2014-04-04.sagews

April 4, 2014

Contents

1 Math 480b Sage Course **1**
- 1.1 April 4, 2014 . . . 1
- 1.2 Python - Control Structures: if/then/else . . . 3
- 1.3 Python - Looping while, for, range . . . 3
- 1.4 Python - Data types str, int and long, float, list, tuple, dict, etc. . . . 5
- 1.5 Python - Functions (again) . . . 5
- 1.6 Python - Variables are references ! . . . 5
- 1.7 Putting code in files/modules and loading it . . . 5

1 Math 480b Sage Course

1.1 April 4, 2014

Screencast: `http://youtu.be/T7aXPTQx3hI`

Plan

- Questions?
- Python
  - control structures: if/then/else
  - looping: while, for, range
  - data types: string, integer, list, tuple, dict
  - functions
  - Python variables are references
  - putting code in files/modules and loading them
- Homework 2

```
R.<x> = QQ[]
R
```

```
Univariate Polynomial Ring in x over Rational Field
```

```
S.<y> = PolynomialRing(RR)
S
```

```
Univariate Polynomial Ring in y over Real Field with 53 bits of precision
```

```
3/5*y + y^5
```
```
y^5 + 0.600000000000000*y
```

```
f = (x^3 - 1)*(x-6/7)^2*(x-5)
f
```
```
x^6 - 47/7*x^5 + 456/49*x^4 - 229/49*x^3 + 47/7*x^2 - 456/49*x + 180/49
```

```
v = f.roots()
type(v)
```
```
<type 'list'>
```

```
v[0]
```
```
(5, 1)
```

```
v[0][0]
```
```
5
```

```
v
```
```
[(5, 1), (1, 1), (6/7, 2)]
```

```
for r in v:
 print "hi there"
 print r[0]

print "Hmm. I wonder what r is?"
print r
```
```
hi there
5
hi there
1
hi there
6/7
Hmm. I wonder what r is?
(6/7, 2)
```

```
5/4
```
```
5/4
```

```
5//4
```
```
1
```

```
f
```
```
x^6 - 47/7*x^5 + 456/49*x^4 - 229/49*x^3 + 47/7*x^2 - 456/49*x + 180/49
```

```
CC
```
```
Complex Field with 53 bits of precision
```

```
ComplexField(200)
```
```
Complex Field with 200 bits of precision
```

```
f.roots(ring=CC)
```
```
[(1.00000000000000, 1), (5.00000000000000, 1), (-0.500000000000000 - 0.866025403784439*I,
```

```
1), (-0.500000000000000 + 0.866025403784439*I, 1), (0.857142857142856 -
1.70882477893134e-8*I, 1), (0.857142857142856 + 1.70882477893134e-8*I, 1)]
```

```
a = 5
b = 7
a+b
3
```

```
b
```

```
Error in lines 1-1
Traceback (most recent call last):
  File "/projects/edf7b34d-8ef9-49ad-b83f-8fa4cde53380/.sagemathcloud/sage_server.py",
line 733, in execute
    exec compile(block+'\n', '', 'single') in namespace, locals
  File "", line 1, in <module>
NameError: name 'b' is not defined
```

```
def f(n,m):
    return n+m
```

```
f(2,5)
```

```
10
```

1.2 Python - Control Structures: if/then/else

```
# come up with examples here
```

```
if "":

if "false":
```

```
if 2 == 1+10:
    print "good"

elif 3 == 1+2:
    print "yes"
else:
    print "very bad"
```

```
very bad
```

```
# like "?:"

a = "good" if 2==1+1 else "bad"
a
```

```
'good'
```

```
a = "good" if 2==5+1 else "bad"
a
```

```
'bad'
```

1.3 Python - Looping while, for, range

```
range(5)
```
```
[0, 1, 2, 3, 4]
```

```
for n in range(5):
    print n
```
```
0
1
2
3
4
```

```
for n in range(10^8):   # scary
    print n
    if n > 5:
        break
```

```
for n in xrange(10^8):   # good
    print n
    if n > 5:
        break
```
```
0
1
2
3
4
5
6
```

```
v = ['william', 'simon', pi, e, sin(x)]
v
```
```
['william', 'simon', pi, e, sin(x)]
```

```
for z in v:
    print 2*z, z + z
```
```
williamwilliam williamwilliam
simonsimon simonsimon
2*pi 2*pi
2*e 2*e
2*sin(x) 2*sin(x)
```

```
a = "simon"; b = "william"
a + b
b + a
```
```
'simonwilliam'
'williamsimon'
```

```
a*b   # this should be concat; sigh...
```
```
Error in lines 1-1
Traceback (most recent call last):
```

```
  File "/projects/edf7b34d-8ef9-49ad-b83f-8fa4cde53380/.sagemathcloud/sage_server.py",
line 733, in execute
    exec compile(block+'\n', '', 'single') in namespace, locals
  File "", line 1, in <module>
TypeError: can't multiply sequence by non-int of type 'str'
```

```
a = 10
while a > 3:
    print a^5
    a = a - 2
```

```
100000
32768
7776
1024
```

1.4 Python - Data types str, int and long, float, list, tuple, dict, etc.

1.5 Python - Functions (again)

1.6 Python - Variables are references !

There is a major difference between how the math-oriented programming languages (Matlab, Pari, Maple, etc.) work, and how most modern mainstream programming languages, such as Python, work.

```
octave = Octave()
```

```
%octave
v = [3,4]; w = v; w(1)=2014
```

```
w =

 2014 4
```

```
%octave
v
```

```
v =

 3 4
```

```
#python
v = [3,4]; w = v; w[0]=2014
w
```

```
[2014, 4]
```

```
v
```

```
[2014, 4]
```

1.7 Putting code in files/modules and loading it

WARNING: The Sage preparser!

- import
- reload
-

```
%default_mode python
```

```
2/3
```

```
0
```

```
3^8
```

```
11
```

```
%default_mode sage
```

```
2/3
```

```
2/3
```

```
%python
2/3
```

```
0
```