the Jupyter notebook

1. the Jupyter notebook
   - documenting your work
   - the four major operations
   - formatting with restructured text
2. Exact, Symbolic, and Approximate Computations
   - three types of computations
   - using the working precision

MCS 320 Lecture 2
Introduction to Symbolic Computation
Jan Verschelde, 10 June 2024

the Jupyter notebook

the Jupyter notebook

The Jupyter notebook is a web based interface for computations.

Jupyter stands for Julia, Python, R, and many others.

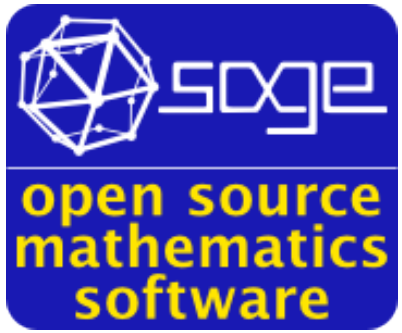
- Input cells can contain code or text (markdown, LaTeX).
- Output cells hold numerical data, symbolic expressions, or plots.
- We use Jupyter to document our computations.

Jupyter is the default interface for SageMath.

A well structured notebook runs as a computer program.

All your work must be submitted in a Jupyter notebook.

the Jupyter project is separate from SageMath



T. Kluyver, B. Ragan-Kelley, F. Pérez, B. Granger, M. Bussonnier, J. Frederic, K. Kelley, J. Hamrick, J. Grout, S. Corlay, P. Ivanov, D. Avila, S. Abdalla, C. Willing, and Jupyter Development Team: **Jupyter Notebooks — a publishing format for reproducible computational workflows.**
In F. Loizides and B. Schmidt, editors, *Positioning and Power in Academic Publishing: Players, Agents, and Agendas*, pages 87–90. IOS Press, 2016.

Visit `https://jupyter.org` for more information.

the Jupyter notebook

the four major operations

Making, downloading, uploading, and printing:

1. When making a Jupyter notebook, we select a kernel.
2. Download (or saving) a Jupyter notebook to your disk.
   Be aware that many browsers will add the `.txt` extension to the `.ipynb` file.
3. Upload a notebook into a SageMath session.
   The end of the `.ipynb` file contains meta information, about the kernel and the version of the notebook.
4. Printing to a pdf file.
   This operation is useful to generate a technical report.

the Jupyter notebook

formatting with restructured text

The Markdown cells occur in two types:

1. headers: section (`#`), subsection (`##`), subsubsection (`###`),
2. paragraphs of text.

To edit a formatted Markdown cell, convert to `Raw NBConvert`.

To learn more, visit
`https://docutils.sourceforge.io/rst.html`

the Jupyter notebook

1. the Jupyter notebook
   - documenting your work
   - the four major operations
   - formatting with restructured text

2. Exact, Symbolic, and Approximate Computations
   - three types of computations
   - using the working precision

Exact, Symbolic, and Approximate Computations

We distinguish between three types of computations:

1. An *exact* computation is free from errors.

   Example: $\frac{1}{2} + \frac{1}{4} = \frac{3}{4}$.

2. A *symbolic* computation operates on symbols.

   Example: $\sin(\pi/3) = \sqrt{3}/2$.

   Note $\sqrt{3}$ is the symbol for the unevaluated function `sqrt(3)`.

3. An *approximate* computation works in limited precision.

   Example: $\sin(3.1415/3) = 0.8660$.

the Jupyter notebook

1. the Jupyter notebook
   - documenting your work
   - the four major operations
   - formatting with restructured text

2. Exact, Symbolic, and Approximate Computations
   - three types of computations
   - using the working precision

Using the Working Precision

Consider the following computation: `Pi10 = pi.n(digits=10)`

The `Pi10` evaluates to `3.141592654`, an approximation for π, accurate with 10 decimal places.

The statement `delta = pi - Pi10`
shows `pi - 3.141592654`, which is a symbolic expression.

Now consider the following two statements:

1. `delta.n(digits=10)` evaluates to `0.0000000000`
2. `delta.n(digits=11)` evaluates to `3.6379788071e-12`

Which outcome is correct? Explain.