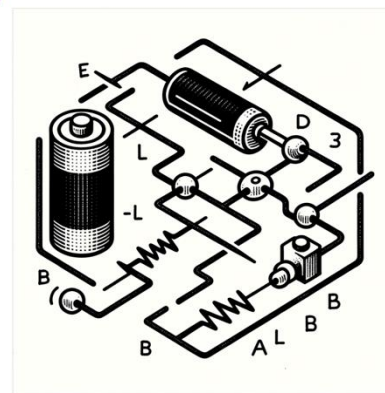


# Teaching Digital Topics: A Practical Report with Dos and Don'ts

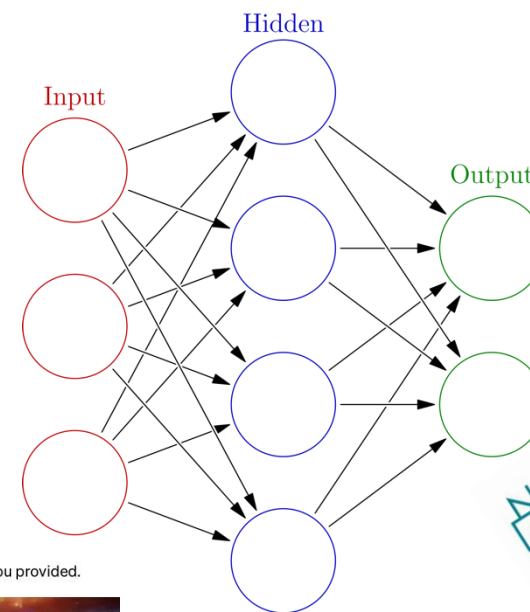


ErUM Data Hub  
Deep Learning Train-the-Trainer  
September 19, 2025

ChatGPT



Here's a redrawn, cleaner version of the electric circuit schematic you provided.



perplexity

ALEX  BRITZ  
FACILITATING SCIENCE



# Disclaimer



These slides primarily reflect my experiences and opinions from about five years of teaching digital topics.

What I've compiled here is likely incomplete and includes many personal perspectives, so please view this as a trigger for personal reflection rather than strict guidance.

If you have any doubts or questions, please contact me under [mail@alexbritz.de](mailto:mail@alexbritz.de).



*Let's Play!*  
**Myth or Fact**

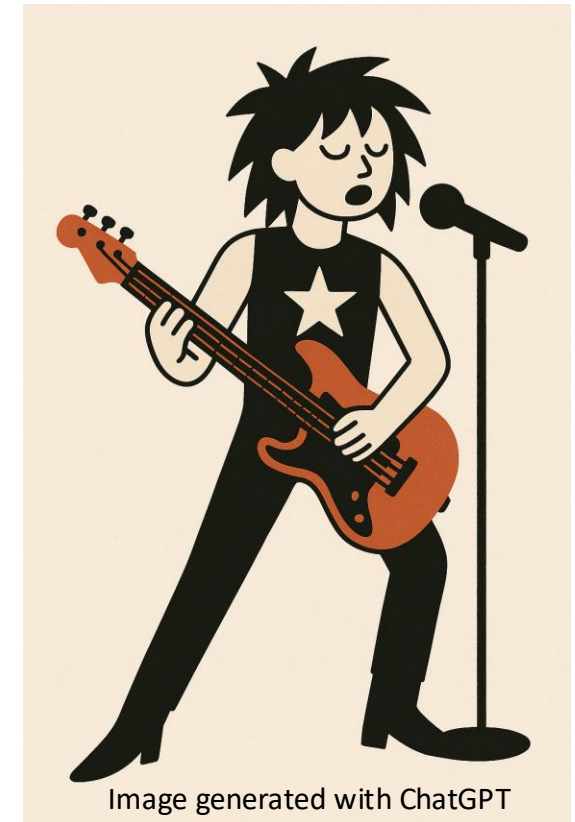
# Myth or Fact I: It Requires 10000 Hours to Reach Excellence



MYTH!

The psychologist Anders Ericsson published this number in 1993 about music students in Berlin.

But: You can't reach excellence just by practicing brute force for 10000 hours. Other factors are also important, for example, goal-oriented practice.



# Myth or Fact II: **Videos are the Most Effective Way to Learn.**



## MYTH!

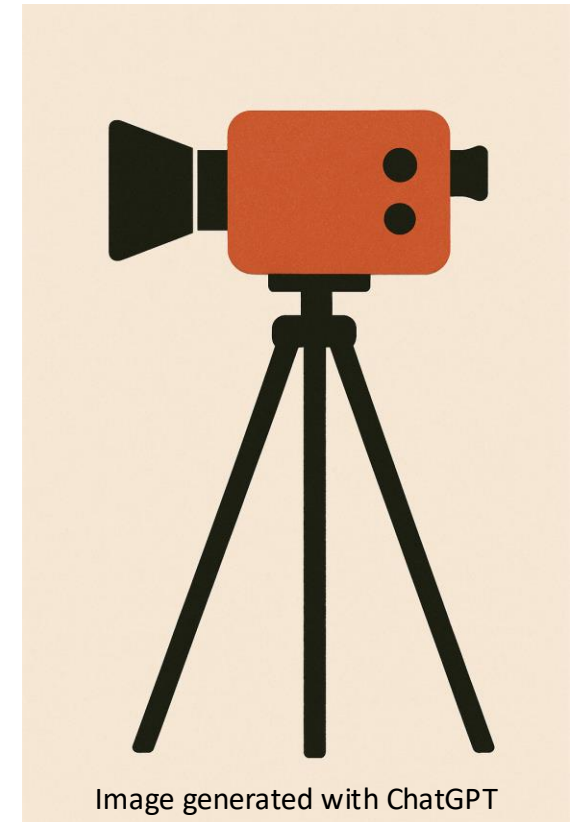
Pure video courses and long learning videos have a large abortion rate.

Consumption is passive.

Content often lacks fit and is inflexible.

Lack of interaction.

Distraction occurs due to multitasking.



# Myth or Fact III: Digital Natives Navigate The Digital Sphere Naturally and Easily.



MYTH!

Digital literacy  $\neq$  Digital familiarity.

We need to distinguish usage and competence – there's no clear correlation.

Digital skills need to be developed and are not inherited.





*Who am I?*

**And what am I going to talk about.**

# Introduction of Myself



- By education: Experimental Physicist
- Research Career: Physical Chemistry and Materials Science at Synchrotrons, XFELs, and UED.
- Now: Trainer and Facilitator
- Taught about  
**350+** PhD candidates, postdocs and group leaders in Python programming and  
**150+** in Generative AI use in science and academia  
(and many more in communication skills)





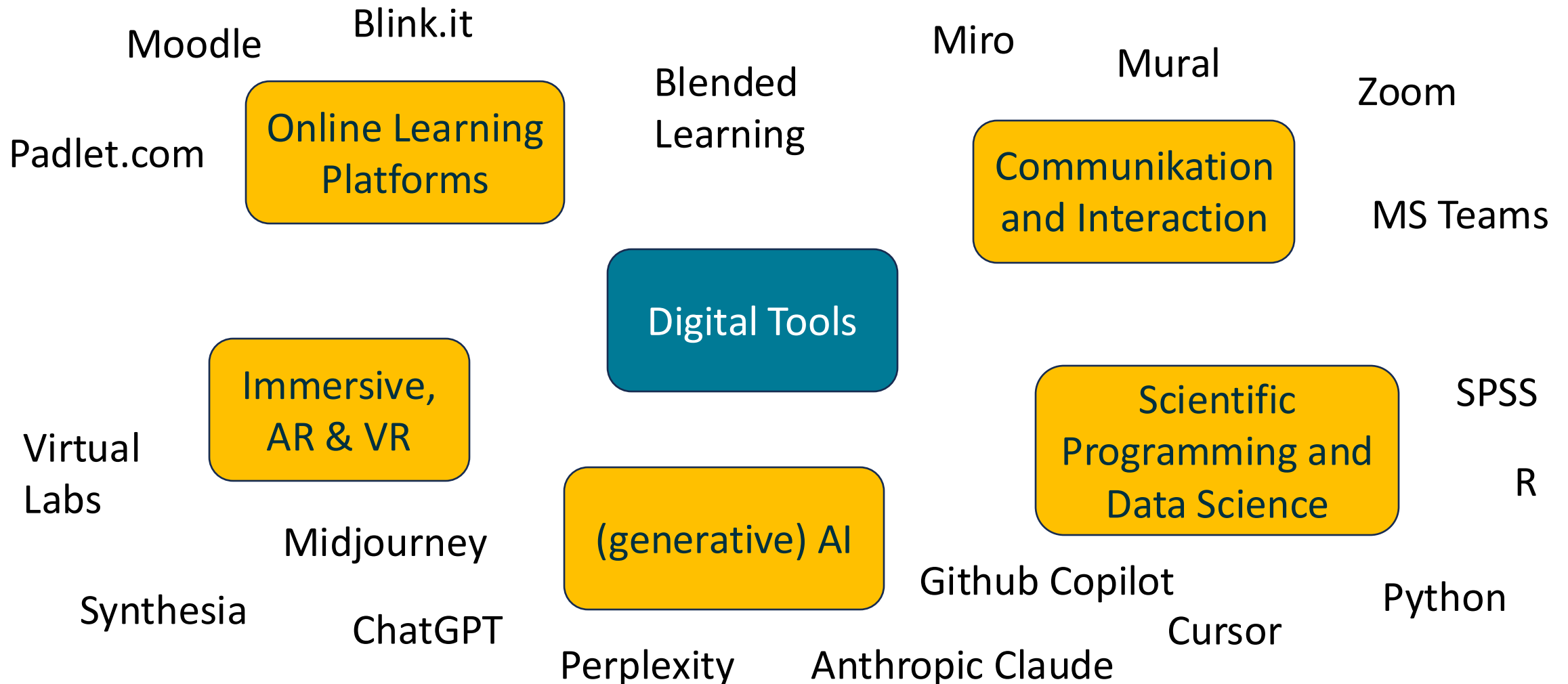
# What Are Typical Digital Topics?

## Examples of Courses Which I Teach & Taught



- Generative AI in Science and Academia
- Python Programming
- Data Analysis and Visualization
- Machine Learning with Python
- But also: Project Management
- VLAB – a virtual XFEL Laboratory (lead by C. Bressler at Uni HH)

# Digital Learning and Teaching (in Natural Sciences)



# Teaching Digital Topics: 10 Practical Advices



1. **Adapt to different levels and personalities** → If possible, work with small groups (max 15 students).
2. **Assume no prior knowledge and challenge in baby steps** → Start with basics, accelerate challenges incrementally, finish with advanced concepts. Include optional extensions (bonus challenges).
3. **Avoid distractions and promote commitment** → Lead by example and be committed yourself. Find creative ways to avoid distractions and enhance commitment.
4. **Don't assume motivation** → Make relevance explicit (e.g., *"this method is used in real physics research"*).
5. **Balance practice and theory (~50/50)** → Live coding, teamwork, and projects are as important as lectures.
6. **Distinguish familiarity from literacy** → Not just *"can use"*, but *"understands what it does and why"*.
7. **Promote collaboration** → Teamwork, pair programming, and peer review mirror real scientific practice.
8. **Encourage responsible tool use** → Rely on few, well-understood tools and techniques. Prefer basic over complex tools, promote critical awareness when using GenAI tools.
9. **Assure a fit of the training / workshop content** → Use pre-surveys to calibrate. Consider saying "no".
10. **Prioritize reflection, transfer & application** → Ask *"why does this matter?"*, work on projects with real datasets, emphasize reproducible workflows, and plan for application beyond the teaching format.



# **1. Adapt to Different Skills/Knowledge Levels and Personalities**

# Python Pre-Questionnaire

When you submit this form, it will not automatically collect your details like name and email address unless you provide it yourself.

1. What's your name (optional)?

2. And your email address (optional)?

3. What's your research topic (optional)?

4. Please describe your programming knowledge. Which programming languages do you already know? And to which level (beginner, intermediate, expert)?

5. Did you already use Python in the past? If so, please name packages that you typically use and feel comfortable with.

6. Please describe what you would like to learn in this course!

7. Please describe how you would like to use Python in the future!

8. Do you have data from experiments or simulations which you could work with during this course? If so, please describe the data in a few words (type of data, file formats, size, etc.)

9. What should NOT happen during the course?

10. Please fast forward in time. What should happen in this course in order for it to be successful to you?

# Challenges with Self-Assessment and Prior Knowledge



Scenario 1: A student signs up for the course „Machine Learning with Python” but has never written any Python code before.

Scenario 2: A student in the “Fundamentals of Python” course has already participated in a similar program half a year ago. They believe they didn’t learn anything but can complete all exercises quickly (and even help me teach the class).



## **2. Don't Count on Prior Knowledge and Challenge in Baby Steps**

# I Believe I Can Learn It!

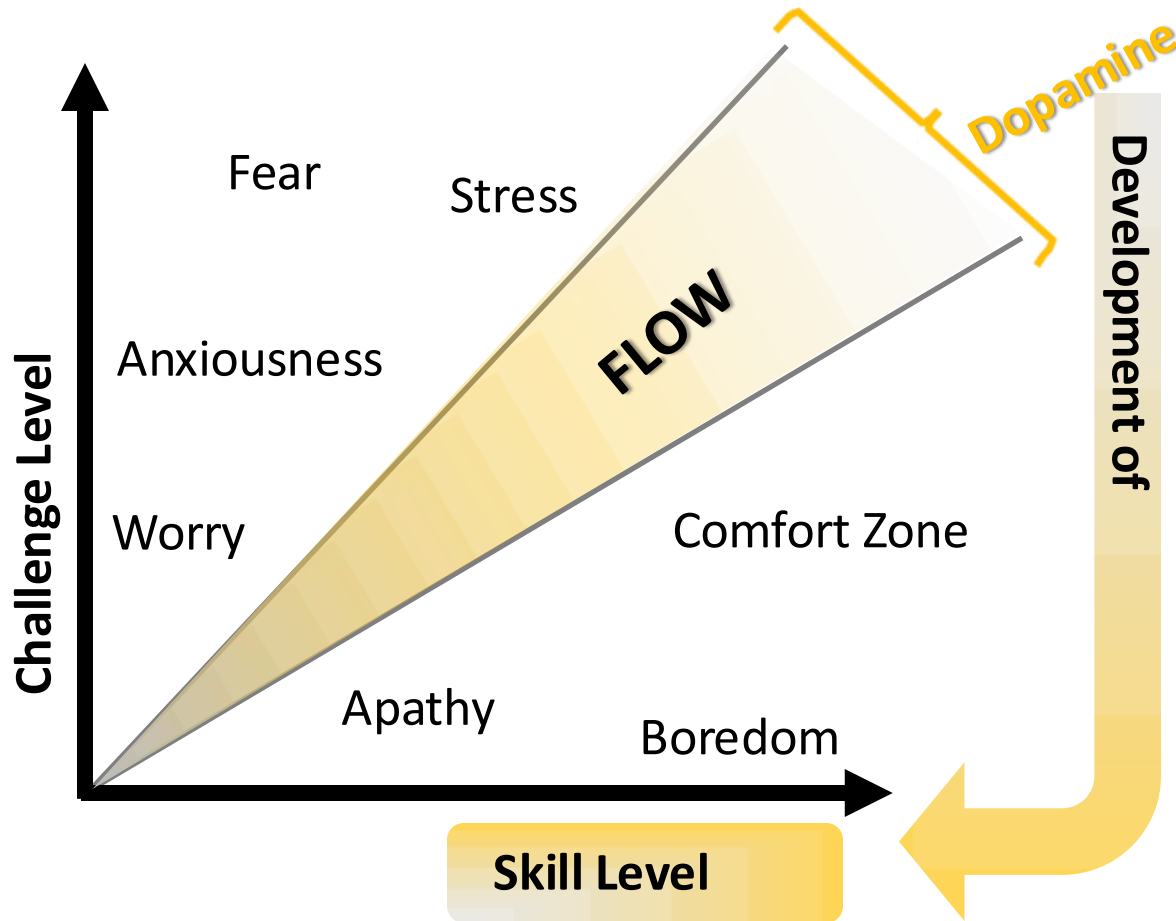


The student says: „*I believe I can do It!*“

- Self-efficacy plays a major role in learning: If I can do it, I will make it.
- How can we enhance self-efficacy:
  - Find role models.
  - Celebrate small successes.
- How can we implement this? Take baby steps and challenge incrementally – one step at a time.



# Enhance „Flow“ (Montessori & Csíkszentmihályi)

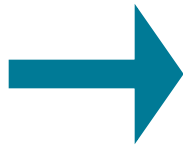


According to Steven Kottler the Challenges should be 4% above the Skills  
<https://www.flowresearchcollective.com/blog/flow-trigger-the-challenge-skill-ratio>

# Baby Steps in a Deep Learning Exercise



Build and train a neural network with PyTorch to classify handwritten digits from the MNIST dataset.



1. Set up a fresh Python environment and install torch, torchvision, and matplotlib.
2. Import PyTorch/torchvision; configure device detection (cuda if available) and set random seeds.
3. Download and load MNIST with torchvision.datasets.MNIST, applying ToTensor() and normalization; create train/test DataLoaders.
4. Visualize a small batch (e.g., 25 images) with labels to sanity-check the input pipeline.
5. Define a simple model (start with a CNN: Conv→ReLU→Pool→Conv→ReLU→Pool→FC→Dropout→FC); print a parameter count summary.
6. Choose CrossEntropyLoss and an optimizer (e.g., Adam), set reasonable hyperparameters (batch size, learning rate, epochs).
7. Implement the training loop: forward pass, compute loss, backprop, optimizer step, and track running loss/accuracy per epoch.
8. Implement an evaluation loop on the test set (no grad), reporting overall accuracy and a confusion matrix.
9. Run at least two experiments (e.g., change learning rate or add BatchNorm/Dropout) and compare metrics; plot loss/accuracy curves.
10. Save the best model with torch.save, reload it to verify inference on a few samples, and write a short paragraph summarizing results and next steps.



# **3. Avoid Distractions and Promote Commitment**

# Challenges with Blended Learning



We once designed a blended learning format about data analysis with python for PhD candidates in physics. It was a flipped classroom format with asynchronous video lectures, and we filmed a total of 10 hours of video material, ranging from fundamentals to advanced and expert levels. Although participation in the live session was great and the feedback positive, the video material was not as effective as expected. Of the 20 videos produced, students watched only the initial/fundamental ones; few students watched the later ones.

I have had similar experiences with online content of communication skills workshops.

→ Blended learning is challenging. Learning from asynchronous content requires dedication and commitment.

# Distractions due to Emails



Instructor („Me“): „This is how you can do xyz...”

Student (looks up from their screen. „Can we also do xyz? That would be really interesting to know...”

Distractions, even if happening in a few seconds, kill our concentration.

# Success Story: Forcing Commitment.



This year, I designed a new asynchronous Python Kickstarter course (with a total workload of about 8 hours).

I offered this course for free in exchange for

- a) Feedback after completion and
- b) Weekly updates from the students about their progress.

The course's completion rate was about 50%, which is extremely high for a fully asynchronous format.



## **4. Don't Assume Motivation**

# A Motivated Participant?!



„Machine Learning is a great skill to have on my CV. I saw this in the course catalogue of the university and signed up. It's free. Cool, right?“

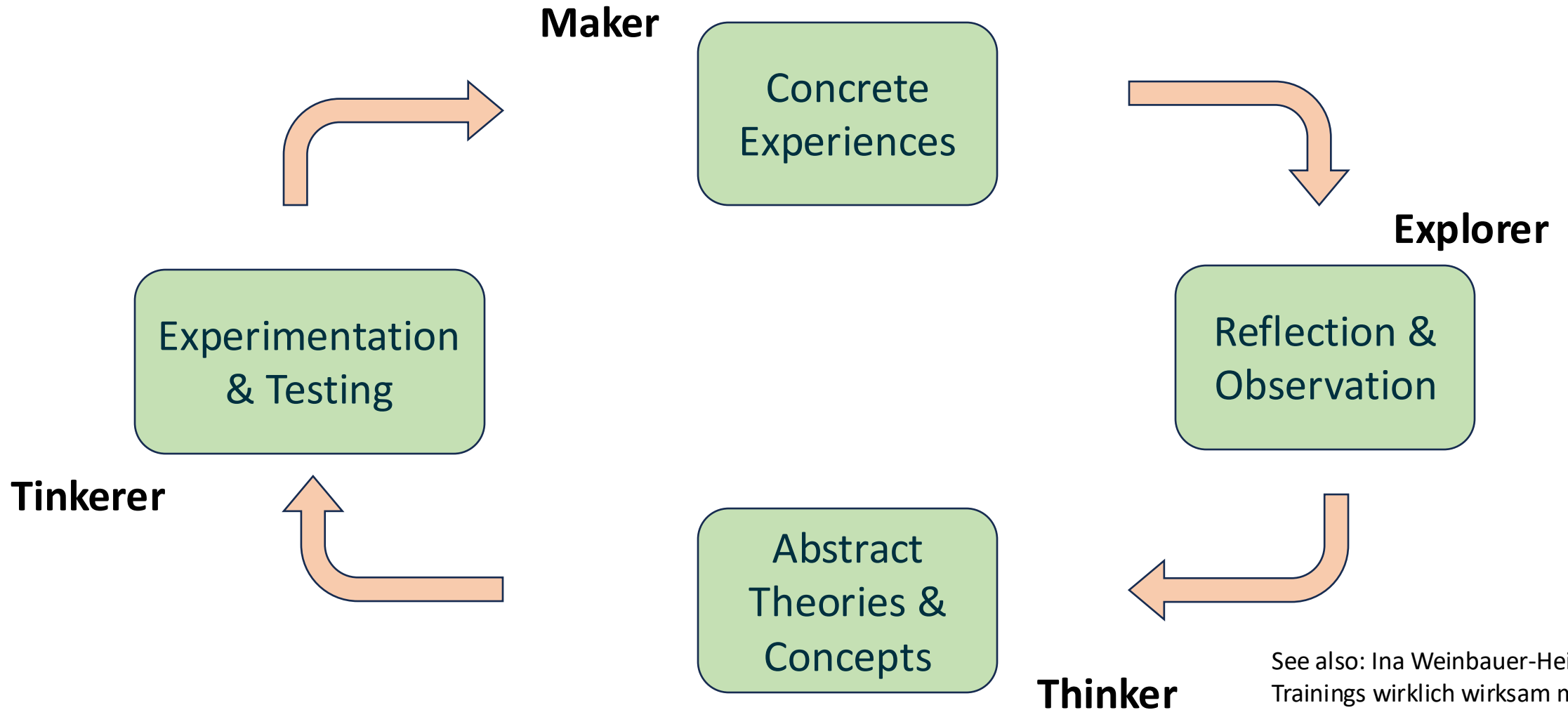
*Participant in the intro round.*





## **5. Balance Theory and Practice**

# Learning by Doing



See also: Ina Weinbauer-Heidel „Was Trainings wirklich wirksam macht: 12 Stellhebel der Transferwirksamkeit“

# Balance between Theory Input and Exercise / Application



Active Practice should make up  
**30% – 50%**  
of any Training or Workshop



## **6. Don't Mistake Familiarity with Literacy**

# Familiarity or literacy?



Instructor: „How often do you use AI tools at work?“

Student: „Daily. It’s fully incorporated into my workflows“

Instructor: „Amazing. What is your formula to prompt effectively?“

Student: „No formula, I just ask questions. Sometimes ChatGPT makes up citations. I don’t know why...”



## **7. Promote Collaboration**

# Promote Collaboration and Team Work



Because it's fun.

And effective.



## **8. Responsible Tool Use**



# Battle of the Tools



THE ULTIMATE AI TOOL CHEAT SHEET					
LLM / TEXT	IMAGES	VIDEOS	AUDIO		
ChatGPT	Midjourney	Runway	Murf	Eleven Labs	
Claude	DALL-E 3	Pictory	Other	Speechify	
Bing Chat	Leonardo	Veed			
Google Bard	Firefly	HeyGen			
CHATBOT	MARKETING	PRESENTATIONS	DESIGN		
Dante AI	AdCreative	Tome	Canva	Adobe Firefly	
	Claude	Simplified	Designs.ai	Makker	
	Jasper	Beautiful.ai			
	SEMrush	Synthesia			
RESEARCH	COPYWRITING				
HARPA					
Perplexity					
Glassp					
ChatGPT Plus					

**Aditya Kachave** • 3rd+  
Exploring AI | Be10X | IIT Kharagpur  
1d • ...more

Tested 100+ AI Productivity tools so you don't burn months figuring it out

+ Follow

### AI PRODUCTIVITY TOOLS

- AI CHATBOTS**
  - ChatGPT
  - Claude
  - DeepSeek
  - Gemini
  - Grok
  - Meta AI
  - MS Copilot
  - Perplexity
- AI PRESENTATION**
  - Beautiful AI
  - Gamma
  - Pitch
  - Plus
  - PopAI
  - Presentation AI
  - Slidesgo
  - Tome
- AI CODING ASSISTANCE**
  - AskCodi
  - Codiga
  - Cursor
  - GitHub Copilot
  - Qodo
  - Replit
  - Tabnine
- AI EMAIL ASSISTANCE**
  - Clipit AI
  - Friday
  - Mailmaestro
  - Shortwave
  - Superhuman
- AI MEETING NOTES**
  - Avoma
  - Equal Time
  - Fathom
  - Fellow App
  - Fireflies
  - Krisp
  - Otter
- AI SPREADSHEET**
  - Bricks
  - Formula Bot
  - Gigasheet
  - Rows AI
  - Sheet AI
- AI IMAGE GENERATION**
  - Adobe Firefly
  - DALL-E
  - FLUX.1
  - Ideogram
  - Midjourney
  - Recreate
  - Stable Diffusion
- AI SCHEDULING**
  - Calendly
  - Clockwise
  - Motion
  - Reclaim AI
  - Taskade
  - Trevo AI
- AI KNOWLEDGE MANAGEMENT**
  - Mem
  - Notion
  - Tettra
- AI WORKFLOW AUTOMATION**
  - Integrately
  - Make
  - Monday.com
  - N8n
  - Wrike
  - Zapier
- AI GRAPHIC DESIGN**
  - AutoDraw
  - Canva
  - Design.com
  - Framer
  - Microsoft Des.
  - Uizard
- AI VIDEO GENERATION**
  - Descript
  - Haiper AI
  - Invideo AI
  - Kling
  - Krea AI
  - Luma AI
  - Pika AI
  - Runway
  - Sora
- AI DATA VISUALIZATION**
  - Deckplot
  - Flourish
  - Julius
  - Visme
  - Zing Data

**Maria Gharib** • 3rd+  
The AI Copy Girl  
3d • ...more

99% of entrepreneurs are sleeping on these AI tools.  
You're losing hours every day. ...more

+ Follow

### 55+ AI TOOLS FOR ENTREPRENEURS

- BRAINSTORMING**
  - ChatGPT 5
  - Claude
  - Perplexity
  - Poe
  - Spencer AI
- LEARNING**
  - Google NotebookLM
  - Whimsical
  - Miro AI
  - Eightify
  - QuizGecko
  - Langotalk
  - Glassp
  - Stimuler
- PRESENTATIONS**
  - Gamma
  - Decktopus
  - SlidesAI
  - Pop AI
  - Presentations AI
  - Prezi
  - b. Beautiful AI
  - AI PPT
- ANALYSIS**
  - Rows
  - Julius
  - Quadratic
  - Numerous AI
  - Formula Bot
  - ChatCSV
  - MySheet AI
- MEETINGS**
  - Itadv
  - Otter AI
  - Krisp
  - Fireflies
  - Notion
- RESEARCH**
  - ChatGPT Search
  - Perplexity
  - ChatGPT Deep Research
  - Gemini Deep Research
  - Silatus
  - Scispace
  - Consensus
- DIGITAL WORKSPACE PRODUCTIVITY MANAGEMENT**
  - Notion
  - Clickup
  - Monday
  - Asana
  - Trello
  - Taskade
  - BeforeSunset AI

Subscribe to Mindstream to learn AI for free -> [www.mindstream.news](http://www.mindstream.news)

**AI Apps Central**  
5,218 followers  
1w • ...more

100+ AI Tools to Boost Your Productivity

This is the ultimate resource for anyone looking to save hours of ...more

+ Follow

### 100+ AI TOOLS TO SAVE HOURS OF WORK IN JUST MINUTES

- Ideas**
  - Copilot
  - You
  - Perplexity
  - ChatGPT
  - Claude
- Presentation**
  - Prezi
  - Pitch
  - PopAI Pro
  - Slides AI
  - SlidesAI
  - YOW
- Website**
  - Dora
  - Durable
  - Wegic
  - Framer
  - 10Web
- Writing**
  - FastRead
  - Jasper
  - Copy AI
  - TextBlaze
  - Writesonic
- AI Model**
  - RenderNet AI
  - Glance AI
  - AOB
  - Deepmode
  - AI Herald
- Image**
  - Dora
  - Freepik
  - Phygital
  - Stocking AI
  - Image Create
- Twitter**
  - Metricool
  - PostHive
  - Tribescale
  - TweetFunnel
  - Typicaly
- Web3**
  - Ava
  - Althea
  - Aditi AI
  - Spice AI
  - LNK
- Voice**
  - Udio
  - Suno AI
  - VEEDIO
  - Speechify
  - Eleven Labs
- AI Detector**
  - GPTZero
  - Wordline
  - Copyscape
  - Bypass GPT
  - Grammarly
- Speech To Text**
  - Fluent AI
  - Cockatoo
  - Whisper AI
  - Adapt AI
  - Speechful
- Code Generator**
  - Codex AI
  - Continue
  - VO Dev
  - Codewarp
  - Refraction
- Automation**
  - Phantom
  - Outreach
  - ClickUp
  - Drift
  - Empell
- Video**
  - Syllaby
  - HeyGen
  - Nutface AI
  - Decipher
  - Synthesia
- Design**
  - Looka
  - Clipdrop
  - Autodraw
  - Vance AI
  - Design AI
- Chatbot**
  - Poe
  - Claude
  - Gemini
  - ChatGPT
  - HuggingChat
- Blog Writer**
  - Katib
  - Research
  - Elephas
  - Junia AI
  - Journalist AI
- Meeting**
  - Thiv
  - Krisp
  - Otter
  - Avoma
  - Fireflies

**Gokul Kanagaraj** • 3rd+  
Aspiring Cybersecurity Analyst | SOC Analyst | TryH...  
1w • ...more

100 AI Tools You Can't Ignore in 2025

From coding copilots to faceless video creators, the AI landscape is exploding with innovation. Whether you're building, automating, ...more

+ Follow

### 100 AI Tools You Can't Ignore in 2025

- ChatGPT.com – solves anything
- Claude.ai – conversational assistant
- Perplexity.ai – research with sources
- Midjourney.com – generate breathtaking art
- Runwayml.com – AI video editing
- Replit.com – write and run code
- Cursor.so – AI coding IDE
- Blackbox.ai – AI for developers
- Tabnine.com – coding autocomplete
- Manus.im – build apps with text
- Canva.com – design anything with AI
- Gamma.app – AI-powered presentations
- Tome.app – instant slide decks

# Should I Teach Seaborn or Matplotlib?



- Low-level: Full control for custom plots.
- Requires more code for complex visuals.
- Extremely flexible and customizable.



- High-level: Quick creation of statistical and aesthetic plots.
- Simplified syntax, efficient for complex plots.
- Limited custom options, focused on good defaults.

My Answer: Teach Matplotlib – because it's the foundation of Seaborn. If you know Matplotlib, learning Seaborn is trivial. The other way around difficult.

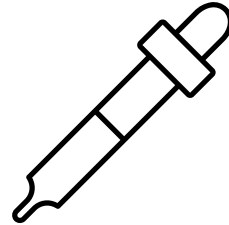
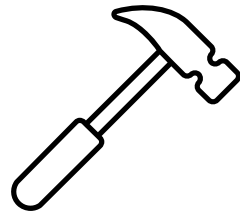


## **9. Does the Training fit the Participant's Needs?**

# If I have a hammer, everything looks like a nail

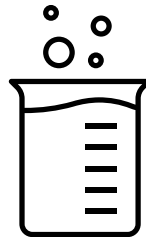
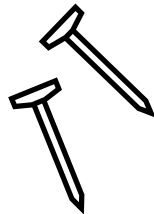


Solution  
(=Tool)



But what is the  
implication for teaching  
Deep Learning?

Problem



→ Teach also other  
methods (e.g. statistical  
analysis, classification and  
regression)

# Examples of Poor Fit of Training



Life scientists participate in „Fundamentals of Python“ course, although in their field and their environment everyone works with R.

Participants learn Deep Learning (PyTorch), but not simple classification and regression algorithms (e.g. with Scikit-Learn) – now they treat every problem with the most complex solution (the sledge hammer).

# Python Pre-Questionnaire

When you submit this form, it will not automatically collect your details like name and email address unless you provide it yourself.

1. What's your name (optional)?

2. And your email address (optional)?

3. What's your research topic (optional)?

4. Please describe your programming knowledge. Which programming languages do you already know? And to which level (beginner, intermediate, expert)?

5. Did you already use Python in the past? If so, please name packages that you typically use and feel comfortable with.

6. Please describe what you would like to learn in this course!

7. Please describe how you would like to use Python in the future!

8. Do you have data from experiments or simulations which you could work with during this course? If so, please describe the data in a few words (type of data, file formats, size, etc.)

9. What should NOT happen during the course?

10. Please fast forward in time. What should happen in this course in order for it to be successful to you?



*What Makes Learning Effective?*

## **10. Prioritize Transfer**



## Evaluation with Smiley-Sheets

→ These only assess the perception/reaction of the students.

**Please reflect upon each statement to which extent it applies to the workshop:**

The workshop met my expectations.

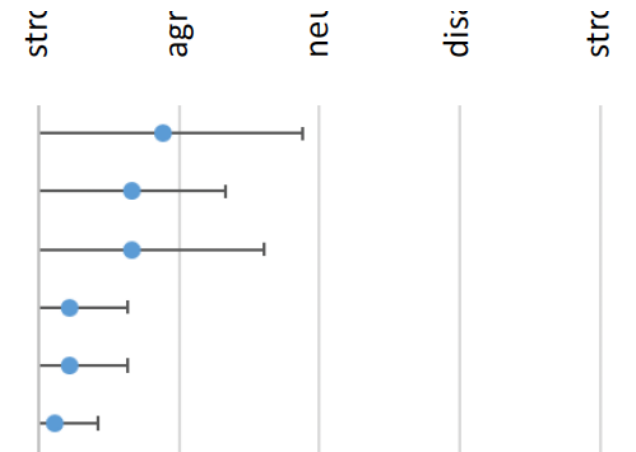
I will be able to use aspects in my future work.

The workshop was well-organized.

I recommend this course to other postdocs.

The workshop should be offered again.

The instructor should be invited again.



**Please assess the trainer concerning the following topics:**

Overall impression

Comprehensibility (easy to understand)

Slides / Presentation

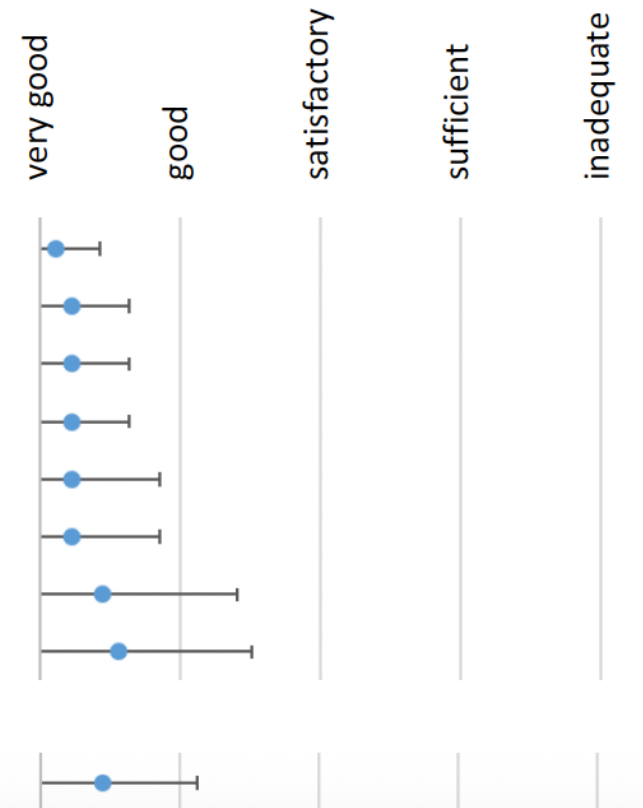
Handouts

Preparation of the trainer

Practical exercises

Feedback from the trainer

Use of pre-workshop questionnaire

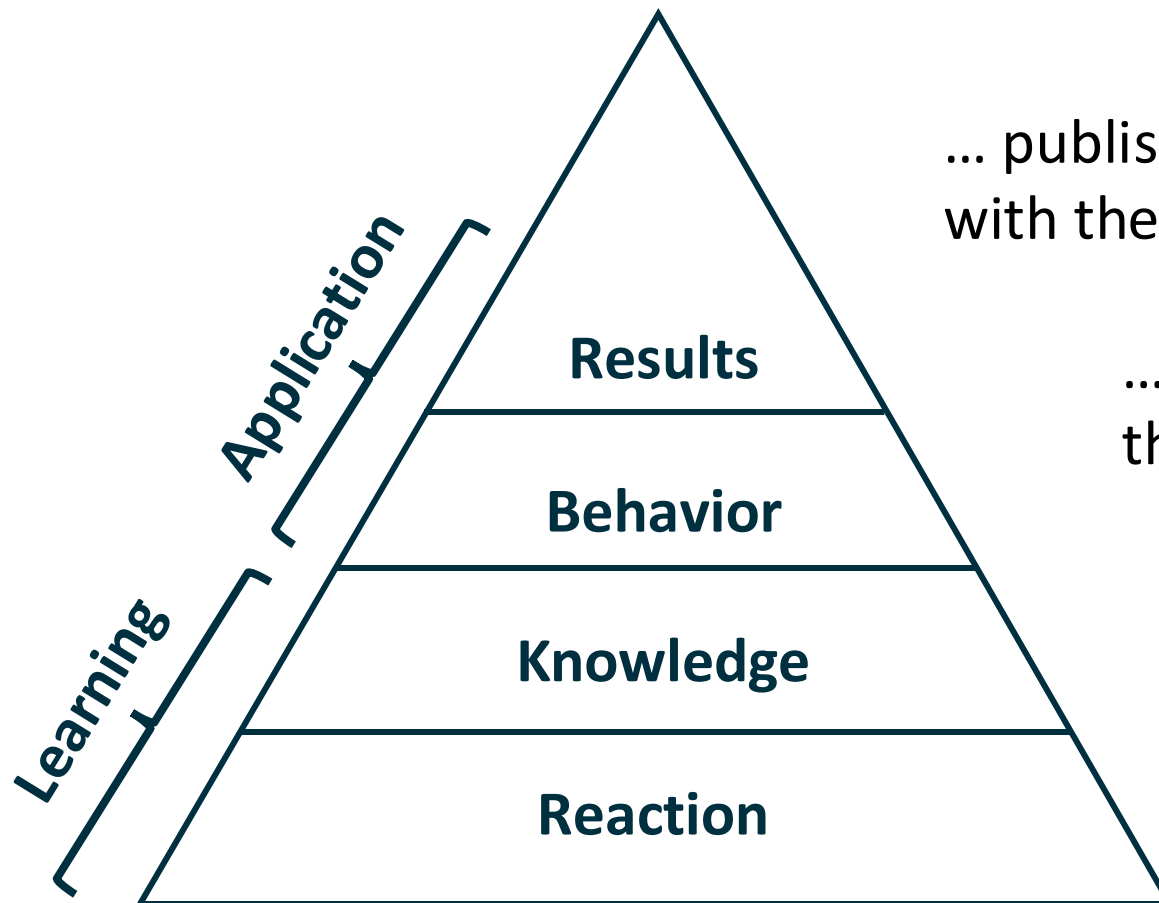


**How would you assess this workshop in total?**





# 4 Levels of Learning Transfer (Kirckpatrick Model)



... publishes a paper  
with these results.

... starts to use nn to model  
their own data sets.

... knows how to  
model with nn.

The student liked  
the dl workshop.

# What Makes Learning Effective?

## 5 Checks for Guidance.



See also: Ina Weinbauer-Heidel „Was Trainings wirklich wirksam macht: 12 Stellhebel der Transferwirksamkeit“

# Teaching Digital Topics: 10 Practical Advices



1. **Adapt to different levels and personalities** → If possible, work with small groups (max 15 students).
2. **Assume no prior knowledge and introduce challenges gradually** → Begin with fundamentals, increase difficulty step by step, and conclude with advanced topics. Include optional extensions (bonus challenges).
3. **Avoid distractions and promote commitment** → Lead by example and stay committed yourself. Find creative ways to avoid distractions and strengthen commitment.
4. **Don't assume motivation** → Make relevance explicit (e.g., *"this method is used in real physics research"*).
5. **Balance practice and theory (≈50/50)** → Live coding, teamwork, and projects are as important as lectures.
6. **Distinguish familiarity from literacy** → Not just *"can use"*, but *"understands what it does and why"*.
7. **Promote collaboration** → Teamwork, pair programming, and peer review mirror real scientific practice.
8. **Encourage responsible tool use** → Rely on a few, well-understood tools and techniques. Prefer basic over complex tools. Promote critical awareness when using GenAI tools.
9. **Assure a fit of the training/workshop content** → Use pre-surveys to calibrate. Consider saying "no".
10. **Prioritize reflection, transfer & application** → Ask *"why does this matter?"*, work on projects with real datasets, emphasize reproducible workflows, and plan for application beyond the teaching format.

# Acknowledgements and References



Vlab Project lead by Prof. Dr. Christian Bressler (Uni Hamburg / European XFEL)

Yvonne Konstanze Behnke „Lernmythen aufgedeckt“

Images generated partially with ChatGPT and Midjourney

Ina Weinbauer-Heidel „Was Trainings wirklich wirksam macht: 12 Stellhebel der Transferwirksamkeit“



# Additional Slides



*Beyond Classroom Lectures and  
Exercises*

**Teaching in a Digital World**

# The Future of Teaching



## 4 Brainstorming Questions:

1. What novel teaching techniques (beyond lecture and exercise) do you know? How effective are they?
2. In a dream world - how would you teach? What tools would be available?
3. How do you think GenAI will change learning? Describe specific scenarios!
4. What challenges regarding teaching do you expect in the future?

Step 1: Split into 4 groups (5 min)

Step 2: Interview participants from other groups about your question. Take notes if needed (20 min)

Step 3: Summarize your group's findings and display on a flipchart (15 min)

Step 4: Let's discuss in the plenary (45 min)

Step 5: Closeout remarks