Problem 2: Advanced cooling

Aims of this exercise

1. Understand that writing cooling functions is a mess
2. Realize that KROME saves your day

- Homework: because 7 hours of school wasn't enough
(Anonymous)

Core of part1

Custom cooling functions go into the chemical network file

Recap for part1

- `http://kromepackage.org/bootcamp/exercises/day1_2.tar.gz`
- NOTE: network is provided in file `network_start`
- TODO: complete the network with custom cooling function
- NOTE: parts for custom cooling are in `code_block` file
- TODO: copy them into the network file
- TODO: add the tokens @something listed in the text
- `@var ntot = get_Hnuclei(n(:))` → `@var:ntot = ...`
- `@cooling`
- `@cooling_start @cooling_stop`

KROME Bootcamp 2014 - Writing the code

2 things worth knowing

1. loop on Tgas
2. for each temperature go to equilibrium

Recap for part1A

- TODO: write a loop on Tgas with log steps from $10 - 10^4$ K
- TODO: inside the loop initialize the species
- `krome_scale_Z` helps
- `python list_user_functions.py`
- NOTE: chemical equilibrium = running KROME at constant temperature for 10^8 yr
- TODO: add `call krome_thermo_OFF()`
- call KROME using `-useThermoToggle`

KROME Bootcamp 2014 - H_2 cooling

This is easy...

- `-cooling=H2` enables H_2 cooling
- do not forget `-noExample` option!

2 more things worth knowing

1. LAMDA is an atomic database (rotational)
2. file `tools/coolCO.dat` contains rotational data

Additional option to use

- `-coolFile=` enables cooling from external data

GOOD WORK!

`http://kromepackage.org/bootcamp/exercises/day1_2.tar.gz`