

Python IV

Luis Pedro Coelho

Programming for Scientists

September 24, 2012





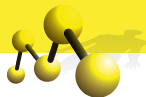
Exceptions

Report errors for higher up.



```
def f(x):  
    return log(x)**2  
  
def g(x):  
    y = f(x)  
    return y+1  
  
def h(x):  
    return g(x+1) + g(4*x)  
  
print h(0)
```

Exceptions



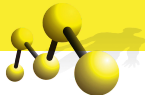
```
def log(x):  
    if x <= 0.:  
        raise ValueError(  
            'log: argument must be greater than zero')  
    ...
```

Try-Except

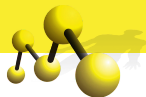


```
try:  
    h(0)  
except:  
    print 'Ooops'
```

Try-Except



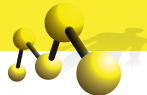
```
try:  
    <line 1>  
    <line 2>  
    <line 3>  
except:  
    <line 1>  
    <line 2>
```



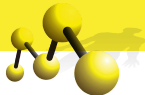
Exceptions

- Exceptions are **objects**.
- Exceptions have **type** and **values**.

Exception Hierarchy



(Nothing here, folks, look at the blackboard)



Exception Handling: Error Handling

```
try:
    <code 1>
    <code 2>
    <code 3>
except IOError, exc:
    print 'I/O Error', exc
except:
    print 'Unspecified error'
```

```

def f(x):
    if x <= 0.:
        raise ValueError(
            'f: argument must be greater than zero')
    return sqrt(x)+2

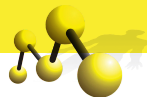
def g(x):
    y = f(x)
    print (y > 2)

try:
    g(1)
    g(-1)
except:
    print 'Exception'

```

This outputs:

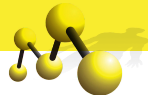
(a)	(b)	(c)	(d)
True	True	False	True
True	False	Exception	Exception



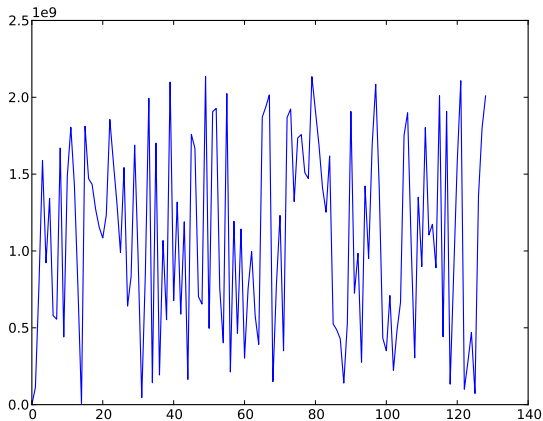
Random numbers

- Truly random numbers
- Pseudo random numbers

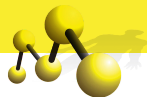
Pseudo Random Numbers



$$x_{i+1} = 48271x_i \mod (2^{31} - 1)$$

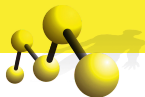


Pseudo Random Numbers



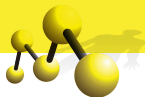
- Are not random
- Some are “more random” than others

Pseudo Random Numbers



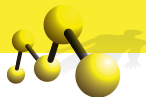
- Are not random
- Some are “more random” than others
- For testing/reproducibility, you want **pseudo**-random numbers.
- For cryptography, you want really random numbers.

Testing with random numbers



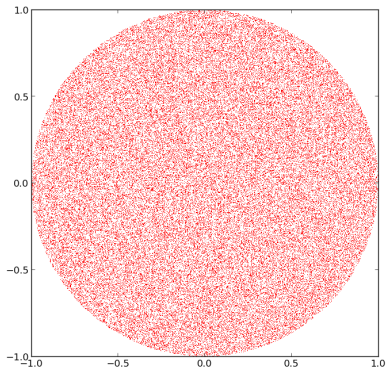
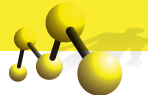
```
import random
random.seed(32)
for i in xrange(16):
    qs = [random.randint(0,40) for j in xrange(100)]
    s,e = trim(qs, 20)
    assert s <= e
    assert np.all(qs[s:e] > 20)
```

Other Random Things

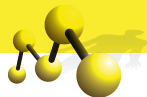


- Random floating point numbers
- Random normally distributed values
- Shuffle arrays
- ...

Random on a circle



More randomness



- Check out `numpy.random`
- Check out `scipy.stats`



```
import pickle
```

```
something = [12, 'hello']
```

```
pickle.dump(something, open('myfile.pkl', 'w'))
```

Later

```
import pickle
```

```
other = pickle.load(open('myfile.pkl'))
```