2014-04-16.sagews

April 16, 2014

Contents

1 Math 480b Sage Course

1.1 April 16, 2014

Screencast: `http://youtu.be/Z5lye6AIC5Y` (part 1) and `http://youtu.be/dp-ovN6J-_M` (part 2)

Plan

- Questions?
- Homework reminder: everything due by Friday at 6pm. If you have permissions issues, restart your project server.
- Python classes: creating your own data types:
  - rational numbers
  - the field of all rational numbers
- Python decorators
- Example python decorator: @interact
- Something different: some 2D graphics in Sage

1.2 Python class: a rational number

```
class RationalNumber:
    def __init__(self, n, d):
        self.n = n
        self.d = d
```

```
t = RationalNumber(3, 2)
```

```
type(t)
```

```
<type 'instance'>
```

```
t
```

```
<__builtin__.RationalNumber instance at 0x782a908>
```

```
t.n
```

```
3
```

```
t.d
```

```
2
```

```
t
```

```
<__builtin__.RationalNumber instance at 0x782a908>
```

```
class RationalNumber:
    def __init__(self, n, d):
        self.n = n
        self.d = d
    def __repr__(self):
        return "%s/%s"%(self.n, self.d)
```

```
t = RationalNumber(78,33484)
```

```
t
```

```
78/33484
```

```
# swap
a,b  =  b,a
```

```
class RationalNumber:
    """
    EXAMPLES::

        sage: RationalNumber(78,-33484)
        -39/16742
    """
    def __init__(self, n, d):
        if d < 0:
            (n, d) = (-n, -d)
        g = gcd(n,d)
        self.n = n // g
        self.d = d // g
    def __repr__(self):
        return "%s/%s"%(self.n, self.d)
```

```
RationalNumber(78,33484)
```

```
39/16742
```

```
RationalNumber(78,-33484)
```

```
-39/16742
```

```
class RationalNumber:
    """
    EXAMPLES::

        sage: RationalNumber(78,-33484)
        -39/16742
    """
    def __init__(self, n, d):
        if d < 0:
            (n, d) = (-n, -d)
        g = gcd(n,d)
        self.n = n // g
        self.d = d // g
    def __repr__(self):
        return "%s/%s"%(self.n, self.d)
    def __add__(self, right):
        if not isinstance(right, RationalNumber):
            raise TypeError
        return RationalNumber(self.n*right.d + self.d*right.n, self.d*\
           right.d)
```

```
s = RationalNumber(2,3)
t = RationalNumber(3,7)
print s, t
2/3 3/7
```

```
s+t
23/21
```

```
s + 'funny thing'
Error in lines 1-1
Traceback (most recent call last):
  File "/projects/74af30b7-ad25-4308-a02e-c71fcd84de6e/.sagemathcloud/sage_server.py",
line 733, in execute
    exec compile(block+'\n', '', 'single') in namespace, locals
  File "", line 1, in <module>
  File "", line 18, in __add__
TypeError
```

```
class RationalNumber:
    """
    EXAMPLES::

        sage: RationalNumber(78,-33484)
        -39/16742
    """
    def __init__(self, n, d):
        if d < 0:
            (n, d) = (-n, -d)
        g = gcd(n,d)
        self.n = n // g
        self.d = d // g
    def __repr__(self):
        return "%s/%s"%(self.n, self.d)
    def __add__(self, right):
        if not isinstance(right, RationalNumber):
            raise TypeError
        return RationalNumber(self.n*right.d + self.d*right.n, self.d*\
           right.d)
    def __mul__(self, right):
        if not isinstance(right, RationalNumber):
            raise TypeError
        return RationalNumber(self.n*right.n, self.d*right.d)
    def __neg__(self):
        return RationalNumber(-self.n, self.d)
    def __pow__(self, k):
        return RationalNumber(self.n^k, self.d^k)
    def __div__(self, right):
        if not isinstance(right, RationalNumber):
            raise TypeError
        return RationalNumber(self.n*right.d, self.d*right.n)
```

```
s = RationalNumber(2,3)
t = RationalNumber(3,7)
s*t
```

```
2/7
```

```
-s/t
```

```
-14/9
```

```
((s+t)*(s*t)) + RationalNumber(3,-7)
```

```
-17/147
```

1.3 Another Class: the field of rational numbers

1.4 Python decorators

- example of making a verbose function
- illustration of a complex decorator: interact

```
def noisy(f):
    def g(*args, **kwds):
        print args, kwds
        ans = f(*args, **kwds)
        print ans
        return ans
    return g
```

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

```
f(3,4)
```

```
(3, 4) {}
7
7
```

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

```
f(3,4)
```

```
(3, 4) {}
7
7
```

1.5 Some graphics for fun

- basics of Sage 2d graphics