why are you teaching a Software or Data Carpentry workshop?* The reason I have engached into SCW is because I think we really need to share some basic skills for less-computationally friendly scientist to adopt techniques that are going to be really helpful. But mostly I find it extremely fun to do.
    - *What should you be careful not to say?* Anything depressing. Like, if I don't start using these skills the week after it finishes everything will be forgotten and your time here will be badly spent.
    - *Share some advice for success*: In order to make the most of these couple of days, my advice would be to try to engage in the class, use your yellow and red postits, request help from the helpers early rather than late and ask any questions that may come up. So that when you go back to the lab on Monday, instead of using your favourite tool and make as if these two days had never happened, open a terminal in your machine and type "jupyter-notebook".

Vasu:
Introduce yourself with your name, where you work, what you do and maybe a fun fact. You need to establish that you are a qualified instructor, so you need to mention your programming background and how you currently use programming in your daily life through an example. Convey your motivations for teaching and how this is also a learning opportunity for you.

Diogo:

- I want to convey that I learned Python by myself, so it isn't that hard, and that it greatly improved my own work. I want to feel more approachable to students, so they feel comfortable asking questions and going back over the same examples again.
- I can say how Python helped me, for example reproducing quality figures for publications, and it is great to have someone who can help us out when learning.
- I'm teaching a SWC workshop to help other researchers/colleagues improve their work productivity similar to how I have improved my own. I like to teach amazing new ways to make things better. I taught myself python when I was a kid and have some experience in using it on a research level, meaning that I might be able to help you out!

- Hi Welcome to the SWC Python workshop ! My name is Diogo Aguiam and I love using Python in my work, just like you will by the end of this course! So. I taught myself python a few years ago and used it throughout my work. First it was to make good figures for publication. I always have to go back and add another detail. And then I used it to process my data! I have got colleagues to learn Python and they now use it daily in their work instead of MATLAB, IDL, and other commercial tools.
- Share some advice for success: Along the course you may have some doubts about what I am explaining or how to do something. You can ask me anytime to go back over some example and that is Okay, use the post-its to signal me!. It is important to get all the basics well understood and do the exercises by yourself! The exercises we present are structured to make you think how you can do things in python. It requires a bit of effort to do them all but please try them. Even if you are experienced, go over them again and you might learn something new! I know I do.
-

Alice:

- Hi all! I'm Alice, one of your instructors for the day, I'm a molecular biologist by formation, and I switched to bioinformatics and computer science during and after my master. So I didn't have any formal coding class, but I now work as a developer at the Earlham Institute in Norwich. I know probably some of you, if you are like my classmates at university at least, don't really enjoy staying at the computer and would rather keep working in the lab. We actually want to help in that so you can automate some tasks and get back to your research as soon as possible. During the workshop stop me if you get lost at any point or use the post-it so someone can come to help you (point to helpers etc).

  I'll be teaching you this morning, and then in the afternoon you'll have PincoPallino for the lecture about (....). We are going to have a coffee break in about one hour and a half, hopefully we stay on schedule :) You can also come talk to us during the breaks, we'll be around, but again, ask for help if you have problems at the moment, so you don't fall behind. And then at 12;30 there will be the lunch break for one hour, and we'll meet again here. The whole day will finish at 5pm.

Yo:

Hi everyone. My name is Yo Yehudi, and I'll be one of your instructors today for the Software Carpentry Workshop. A bit about me: I'm a Software Engineer at the University of Cambridge, working on the Department of Genetics to create software that integrates biological data. Despite this, I'm actually not a biologist. One of the things I like to do is make it easier for scientists to access the computational side of things, and that's part of why I'm here today - helping to convey some of the skills that don't take terribly long to learn, but will save you time and effort in the long term.

This workshop runs for two days, from 9am until around 5pm. Over this time, we'll have one coffee break in the morning around 11, lunch around 12:30, and another break around 2:30PM. We generally expect for you to be available and present when possible, but if you have any requirements to nip out - perhaps to pick up kids from school or anything like that, please feel free to let me know and we'll try to schedule the breaks appropriately.

Exercise (pick one of those and add to your introduction):1:45

- Describe the prerequisites
- Highlight main aspects of the schedule.
- Communicate the workshop structure
- Explain your expectations
- Share some advice for success

Vasu: Workshop structure
The workshop is split into 2 days. We will start with a setup session to make sure everyone has all the software and prerequistes set up correctly. We will have 2 working sessions in the morning with a Coffee break inbetween at 10:30. Lunch will be from 12:30 - 1:30 PM. We will have 2 working sessions in the afternoon with a coffee break at 3:00 PM and will wrap up for the day at 4:30 PM. We will follow the same structure for tomorrow as well.

Icebreakers:
on this page

Practice Your Introduction (until 14:10)

Imagine you have completed instructor training and you are about to teach a full lesson around the material you have been practicing teaching today.

1. Rehearse your introduction in your mind. (3 minutes)
2. Return to your groups of 2 or 3 and each give your 90 second introduction. (5 min)
3. Discuss what you liked about each other's introductions. (6 min: optional, if there is time.)

This exercise will take 8 minutes or 14 minutes depending on whether time is included for feedback or not.

make the most out the first day by Cornegie Mellon University
https://www.cmu.edu/teaching/designteach/teach/firstday.html

This workshop's website:
https://aschuerch.github.io/2017-10-30-open-online/

There are many ways to get connected with the Carpentry community:
- Our websites are:
    - Software Carpentry
        - Blog
        - Get Involved
    - Data Carpentry
        - Blog

        - Get Involved
- 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
    - Software Carpentry on GitHub
- Software and Data Carpentry share public discussion lists that host everything from lively discussion on teaching practices to job postings and general announcements:
    - General Discussion list
    - Other lists
- We publish a joint newsletter.
- Host monthly community calls and weekly instructor discussion sessions:
    - Check out our community calendar
- You can also find us on
    - Twitter:
        - Software Carpentry on Twitter
        - Data Carpentry on Twitter
    - Slack
    - Facebook

Mateusz' list of most useful links:
https://carpentries.github.io/instructor-training/20-carpentries/
http://lists.software-carpentry.org/listinfo/discuss
http://pad.software-carpentry.org/pad-of-pads
http://pad.software-carpentry.org/instructor-discussion

Exercise: Practice with SWC and DC infrastructure, until 14:55

Go to the SWC workshop template repository or the DC workshop template repository and follow the directions to create a workshop website using your local location and today's date. Put the link for your workshop website into the Etherpad.

This exercise should take about 10 minutes.

Bagus: https://bagustris.github.io/2017-10-31-DataITS/
Andrew: https://andrewsanchez.github.io/2017-10-31-nau/
Yo: https://yochannah.github.io/2017-11-10-swcarpentry-workshop-wimblington/
Désirée: https://desireetreichler.github.io/2017-11-14-dummy/
Ahmed: https://ahmedmoustafa.github.io/datacarpentry_workshop_example/
David: https://daviddesancho.github.io/2017-10-31-SCWtest
Vasu: https://vasudevanv.github.io/2018-02-30-serengeti/
Diogo: https://daguiam.github.io/2017-10-31-Lisbon-Portugal/
Alice: https://aliceminotto.github.io/2017-10-31-norwich/

Exercise: Organize Your Knowledge
digital copy here

In groups of 3-4, fill out this worksheet, listing all of the teaching strategies and techniques you would use at each stage of a workshop. Be sure to include implementation details such as how long you would spend or how often you would do a particular strategy as well and any other information you think is important.

When you have a good amount of information in each box, type “done” in the chat.

5 min for thinking about it

5 minutes discussing in groups

back 15:45

Andrew:
Yo + Andrew + Désirée:
Preparation:

- Webpage
- Find instructors
- Ensure everyone knows the roles
- Dry run
- Makes sure location is ok and meets needs
- CoC

Workshop intro

- CoC again
- Introducing yourself and why you’re qualified to teach
- Introduce co-instructors and helpers
- Share content and objectives
- General setup of the day = lunch, breaks, etc.
- Break the ice - intros

Teaching: Formative assessment

- Frequent low-stakes or no stakes quizzes. Can be polling style, allows us to understand whether or not students understand. Keep very frequent!
- Sticky notes?
- Exercises can count as a form of assessment too!

Active learning:

- Designing exercises - live coding, having everyone follow along, typing together in real time.
- Breaking things up - don’t let people fall asleep. Some talk, then exercises.
- Engaging discussion.
- Encouraging interaction *between* students.

Practical skills:

- Basic coding skills, e.g. tidy code, good principles, for loops,
- Trusting your own abilities
- R
- Git
- Python
- Where to find help

Feedback to learners:

- Effort and Improvement-based assessment, rather than performance.

Désirée:
David:

Vasu/Bagus/Ahmed:
Preparation -
Instructor notes, Etherpad, Know your audience, pre-survey, website, sticky notes
Introduction -
Demonstrate qualification, experience and be approachable, ice breakers, accessibility Planning,
code of conduct, workshop structure
Formative assessment -
Multiple choice and misconception checks, mental models
Active learning -
Live coding, concept maps.
Feedback -
1up 1down, improvement based feedback.
End -
Post survey and feedback for instructors and organizers

Diogo + Alice + David:

Preparation:

Read the preworkshop surveys to understand the audience
Prepare a way to make yourself approachable to the audience
Check the syllabus
Send pre requisites
Meet with the co instructors
Prepare the workshop website for the venue, etc
Make sure the pre installed software is there, have a contact person from the different centers

Workshop Introduction:

Be approachable
Introduce yourself tell why you are teaching this and why it is relevant to you
Introduce the syllabus
Motivate the learners towards the tasks of the workshop
Explain the post-it feedback, How to interact, how to ask for help
Explain the schedule, when are the breaks.

Formative assessment:

Become aware of misunderstandings in the audience
Used to refocus the learners after each topic

Use post its to tell when it is done

Active learning:

Live-coding
Solving exercises
Follow the exercises in own laptops

Practical Skills

Solving the exercises
See how the learners can apply this in their work

Feedback to learners

A way for the learners to review their understanding
Using the helpers to help learners around the room

Workshop end

Fill out the post workshop survey
post it feedback
Summarise the workshop
What to do afterwards: workshop syllabus on the web

http://swcarpentry.github.io/shell-novice/

What Are the Challenges?
What are some of the challenges you might expect when teaching learners with a broad range of expertise?

- Be very clear about when you move on to a new topic (so that you catch the attention of those who dropped out)
- Diverse classes: consider asking advanced students to help the slower students
- Make the more advanced help the others
- Give them something fun to learn
- Tell them to take the rest of the day off
- Get helpers to support the learners who are slower and are about to fall behind. Also consider at which point this is happening, don't let anyone behind for the first part that will be needed for understanding everything else.
- Assign helpers to slower students, but not dedicate to a person.
- Survey the students before the class to learn about their backgrounds and plan accordingly.
- If not possible to learn about the background in advance, plan to go along with the slower/less advanced group.
- Ask the more advanced student to assist the less advanced students.
- Encourage interaction between students with different backgrounds and skill levels.

communication, pages, mailing lists:
https://carpentries.github.io/instructor-training/20-carpentries/
-> for the instructor newsletter, click on SCW "get involved", and scroll down to the Conversations section

Finalising the instructor training process:

https://carpentries.github.io/instructor-training/checkout/

Summary, check-out procedure:

- Fill in application instructor application form
- Submit a small contribution (issue, pull request)
- Take part in Discussion Session
- Teach a short Demonstration Lesson

Mentoring groups:
Mentoring › Carpentries Mentorship Program - 2.0
https://software-carpentry.org/blog/2017/10/mentoring-wrap-up.html

Carpentries instructors worldwide:
https://software-carpentry.org/team/

Post-workshop survey:
https://www.surveymonkey.com/r/post-instructor-training

Follow-up of workshops:
https://hackyhourstluc.wordpress.com/
https://science.mozilla.org/programs/studygroups