

# BASICS OF DOCUMENTING YOUR CODE

## PART 02

ALI FARNUDI



دانشگاه صنعتی شریف



The Abdus Salam  
International Centre  
for Theoretical Physics

DAVID

JACOB KAPLAN-MOSS

---

**CREDIT**

# A QUICK REVIEW OF THE LAST LECTURE

---

- ▶ **Less comment**, is not a **bad** thing.
  - ▶ **Don't** comment things that are **obvious**
  - ▶ **Don't** comment **what** your code is doing
  - ▶ **Only comment why** you did something
- ▶ comments are **dead**
- ▶ **Well defined** variables, functions, and inputs.

---

**BETTER DOCUMENTATION WILL MAKE  
YOUR PROJECT MORE SUCCESSFUL.**

# BETTER DOCUMENTATION WILL MAKE YOUR PROJECT MORE SUCCESSFUL.

---

**Why** do people read documentation?

**Who** should write the documentation?

**What** should we document?

**Which** tools should we use?

---

**WHY?**

# WHY?

---

- ▶ **First Contact** - new users
- ▶ **Education** - new users + existing users
- ▶ **Support** - experienced users
- ▶ **Troubleshooting** - annoyed users
- ▶ **Internal** - your fellow developers
- ▶ **Reference** - everyone

---

**DOCUMENTATION IS COMMUNICATION**



# DOCUMENTATION IS COMMUNICATION

---



you

# DOCUMENTATION IS COMMUNICATION

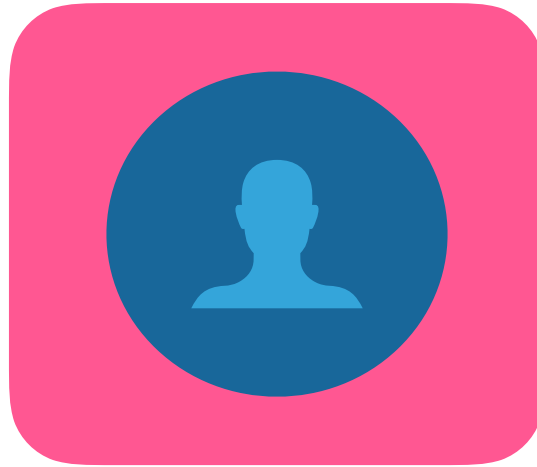
---



developers

# DOCUMENTATION IS COMMUNICATION

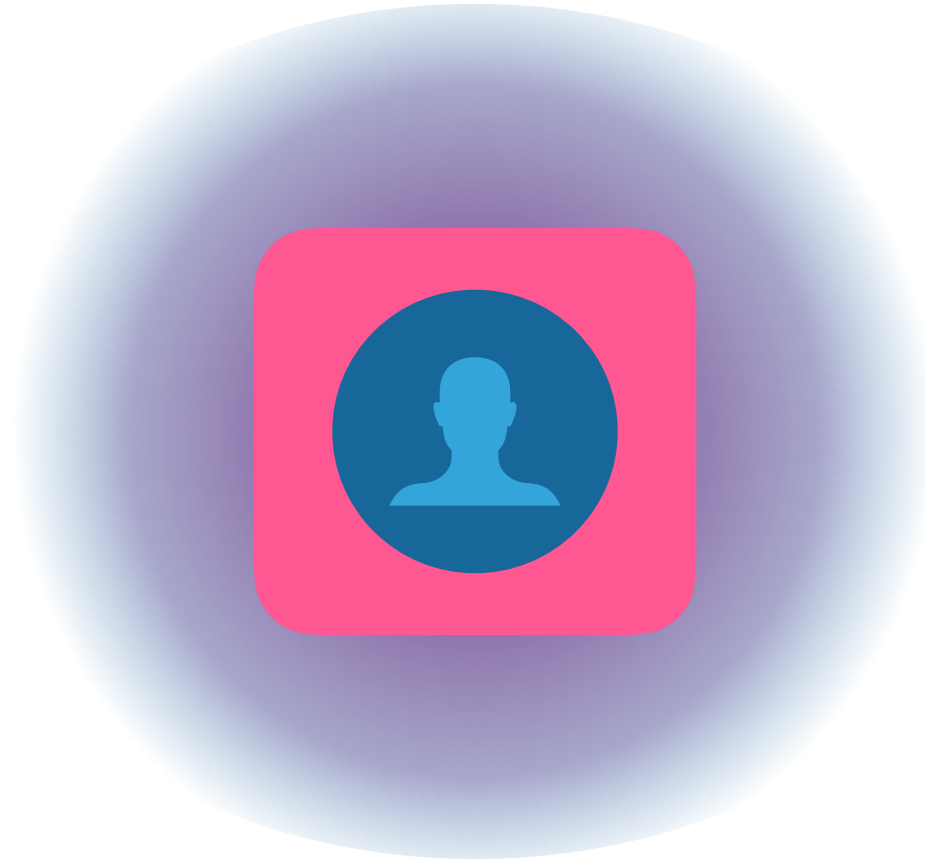
---



Engaged users that  
send you emails

# DOCUMENTATION IS COMMUNICATION

---



invisible users that don't ask  
questions on stack overflow

---

**GREAT DOCUMENTATION HAS TO SERVE  
MULTIPLE CONFLICTING MASTERS**

---

**GREAT DOCUMENTATION HAS TO SERVE MULTIPLE CONFLICTING MASTERS**

**SO NOT JUST DOCUMENT IT BUT  
WRITE IT MULTIPLE TIMES!**

**IS IT STARTING TO SOUND HARD?**

---

**WHO?**

**ANYONE**

# WHO? ANYONE

---

- ▶ Should be obsessed with perfection.
- ▶ You are not being published, so typos and **minor** grammar mistakes are ok.
- ▶ If there is no documentation people will **complain and move elsewhere**.
- ▶ But if **you write it** people will **help you** with the gaps.
- ▶ **Great documentation is written by great developers.**
- ▶ Maybe try **Documentation Driven Development**.



---

**WHAT?**

# WHAT?

---

- ▶ **Tutorials**
- ▶ **Topic Guides**
- ▶ **References**
- ▶ **Troubleshooting**

# WHAT? TUTORIALS

---

- ▶ **Quick** - The user should experience success within 30mins
- ▶ **Easy** - help users feel an epic win
- ▶ **Not too easy** - don't sugar coat the truth
- ▶ Show off how the project **feels**

# WHAT? TUTORIALS TOPIC GUIDES

---

- ▶ **Conceptual** - foster understanding, not parroting.
- ▶ **Comprehensive** - explain in detail.
- ▶ Tell me the **why** of the topic

# WHAT? TUTORIALS TOPIC GUIDES REFERENCE

---

- ▶ **Complete** - if it is not there it didn't happen.
- ▶ Designed for the **experienced user**
- ▶ Give me the **how** of the topic

# WHAT? TUTORIALS TOPIC GUIDES REFERENCE

---

**THERE ARE NO SHORTCUTS FOR DOCUMENTATION.**

**THEY SHOULD BE WRITTEN, ORGANISED, AND  
EDITED BY PEOPLE.**

**WHAT? TUTORIALS TOPIC GUIDES REFERENCE TROUBLESHOOTING**

---

**ANSWER TO QUESTIONS ASKED IN ANGER**

ANSWER TO QUESTIONS ASKED IN ANGER

**FAQs** are good as long as the **Qs** are really **FA'd**.



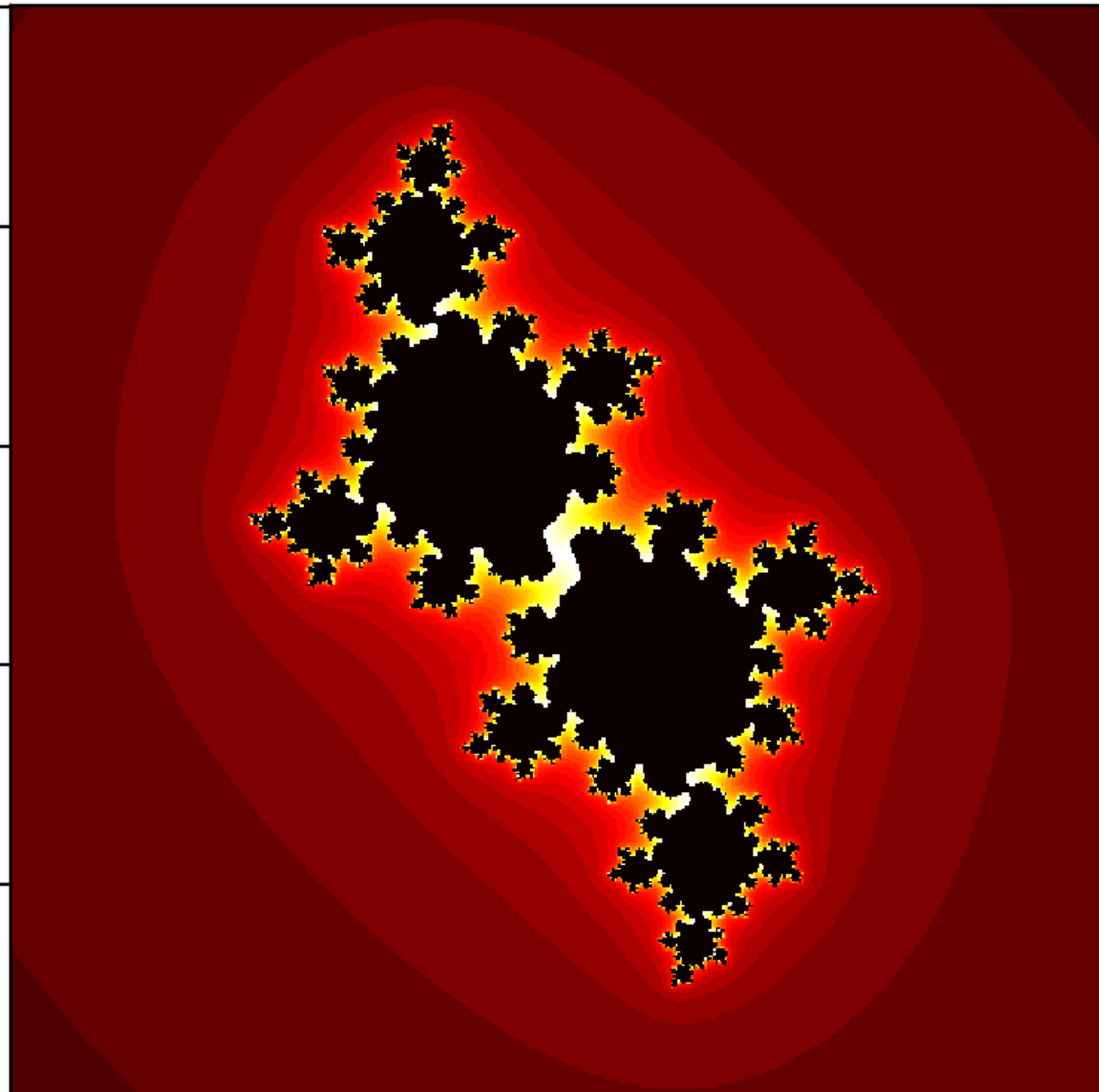
# WHAT? TUTORIALS TOPIC GUIDES REFERENCE TROUBLESHOOTING

---

ANSWER TO QUESTIONS ASKED IN ANGER

**FAQs** are good as long as the **Qs** are really **FA'd**.

**VERY DIFFICULT**



# WHAT? TUTORIALS TOPIC GUIDES REFERENCE TROUBLESHOOTING

	Tutorials	Topic guides	Reference	trouble-shooting
Project	Tutorials, getting started	Guides, how-tos	APIs, Indexes, search	FAQs
Document	Introductory material	How-to guides	See also, next steps	Notes, warnings
Section	Overview	Tasks, examples	Cross-ref, API docs	Common pitfalls
Elements	Examples	Detailed instructions	Cross-ref, API docs	"if it didn't work ..."

---

**WHICH?**

# WHICH?

---

**TOOLS DON'T MATTER  
(FOR THE MOST PART)**

**WHICH?**

---

**TOOLS DON'T MATTER (FOR THE MOST PART)**

**BUT THE MOST IMPORTANT THING IS TO  
GET IT RIGHT**

# WHICH?

---



# SPHINX

PYTHON DOCUMENTATION GENERATOR

# BETTER DOCUMENTATION WILL MAKE YOUR PROJECT MORE SUCCESSFUL.

---

**Why** do people read documentation?

**Who** should write the documentation?

**What** should we document?

**Which** tools should we use?



# BETTER DOCUMENTATION WILL MAKE YOUR PROJECT MORE SUCCESSFUL.

---

**Everybody** reads the documentation.

**Who** should write the documentation?

**What** should we document?

**Which** tools should we use?

# BETTER DOCUMENTATION WILL MAKE YOUR PROJECT MORE SUCCESSFUL.

---

**Everybody** reads the documentation.

**Developers** write great documents

**What** should we document?

**Which** tools should we use?

# BETTER DOCUMENTATION WILL MAKE YOUR PROJECT MORE SUCCESSFUL.

---

**Everybody** reads the documentation.

**Developers** write great documents

Great documentation is like a **fractal**

**Which** tools should we use?

# BETTER DOCUMENTATION WILL MAKE YOUR PROJECT MORE SUCCESSFUL.

---

**Everybody** reads the documentation.

**Developers** write great documents

Great documentation is like a **fractal**

**Tools don't matter** but use good ones



**GO WRITE!**