

Math 724
Fall 2013

Here is an easy Sage routine to calculate factorials recursively.

```
def facto(n):  
    if n == 0:  
        answer = 1  
    else:  
        answer = n * facto(n-1)  
    return answer
```

Some notes:

- To start using Sage quickly, try the Sage Cloud at <https://cloud.sagemath.com/>.
- Sage does have a built-in `factorial` function; this is intended as a quick example.
- Sage cares about indentation. If I wanted Sage to do something additional in the `else` case, it would need to be indented eight space (same as `answer = n * facto(n-1)`). Indenting only four spaces (same as `return answer`) would cause the new line of code to be executed for all n . Indenting a different number of spaces would probably cause a syntax error.
- Note that `=` means assignment and `==` means equality checking.
- If you know what memoizing is, you know that the code above is theoretically inefficient and could be improved by memoizing. However, for the purpose of computing small examples in Math 724, don't worry about it.
- How did I get LaTeX to pay attention to line breaks and indentation? With the `verbatim` environment. Compare the [source .tex file](#) to the PDF document you are now reading.