Modifying a Program

Programming with Julia
for Digital Humanities & Non-technical Sciences

Francisco Coelho, Miguel Avillez, André Fernandes

High Performance Computing Chair

February 9, 2024

Catch-up

- Questions & Answers.
- Hand-on Solutions.

HPC CHAIR
High Performance Computing

HPC CHAIR
High Performance Computing

Modifying Programs 1/2

You may already know something about programming.

Now our job is to modify existing programs.

. . . and see what happens.

Later We'll compose programs from nothing.

HPC CHAIR
High Performance Computing

Modifying Programs

Visit the `modify.ipynb` notebook.

- There you'll find a complete program.
- Your main task is to read the code and figure out what it does.
- You'll do small modifications and observe the effects.

The goal is to draw out insights about key different elements in a program.

HPC CHAIR
High Performance Computing

Code and Data

Some modifications damage the program, others don't.

Code Some parts are **instructions**, operations to be done.

Data Other parts are **values**; A text or a number.

- If we change one value, the program still runs.
- If we change an instruction, we are very likely to break the program.

Telling instructions from values hints the effect of modifications.

HPC CHAIR
High Performance Computing

Errors and errors 1/2

When developing a computer program, the goal is always twofold:

- Develop a **working program**;
- Have the program **work in the intended way**.

HPC CHAIR
High Performance Computing

Errors and errors 2/2

Learning from Errors

Visit the `errors.ipynb` notebook.

- Error messages help programmers to fix their programs.
- But not all errors are created equal.
- A *valid* program might still have errors.

HPC CHAIR
High Performance Computing

Syntax and Semantics

We found two types of errors:

Syntax errors result from *invalid* programs; error messages make them easy to fix.

Semantic errors when a *working* program is not working in the intended way.

HPC CHAIR
High Performance Computing

HPC CHAIR
High Performance Computing

Double, double

Double, double

Visit the `double.ipynb` notebook.

- This is the first visit to this notebook. Others will follow soon.
- For now you just need to type some code and become aware that code has structure, denoted by the indentation.

HPC CHAIR
High Performance Computing

Function Template

Function definitions are made using a particular **template**:

```
function ____(____)
    ____
end
```

- The first blank contains the function name.
- The second blank contains zero or more arguments.
- The third blank, which may contain one or more lines of code, is the function body.

Punctuation (e.g. parentheses) and **indentation** are not decorative.

HPC CHAIR
High Performance Computing

Try out the Function

Double, double

Visit the `double.ipynb` notebook *again*.

- To "call" (or "invoke") the function from the previous visit.
- Notice code abstraction: The same block of code reused to perform many computations.
- Formulate expectations about the behaviour of your code.

HPC CHAIR
High Performance Computing

Mental Model

People can operate computers only because they have **a mental model** of how they work.

- To a person with no expectation about `2 + 2` the output might be `17` or `"Hello world."` or a video playing.
- If any result seems equally likely, the system would be inscrutable.

It is extremely helpful to form an **expectation about what will happen** when you invoke a function or run a program.

HPC CHAIR
High Performance Computing

Modifying "Double, double"

Double, double

Visit the `double.ipynb` notebook **a third time**.

- Now you are asked to modify the `double()` function.
- This is an *open* exercise: there are no right or wrong answers. Just follow your will and some hints.

HPC CHAIR
High Performance Computing

Describe the Function

We need to be discerning about how functions work.

- What is the function's *interface*? It's *inputs* and *result*?
- Are the arguments *modified*?

Describe the Function

Visit the `describe.ipynb` notebook to describe the `double()` function.

- A *description* enables expectations.
- Variables in the function are *scoped*.

HPC CHAIR
High Performance Computing

HPC CHAIR
High Performance Computing

Concepts

Data and Code (values and instructions) are different.

Syntax Errors help us attain formal validity.

Valid and Intentional (syntax and semantics) are different.

- **Variables** hold values.
- **Templates** structure code.
- **Functions** bundle code.
- **Interfaces** are part of functions.
- **Scope** limits variables in functions.

HPC CHAIR
High Performance Computing

HPC CHAIR
High Performance Computing

- Notebooks with **conventional starter programs** are presented to type in and run programs.
- The notebooks explain **step-by-step** how each program works and uses these programs to introduce self-sufficient programming.

HPC CHAIR
High Performance Computing

Greetings

Making the computer "speak", and say hello, is often the first task done when learning to program in an imperative programming language — such as `Julia`.

Example/Exercise

Visit the `greetings.ipynb` notebook to review simple string operations and the process of code abstraction into a function.

HPC CHAIR
High Performance Computing

Temperature Conversion

- Converting between Fahrenheit and Celsius is often the first mathematical function implemented by new programmers.
- **A conversion program demonstrates how computation is general**—how it can work on arbitrary values.

Temperatures

Visit the `temperatures.ipynb` notebook to create conversion functions between Fahrenheit and Celsius and practice programming numeric calculus.

HPC CHAIR
High Performance Computing

HPC CHAIR
High Performance Computing

Accumulate

- Another "modify a program" exercise, to explore *iteration*.
- *Iteration* will be covered in more depth in the next session.

Accumulate

This exercise asks you to adapt the iteration, initialization and accumulation ideas of the `double()` and `mean()` functions to solve other problems.

Visit the `accumulate.ipynb` notebook.

HPC CHAIR
High Performance Computing

Unit Conversions

- Working with large quantities of related functions asks for related implementations.

Unit Conversions

The `conversions.ipynb` notebook asks you to adapt the structure and calculations of the *Temperature Conversion* exercise to other measurements.

HPC CHAIR
High Performance Computing