Random Magma knowledge

This is a compilation of some useful Magma tricks that I have learned through the years. Thank you to all the wonderful people who have shared their knowledge with me: Eran Assaf, Edgar Costa, Sachi Hashimoto, Avi Kulkarni, Sam Schiavone, Pim Spelier, Drew Sutherland, and John Voight.

- Useful links: Documentation and general examples.
- You can write the first letter of a function and tab complete twice to get a list of possible functions.
- To kill a process, use `control` + `c`. To kill Magma, do this twice.
- To ignore `>`, use `SetIgnorePrompt(true);`. This is very helpful when you are copying code from the terminal.
- `$1` denotes the last printed result (you can also call `$2` and `$3`).
- The rational numbers are not a number field in Magma. You can instead call `RationalsAsNumberField();`.
- When you are debugging functions, you can `SetDebugOnError(true);`. This will give you access to the "inside" of your function, up to where Magma got stuck. You print things by `p whateverYouWant`. Use `q` to quit back to the Magma terminal. You can see all the functions that were used in your computation using `bt` (shows you the "frames"). Go to a specific frame by writing `f theNumberYouWant`. Warning: do not use `;`.
- In your home directory you can create a `file.Magmarc`, if does not exist, to have certain commands to run on start. For example, you can set `QQ := Rationals();`
- Use `control` + `e` to get to the last character of a line in the terminal. Use `control` + `a` for the first one. Do `control` + `k` to delete the line. You can also find other combinations here.
- Do `%p` to print all the Magma session.
- You can search only for signatures without inheritance:

  `ListSignatures(ModFrmHilElt : Isa := false);`. Moreover, you can also just look for functions where your type is an argument or a return values (very useful when you try to find a function producing the type that another function needs...):

  `ListSignatures(ModFrmHilElt : Search := "ReturnValues", Isa := false);`.
- Magma is fussy about types (integers and rational number are different types, for example) and elements may need to be explicitly coerced to have the type you want. To change the rational number (which will simply produce when printed) into the integer , use `Integers()!r`.

- The function `Discriminant(K)` returns the discriminant of the polynomial used to define, not the discriminant of K. Use `Discriminant(RingOfIntegers(K))` to get the discriminant of K.
- Use for verbose: `vprintf VerboseLevel : "whatever you want to print";`

Do you have more random knowledge? Please email it to me and I will add it to this list!