Sage Introduction

The sage server is located at `https://math-158.unl.edu:8433`
Create a new account using your *University ID* (for example, "s-jdoe7").
Remember your password!
By logging in, you are agreeing to not abuse the system and to abide by University guidelines.

There is a public Sage server located at `http://www.sagenb.org`. Note that accounts on this server are reset every few hours, so download your work to save it!

```
%hide
%html
Click "New worksheet"
```

Click "New worksheet"

Click on "Help" at the top right---this gives an overview of the features of the Sage worksheet. There is also a tutorial; the sections on basic algebra, calculus, and linear algebra are useful for us.

Basic Arithmetic

```
100*(32-5)/5
```

```
540
```

```
# Sage has support for exact arithmetic with large
integers and rationals.
print 2^100
print 2^100/factorial(30)
```

```
1267650600228229401496703205376
188894659314785808547843952575621190533915703125
```

```
# Evaluating to a floating point number
print n(pi)
print n(pi,200)  # 200 bits of floating point precision
```

```
3.14159265358979
3.14159265358979323846264338327950288419716939937510582
```

```
# Defining a function:
x,y=var('x,y')
f=sin(x)*y^2
g(x,y)=f
```

```
f
```

```
    sin(x)*y^2
```

```
g
```

```
    (x, y) |--> sin(x)*y^2
```

```
f.derivative(x)
```

```
    cos(x)*y^2
```

```
fgrad=f.gradient()
fgrad
```

```
    (cos(x)*y^2, 2*sin(x)*y)
```

```
solve([fgrad[0]==0,fgrad[1]==0],x,y)
```

```
    [[x == r1, y == 0]]
```

Linear Algebra

```
A=matrix(RDF,3,2,[1,2,3,4,5,6])  # RDF is the "real
double field", ie, floating point numbers
A
```

```
    [1.0 2.0]
    [3.0 4.0]
    [5.0 6.0]
```

```
identity_matrix(5)
```

```
    [1 0 0 0 0]
    [0 1 0 0 0]
    [0 0 1 0 0]
    [0 0 0 1 0]
    [0 0 0 0 1]
```

```
A=matrix(2,2,[1,2,3,4]); A
```

```
    [1 2]
    [3 4]
```

```
A[1,0]  # all indices are zero-indexed!
```

```
    3
```

```
B=A.inverse(); B
```

```
    [  -2    1]
    [ 3/2 -1/2]
```

```
A*B
```

```
[1 0]
[0 1]
```

```
A*B==identity_matrix(2)
```

```
True
```

```
A.transpose()
```

```
[1 3]
[2 4]
```

```
A.determinant()
```

```
-2
```

```
v=vector([1,2]); v
```

```
(1, 2)
```

```
A*v
```

```
(5, 11)
```

Programming

```
# Sage programming is just Python
# New lines or semicolons separate commands.
# Level of indentation indicates blocks
```

```
# Compute the sum of the numbers from 1 to 10
s=0
for i in range(1,11):  # Note the syntax on range
 s=s+i
 print "i=",i,"s=",s
print s
```

```
i= 1 s= 1
i= 2 s= 3
i= 3 s= 6
i= 4 s= 10
i= 5 s= 15
i= 6 s= 21
i= 7 s= 28
i= 8 s= 36
i= 9 s= 45
```

```
    i= 10 s= 55
    55
```

```
# Defining a function; ie, a "programming" function
def my_factorial(i):
 if i==1:
   return 1
 else:
   return i*my_factorial(i-1)
```

```
my_factorial(3)
```

```
    6
```

```
for i in range(1,10):
 print
i,factorial(i),my_factorial(i),my_factorial(i)==factorial(
```

```
    1 1 1 True
    2 2 2 True
    3 6 6 True
    4 24 24 True
    5 120 120 True
    6 720 720 True
    7 5040 5040 True
    8 40320 40320 True
    9 362880 362880 True
```

```
```