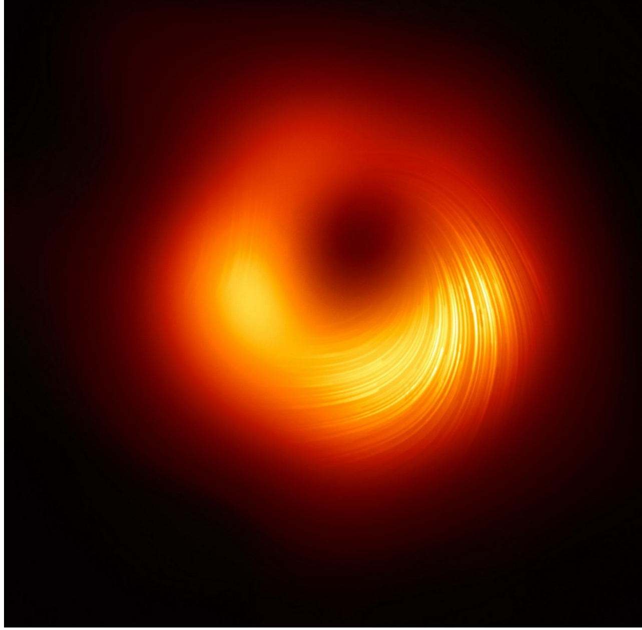
How to lower the entry barrier to your scientific software

Gabriele Bozzola, PhD

gbozzola@caltech.edu

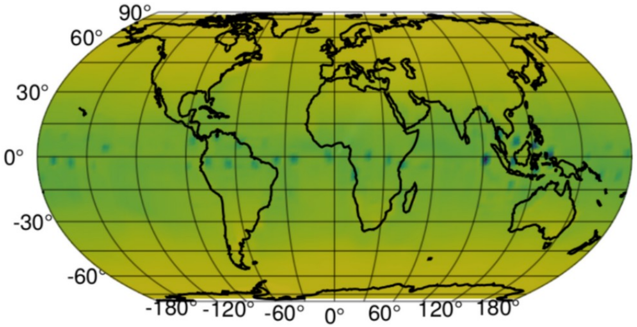
(Views are my own)

I am a (Scientific) Software Engineer



CliMA

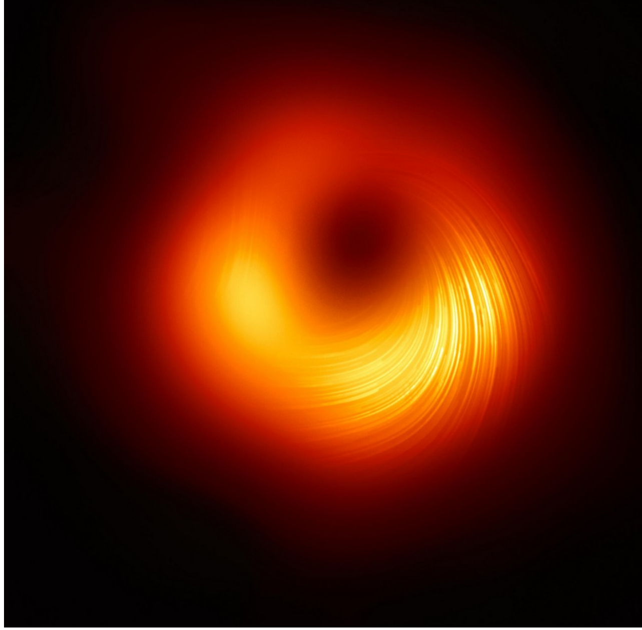
CLIMATE MODELING ALLIANCE



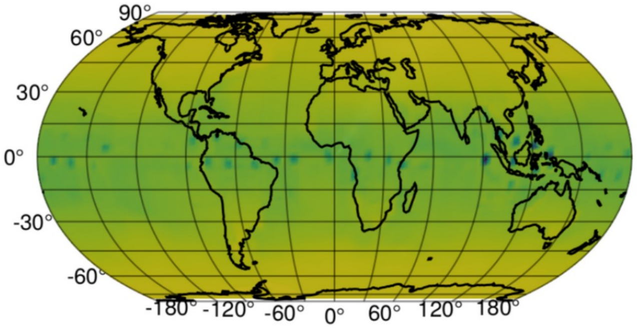
An open-source climate model in Julia

Ask me more about any of these!

I am a (Scientific) Software Engineer



CliMA
CLIMATE MODELING ALLIANCE



An open-source climate model in Julia

I also review scientific open-source software

JOSS
The Journal of Open Source Software

Ask me more about any of these!

This talk in one slide

This talk in one slide

Scientific software has fundamentally unique characteristics and needs …

This talk in one slide

Scientific software has fundamentally unique characteristics and needs …

… that lead to unique risks and opportunities …

This talk in one slide

Scientific software has fundamentally unique characteristics and needs …

… that lead to unique risks and opportunities …

… that we can start addressing with a user/dev/scientist centered approach to documentation …

This talk in one slide

Scientific software has fundamentally unique characteristics and needs …

… that lead to unique risks and opportunities …

… that we can start addressing with a user/dev/scientist centered approach to documentation …

… that we can elevate code from just a tool to a self-contained contribution …

This talk in one slide

Scientific software has fundamentally unique characteristics and needs …

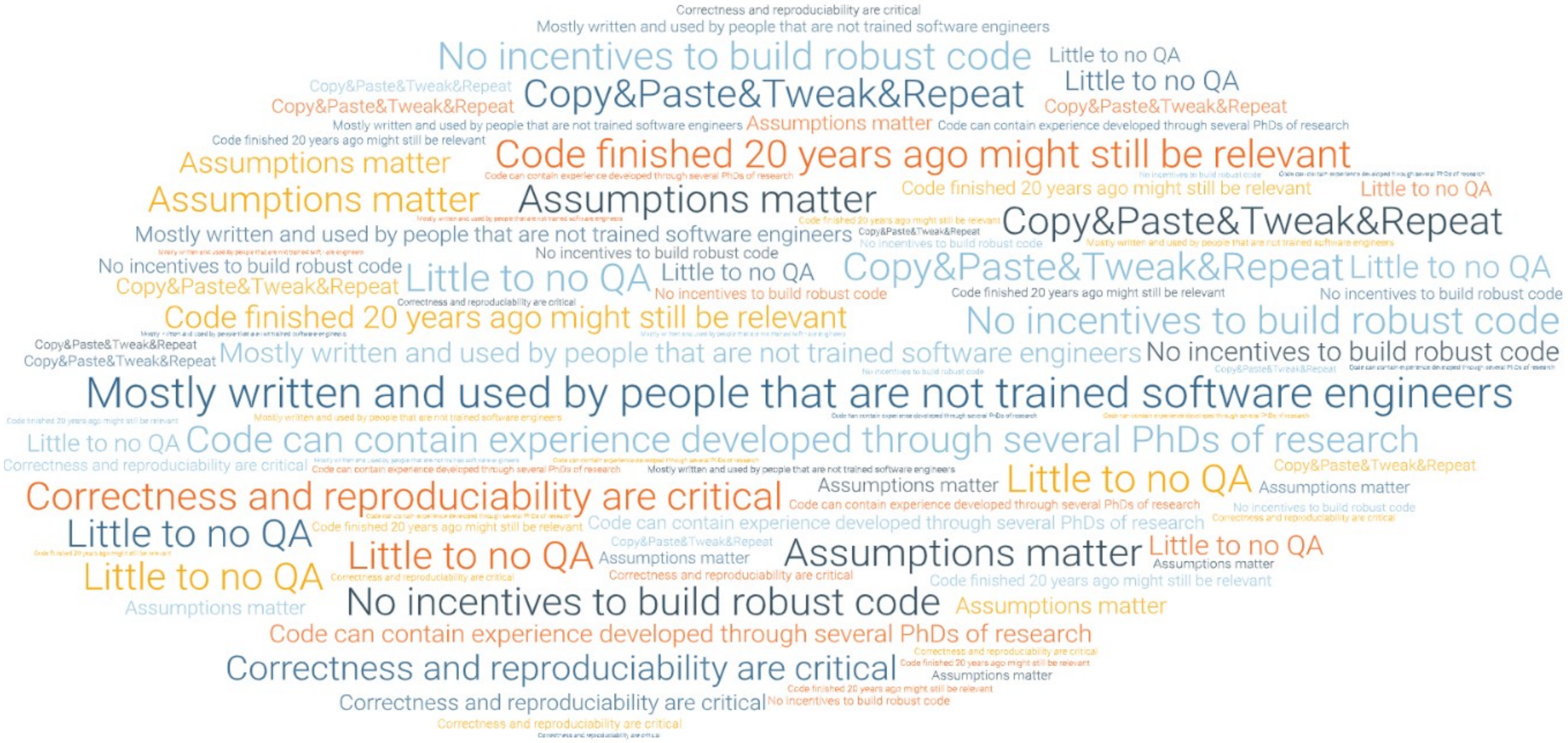
… that lead to unique risks and opportunities …

… that we can start addressing with a user/dev/scientist centered approach to documentation …

… that we can elevate code from just a tool to a self-contained contribution …

… to accelerate science!

Scientific software has unique characteristics and needs


Correctness and reproduciability are critical
Mostly written and used by people that are not trained software engineers
No incentives to build robust code
Little to no QA
Copy&Paste&Tweak&Repeat
Assumptions matter
Code can contain experience developed through several PhDs of research
Code finished 20 years ago might still be relevant

POV: You are an average piece of scientific software

- You have been developed by a graduate student over 5 years (or their advisor 25 years ago)
- People learn how to use you through oral tradition
- You have no automated tests
- You lead to reproducible results only on even days
- You are required to be correct

POV: You are an average piece of scientific software

- You have been developed by a graduate student over 5 years (or their advisor 25 years ago)
- People learn how to use you through oral tradition
- You have no automated tests
- You lead to reproducible results only on even days
- You are required to be correct

- You were used to drive new science and publish papers
- You contain lots of lessons learned
- You could be used for future projects
- You have lots of potential!

Problem:
There is a lot of unrealized scientific potential!

Science killers:

- Time spent understanding the code
- Duplication of efforts
- Incorrect code/usage
- Unknown mismatch in assumptions
- …

Problem:
There is a lot of unrealized scientific potential!

Science killers:

- Time spent understanding the code
- Duplication of efforts
- Incorrect code/usage
- Unknown mismatch in assumptions
- …

Opportunity:
Code as a self-contained scientific contribution

Science enabler:

- Openness
- Ease of use and extension
- No hidden knowledge
- Sharing the “lessons learned”
- …

Problem:
There is a lot of unrealized scientific potential!

Science killers:

- Time spent understanding the code
- Duplication of efforts
- Incorrect code/usage
- Unknown mismatch in assumptions
- …

Opportunity:
Code as a self-contained scientific contribution

Science enabler:

- Openness
- Ease of use and extension
- No hidden knowledge
- Sharing the “lessons learned”
- …

Documentation mitigates risks and contributes to elevating status of code from **tool to scientific contribution**

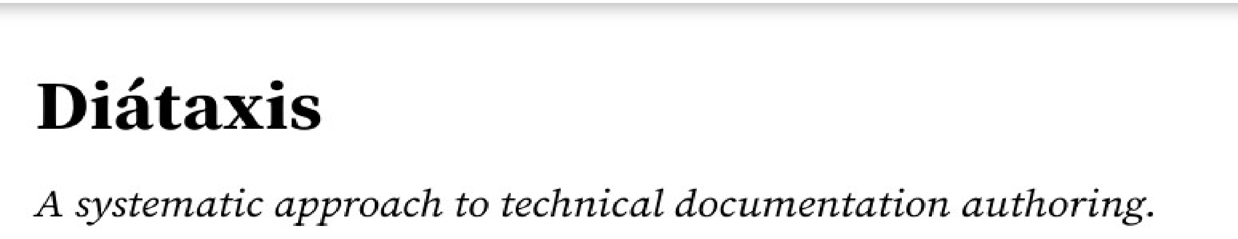
A practical framework for effective technical documentation

Gabriele Bozzola, PhD

gbozzola@caltech.edu

(Views are my own)

Extending Daniele Procida’s diataxis.fr for scientific code



Diátaxis

A systematic approach to technical documentation authoring.

Two core principles:

Explicitly address **users/developers/maintainers/scientists needs**

Explicitly address **how people learn and seek information**

Doc Braun's lab is studying time travel

The lab is developing `TheLorean.jl`:
A Julia code to compute flux capacitance

Doc Braun’s lab is studying time travel

The lab is developing `TheLorean.jl`:
A Julia code to compute flux capacitance

Marty McDrive

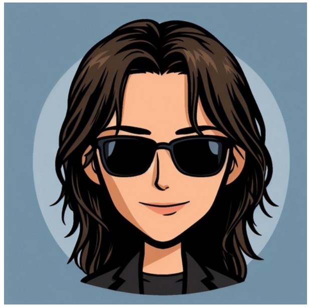
Graduate student
Mostly a user

Doc Braun’s lab is studying time travel

The lab is developing `TheLorean.jl`:
A Julia code to compute flux capacitance

Marty McDrive

Graduate student
Mostly a user



Tera Connor

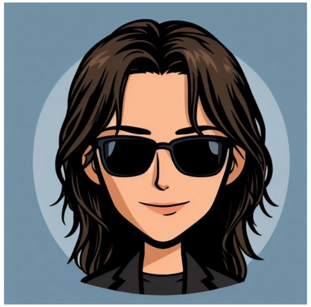
Postdoc
User and developer

Doc Braun’s lab is studying time travel

The lab is developing `TheLorean.jl`:
A Julia code to compute flux capacitance

Marty McDrive

Graduate student
Mostly a user



Tera Connor

Postdoc
User and developer

Dr Barnsworth

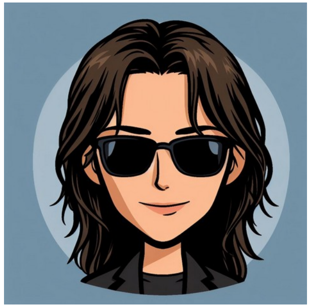
“Competing” researcher
Interested in methods and capabilities

Doc Braun’s lab is studying time travel

The lab is developing **`TheLorean.jl`**:
A Julia code to compute flux capacitance

Marty McDrive

Graduate student
Mostly a user



Tera Connor

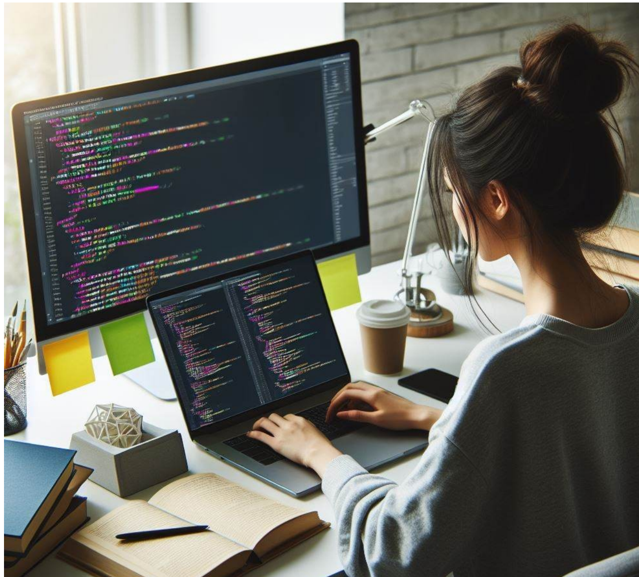
Postdoc
User and developer

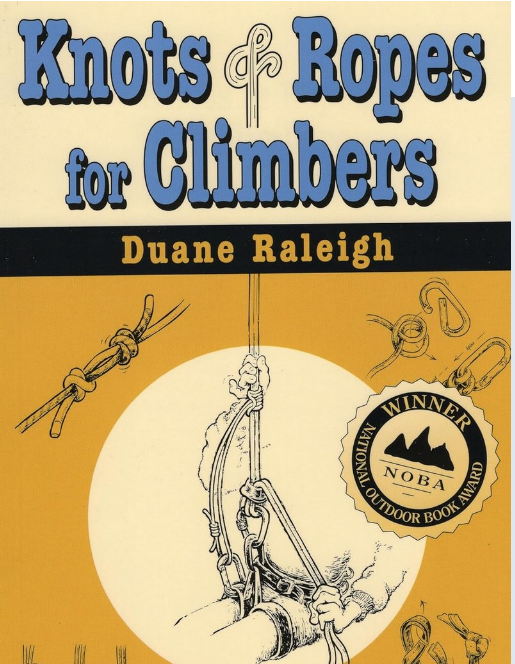
Dr Barnsworth

“Competing” researcher
Interested in methods and capabilities

To **use**, to **extend**, to **learn from**

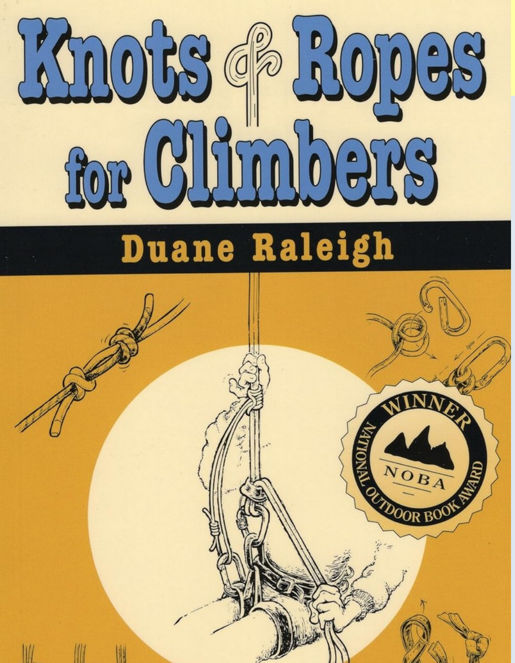
So, you decided you want to learn to <insert craft>





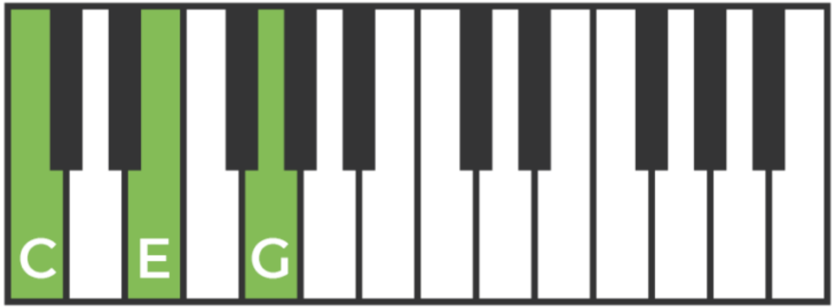
THEORETICAL KNOWLEDGE

PRACTICAL SKILL

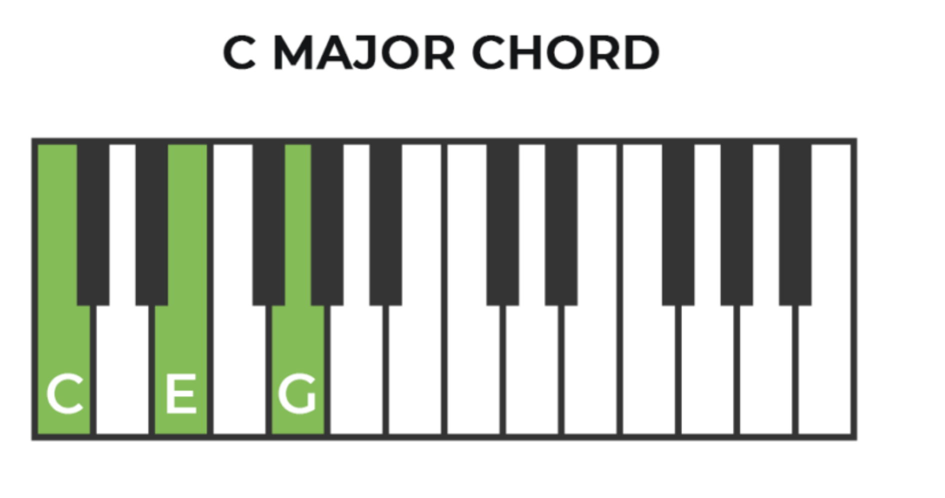


THEORETICAL KNOWLEDGE

C MAJOR CHORD

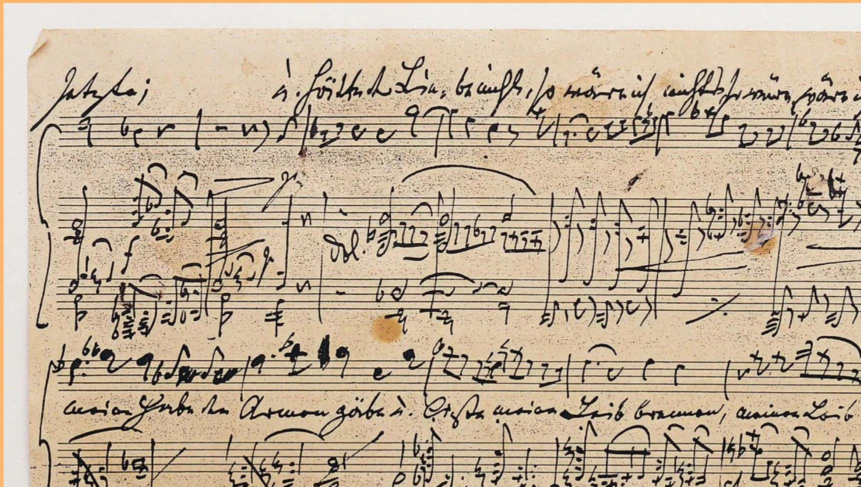


WHILE LEARNING



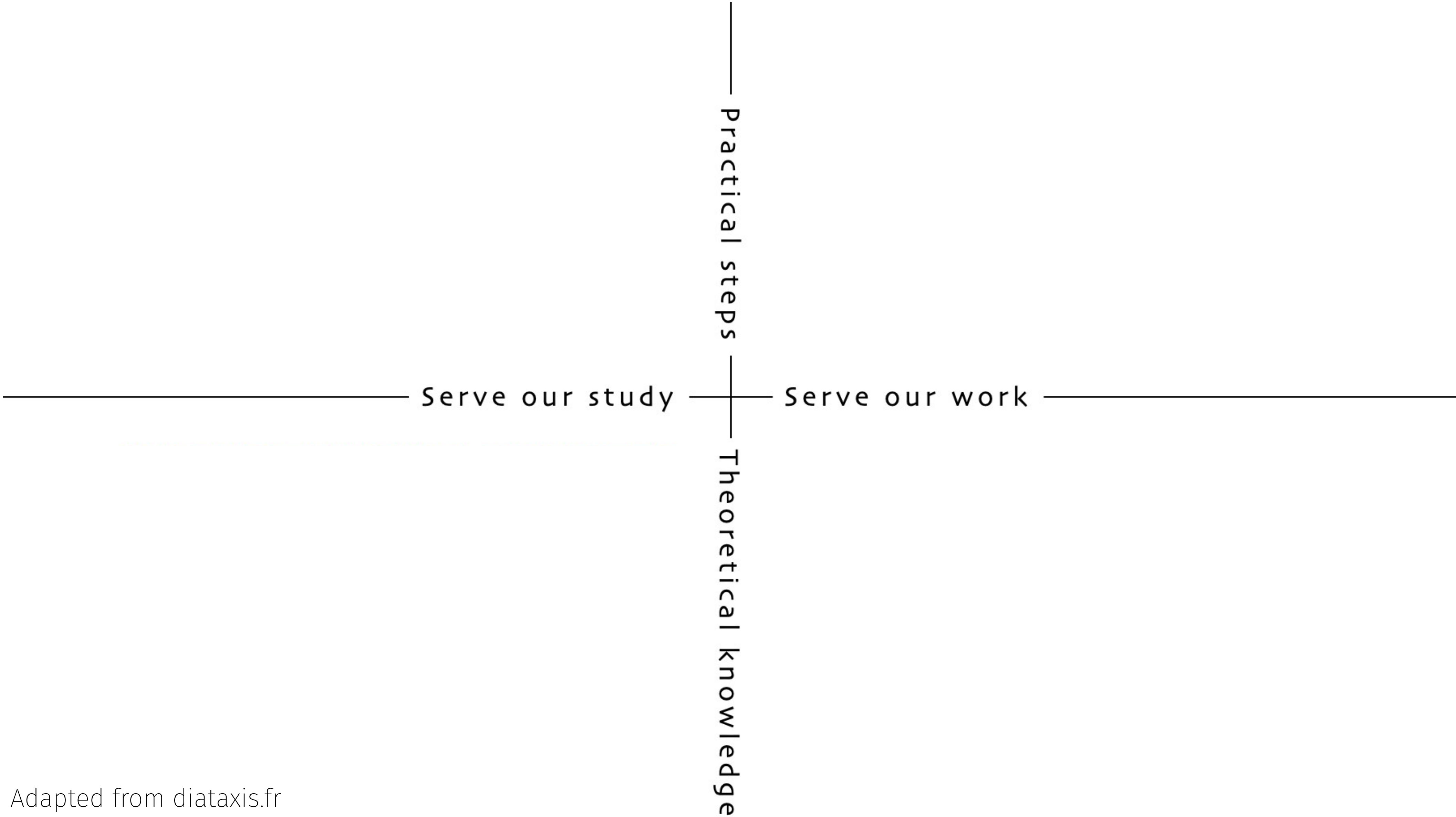
WHILE LEARNING

WHILE DOING



The key idea:

DOCUMENTATION IS NOT ONE THING, AND IS NOT FOR ONE PERSON



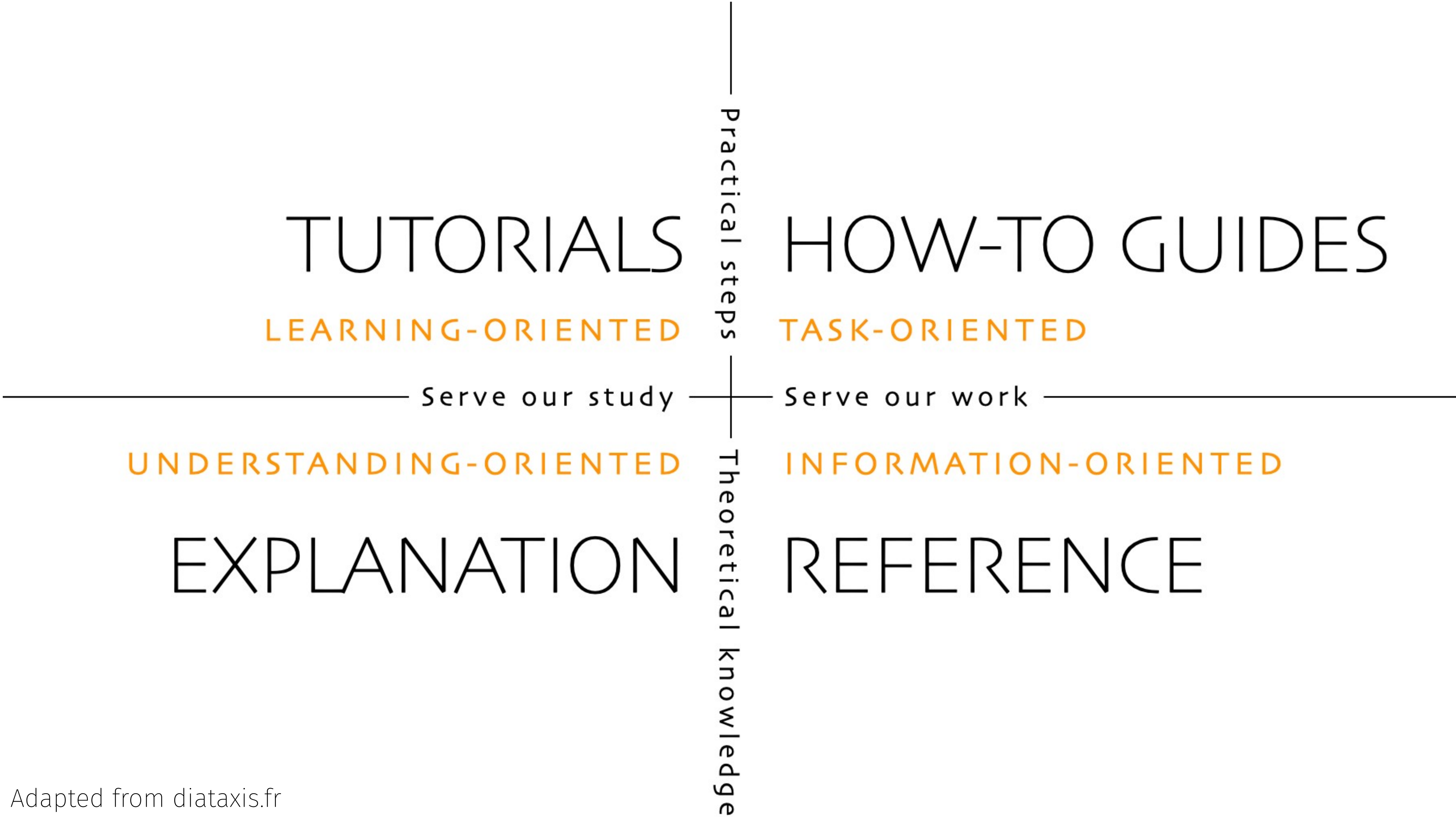
Practical steps

Serve our study

Serve our work

Theoretical knowledge

Adapted from diataxis.fr



Practical steps

TUTORIALS

LEARNING-ORIENTED

HOW-TO GUIDES

TASK-ORIENTED

Serve our study

Serve our work

UNDERSTANDING-ORIENTED

EXPLANATION

INFORMATION-ORIENTED

REFERENCE

Theoretical knowledge

Adapted from diataxis.fr

Practical steps

TO USE

TUTORIALS

LEARNING-ORIENTED

TO USE

HOW-TO GUIDES

TASK-ORIENTED

Serve our study

Serve our work

Theoretical knowledge

UNDERSTANDING-ORIENTED

EXPLANATION

TO EXTEND

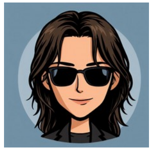
INFORMATION-ORIENTED

REFERENCE

TO USE

TO EXTEND

TO LEARN FROM



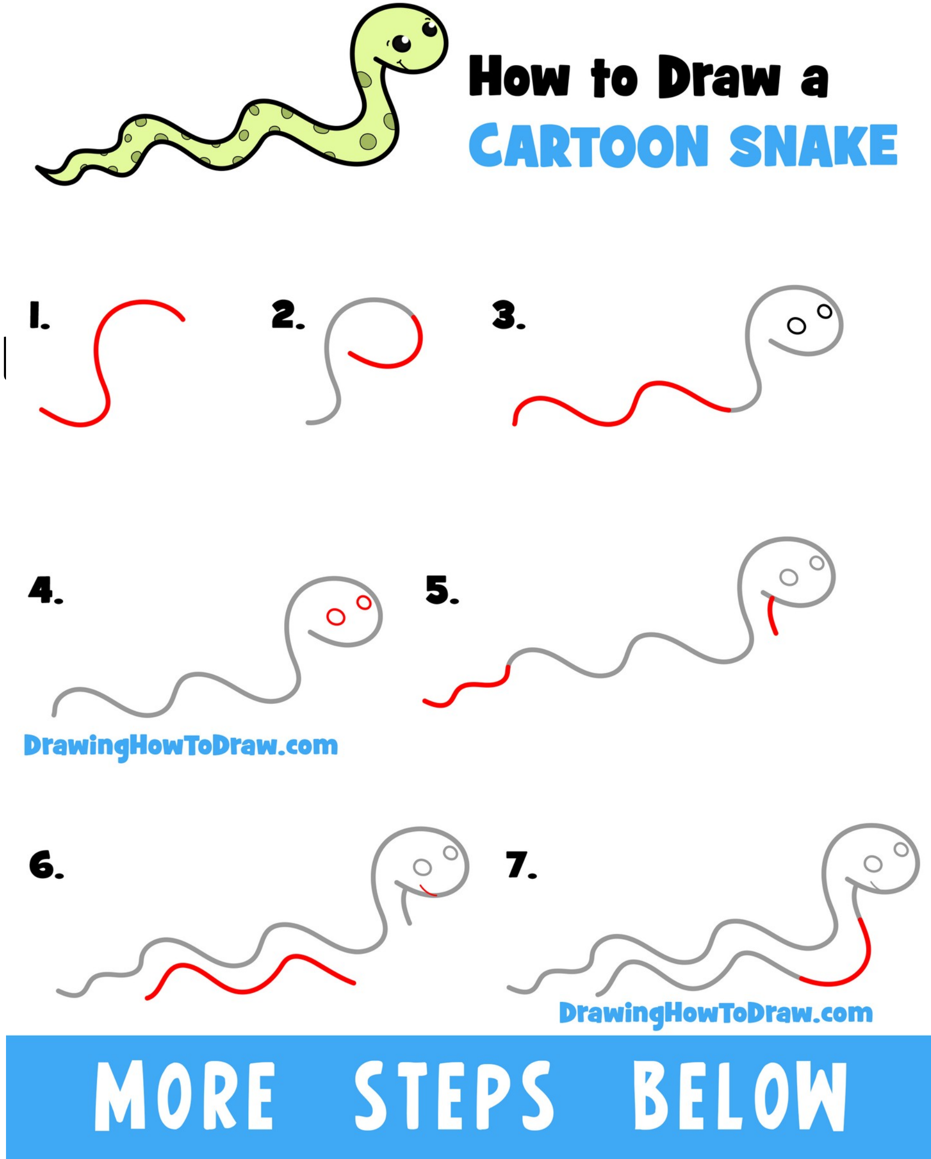
Adapted from diataxis.fr

TUTORIALS

GOAL: Get the user started,
provide familiarity with the vocabul

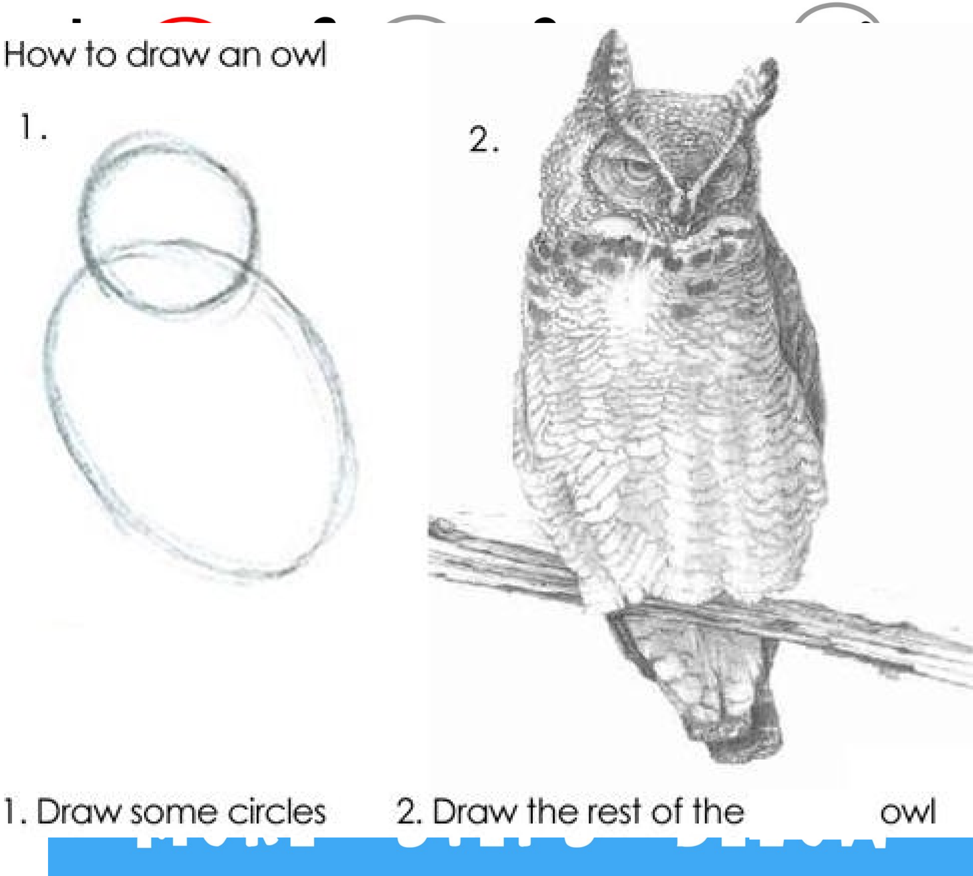
- Provide a complete picture before they start
- Ensure the user sees results immediately
- Describe concrete steps, not abstract concepts
- Offer only minimum, necessary, explanation
- Ignore options and alternatives

Tutorials are learning-oriented



TUTORIALS

GOAL: Get the user started,
provide familiarity with the voca



- Provide a complete picture before they start
- Ensure the user sees results immediately
- Describe concrete steps, not abstract concept
- Offer only minimum, necessary, explanation
- Ignore options and alternatives

Tutorials are learning-oriented

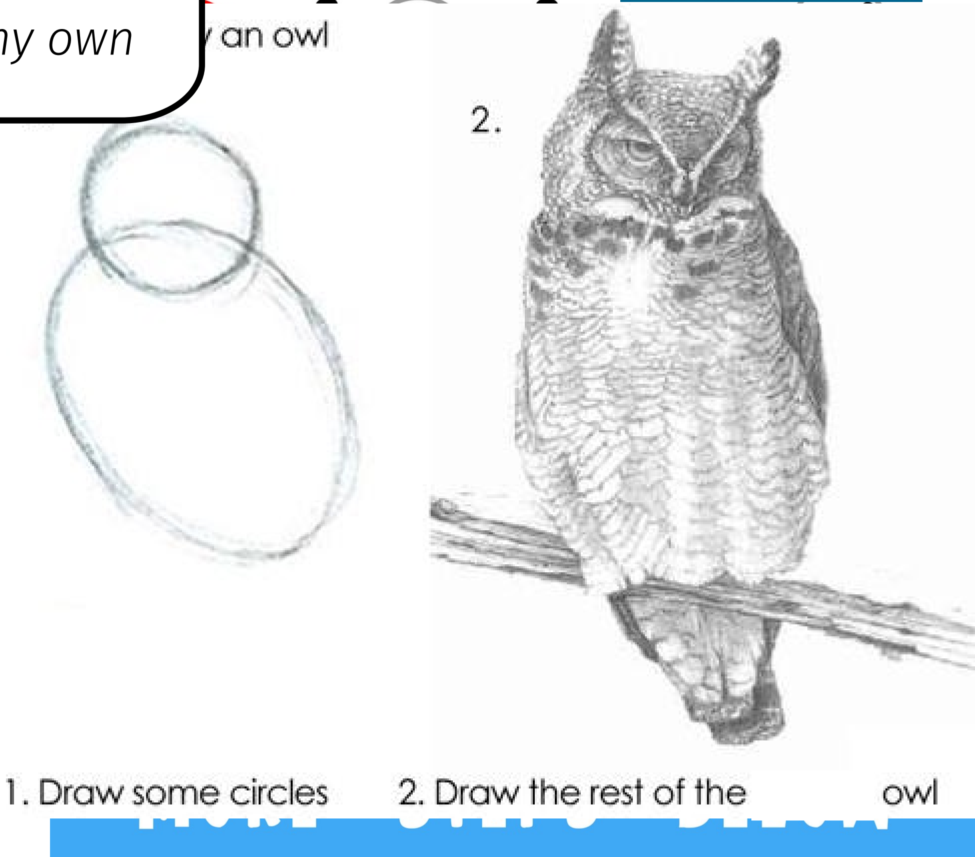
TUTORIALS

GOAL: Get the user started,
provide familiarity with

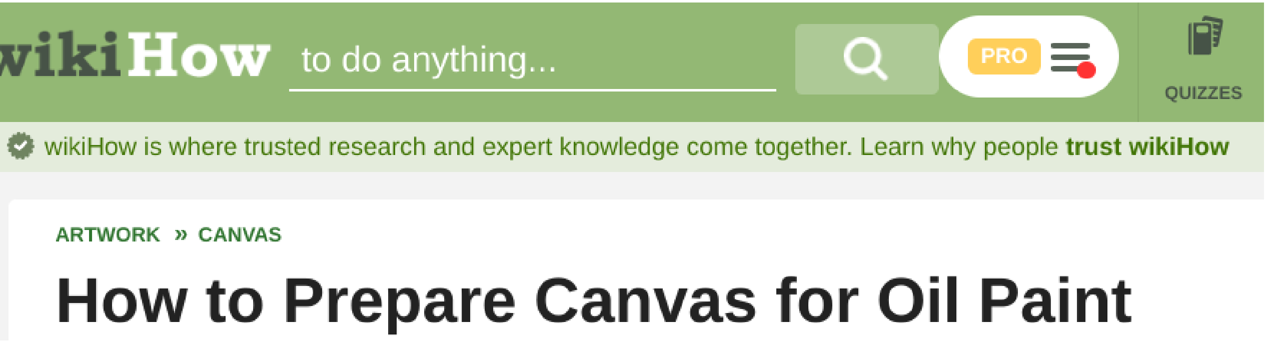
- Provide a complete picture before they start
- Ensure the user sees results immediately
- Describe concrete steps, not abstract concept
- Offer only minimum, necessary, explanation
- Ignore options and alternatives

Tutorials are learning-oriented

I like this because it allows me to get familiar with the code on my own



HOW-TO GUIDE



wikiHow is where trusted research and expert knowledge come together. Learn why people **trust wikiHow**

ARTWORK » CANVAS

How to Prepare Canvas for Oil Paint

How to Tie a Tie
1
2
3
4
5
6
7

GOAL: Provide steps to accomplish something

- Describe a sequence of actions
- Solve a problem
- Don’t explain concepts
- Be flexible
- Omit the unnecessary

How-to guides are goal oriented

wi
do anything...
PRO
QUIZZES
search and expert knowledge come together. Learn why people trust wikiHow
ARTWORK » CANVAS
How to Pre re Canvas for Oil Paint
Hov
I like this because I want to use this code for specific scientific applications
1
2
5
6
7

HOW-TO GUIDE

GOAL: Provide steps to accomplish something

- Describe a sequence of actions
- Solve a problem
- Don’t explain concepts
- Be flexible
- Omit the unnecessary

How-to guides are goal oriented

A tool to bridge learning-doing bridge: EXAMPLES

GOAL: Demonstrate real-life usage

- Showcase and provide idiomatic implementations
- Be starting point for typical use cases
- Maintain a pedagogical spirit
- Can be small or large
- Must-have in scientific software

© Arduino

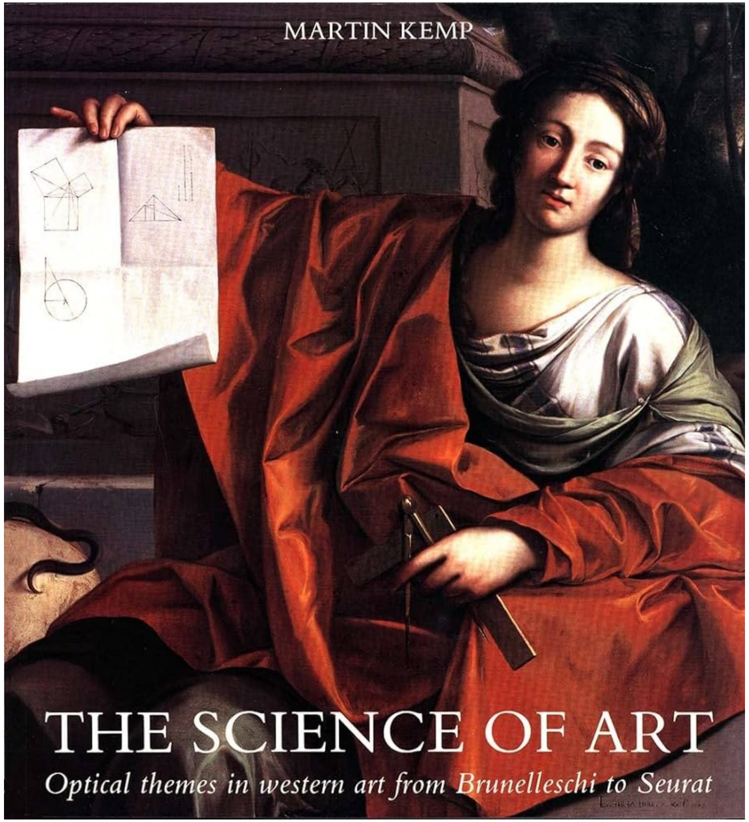
Examples bridge learning and doing

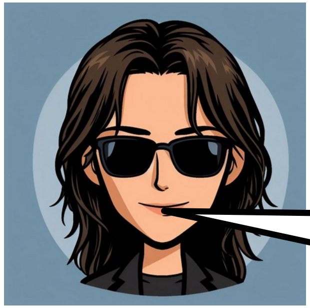
EXPLANATION

GOAL: Provide holistic understanding, Clarify the Whys

- Make connections
- Provide context
- Talk about the subject
- Discuss alternatives and opinions
- Don’t instruct

Explanation is understanding oriented





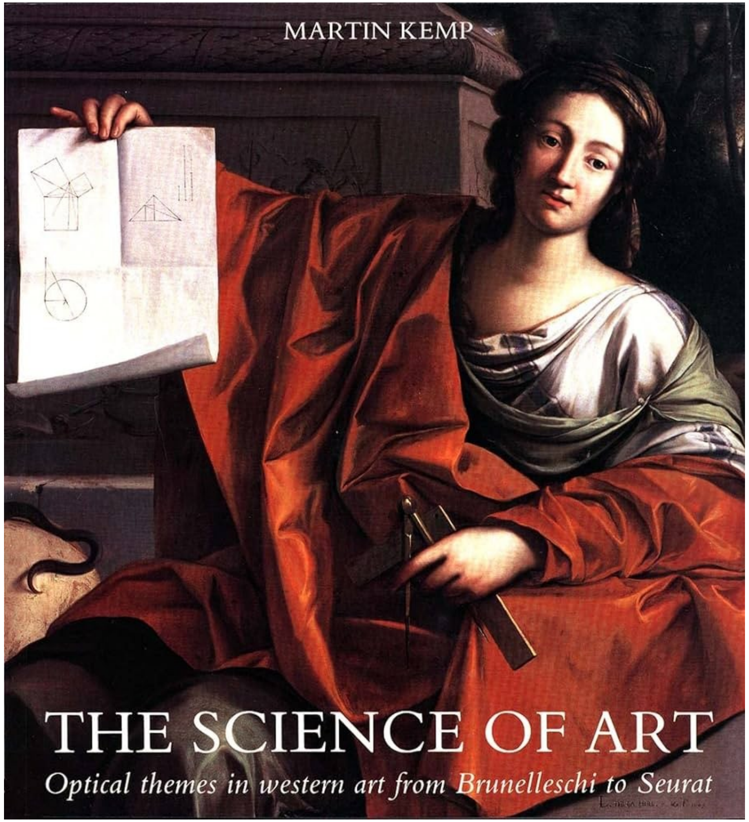
I like this because it gives me confidence to make large changes

EXPLANATION

GOAL: Provide holistic understanding, Clarify the Whys

- Make connections
- Provide context
- Talk about the subject
- Discuss alternatives and opinions
- Don’t instruct

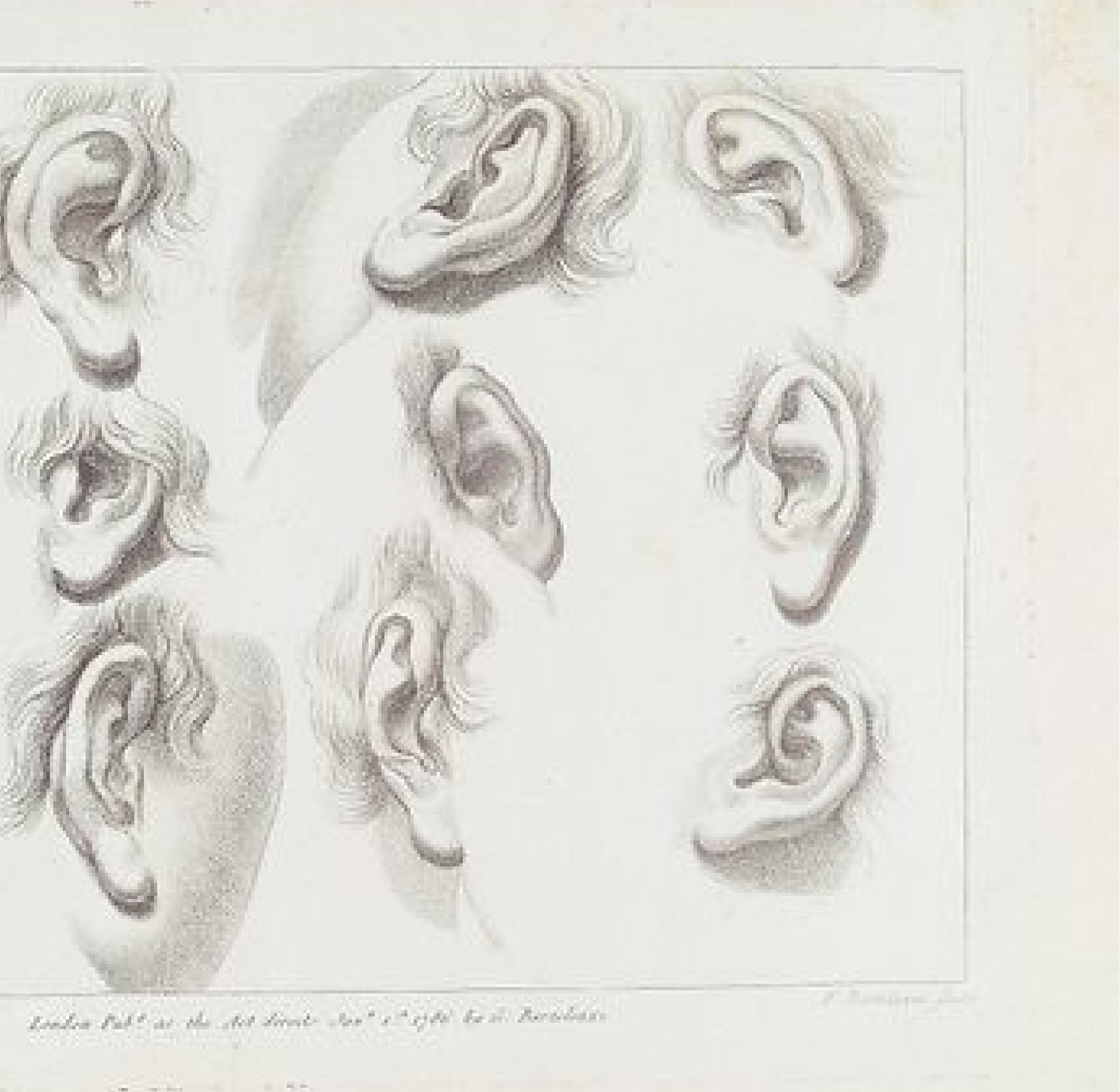
Explanation is understanding oriented



MARTIN KEMP

THE SCIENCE OF ART

Optical themes in western art from Brunelleschi to Seurat

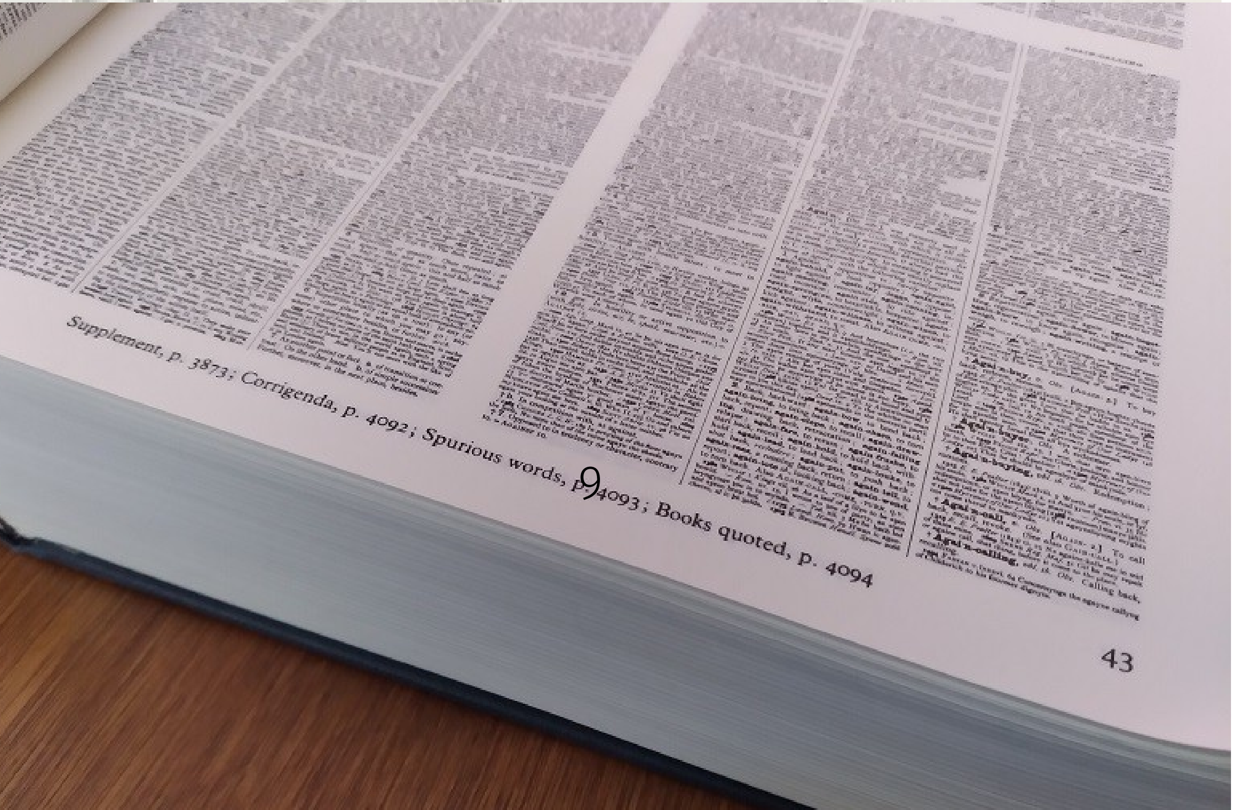


REFERENCE

GOAL: Provide authoritative information on how things are and work

- Do nothing but describe
- Be consistent
- Be accurate

Reference is consultation oriented



Supplement, p. 3873; Corrigenda, p. 4092; Spurious words, p. 4093; Books quoted, p. 4094

43

REFERENCE

GOAL: Provide authoritative information on how things are and work

- Do nothing but describe
- Be consistent
- Be accurate

Reference is consultation oriented

REFERENCE

- Function signatures
- Public APIs
- System architecture
- Object hierarchy
- …

REFERENCE

- Function signatures
- Public APIs
- System architecture
- Object hierarchy
- …

+

REFERENCE for scientific software

- Spell out assumptions, methods, formalism, equations
- Define acronyms
- List and link relevant papers
- List all the working features (in one place)

REFERENCE

- Function signatures
- Public APIs
- System architecture
- Object hierarchy
- ...

+

REFERENCE for scientific software

- Spell out assumptions, methods, formalism, equations
- Define acronyms
- List and link relevant papers
- List all the working features (in one place)

I like this because it allows me to get a high-level view on the science

The documentation matrix

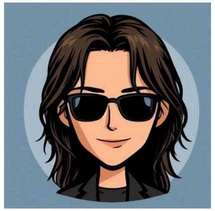
	TUTORIALS	HOW-TO GUIDES	EXPLANATION	REFERENCE
GOAL	Get started, Become familiar	Accomplish a specific task	Build understanding	Describe
ORIENTED TO	Learning	Tasks	Grokking	Consultation
TARGETING	New users	Users	Advanced users/developers	Everyone (+ scientific community)

EXAMPLES

Adapted from diataxis.fr

How to lower the entry barrier to your scientific software

This framework provide natural entry points for **everyone** at **any stage**



Important: This is not the final word!

Frequently Asked Questions

> I only want to spend 15 minutes on documentation, what should I prioritize?

One paragraph where you explain what your code is supposed to accomplish, what input it expects, and what outputs it produces.

Frequently Asked Questions

> I only want to spend 15 minutes on documentation, what should I prioritize?

One paragraph where you explain what your code is supposed to accomplish, what input it expects, and what outputs it produces.

> I only want to spend one hour on documentation, what should I prioritize?

A **features** page/section

- Immediately useful to everyone that is not you
- Clearly identifies what features are supposed to work

Frequently Asked Questions

> I only want to spend 15 minutes on documentation, what should I prioritize?

One paragraph where you explain what your code is supposed to accomplish, what input it expects, and what outputs it produces.

> I only want to spend one hour on documentation, what should I prioritize?

A **features** page/section

- Immediately useful to everyone that is not you
- Clearly identifies what features are supposed to work

> I want to write the best documentation ever, what should I do?

I don't know. Think about the core principles. :)

Assess your project with these at-home tests

The survivability test:

Would someone with reasonable experience be able to independently use and extend my code?

Assess your project with these at-home tests

The survivability test:

Would someone with reasonable experience be able to independently use and extend my code?

The accessibility test:

Would a beginner graduate student be able to use and extend my code with proper guidance?

My call for action

Think of scientific software as a standalone scientific contribution

Design your documentation by keeping in mind that documentation is not one thing and not for one person

Get in touch with me:
gbozzola@caltech.edu
Linkedin: gabrielebozzola