

Computational thinking

François Marechal

- Forget about...
 - listening religiously to your professor
 - take the exercise &... the correction and go home
 - Reproduce
- You are entering the Active learning world !
 - Theory is given on how to solve the problems => structure your reasoning
 - The project scenario is built to learn & apply the theory
 - A-G-I-R
 - Project steps are used to guide/inspire your reasoning
 - Problem solving : facing problems by solving problems
 - Example are given as an inspiration and to speed-up the application
- Computational thinking
 - Computer Aided Engineering
 - Professional tools : MS word and excel should not be the tools for your professional life

Analyse

Generate

Interpret

Report

- **A : Analyse** : State the problem and Activate your knowledge

- Define the purpose of the calculations

- Define the level of details needed
- Identify your knowledge
- Identify the data you need (ref : Zotero)

=> we show you the way similar problems have been solved

- Activate the computational tools needed

- Python scripts - flowsheet simulation - CFD

=> be inspired by the examples

- Describe what you are looking for : for you and the others

```
M = [10,12,14] [kg/s] # flow of M entering the unit u  
for the state 1,2,3 measured in kg/s
```

=> be inspired by the examples reports

- **A : Analyse** : State the problem and Activate your knowledge
- **G :Generate** : Generate numbers by computer tools
 - Program the generation of numbers with comments and reporting
 - => duplicate and adapt the codes
 - => document and report your assumptions
 - => reference the data sources
 - Define the algorithms of the solving procedure
 - Structure => define functions and steps (data flow)
 - Solve
 - Extract numbers
- Programming language are needed
 - => Python or R : elementary levels is required for manipulating data
 - => AMPL is used because it allows you to “think optimisation”

- **A : Analyse** : State the problem and Activate your knowledge
- **G :Generate** : Generate numbers by computer tools
- **I : Interpret** the generated results
 - Transform numbers into engineering meaning :
 - Program the interpretation of the generated numbers
 - Graphs
 - Tables
 - Numbers
 - ⇒ we use Quarto : reproducible science & open source
use Jupyter / Python scripts
- Do not forget
 - Physical units
 - Axis titles
 - Captions
 - References

- **A : Analyse** : State the problem and Activate your knowledge
- **G : Generate** : Generate numbers by computer tools
- **I : Interpret** the generated results
- **R : Report** for the rest of the world and your peers (your boss)...
 - Make a summary of your main results
 - highlight what is important in your findings
 - Make a documentation useful to the others
 - Allow others to reproduce your work : assumptions, limits, ...
 - Add references

- **Using computer softwares to solve engineering problems**
 - Finding and accessing data
 - search, find and cite your data sources
 - Programming calculations and reports
 - Variables with an engineering meaning
 - Variable = name + description + value + [physical units]
 - e.g, TempFlow1=25 #[C] temperature of the flow 1
 - Reproducible science
 - Use Professional tools
 - Developed by others but used by you
 - Reporting
 - Extracting data
 - Presenting data (graphs, tables)
 - Programming reports
 - Project notebook
 - Documenting the calculations
 - Reviewing (do I understand what I'm writing)

EPFL Breacking the ICE

- On quarto, develop of book of CV of the CAS team
 - Develop the CV of one of your team mate
 - Your CV
 - Add a picture
 - Report your CO_2 impact (total over your life)
 - what were the emission on your birthday
 - what is the integral of your consumption based emissions as a mean citizen
 - What is the amount of money you should have put aside to clean your impact (social cost of CO_2)
 - References for the calculations