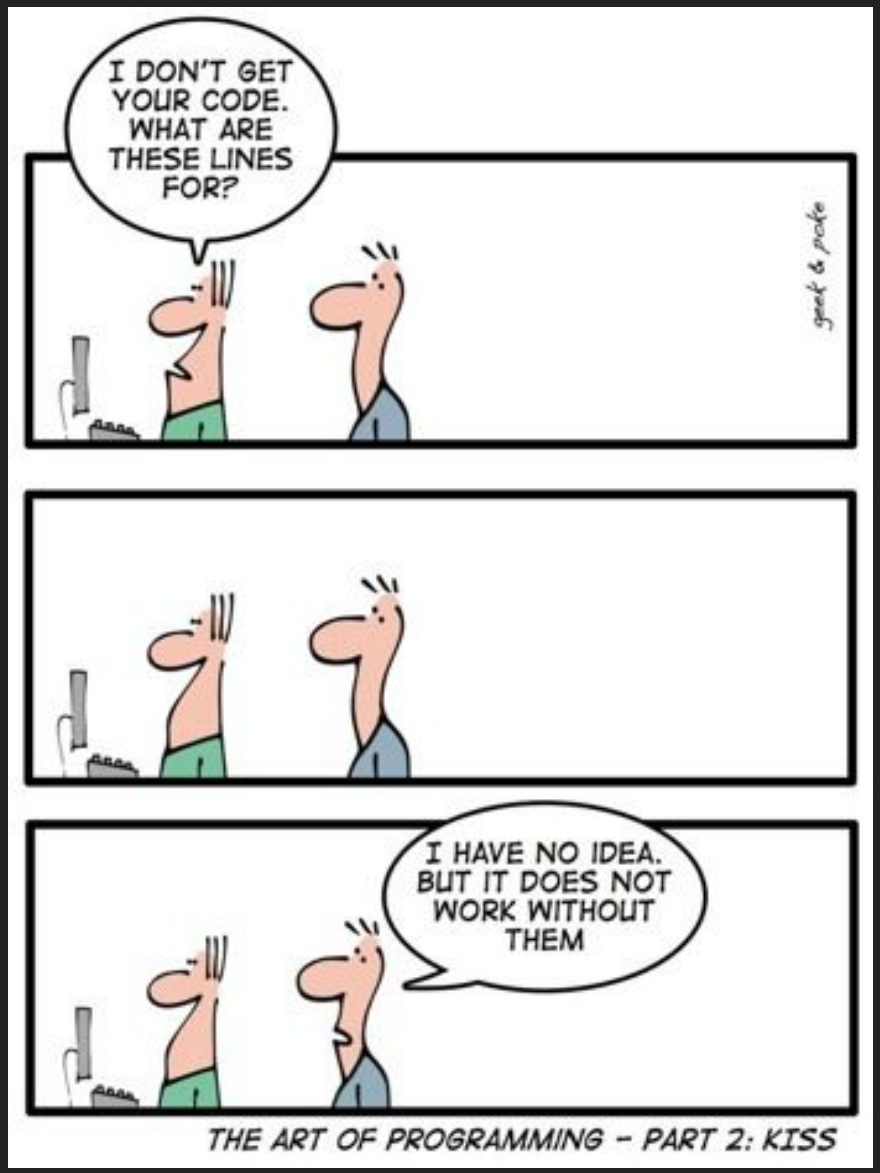


BASICS OF DOCUMENTING YOUR CODE PART 01

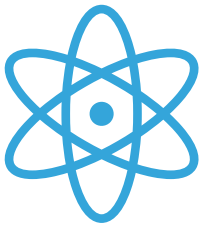
ALI FARNUDI

دانشگاه صنعتی شریف

The Abdus Salam International Centre for Theoretical Physics

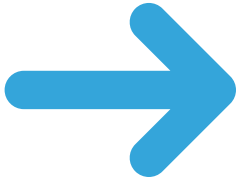
BETTER DOCUMENTATION WILL MAKE
YOUR PROJECT MORE SUCCESSFUL.

BETTER DOCUMENTATION WILL MAKE YOUR PROJECT MORE SUCCESSFUL.

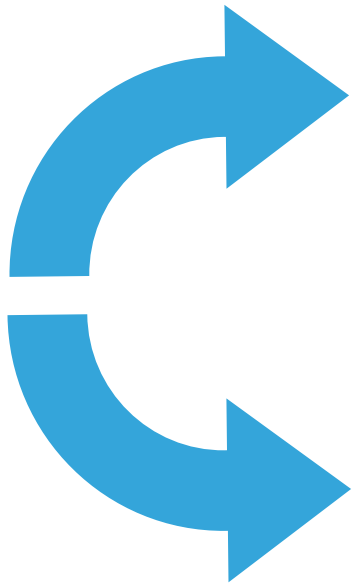


Scientists
write code
for a project

after a couple of
months/years



Write a
thesis/report



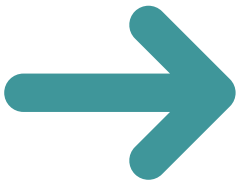
Other group
members will
further develop it

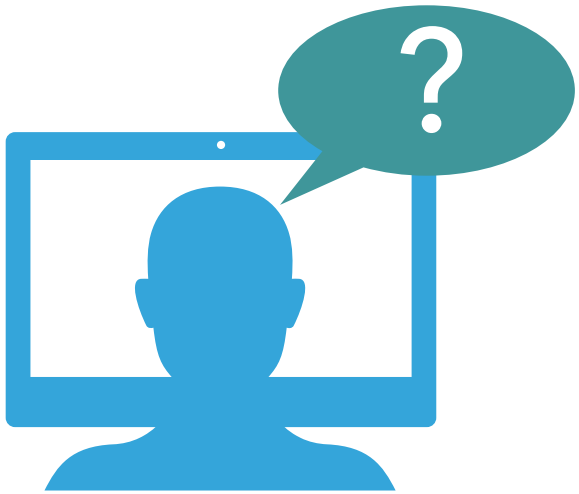
Release the code
to the scientific
community

BETTER DOCUMENTATION WILL MAKE YOUR PROJECT MORE SUCCESSFUL.

MUCH SIMPLER CASE

You are working in a group





Other group members should understand what you are doing.

BETTER DOCUMENTATION WILL MAKE YOUR PROJECT MORE SUCCESSFUL.

Should you comment on every line?

BETTER DOCUMENTATION WILL MAKE YOUR PROJECT MORE SUCCESSFUL.

PROS

- Makes the code **less confusing**.
- Makes it **easier to remember** the next time we look at it.
- Notes and reminders for the algorithm.
- Bug fixes

Should you comment on every line?

BETTER DOCUMENTATION WILL MAKE YOUR PROJECT MORE SUCCESSFUL.

PROS

- Makes the code **less confusing**.
- Makes it **easier to remember** the next time we look at it.
- Notes and reminders for the algorithm.
- Bug fixes

CONS

- Comments **quickly** become **obsolete** when we change the code.
- **Too much** comment can make your code **very confusing**.

TIPS ON COMMENTING

- Use **self-explanatory variable names** and use **consistent capitalisation**
  - sample file 01
  - sample file 02
- Keep functions **short** (10-20 lines)
- **Don't** comment things that are **obvious**
  - (example: list_a.append(1) #adding 1 to the list)
- **Don't** comment **what** your code is doing
- **Only comment why** you did something

WHAT ARE GOOD COMMENTS?

- Bug fixes that are related to the framework or structure of the problem you are solving.
- Why you chose to write a complicated algorithm (I could have used an array but I chose a recursive solution)
- To-Do comments, for example ‘improvements’

SO WHAT SHOULD I TAKE AWAY FROM THIS LECTURE?

- **Less comment**, is **not a bad** thing.
- comments are dead
- Well defined **variables**, **functions**, and **inputs**.

‘ALWAYS CODE AS IF THE GUY WHO ENDS UP MAINTAINING YOUR CODE WILL BE A VIOLENT PSYCHOPATH WHO KNOWS WHERE YOU LIVE.’

Martin Golding