Programming Environments

Presenter: Steve Baskauf
steve.baskauf@vanderbilt.edu

DISC DIGITAL SCHOLARSHIP AND COMMUNICATIONS

Jean & Alexander Heard LIBRARIES

CodeGraf landing page

- vanderbi.lt/codegraf

What is an environment?

DISC DIGITAL SCHOLARSHIP AND COMMUNICATIONS

Jean & Alexander Heard LIBRARIES

Coding environment

- The definition of "environment" is a bit murky
- We can consider an environment to include:
  - the value of defined variables
  - functions available to be used in our code
  - knowledge about position in file directory structure and other computer-wide parameters

Accessing via the shell

- Python example
- R example

Integrated development environment (IDE)

Jean & Alexander Heard
LIBRARIES

What is an integrated development environment (IDE)?

- An IDE is a graphical user interface (GUI) for developing code
- An IDE includes:
  - a code editor
  - a shell
- An IDE might include:
  - tools for examining the environment
  - formatting help and syntax checking
  - mechanisms for debugging code
  - a package manager

Thonny example

- Thonny is a simple Python IDE

Spyder IDE for Python

DISC DIGITAL SCHOLARSHIP AND COMMUNICATIONS

Jean & Alexander Heard LIBRARIES

RStudio IDE for R

Jean & Alexander Heard
LIBRARIES

Literate programming with Jupyter notebooks

DISC DIGITAL SCHOLARSHIP AND COMMUNICATIONS

Jean & Alexander Heard LIBRARIES

Literate programming

- Programming paradigm for making code **understandable to humans**
- Mix text, images, links with code.
- Implementable in a primitive fashion with **comments** (#)
- Implementable in a robust way with **Jupyter notebooks** and R Markdown

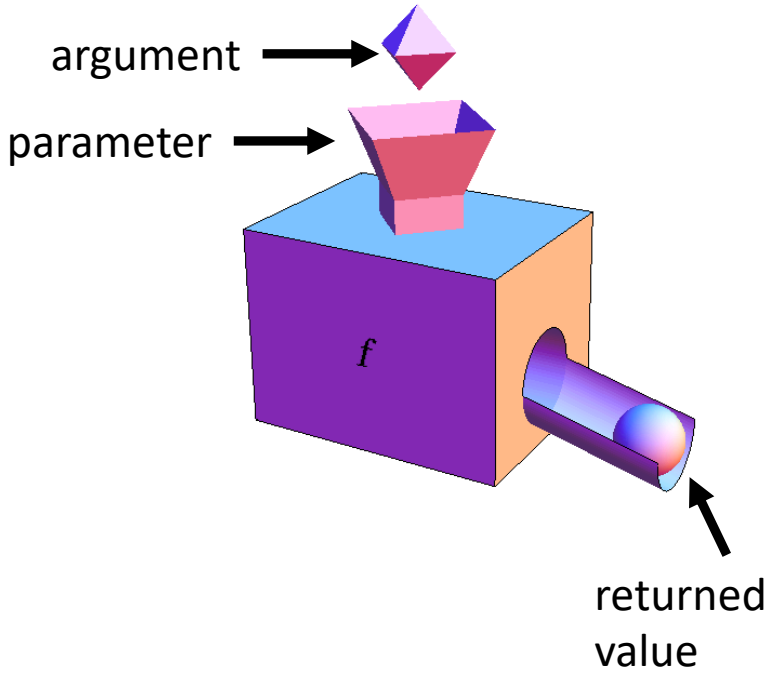
Example: Jupyter notebooks

- Formerly known as "iPython notebooks" (.ipynb file extension)
- Now usable with **Python**, **R**, and other programming languages
- Runnable in a browser when connected to a server
- Viewable in GitHub (but not runnable)

Functions

Jean & Alexander Heard
LIBRARIES

Functions



- A function defines a block of code.
- We pass arguments into functions:
  - functionName(argument1, argument2, ...)
- It's good to name functions by what they do. Example:

```
my_latte = make_latte(beans, milk, water)
```

- Functions can be:
  - built-in
  - defined by you in your code
  - defined by somebody else in a module

Using functions

- Use a function whenever code needs to be **repeated** more than once.
- It isn't necessary to understand how a function works, just:
  - understand what needs to **go in** (arguments), if anything
  - understand what to expect will **come out** (return value), if anything
- Functions leverage the power of **open source** coding
  - We can use the code of others
  - We can make our code available to others.
- Functions keep the language lean by **importing** some code only when its needed

Function example

- We have seen built-in functions like **`input()`** and **`print()`**.
- User-defined example in script: **`reverse_names()`**

Libraries

DISC DIGITAL SCHOLARSHIP AND COMMUNICATIONS

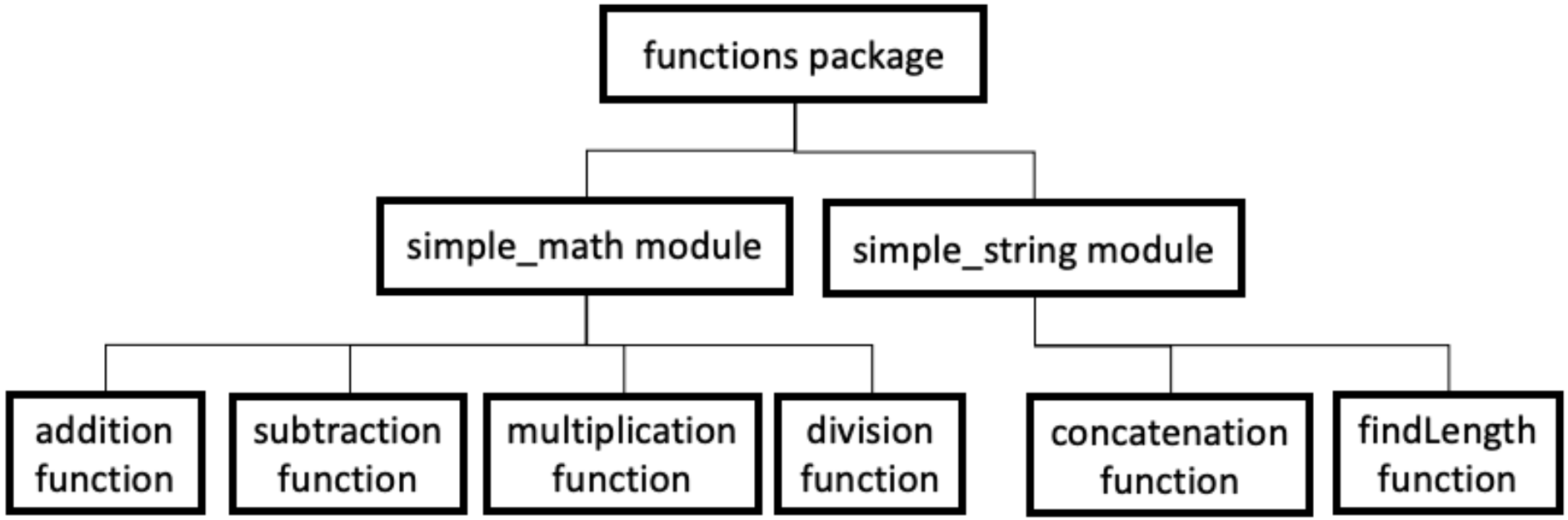
Jean & Alexander Heard LIBRARIES

Importing functions

- Reusable code stored in a separate file
- Code not available in environment unless **imported**
- Some functions are part of the language's **standard library** and can be imported with no additional work
- Some functions aren't included in the standard library
  - must be **downloaded** as a package
  - must be **installed** before they are used
- Platforms (CLI or GUI) usually have a **package manager** to help

Organization of imported functions

- Functions can be organized in a **hierarchical** way
- In Python:
  - related **functions** are grouped in **modules**
  - related **modules** are grouped in **packages**



Import example

- In Python:
  - math module
  - datetime package

Package managers

Jean & Alexander Heard
LIBRARIES

What are package managers?

- Package managers retrieve packages from well-known **repositories**
- They keep track of where the extracted libraries are stored in the computer
- They make the storage information available to the software environment so functions can be located.
- If one package has a dependency on another package, the package manager can automatically retrieve the other package.

How do you access a package manager?

- Python CLI package managers check the Python Package Index (**PyPI**):
  - **PIP** (Preferred Installer Program)
  - **Conda** (Anaconda package manager)
- R packages managed centrally through Comprehensive R Archive Network (**CRAN**) and the built-in **`install.packages()`** function
- Package managers may be built into IDEs.

Separation of environments

- You can keep environments separate if one installed library conflicts with another (virtual environments in Python)
- Installing a package in one application (e.g. Thonny) won't necessarily make it available in another (e.g. Spyder).

Remote Support for Teaching and Research Needs

- Access to digital collections 24/7
- Skype consultations with your subject librarian
- Ask a Librarian: an easy way to submit a question via email
- Live chat available from the Library home page

Jean & Alexander Heard
LIBRARIES

NEED HELP? ASK A LIBRARIAN!

https://www.library.vanderbilt.edu/ask-librarian.php

Jean & Alexander Heard
LIBRARIES