032CM - 2026

PROGRAMMING FOR COMPUTATIONAL CHEMISTRY

Functions in Python

Gianluca Levi

gianluca.levi@units.it, giale@hi.is

Office: Building C11, 3rd floor, Room 329

Fall 2026

Specific vs. general implementation

Midpoint method: $$\int_a^b f(x)\,dx \approx h\sum_{i=0}^{n-1} f(x_i) \qquad x_i = \left(a + \frac{h}{2}\right) + ih$$

Specific implementation (no functions)

Code designed for integral of one particular problem

- **Error-prone**: Formulating the general problem for our specific problem leads to mistakes
- **Repeated formula**: Integrand might need to be repeated in several places
- **Hard to change**: Integral of a different function involves many edits (and new errors)

Specific vs. general implementation

Midpoint method:

$$\int_a^b f(x)\,dx \approx h\sum_{i=0}^{n-1} f(x_i) \qquad x_i = \left(a + \frac{h}{2}\right) + ih$$

Specific implementation (no functions)

Code designed for integral of one particular problem

- **Error-prone**: Formulating the general problem for our specific problem leads to mistakes
- **Repeated formula**: Integrand might need to be repeated in several places
- **Hard to change**: Integral of a different function involves many edits (and new errors)

General implementation (using functions)

Algorithm implemented with **functions** that can be **reused** for multiple specific problems

- Requires some **abstract thinking**
- Code can be used **over and over again**
- Improve **portability** and to **avoid repeating lines** of code
- more **understandable** and **maintainable** code

General structure of a function

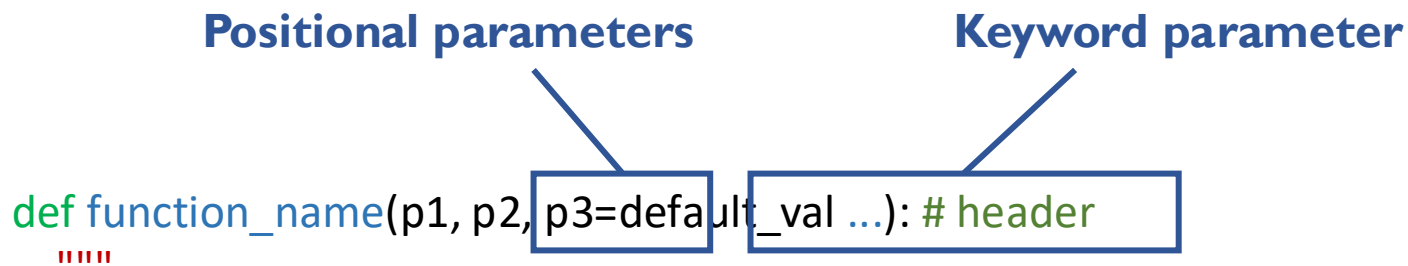
```
def function_name(p1, p2, p3=default_val ...): # header
    """
    This is a docstring
    """

    var_loc = p1 + +p2
    <expression>
    <expression>
        ...
        ...

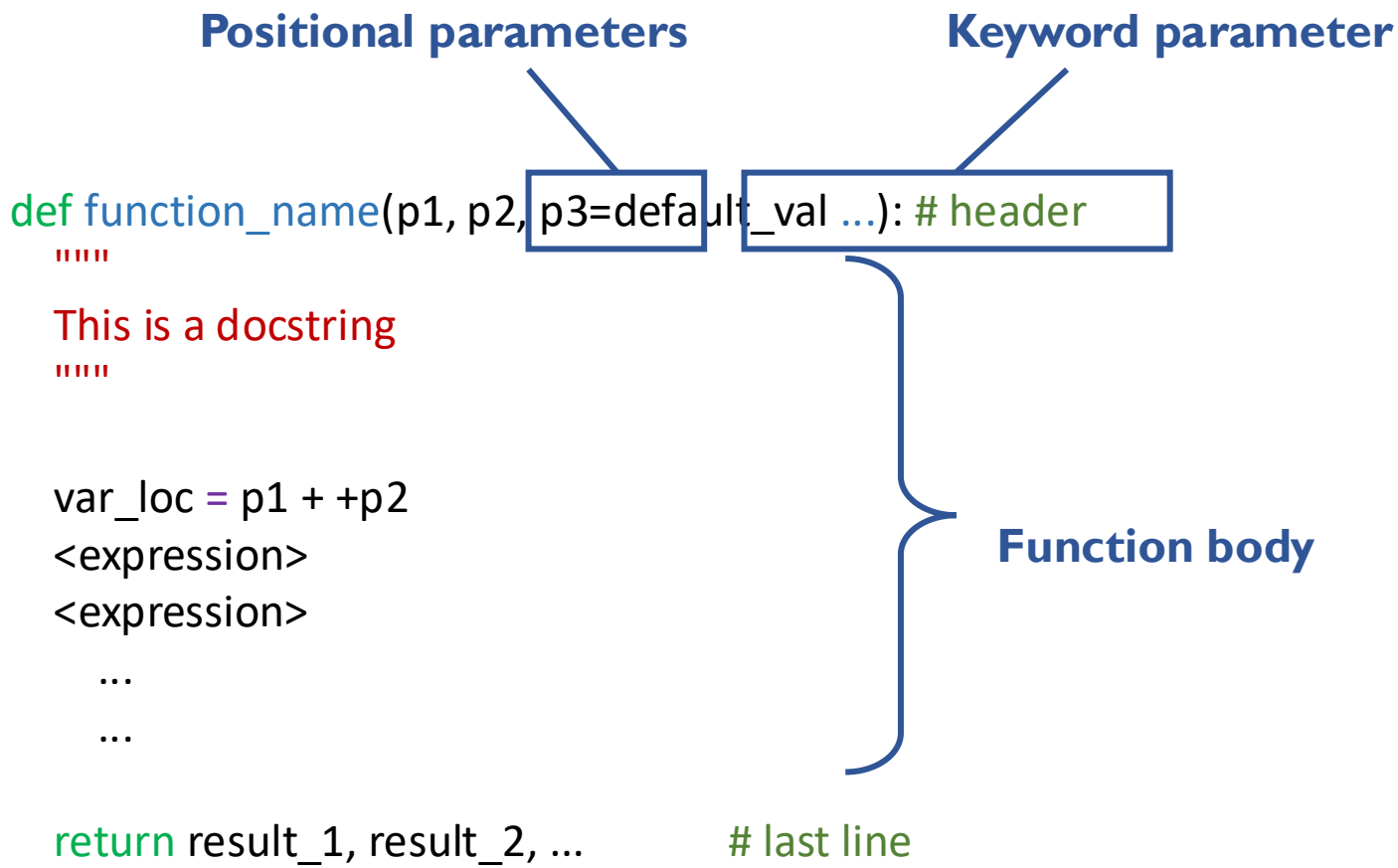
    return result_1, result_2, ...          # last line

var_glob = value
```

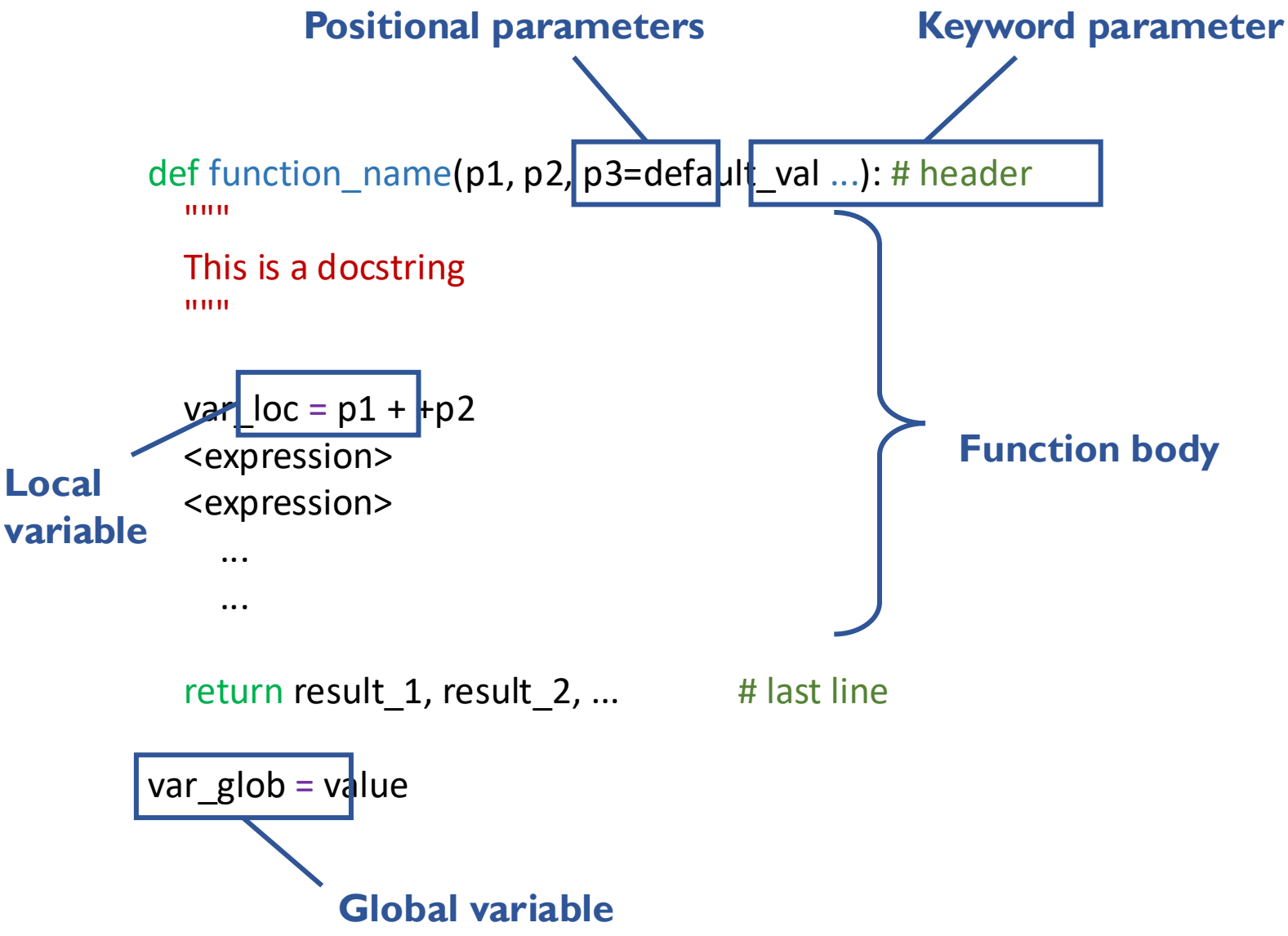
General structure of a function



General structure of a function



General structure of a function



Assignment 4

Problem 1

The Morse potential can be used to approximate the interatomic potential of a diatomic molecule and is defined as

$$V(r) = D_e \left[1 - e^{-\alpha(r-r_e)}\right]^2,$$

where D_e is the well depth, α controls the width of the potential, and r_e is the equilibrium bond length. The external force required to stretch the bond is

$$F_{\text{ext}}(r) = \frac{dV}{dr} = 2\alpha D_e e^{-\alpha(r-r_e)} \left[1 - e^{-\alpha(r-r_e)}\right].$$

Use the following parameters for the Morse potential of carbon monoxide (CO):

$$D_e = 11.22 \text{ eV}, \qquad r_e = 1.128 \text{ Å}.$$

Compare two values of the parameter α:[1]

$$\alpha_1 = 2.2994 \text{ Å}^{-1}, \qquad \alpha_2 = 1.0 \text{ Å}^{-1}.$$

(a) Create an array of 1000 equally spaced r values between 0.75 and 3.0 Å using `np.linspace()`. For each value of α, calculate arrays containing the Morse potential $V(r)$ and the corresponding external force $F_{\text{ext}}(r)$.

(b) Plot the two Morse potentials in the same figure, and the two external forces in a second figure. Label the axes with the appropriate units and include legends.

(c) Write a Python implementation of the midpoint method in vectorized form to integrate the external force.

(d) For each potential, compute the work required to stretch the bond from r_e to 2.0 Å:

$$W = \int_{r_e}^{2.0 \text{ Å}} F_{\text{ext}}(r)\, dr.$$

Use increasing numbers of subintervals $n = 2, 4, 8, 16, \ldots, 2^8$. The analytical value is

$$W_{\text{exact}} = V(2.0 \text{ Å}) - V(r_e).$$

In the same figure, plot $W(n) - W_{\text{exact}}$ as a function of n for the two cases.

(e) Compare the convergence of the two curves. How does increasing the number of subintervals affect the accuracy? Which potential requires a finer grid to reach the same absolute error?

Hints: The midpoint method for n equally spaced subintervals computes an integral according to

$$I \approx h \sum_{i=0}^{n-1} f(x_i), \qquad x_i = a + \left(i + \tfrac{1}{2}\right) h, \qquad h = \frac{b-a}{n}.$$

[1] The first value is taken from A. K. Roy, "Accurate ro-vibrational spectroscopy of diatomic molecules in a Morse oscillator potential," *Results in Physics* **3**, 103–108 (2013), doi:10.1016/j.rinp.2013.06.001.

Assignment 4

Problem 2

Load the data from `october_temperatures_trieste_reykjavik_1961_2025.csv` using `np.genfromtxt()`. The file is available in `/home/pcc/gianluca/assignment4/`. Each row contains one year and the mean temperature during October in Trieste and Reykjavík, in °C. The data cover 1961–2025 and are arranged chronologically.

(a) Plot the mean October temperature as a function of year for both cities. Label the axes, include temperature units, and add a legend.

(b) For each city, calculate the average of the October temperatures over 1961–1970 and 2016–2025, the first and last ten years of the dataset. Print the two averages and their difference in °C.

(c) Identify the warmest and coldest Octobers in the dataset for each city. Print the year and the corresponding mean October temperature in each case.

(d) Create a bar plot showing the four averages calculated in (b). Group the bars by city, with two bars for each city representing the two periods. Label the axes and include a legend identifying the periods.

Hint: You can read the file and skip the header line using the command:

```
data = np.genfromtxt('filename.csv', delimiter=',', skip_header=1)
```

Then access the columns using:

`years = data[:, 0]`, `trieste = data[:, 1]`, `reykjavik = data[:, 2]`.

Use `np.mean()`, `np.argmax()`, and `np.argmin()` to compute averages and find extreme values. `np.argmax()` and `np.argmin()` return array indices, use these indices to obtain the corresponding years.

Assignment 4

Problem 3

The potential energy of a carbon dioxide (CO_2) molecule confined to one dimension along the molecular axis can be approximated as the sum of two Morse potentials, one for each C–O bond:

$$V(r_{CO_1}, r_{CO_2}) = V_{CO_1}(r_{CO_1}) + V_{CO_2}(r_{CO_2}),$$

where each Morse potential is given by

$$V_{CO}(r) = D_{CO}\left(1 - e^{-\alpha_{CO}(r_{CO} - r_e)}\right)^2.$$

The parameters can be chosen as: $D_{CO} = 7.65$ eV, $r_e = 1.162$ Å, $\alpha_{CO} = 2.5$ Å^{-1}.

(a) Define a Python function that calculates the Morse potential for a single bond and a Python function that sums two Morse potentials to give the potential energy of the CO_2 molecule:

$$E_{CO_2}(r_1, r_2) = D_{CO}\left(1 - e^{-\alpha_{CO}(r_1 - r_{CO})}\right)^2 + D_{CO}\left(1 - e^{-\alpha_{CO}(r_2 - r_{CO})}\right)^2.$$

(b) Generate a grid of r_1 and r_2 values from 0.75 Å and 3.0 Å using `np.linspace()` and `np.meshgrid()` and calculate the CO_2 potential energy over this grid.

(c) Create a contour plot of the potential energy surface using `contourf()`. Label the axes appropriately, add a colorbar, and give the plot a title.

(d) Can you identify the position of the minimum of the potential energy surface? In which direction does the energy rise more steeply, moving one bond while the other is fixed or stretching both bonds simultaneously?