Teaching Sciences & Engineering with Jupyter Notebooks

Cécile Hardebolle
cecile.hardebolle@epfl.ch

7 March 2023

Learning goals

At the end of this session, you should be able to

- Explain what **Jupyter Notebooks** are
- Describe how notebooks can be used **for teaching and learning**
- Analyze which **features** can **foster learning** in a notebook
- **Design notebooks** for teaching and learning activities

What make notebooks effective for learning?

Let’s do some physics with a notebook ✍

First, let me briefly introduce how to use the notebook

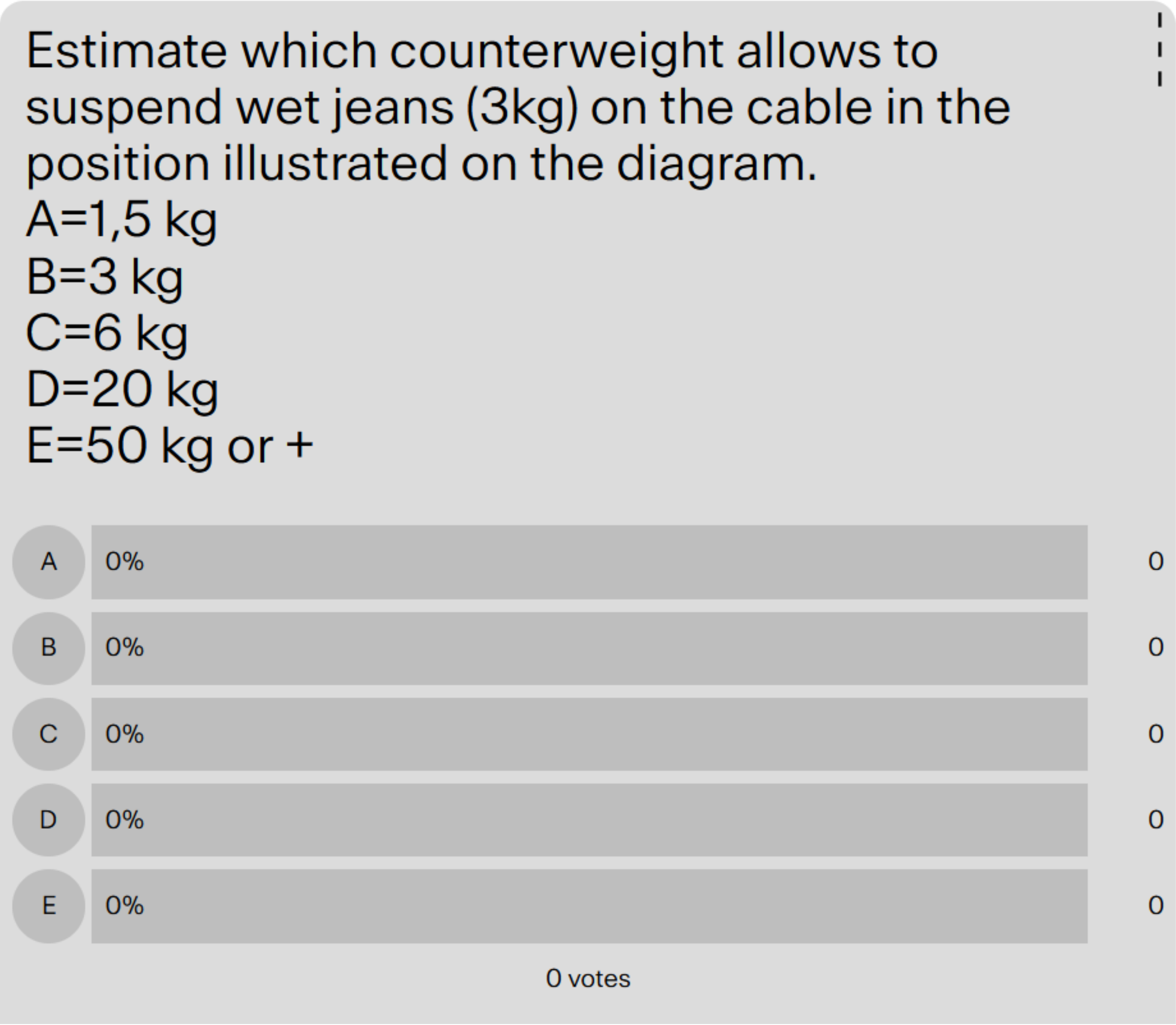
Go ahead! → **https://go.epfl.ch/nbw**

Use Firefox or Safari
(avoid Chromium)

There are **4 activities** including

- Some basic programming
- Interactive questions based on SpeakUp

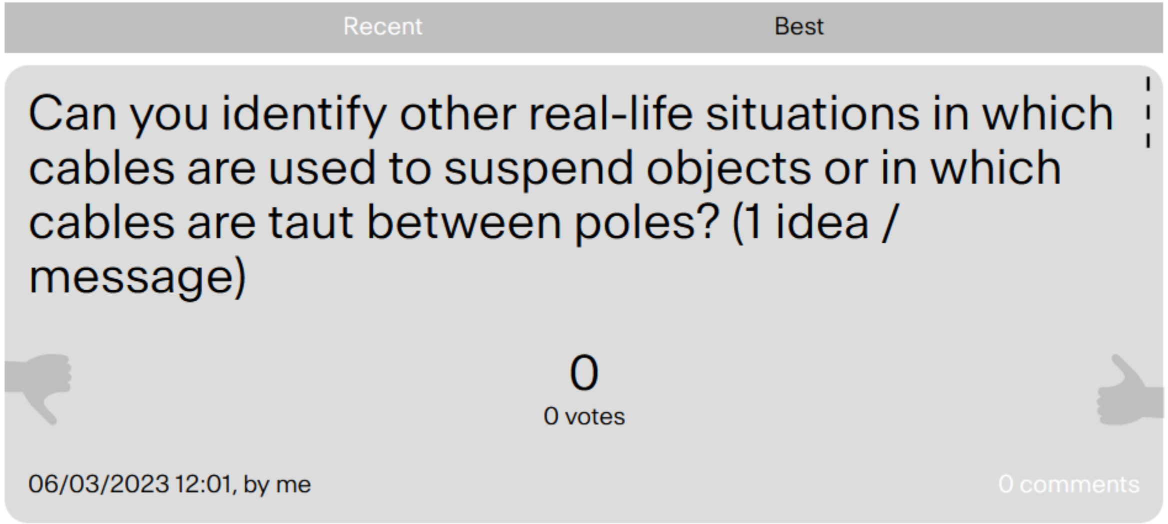
Let’s debrief the physics



Estimate which counterweight allows to suspend wet jeans (3kg) on the cable in the position illustrated on the diagram.
A=1,5 kg
B=3 kg
C=6 kg
D=20 kg
E=50 kg or +

A 0% 0
B 0% 0
C 0% 0
D 0% 0
E 0% 0

0 votes



Can you identify other real-life situations in which cables are used to suspend objects or in which cables are taut between poles? (1 idea / message)

0
0 votes

06/03/2023 12:01, by me

0 comments

https://speakup.epfl.ch/

Poll room: 14804

Chat room: 45263

Which elements in this notebook did you find the most helpful for learning?

URL: http://ttpoll.eu
Session ID: noto

Select <u>up to 3 features</u> :

0% a. Integrated poll (“what does your intuition say”)

0% b. Explanatory text, diagrams and equations

0% c. Interactive visualization

0% d. Mini-activities reading/modifying/writing Python code

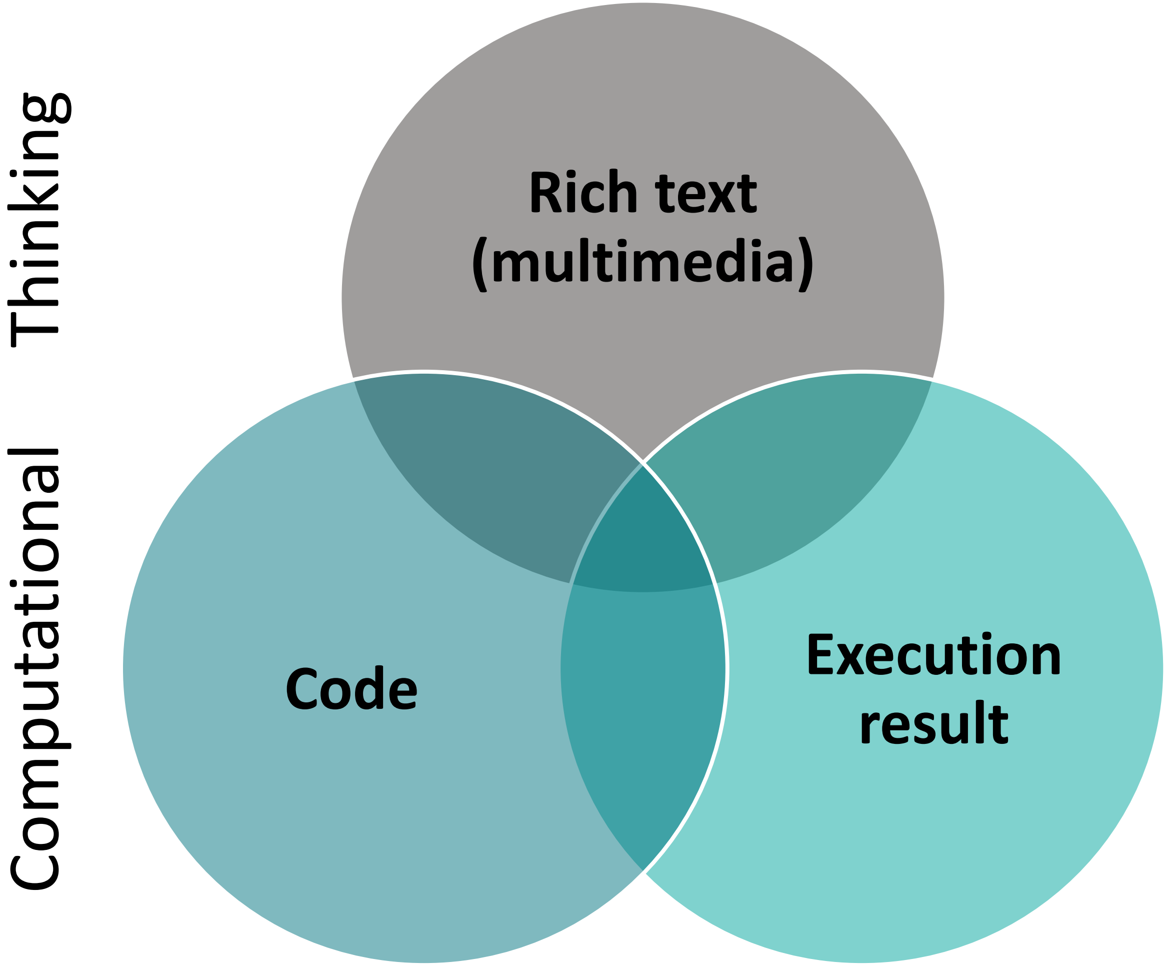
0% e. Solutions of the activities

0% f. Ability to take notes/add/modify text cells

0% g. Integrated chat (“what have you learned so far”)

0% h. Other

What make notebooks effective for learning?



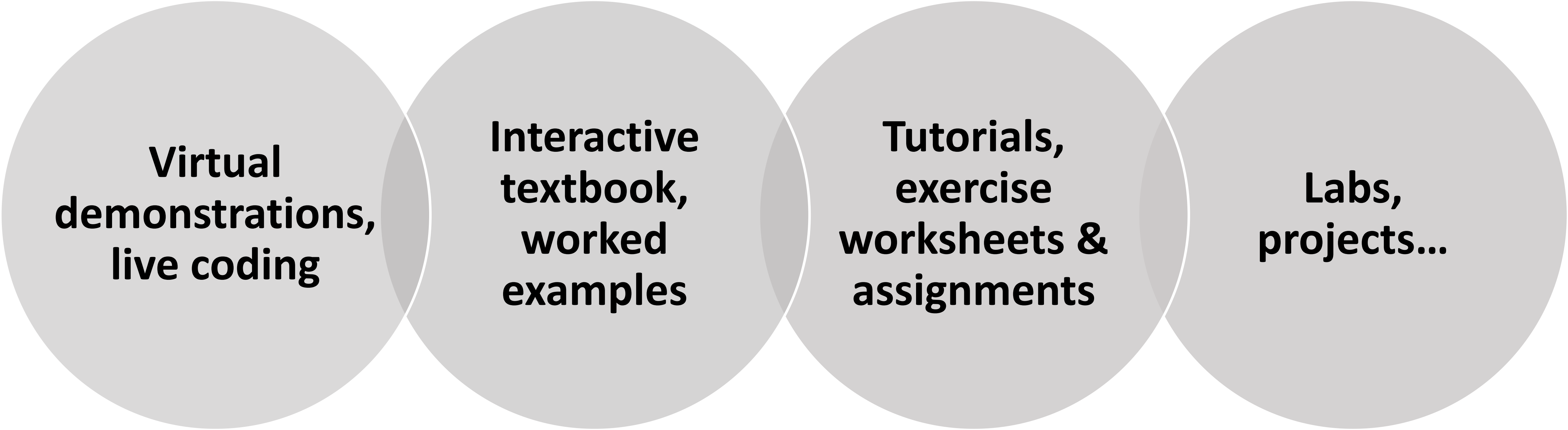
- Multiple **representations**
- **Interaction** and manipulation of representations

Effective for learning **<u>under certain conditions</u>**

[Knuth, 1984 ; Millman & Pérez, 2018]

[Ainsworth, 1999 ; Mayer, 2009]

When / for what to use notebooks?



Virtual demonstrations, live coding

Interactive textbook, worked examples

Tutorials, exercise worksheets & assignments

Labs, projects…

Designing a notebook for your course / your MOOC

Creating a new notebook

1. Create a new notebook:
   - Click on the blue “+” button at the top left of the workspace
   - Select “Python 3” (or another language if you prefer) in the notebook category

2. Rename your notebook:
   - Right-click on your notebook, then choose “Rename” in the menu
   - Change the name to, then press enter

3. Add some content:
   1. Add a **new cell** to your notebook (“+” button at the top of the notebook)
   2. **Convert it to a Markdown cell** (dropdown menu at the top of the notebook)
   3. Write some text and then “execute” the cell to **render** it

Is programming part of your learning goals?

URL: http://ttpoll.eu
Session ID: noto

a. Yes, learning programming is a core goal of the course

b. Yes, but it is not the core goal of the course
(part of the assessment may include some programming, but I focus on teaching some other content)

c. No, programming is used only to illustrate parts of the course
(students are not assessed on programming)

d. Other

How much programming experience do your students have?

0% a. None

0% b. Limited experience (e.g. 1 semester)

0% c. Some experience, but with another programming language

0% d. Some experience with the language I intend to use

0% e. Extensive experience (e.g. with multiple languages)

0% f. I don’t know

0% g. Other

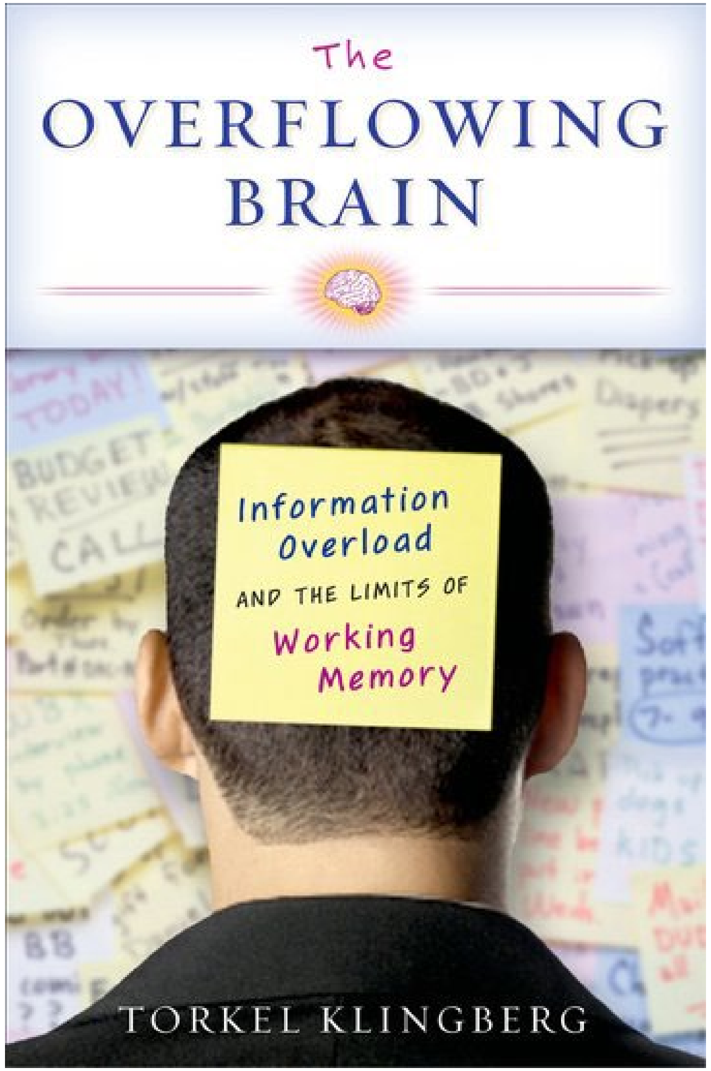
Too much, too quickly

"Overflowing Brain": We have **limited capacity to cope with new information**

Juggling

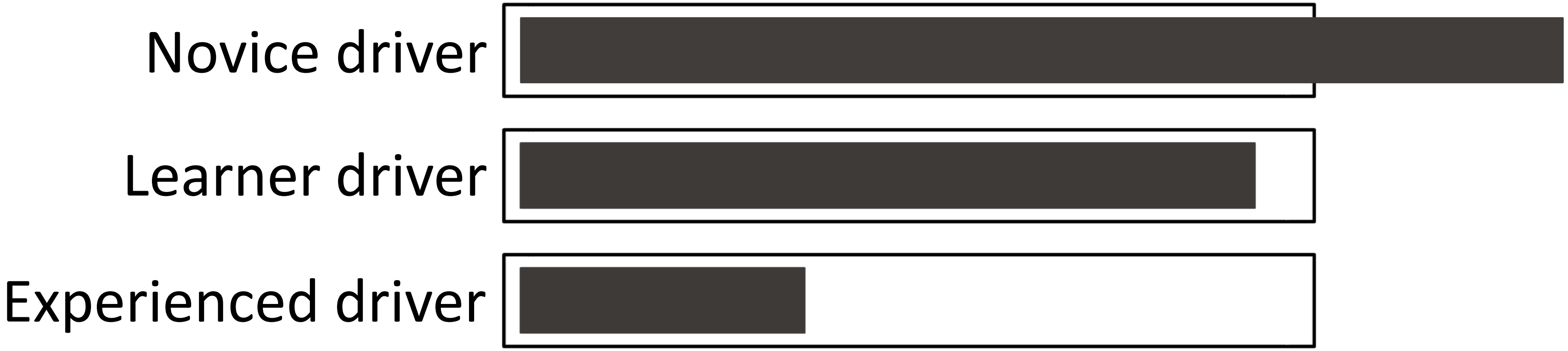
- Concepts
- Notations
- How concepts link/relate
- Strategies for solving
- Calculations…

Easily overwhelmed



Cognitive load
= cost of a task for the cognitive system

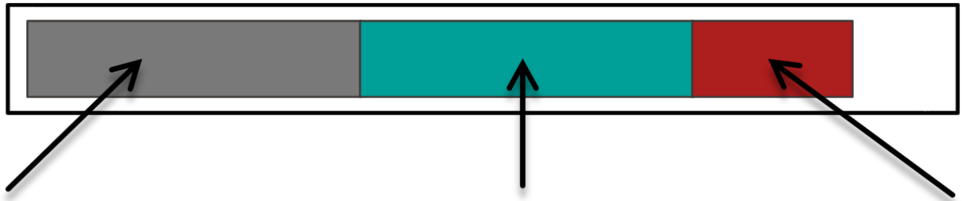
Depends on the **previous ability** of the person on the task:



[Sweller, 1998]

Cognitive load
= cost of a task for the cognitive system

Depends on the **task itself**:



Intrinsic load
= **complexity** of
the task/content

scaffold

Germane load
= necessary **effort**
for learning

optimize

Extraneous load
= **issues** related to
presentation / format

limit

[Sweller, 1998]

Scaffold the intrinsic load: importance of progression

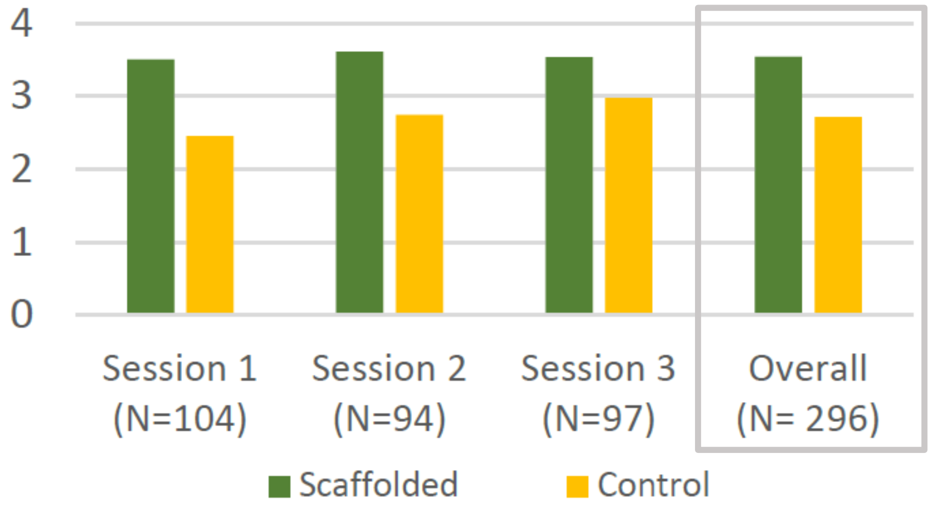
Designing a notebook for your course / your MOOC

Progression in exercises

Comparison:

- 296 EPFL students in 2 groups
  - Control: doable, slightly complex exercise
  - Scaffolded: 2 simpler exercises
    + the same slightly complex exercise as others
- Equal time to both groups
- Scored only on complex task

Average score on geometry task for scaffolded and control groups



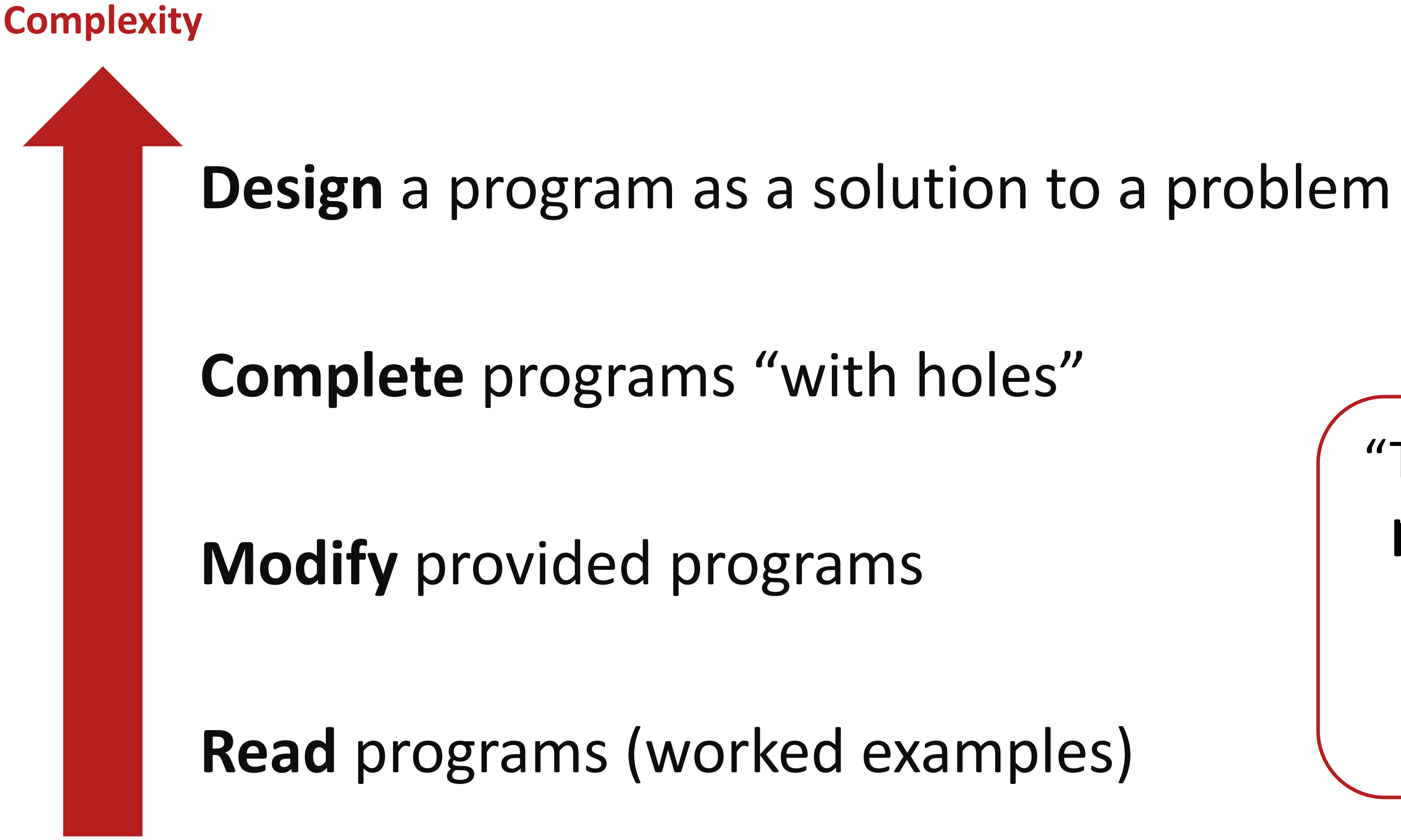
Result: **significantly higher attainment in scaffolded group**, despite effectively having less time

Scaffolded = ‘practice’ exercises

- **remember what they know**
- **build fluency**

[Babel, Benketaf & Schukraft, 2014]

Scaffolding programming activities



Complexity

Design a program as a solution to a problem

Complete programs “with holes”

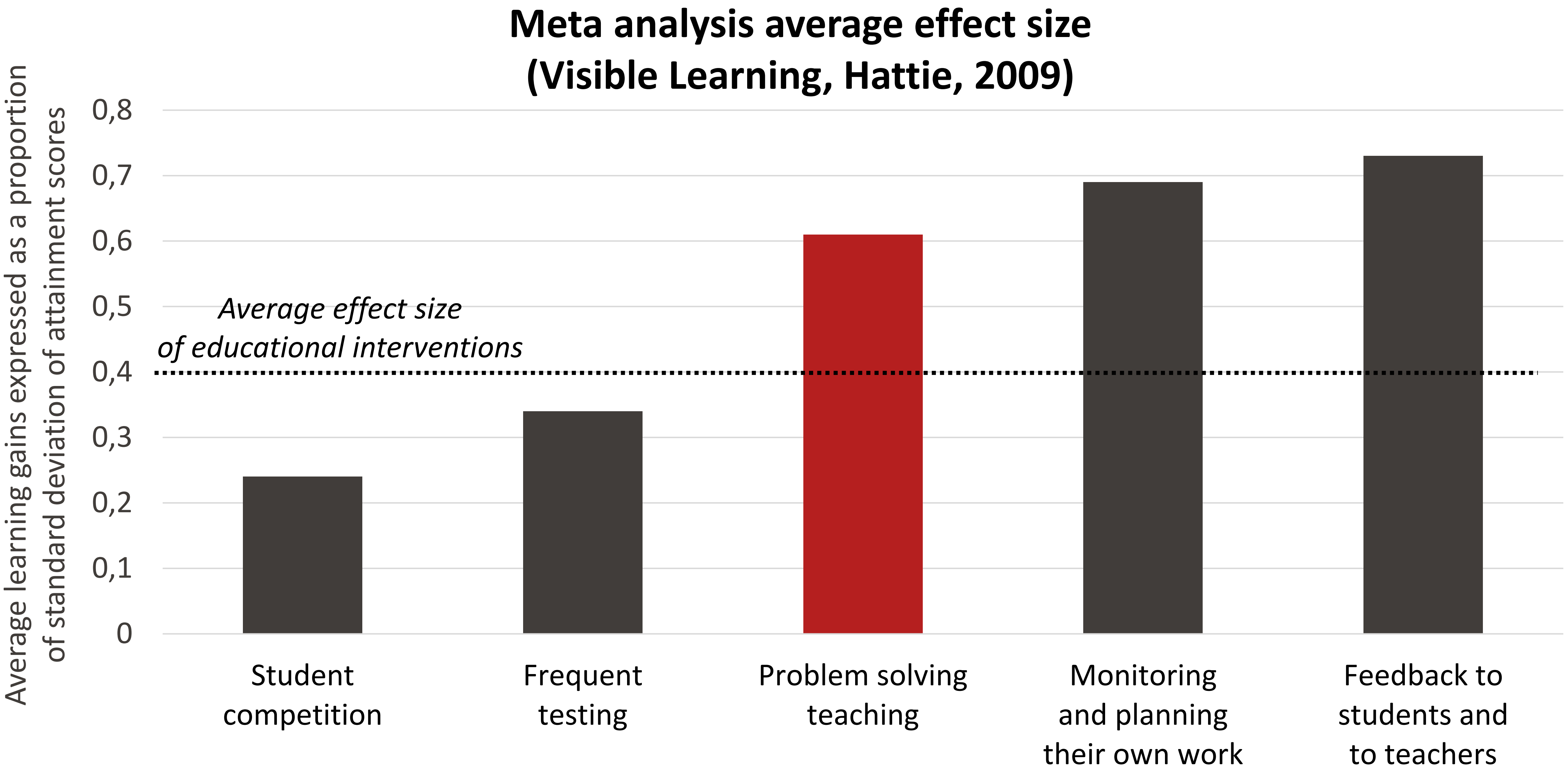
Modify provided programs

Read programs (worked examples)

“These activities emphasize the **pedagogical value of reading code**, as opposed to merely designing and writing it (Sorva, 2012). ”

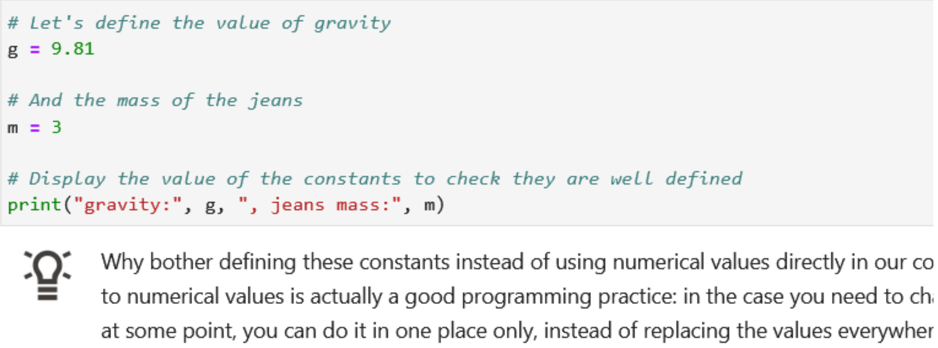
[Robins 2019]

Evidence on teaching strategies



Modeling expert thinking in notebooks

- **Problem solving narrative**
- Illustrated with:
  - Diagrams/sketches
  - Equations
  - **Code**
  - Interactive visualizations
  - …

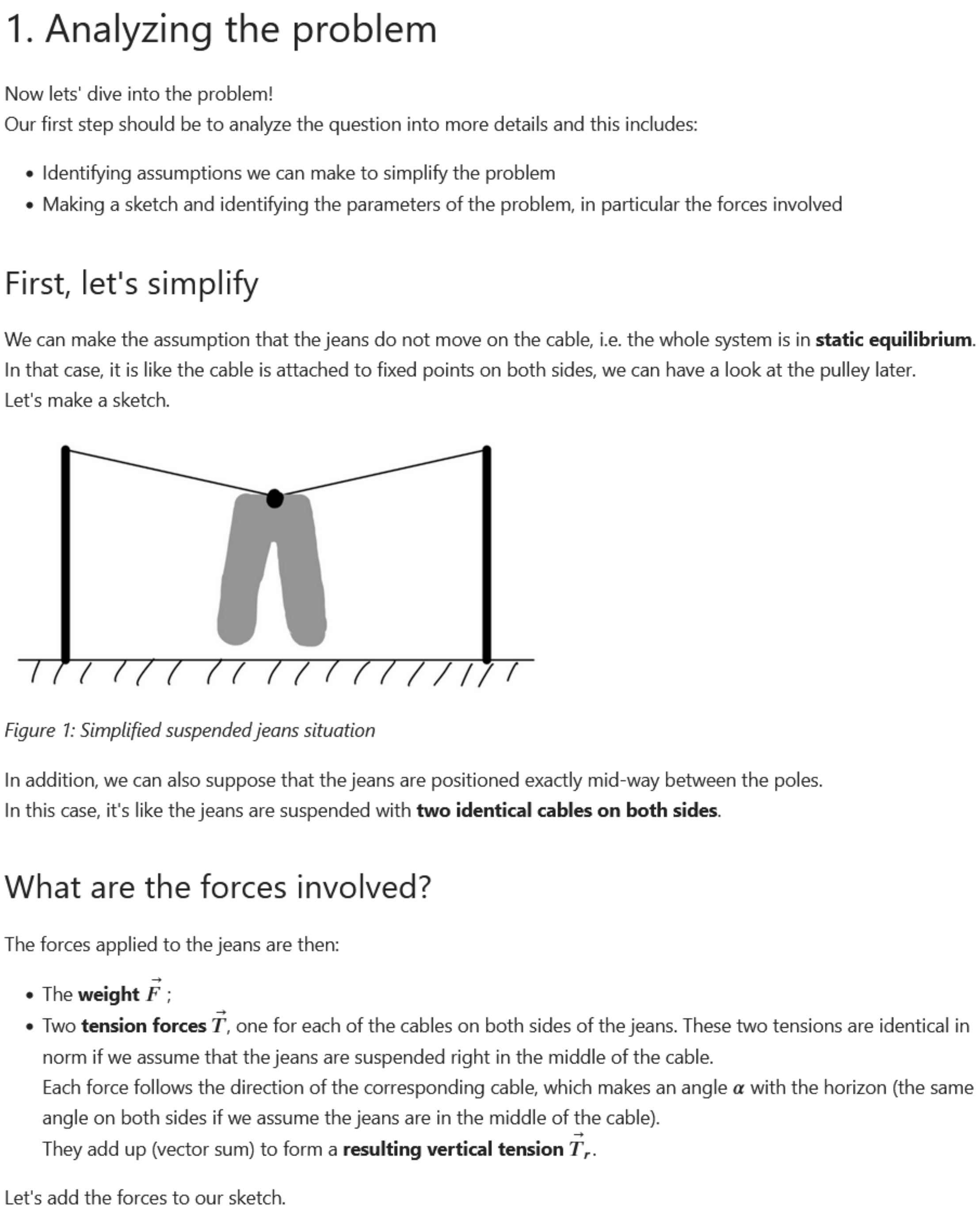


```
# Let's define the value of gravity
g = 9.81

# And the mass of the jeans
m = 3

# Display the value of the constants to check they are well defined
print("gravity:", g, ", jeans mass:", m)
```

Why bother defining these constants instead of using numerical values directly in our co
to numerical values is actually a good programming practice: in the case you need to ch
at some point, you can do it in one place only, instead of replacing the values everywher



1. Analyzing the problem

Now lets' dive into the problem!
Our first step should be to analyze the question into more details and this includes:

- Identifying assumptions we can make to simplify the problem
- Making a sketch and identifying the parameters of the problem, in particular the forces involved

First, let's simplify

We can make the assumption that the jeans do not move on the cable, i.e. the whole system is in **static equilibrium**.
In that case, it is like the cable is attached to fixed points on both sides, we can have a look at the pulley later.
Let's make a sketch.

Figure 1: Simplified suspended jeans situation

In addition, we can also suppose that the jeans are positioned exactly mid-way between the poles.
In this case, it's like the jeans are suspended with **two identical cables on both sides**.

What are the forces involved?

The forces applied to the jeans are then:

- The **weight** $\vec{F}$;
- Two **tension forces** $\vec{T}$, one for each of the cables on both sides of the jeans. These two tensions are identical in norm if we assume that the jeans are suspended right in the middle of the cable.
  Each force follows the direction of the corresponding cable, which makes an angle α with the horizon (the same angle on both sides if we assume the jeans are in the middle of the cable).
  They add up (vector sum) to form a **resulting vertical tension** $\vec{T_r}$.

Let's add the forces to our sketch.

Scaffolding activities in your notebook

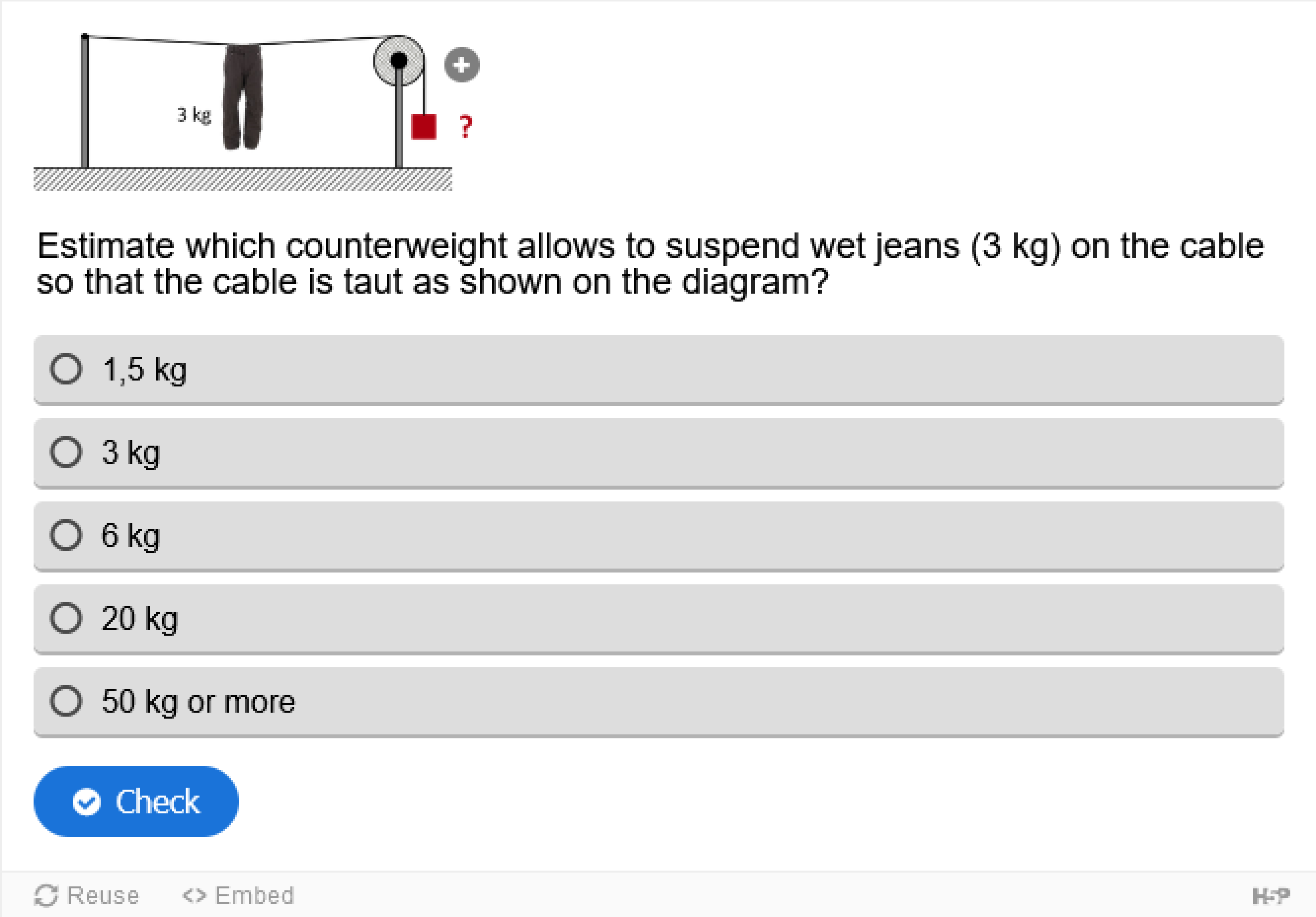
Think about the most complex task/exercise you would like students to be able to do in your notebook.
Which **preparatory activities** could help students to:

- Remember what they know (e.g. quiz)
- Learn about problem solving strategies (e.g. worked example to analyze)
- Build fluency (e.g. simpler questions)

Note down 1 or 2 ideas, trying to be specific (e.g. on what topic would the quiz be? What type of worked example would you provide?).

→ https://go.epfl.ch/nbw-doc

Integrating a quiz into a notebook



Estimate which counterweight allows to suspend wet jeans (3 kg) on the cable so that the cable is taut as shown on the diagram?

○ 1,5 kg

○ 3 kg

○ 6 kg

○ 20 kg

○ 50 kg or more

Integrating a quiz into a notebook

1. Create an “interactive content” activity in moodle (H5P)
2. Check the visibility of the activity
   - Who has access to the course page?
   - Is the activity visible?
3. Find the HTML code to embed the activity (iFrame)

```
<iframe src="https://moodle.epfl.ch/mod/hvp/embed.php?id=1213682"
width="1556" height="310" frameborder="0" allowfullscreen="allowfullscreen"
title="MonthsQuestion"></iframe>
```

4. Create a (Python) code cell which displays the iFrame

```
from IPython.display import IFrame
IFrame('https://moodle.epfl.ch/mod/hvp/embed.php?id=1213682', 500, 350)
```

More details on https://go.epfl.ch/noto-quiz

Inserting an image into a notebook

Option 1:

- Navigate to a folder into your workspace
- Drag-and-drop the image file onto your workspace to upload it
- Use Markdown or HMTL to insert the image into your notebook using the path to the image file:

Markdown: `![alternative text](path-to-image)`
HTML: `<img src="path-to-image" alt="Alternative text" />`

Option 2:

- Edit a Markdown cell
- Drag-and-drop your image directly onto that cell

Optimize the germane load: effort that pays back

Designing a notebook for your course / your MOOC

Learning from doing

“We do not learn from experience.
We learn from **reflecting** on experience.”
(Dewey, 1933)

Questions to trigger reflection

Help students make sense of what they are doing by asking **why**, **what** and **how**, **before / after** the learning activities

CC 2.0 BY Sam on Flickr

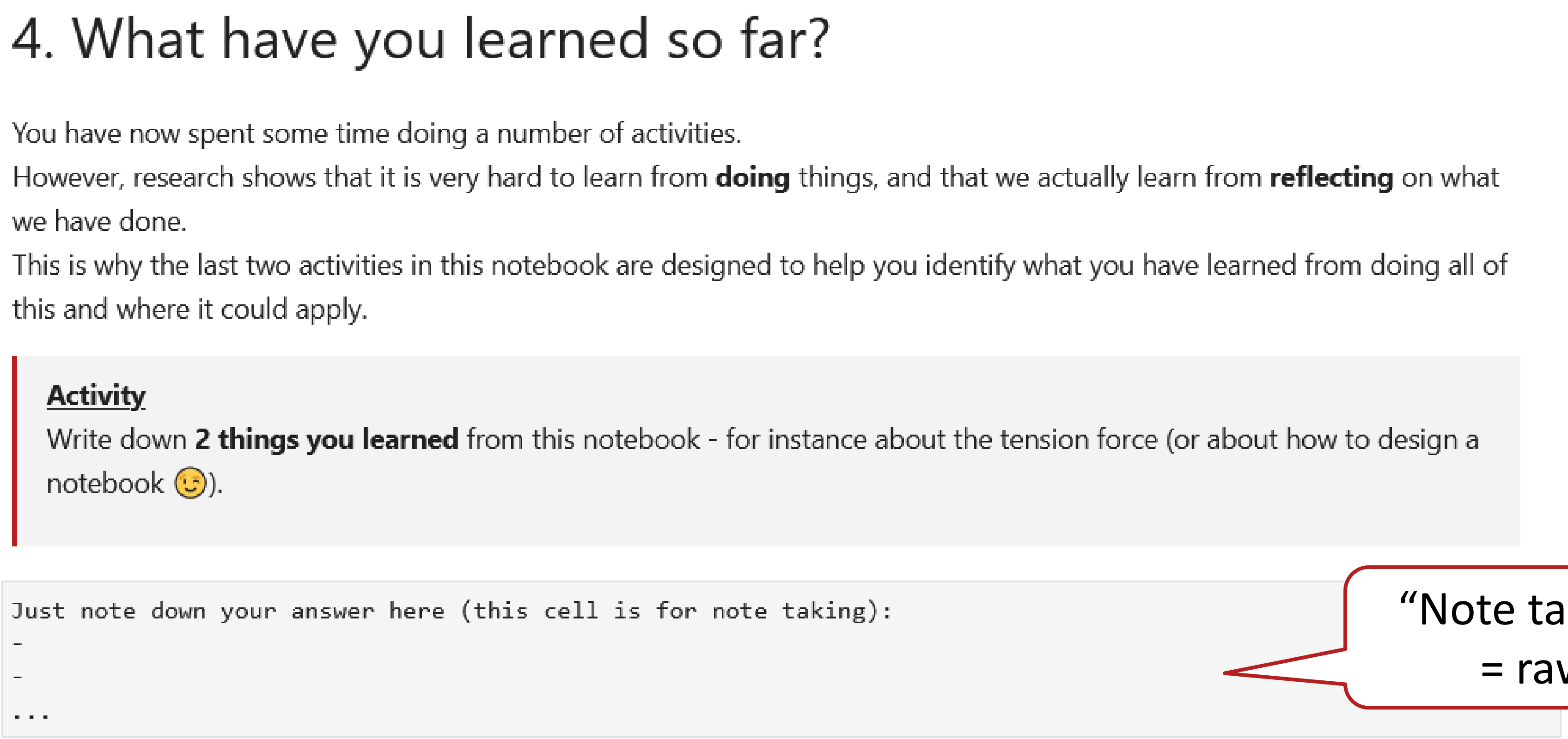
Examples:

- “Why are you asked to do this?”
- “How do you plan to do this?”

- **“What have you learned from doing this?”**
- “How does this relate to X?”, “How could this be used in your project?”
- “What would you do differently next time?”

[Panadero, Klug & Järvelä, 2016]

Reflection questions in action



4. What have you learned so far?

You have now spent some time doing a number of activities.
However, research shows that it is very hard to learn from **doing** things, and that we actually learn from **reflecting** on what we have done.
This is why the last two activities in this notebook are designed to help you identify what you have learned from doing all of this and where it could apply.

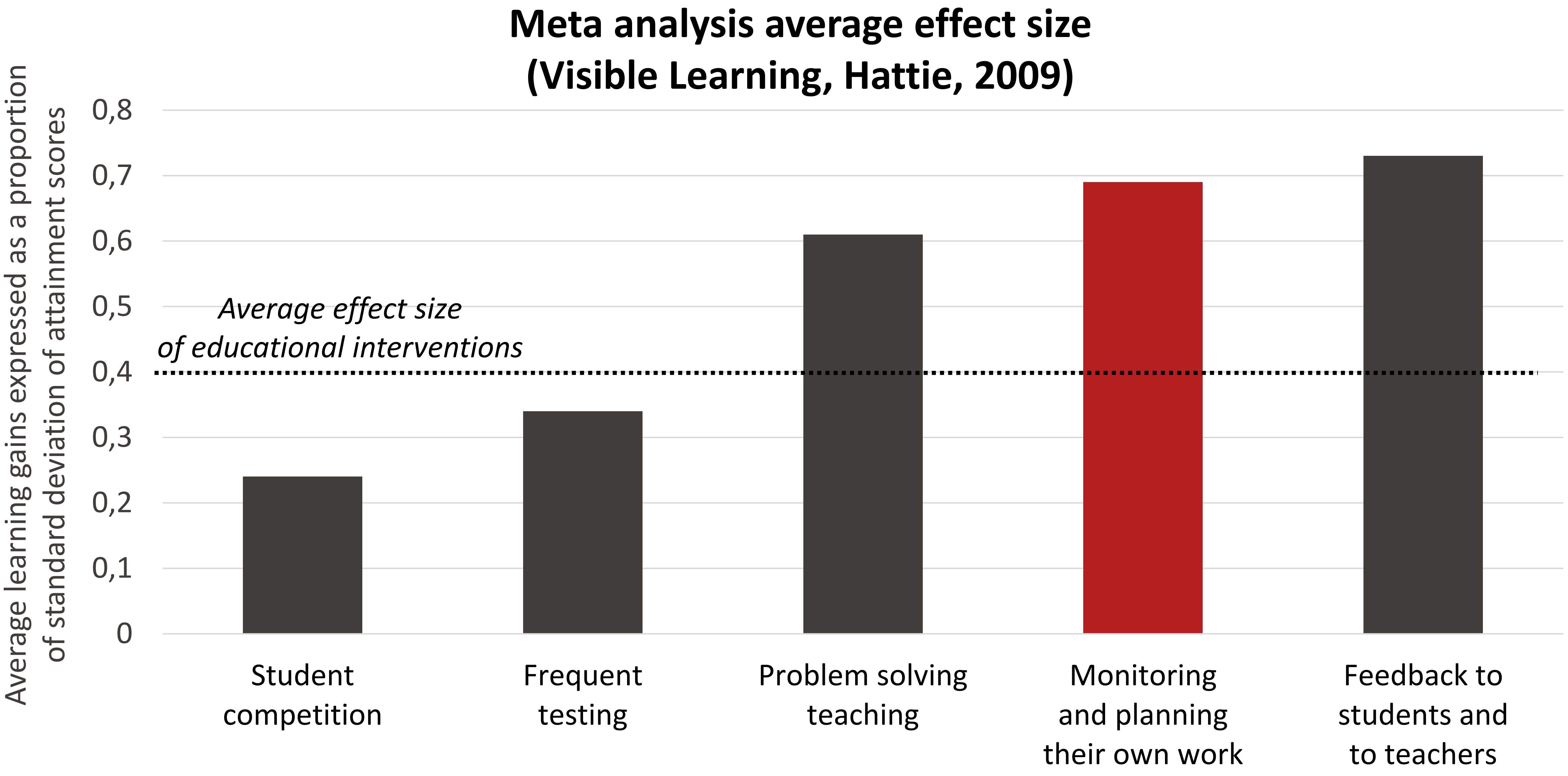
> **Activity**
> Write down **2 things you learned** from this notebook - for instance about the tension force (or about how to design a notebook 😉).

```
Just note down your answer here (this cell is for note taking):
-
-
...
```

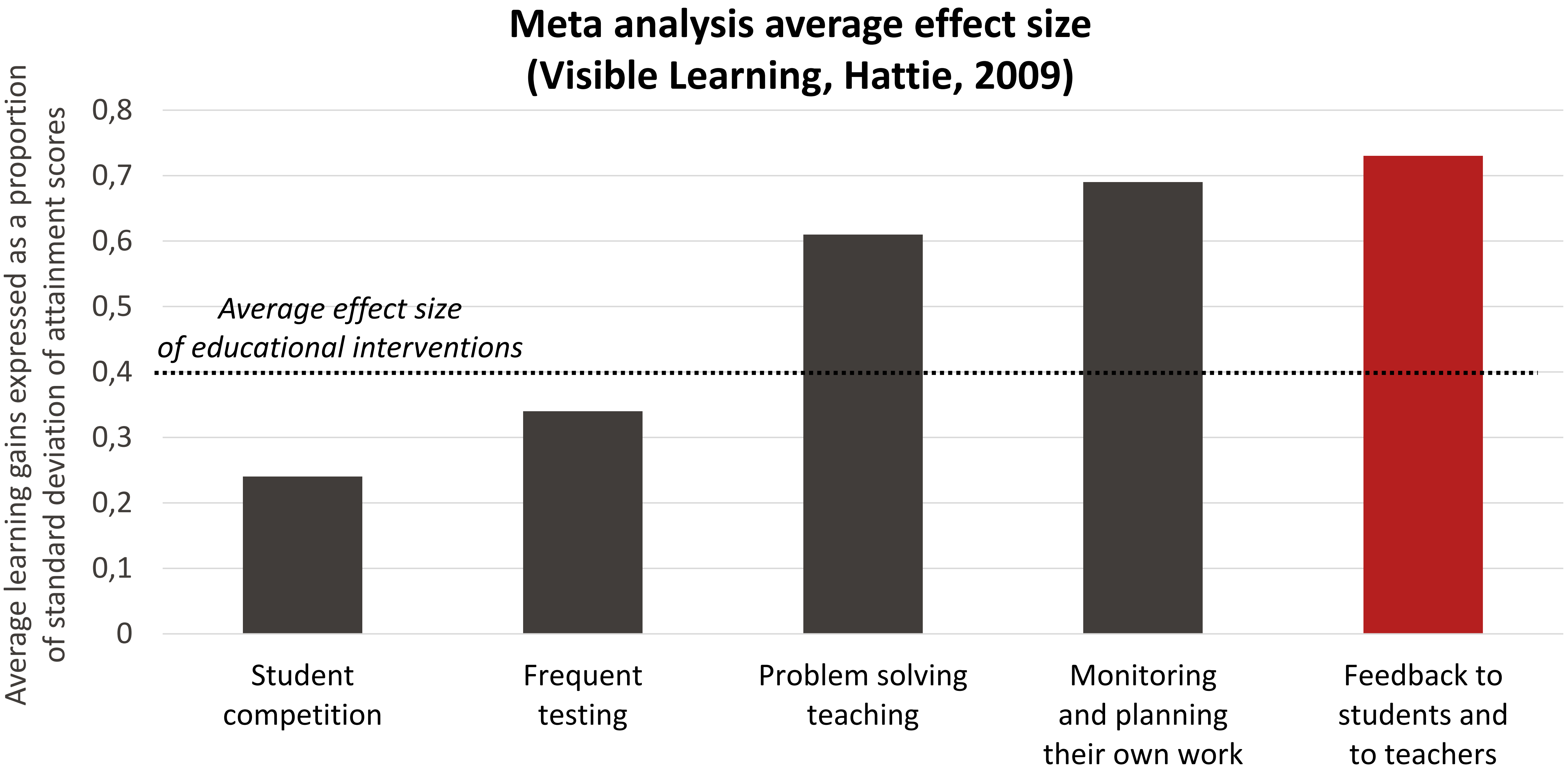
"Note taking" cell
= raw cell

Which **reflection questions** could you use in your notebook?
Note down 1 or 2 ideas.

Evidence on teaching strategies



Evidence on teaching strategies



How can *students* get feedback on their learning in notebooks? ✍

List different ways in which **you** got **feedback** on your learning when working in the physics notebook as a student:

Solutions in notebooks

Option 1: hidden Markdown or Code cell

- Immediately available
- /!\ Check the state before sharing

Option 2a: separate file

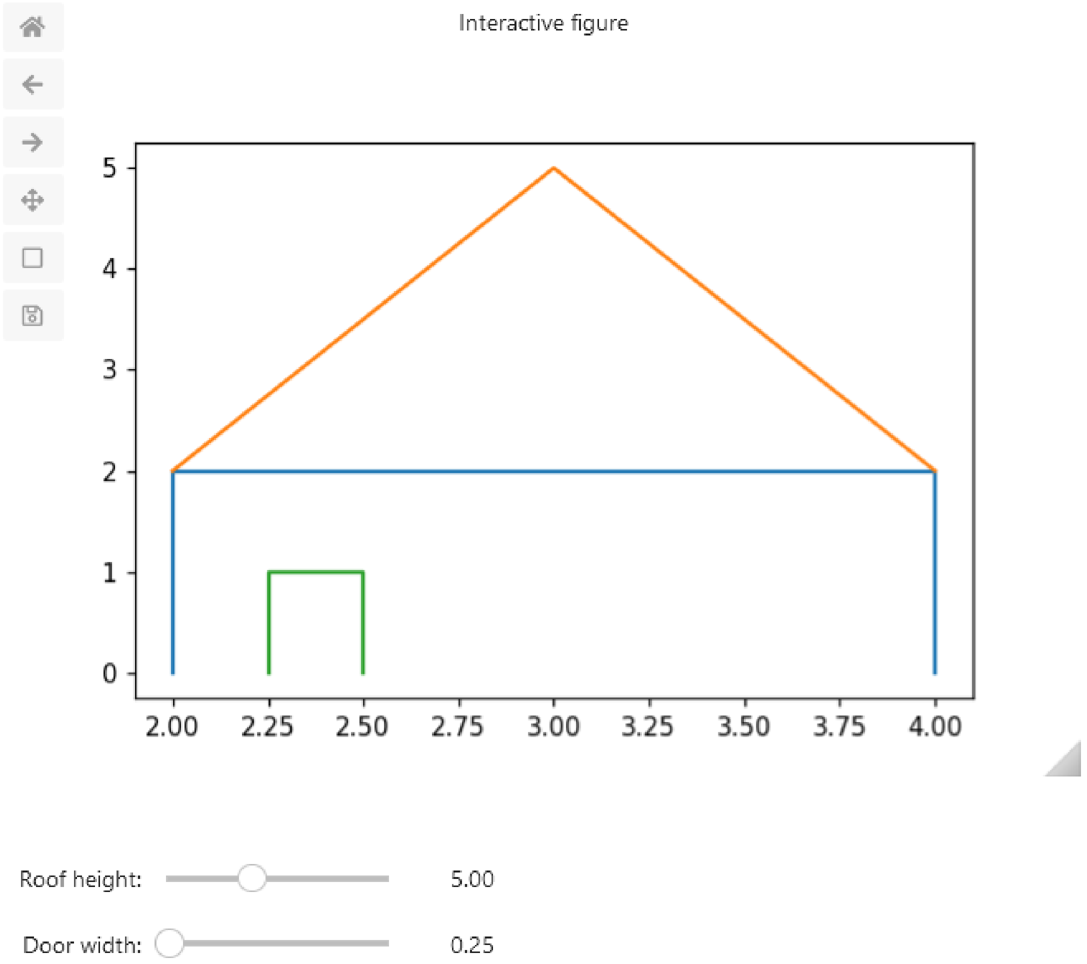
- Can be made available with delay
- /!\ Harder to maintain

Option 2b: separate file generated automatically

- Easier to maintain
- Can be made available with delay
- /!\ Requires the use of tags & command line

Interactive visualizations with Python

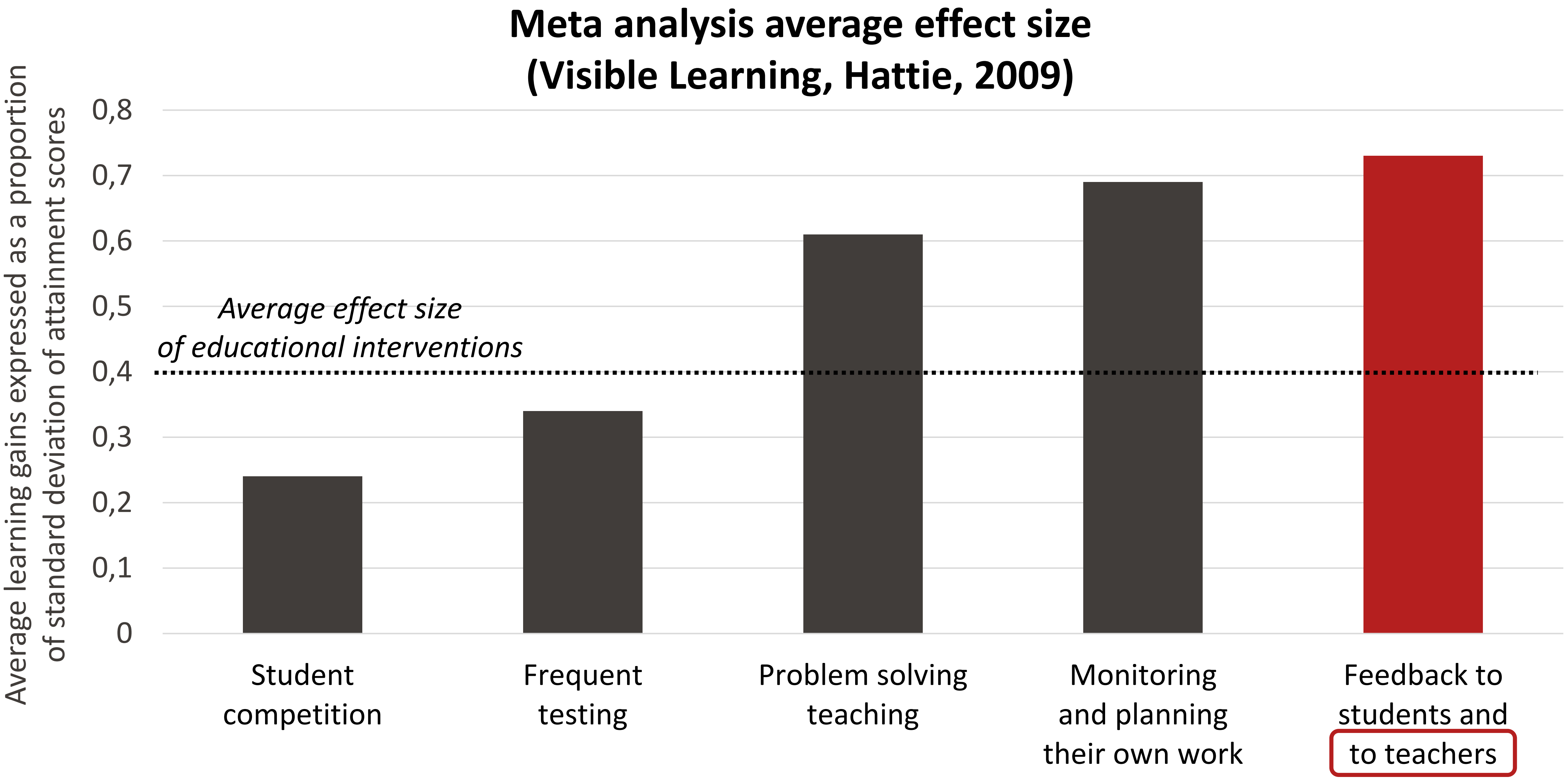
1. Open the notebook GettingStarted-02-InteractiveVisualization.ipynb
2. Follow the worked example
3. [Optional] Extend the example with the activity at the end



Roof height: 5.00

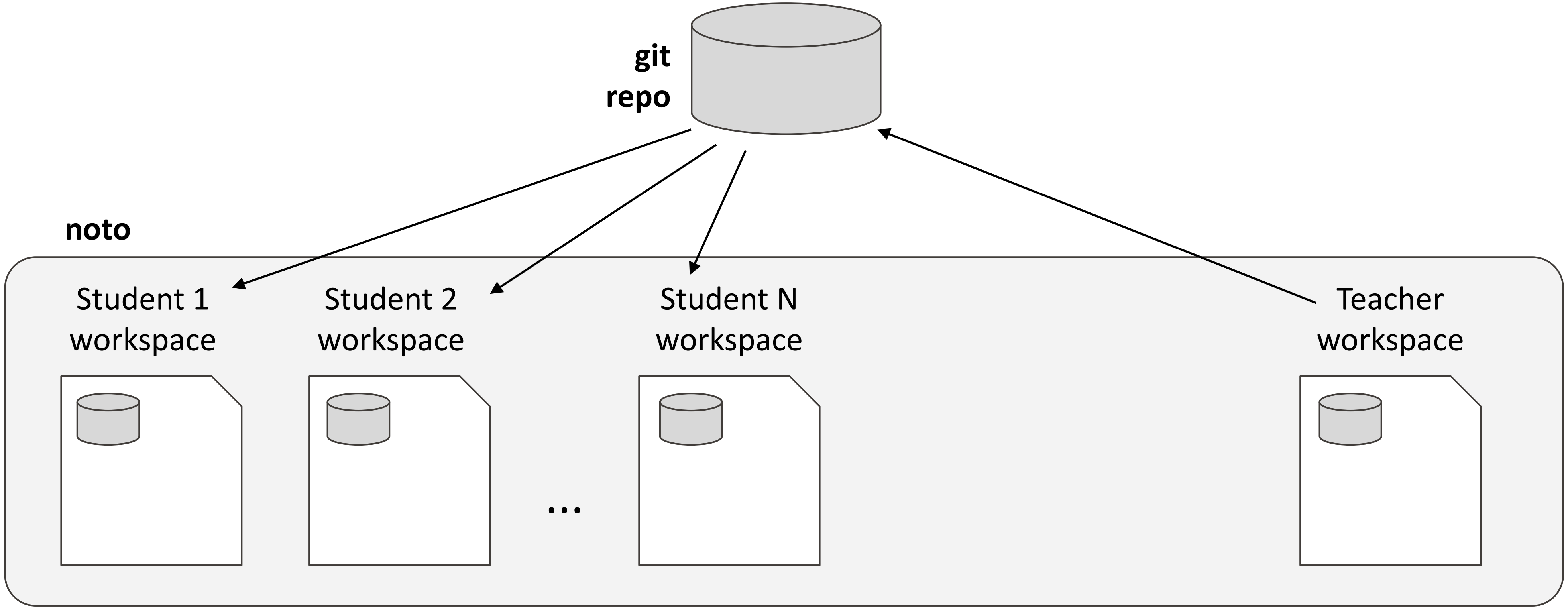
Door width: 0.25

Evidence on teaching strategies



Feedback to teachers in notebooks

Challenge: currently, when students are working on notebooks, by default the teacher doesn't get any information on what they are doing



As a teacher, how can *you* get feedback on how student work with notebooks?

Integrated tools

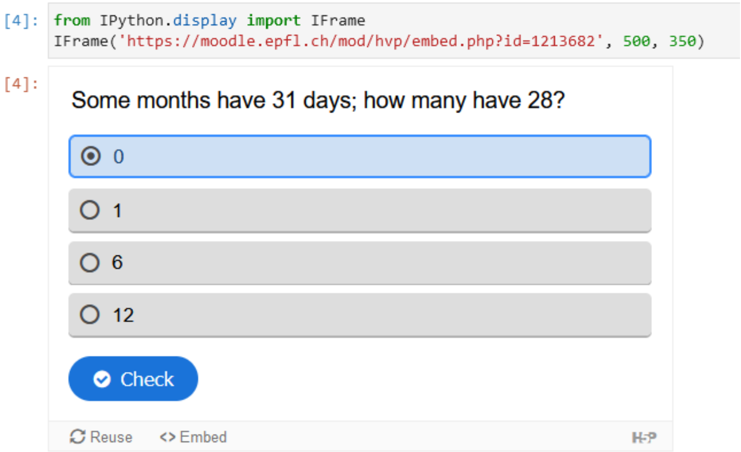
- **Poll** (SpeakUp)
- **Chat** (SpeakUp)
- **Quiz** (H5P/moodle)
- **Survey** (SurveyMonkey)
- Code/assignment **submission**

[FUTURE] Teacher analytics dashboard

Other external tools

- Clickers
- Statistics (go.epfl.ch)
- Questionnaires

Asynchronous quiz with H5P/moodle

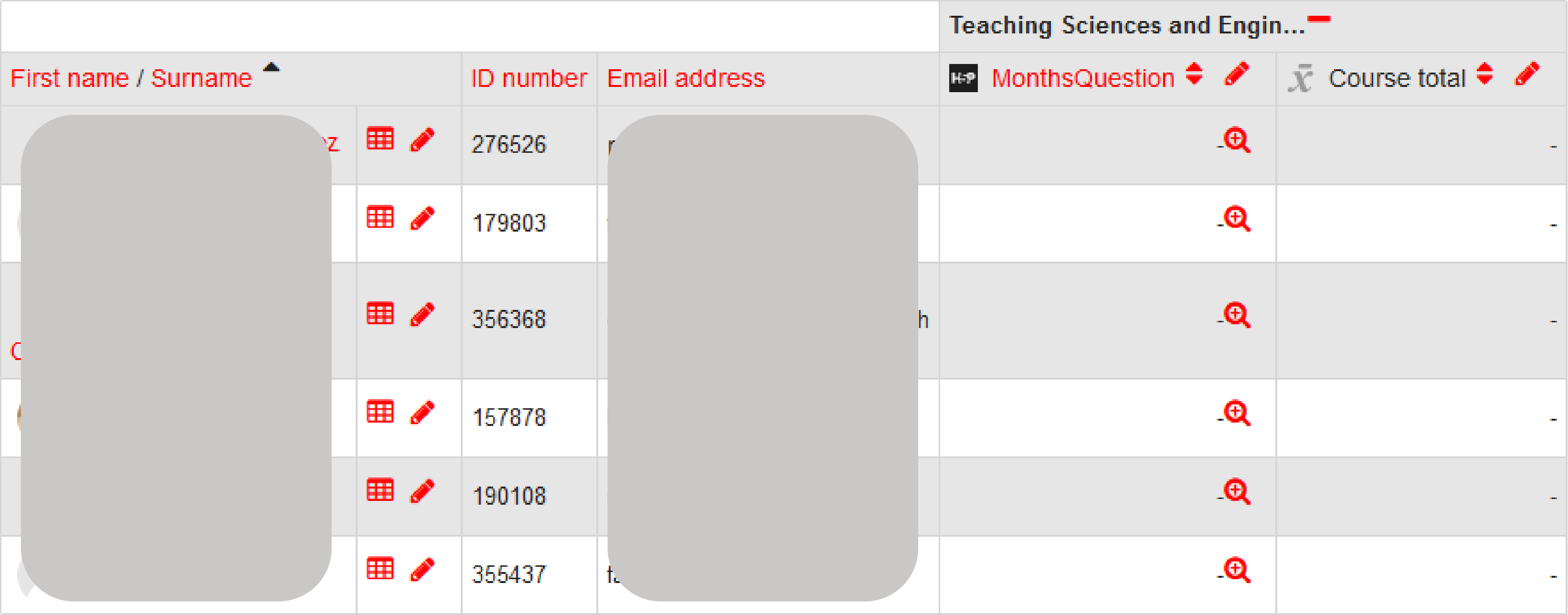


```
[4]: from IPython.display import IFrame
     IFrame('https://moodle.epfl.ch/mod/hvp/embed.php?id=1213682', 500, 350)
```

[4]:

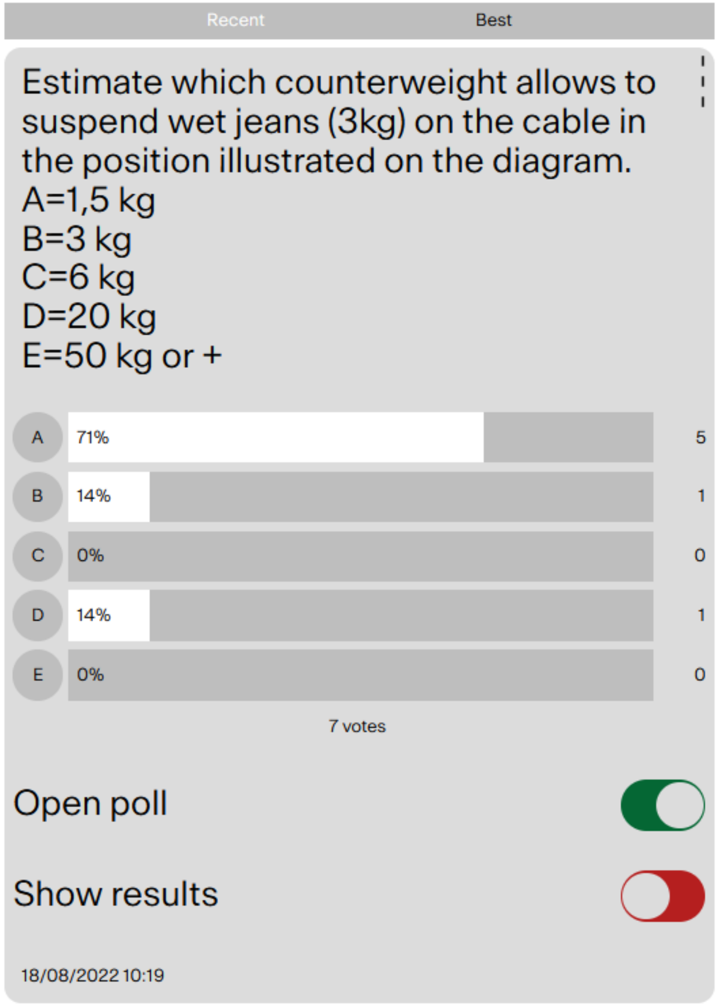
Some months have 31 days; how many have 28?

- ● 0
- ○ 1
- ○ 6
- ○ 12



				Teaching Sciences and Engin...	
First name / Surname		ID number	Email address	MonthsQuestion	Course total
[redacted]		276526	[redacted]	-	-
[redacted]		179803	[redacted]	-	-
[redacted]		356368	[redacted]	-	-
[redacted]		157878	[redacted]	-	-
[redacted]		190108	[redacted]	-	-
[redacted]		355437	[redacted]	-	-

Synchronous activities with SpeakUp



Estimate which counterweight allows to suspend wet jeans (3kg) on the cable in the position illustrated on the diagram.
A=1,5 kg
B=3 kg
C=6 kg
D=20 kg
E=50 kg or +

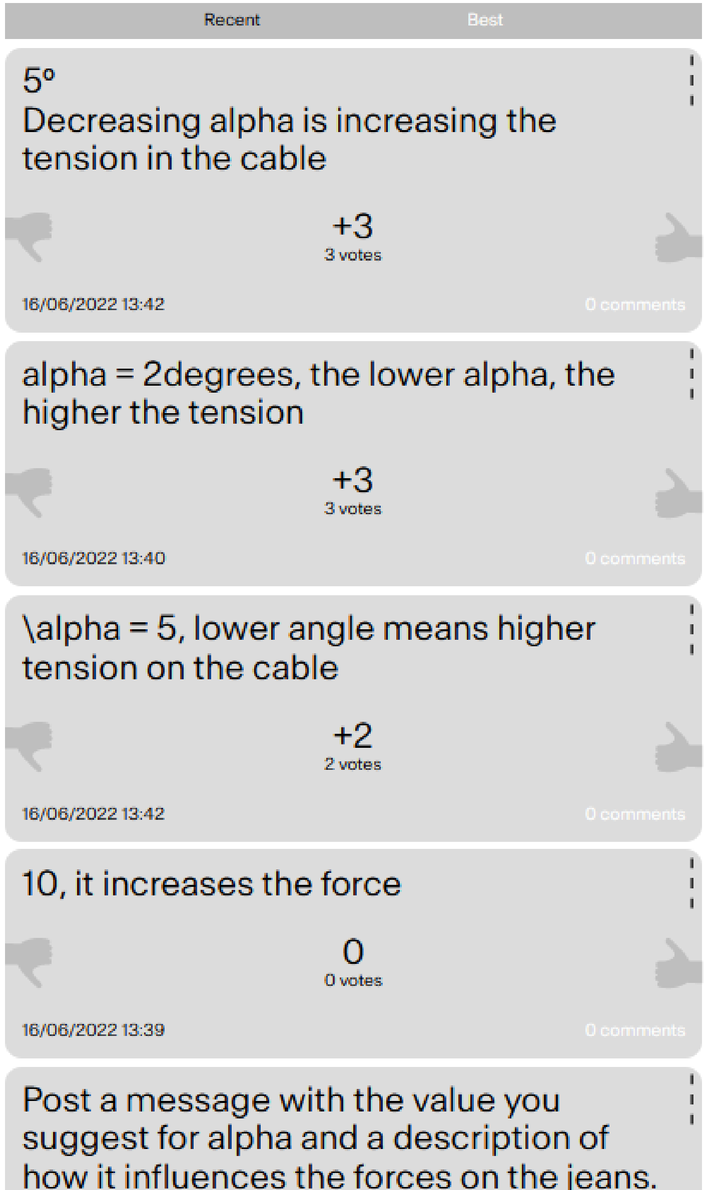
A	71%	5
B	14%	1
C	0%	0
D	14%	1
E	0%	0

7 votes

Open poll

Show results

18/08/2022 10:19



5°
Decreasing alpha is increasing the tension in the cable

+3
3 votes

16/06/2022 13:42
0 comments

alpha = 2degrees, the lower alpha, the higher the tension

+3
3 votes

16/06/2022 13:40
0 comments

\alpha = 5, lower angle means higher tension on the cable

+2
2 votes

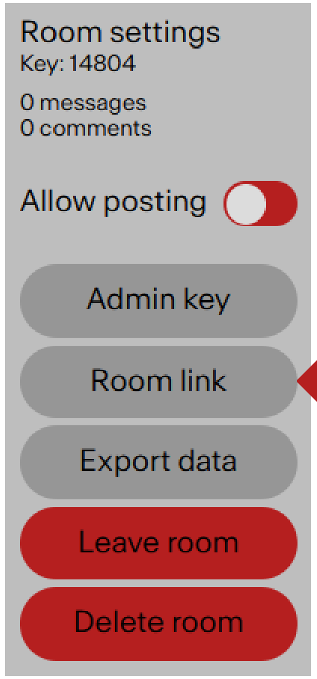
16/06/2022 13:42
0 comments

10, it increases the force

0
0 votes

16/06/2022 13:39
0 comments

Post a message with the value you suggest for alpha and a description of how it influences the forces on the jeans.



Room settings
Key: 14804

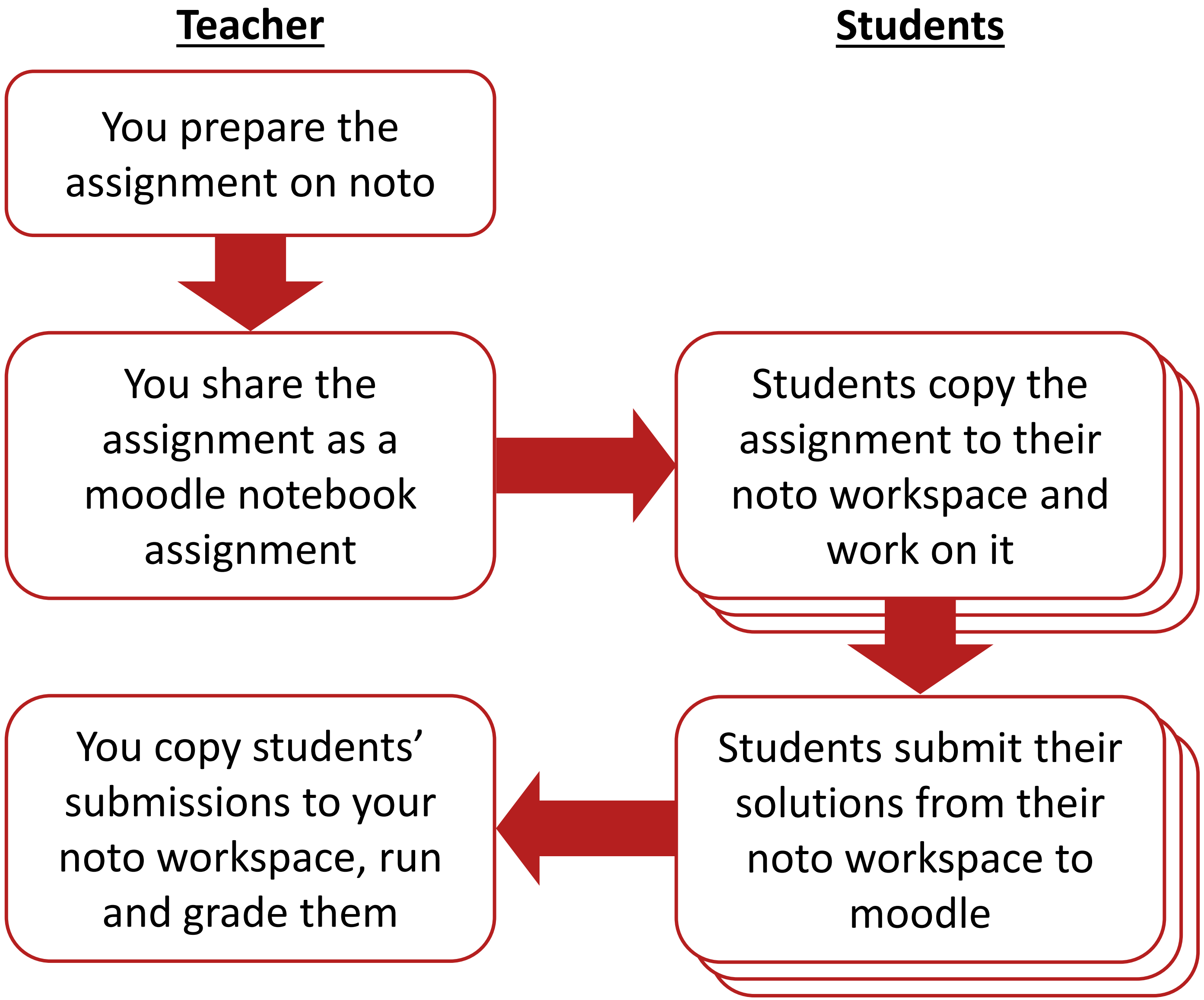
0 messages
0 comments

Allow posting

```
from IPython.display import IFrame
IFrame('https://speakup.epfl.ch/room/join/14804', 640, 360)
```

More details on
https://go.epfl.ch/noto-poll

Notebook assignments with moodle



Teacher

Students

You prepare the assignment on noto

You share the assignment as a moodle notebook assignment

Students copy the assignment to their noto workspace and work on it

Students submit their solutions from their noto workspace to moodle

You copy students' submissions to your noto workspace, run and grade them

More details on https://go.epfl.ch/notebookassignment-teacherdoc

Feedback to students and to you ✍

What type of **feedback mechanisms** could you **integrate into your notebook** so that both students and you get useful information on students’ learning process?
Note down 1 or 2 ideas.

Limit the extraneous load: issues with presentation/format

Designing a notebook for your course / your MOOC

The issue with text

It is very easy to get lost in a notebook, especially when it is long…

- Issues with **attention**
- Issues with **memorization**

How to help:

- Limit the **length** (split into several notebooks)
- Use an **explicit structure**
- Provide an **overview** (e.g. diagram)
- **Highlight** important parts

Markdown / HTML

Formatting text

Create a **highlighted “activity” paragraph** like in the Physics notebook

- Using Markdown:

```
> **Important instruction**: do this.
```

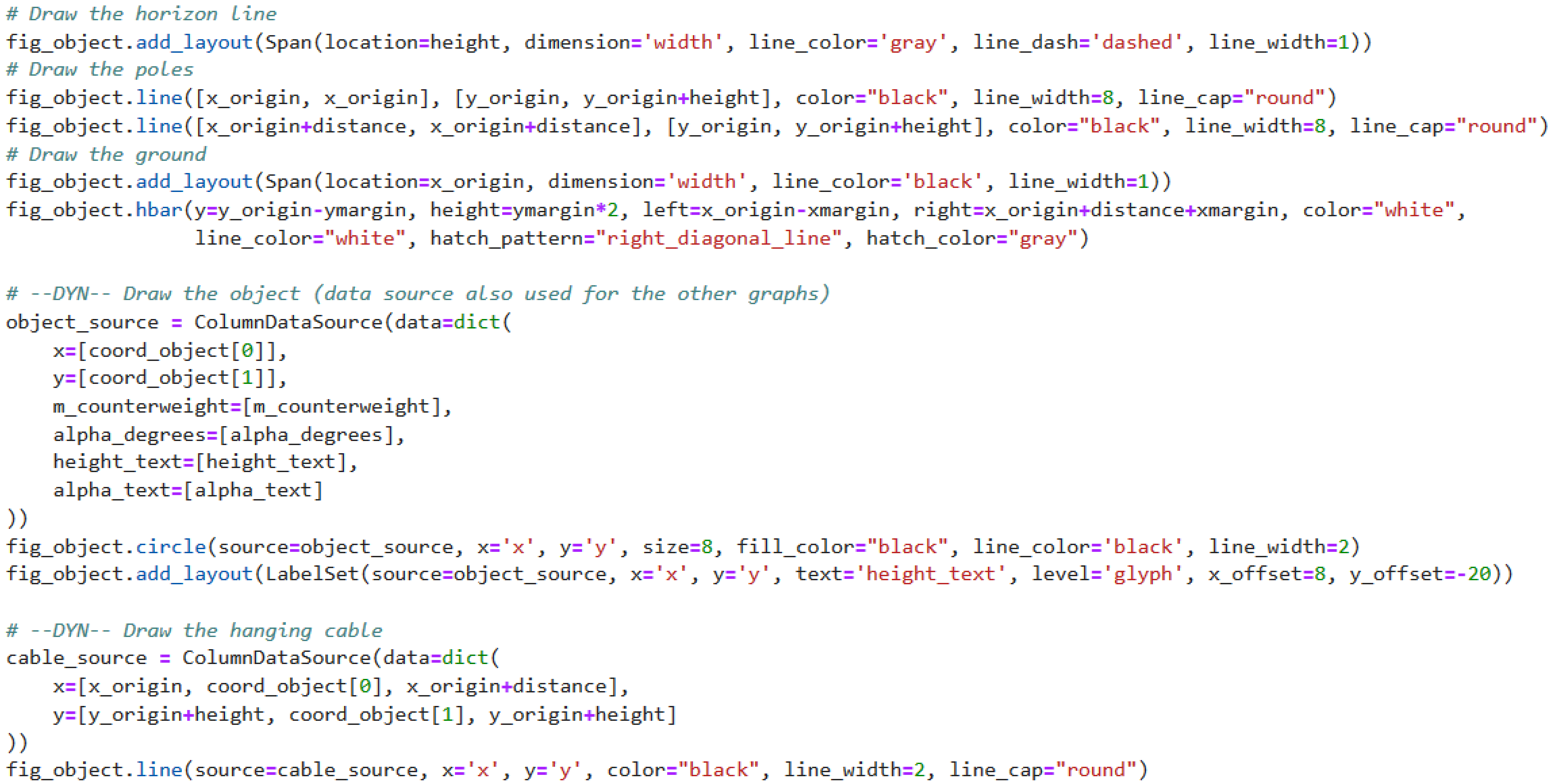
- Using HTML:

```
<div style="padding:8px 0 8px 15px; border-left:3px solid #B51F1F;
background-color:#F3F3F3;">

**Important instruction**: do this.

</div>
```

The issue with code



```
# Draw the horizon line
fig_object.add_layout(Span(location=height, dimension='width', line_color='gray', line_dash='dashed', line_width=1))
# Draw the poles
fig_object.line([x_origin, x_origin], [y_origin, y_origin+height], color="black", line_width=8, line_cap="round")
fig_object.line([x_origin+distance, x_origin+distance], [y_origin, y_origin+height], color="black", line_width=8, line_cap="round")
# Draw the ground
fig_object.add_layout(Span(location=x_origin, dimension='width', line_color='black', line_width=1))
fig_object.hbar(y=y_origin-ymargin, height=ymargin*2, left=x_origin-xmargin, right=x_origin+distance+xmargin, color="white",
                line_color="white", hatch_pattern="right_diagonal_line", hatch_color="gray")

# --DYN-- Draw the object (data source also used for the other graphs)
object_source = ColumnDataSource(data=dict(
    x=[coord_object[0]],
    y=[coord_object[1]],
    m_counterweight=[m_counterweight],
    alpha_degrees=[alpha_degrees],
    height_text=[height_text],
    alpha_text=[alpha_text]
))
fig_object.circle(source=object_source, x='x', y='y', size=8, fill_color="black", line_color='black', line_width=2)
fig_object.add_layout(LabelSet(source=object_source, x='x', y='y', text='height_text', level='glyph', x_offset=8, y_offset=-20))

# --DYN-- Draw the hanging cable
cable_source = ColumnDataSource(data=dict(
    x=[x_origin, coord_object[0], x_origin+distance],
    y=[y_origin+height, coord_object[1], y_origin+height]
))
fig_object.line(source=cable_source, x='x', y='y', color="black", line_width=2, line_cap="round")
```

Limiting unnecessary complexity

- **Hide** the parts of the code that are *not* related to learning goals
- **Structure** and **comment** the code you choose to show

Hiding Python code

General principles:

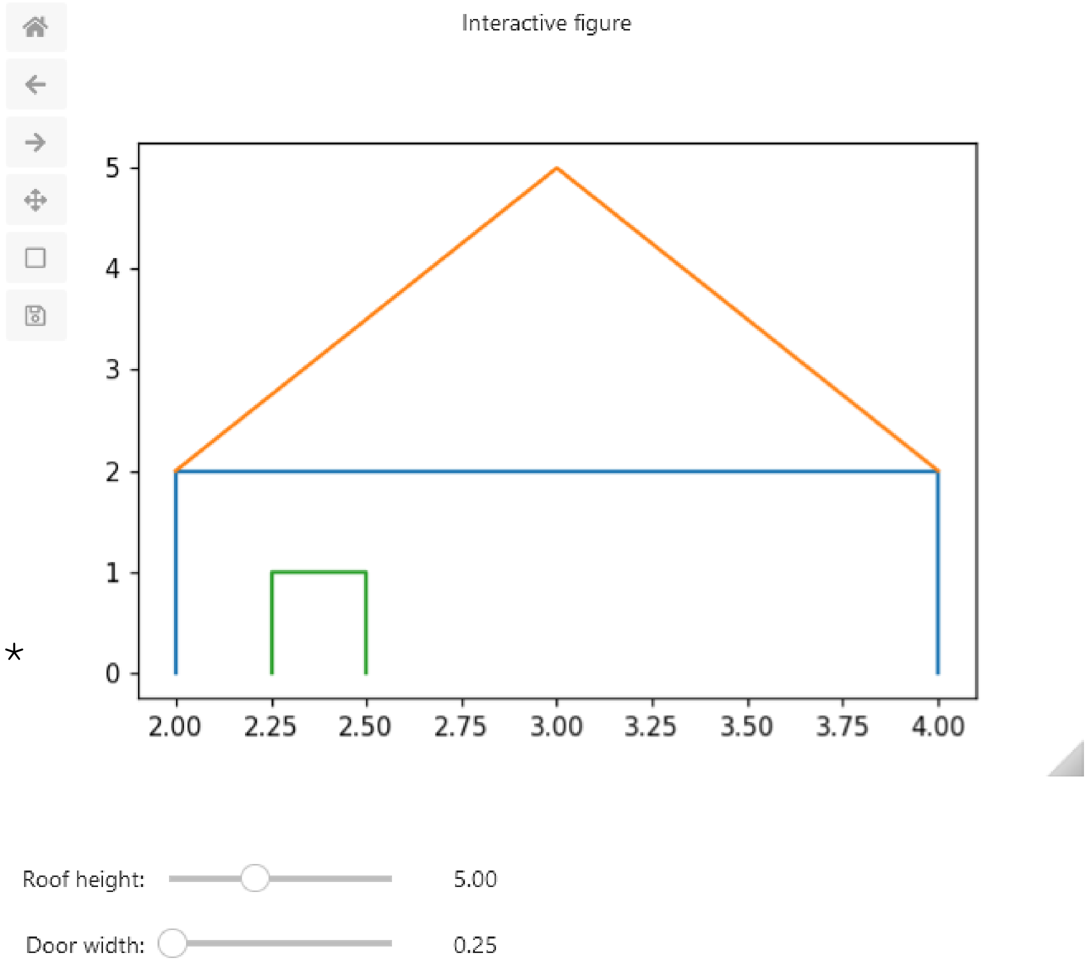
- Structure the code as a function
- Put the code into an external Python file:

  `lib/interactivevisualization.py`

- Import the file

  ```
  from lib.interactivevisualization import *
  ```

- Call the function



Roof height: 5.00

Door width: 0.25

Limiting unnecessary complexity

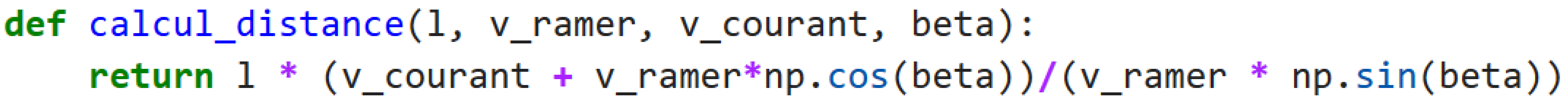
- **Hide** the parts of the code that are *not* related to learning goals
- **Structure** and **comment** the code you choose to show

- **Relate** the code to other representations in the notebook: “pivot” elements

[Ainsworth, 2006]

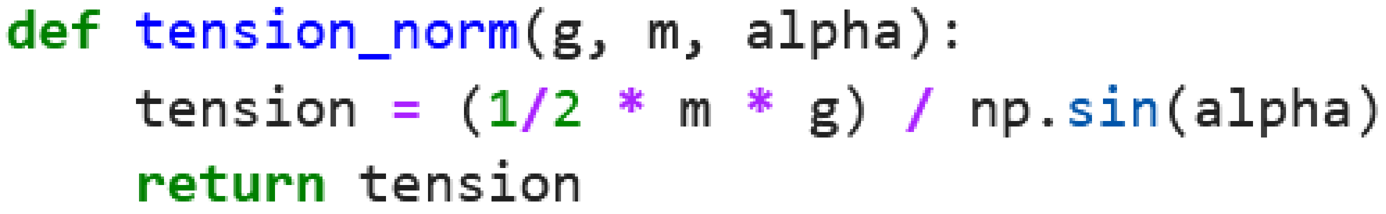
Example: pivot elements

$$\boxed{d = l\frac{v_c + v' \cos\beta}{v' \sin\beta}}$$



```python
def calcul_distance(l, v_ramer, v_courant, beta):
    return l * (v_courant + v_ramer*np.cos(beta))/(v_ramer * np.sin(beta))
```

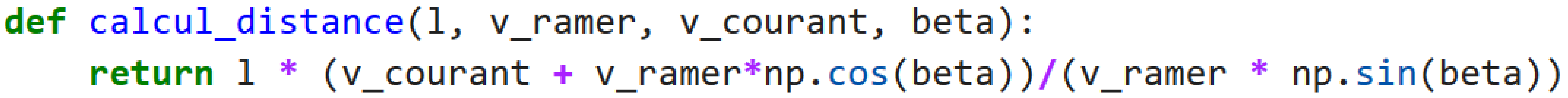
$$|\vec{T}| = \frac{\frac{1}{2}.m.g}{sin(\alpha)}$$



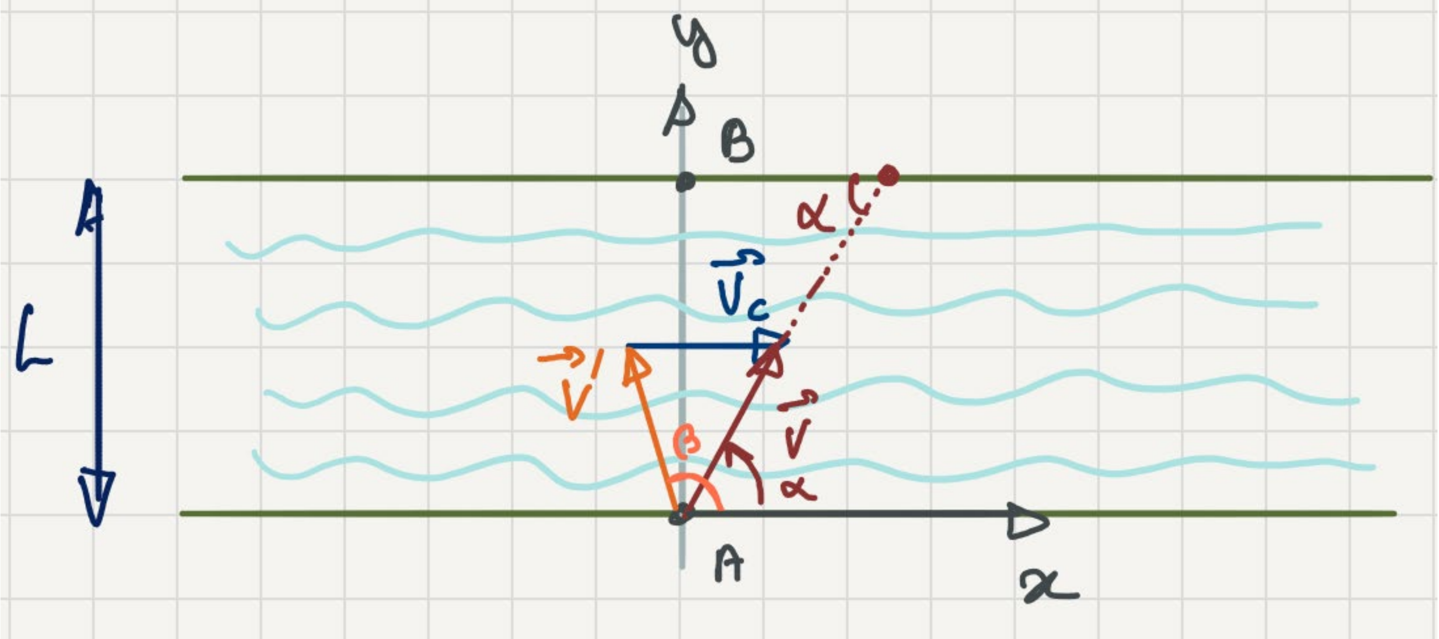
```python
def tension_norm(g, m, alpha):
    tension = (1/2 * m * g) / np.sin(alpha)
    return tension
```

Example: pivot elements

$$\boxed{d = l\frac{v_c + v' \cos\beta}{v' \sin\beta}}$$



```
def calcul_distance(l, v_ramer, v_courant, beta):
    return l * (v_courant + v_ramer*np.cos(beta))/(v_ramer * np.sin(beta))
```



Conclusion

Summary

Write down for yourself:

- **3 things you have learnt** about teaching and learning with Jupyter Notebooks
    -
    -
    -

- **1 thing you would like to experiment** in your notebooks
    -

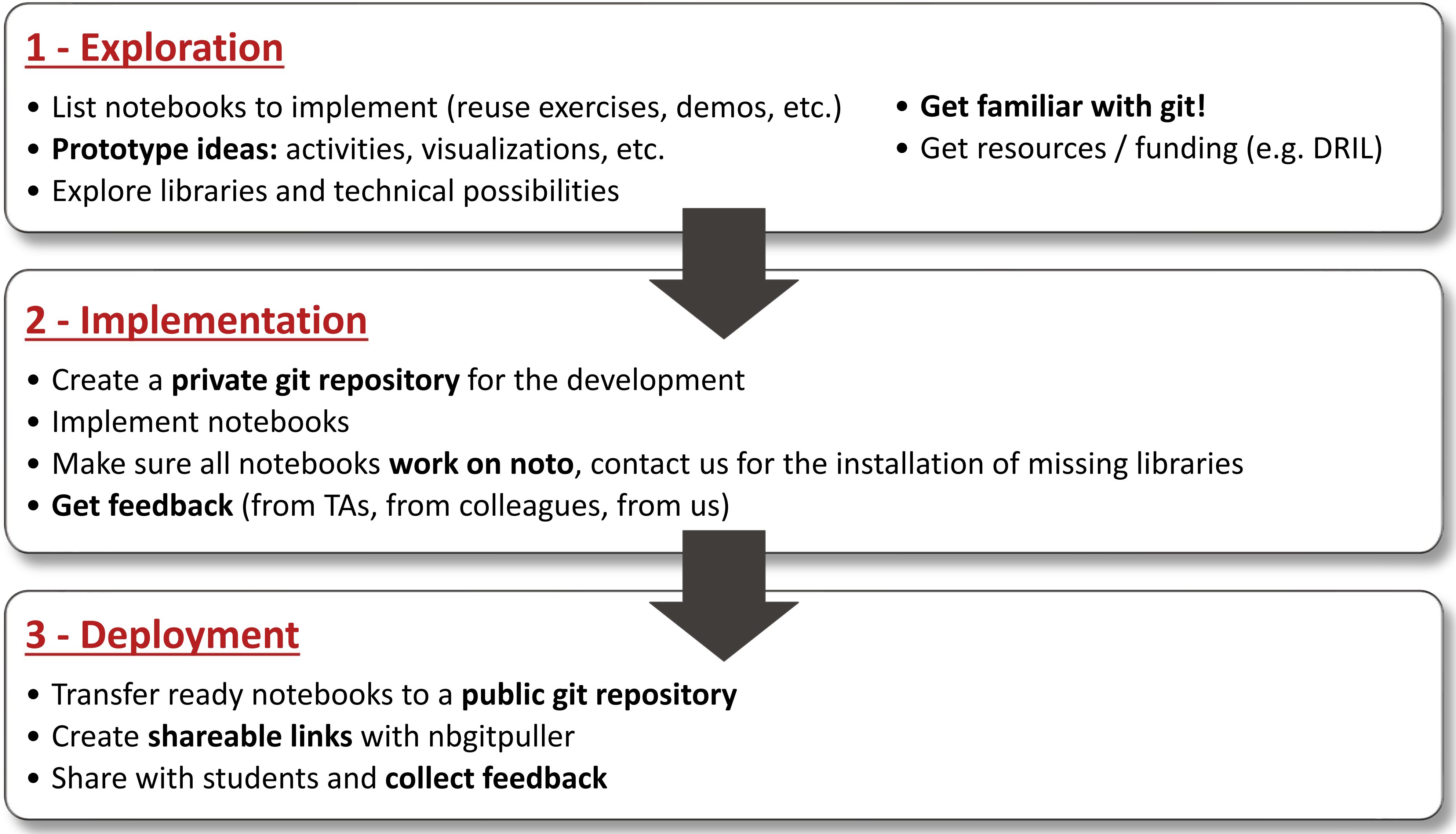
Summary:
3 keys to design effective notebooks

1. Scaffold intrinsic load: importance of **progression**
   - Include preparatory activities before complex tasks
   - Teach problem solving strategies explicitly

2. Optimize the germane load: **effort that pays back**
   - Use reflection questions
   - Provide feedback

3. Limit the extraneous load: issues with **presentation / format**
   - Limit the length of your text, use an explicit structure and highlight important points
   - Hide unnecessary code and relate code to other representations

What to do next?



1 - Exploration

- List notebooks to implement (reuse exercises, demos, etc.)
- **Prototype ideas:** activities, visualizations, etc.
- Explore libraries and technical possibilities
- **Get familiar with git!**
- Get resources / funding (e.g. DRIL)

2 - Implementation

- Create a **private git repository** for the development
- Implement notebooks
- Make sure all notebooks **work on noto**, contact us for the installation of missing libraries
- **Get feedback** (from TAs, from colleagues, from us)

3 - Deployment

- Transfer ready notebooks to a **public git repository**
- Create **shareable links** with nbgitpuller
- Share with students and **collect feedback**

We need your feedback!

Please take 5 minutes to fill out our **anonymous survey** and give us your opinion about this workshop!

https://go.epfl.ch/nbw-survey

Some inspiration to start?

Contact us for a short chat:
noto-support@groupes.epfl.ch

Resources from the workshop folder!

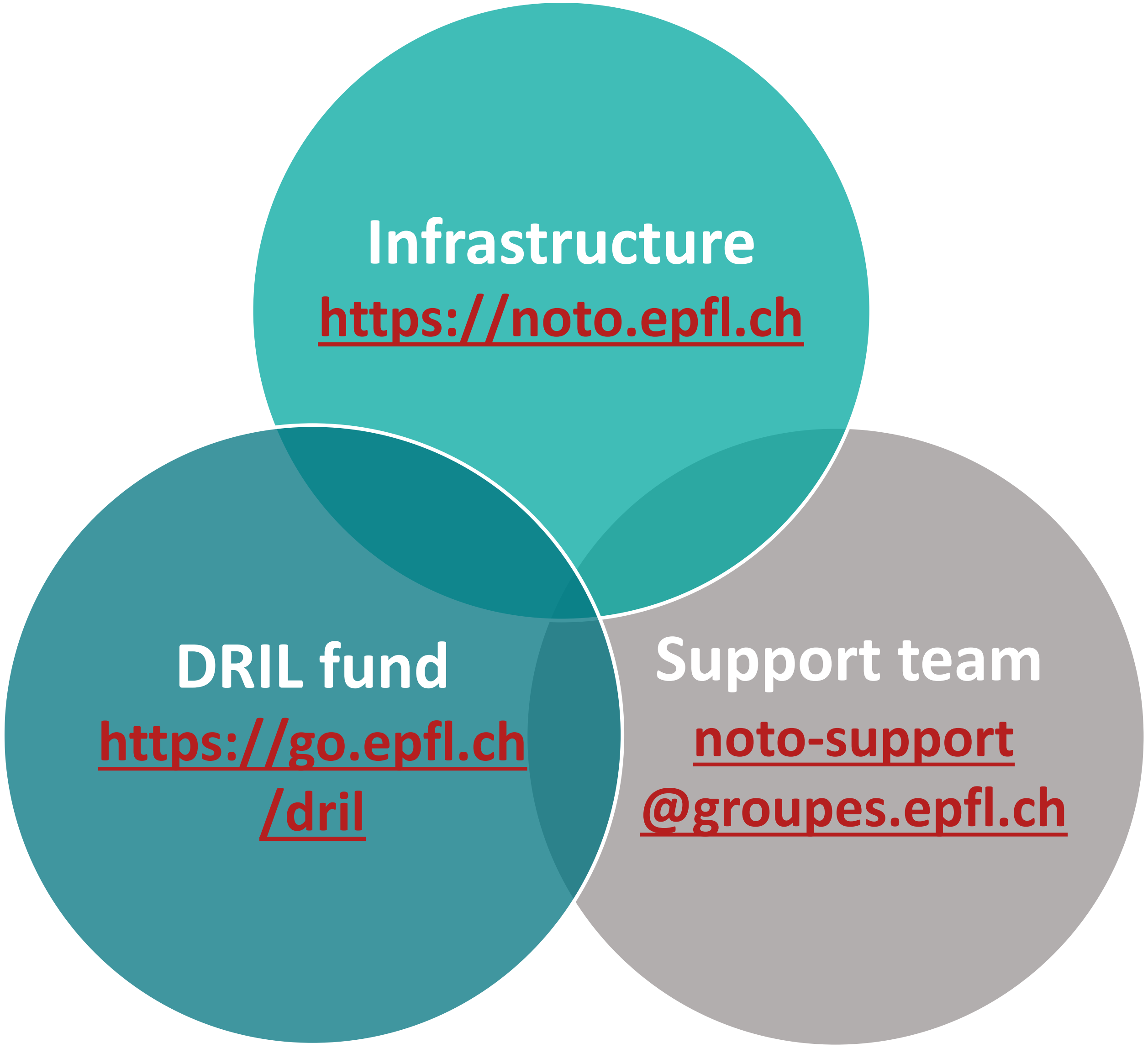
On our Jupyter notebooks for education **website**:

- Interactive textbooks: https://go.epfl.ch/interactivetextbooks
- Exercise worksheets: https://go.epfl.ch/exerciseworksheets
- Notebooks developed by other EPFL teachers: https://go.epfl.ch/notebookexamples

In our **documentation**:

- Sharing notebooks with your students on noto: https:/go.epfl.ch/noto-share
- Embedding SpeakUp polls (or chats) into notebooks: https:/go.epfl.ch/noto-polls
- Collecting student feedback into notebooks using an integrated survey: https://go.epfl.ch/noto-feedback

Support for developing and using notebooks



Infrastructure

https://noto.epfl.ch

DRIL fund

https://go.epfl.ch/dril

Support team

noto-support@groupes.epfl.ch

One website:

https://go.epfl.ch/**notebooks**

Start with Python
/ Have my students start with Python

- “4-Hours Python Quick Start” (basis of Python): https://github.com/hrzn/4hours-python-intro
- “Mathematical Python” (mathematical computing): https://personal.math.ubc.ca/~pwalls/math-python/
- Software Carpentry “Programming with Python” (data analysis): https://swcarpentry.github.io/python-novice-inflammation/
- “EPFL-BIO-210” (data analysis, machine learning): https://github.com/amathislab/EPFL-BIO-210

- Software Carpentry “Version control with git”: https://swcarpentry.github.io/git-novice/