Getting Started with SageMath

Getting Connected to the SDSU VERNE cluster

- Login to VERNE using your usual SDSU credentials at jupyterhub.sdsu.edu

Learning About the Environment

- To get started, see the Jupyter notebook tutorial
  https://github.com/SDSU-Research-CI/ic-intro
  The first two pages give general information about getting started and describe the Jupyter Notebook. The third page explains Markdown, which is a way to gussy up your notebook. One of the key things is one the first page. It tells you how to login to VERNE (which you did in the first step above). Then, how to launch a *terminal* window and issue the command
  git clone https://github.com/SDSU-Research-CI/ic-intro.git
  This creates a copy of the the Jupyter notebook tutorial in your account on VERNE.

- This page
  https://gist.github.com/discdiver/9e00618756d120a8c9fa344ac1c375ac
  has more information about Jupyter keyboard shortcuts. Refer to it as you get more experience.

- This page has a Markdown cheatsheet as well as links to other tutorials about Jupyter
  https://www.ibm.com/docs/en/watson-studio-local/1.2.3?topic=notebooks-markdown-jupyter-
  Again, refer to it after you have some experience.

- Sage tools: Here are some brief thoughts, but read the tutorial (see below) to get started
  - <shift> <enter> to implement a command
  - <tab> to get information about an object or a command,
  - Use ? at the end of a function name to get information about the function.
  - Sage defines these ZZ, QQ, RR, CC as what you think they should be.
  - Coercion is a key idea. RR(7/3) converts an element of QQ to a real number, which means it is no longer exact, since the decimal is repeating.
  - These have Boolean output: ==, <, >, and, or, in, as do several functions that start with is_ like is_prime .

Work through the SDSU SageMath tutorial

Launch a Jupyter notebook with Sage 10.0 Please get started with this tutorial
https://mosullivan.sdsu.edu/Teaching/sdsu-sage-tutorial/about.html
In particular, work through

- About This Tutorial
  - How to use this tutorial
  - Getting Started
- SageMath as Calculator
  - Basic Arithmetic and Functions
- Programming in SageMath
  - SageMath objects
  - Programming Tools (do this a bit later)
- Mathematical Structures
  - Integers mod n
  - Groups: Symmetric and Abelian
  - Fields
  - Rings

Some exercises

Try out these functions on an integer (element of ZZ) and then on $k[x]$ for k a field (perhaps `FiniteField(p)` for p a prime). (Example `divisors(86)`)

- `divisors`
- `factor`
- `prime_divisors`
- `radical`
- `sqrt`
- `gcd`
- `lcm`
- `xgcd`
- `abs`
- `next_prime` `next_prime_power`
- `previous_prime`
- `previous_prime_power`

These are *methods* that you can apply to an integer input this `a=75`, then do

- `a.binary()`
- `a.oct()`
- `a.hex()`
- `a.quo_rem()`
- `a.radical()`
- `a.squarefree_part()`

Functions for rational numbers and real numbers

- `ceil`
- `floor`
- `round`
- `random`
- `log`
- `exp`

For practice programming, write a function to compute the gcd of two integers. Test it on two polynomials in $\mathbb{F}_p[x]$ for some prime p. Write a function to compute the linear combination that gives the gcd (this is what the function xgcd does, but try writing your own). Add print statements to have it show each step.

Test some of the `is_` functions on rings, groups or elements of them type `is_ <tab>` to get a list of possible functions.

- `is_unit`
- `is_nilpotent`
- `is_zerodivisor`
- If R is some ring (or field or...) `R.random_element()}` gives a "random" element. Try `R.random_element?` for information.