

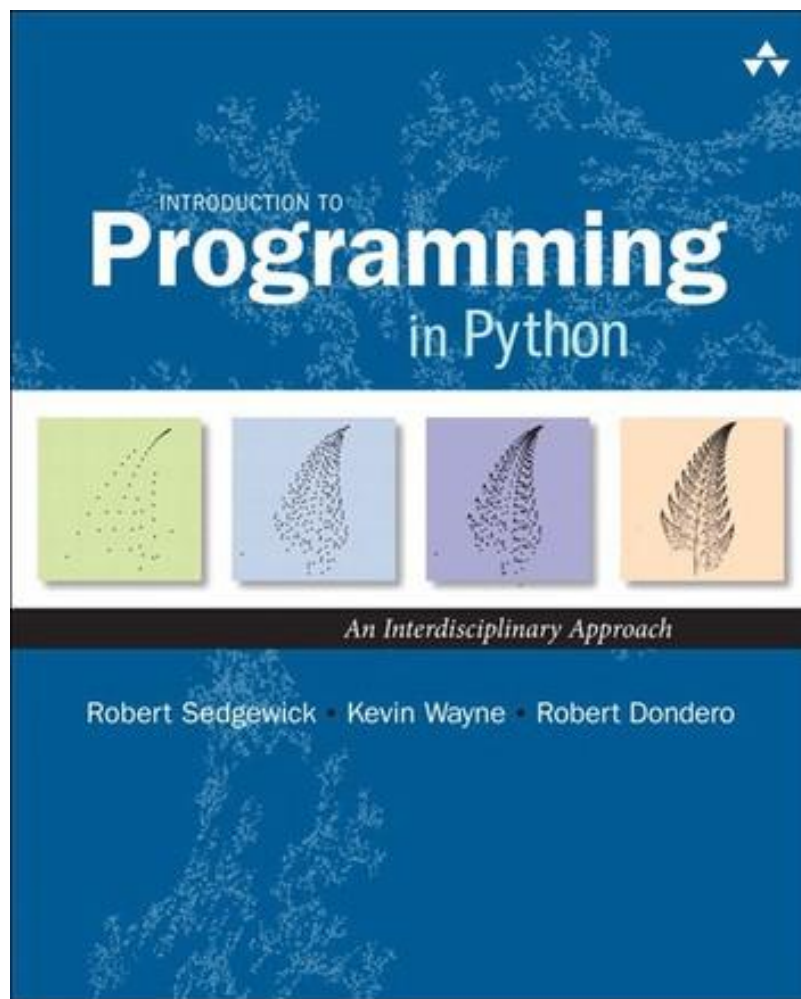
عناصر برنامه نویسی: ورودی و خروجی

سید ناصر رضوی www.snrazavi.ir

۱۳۹۸

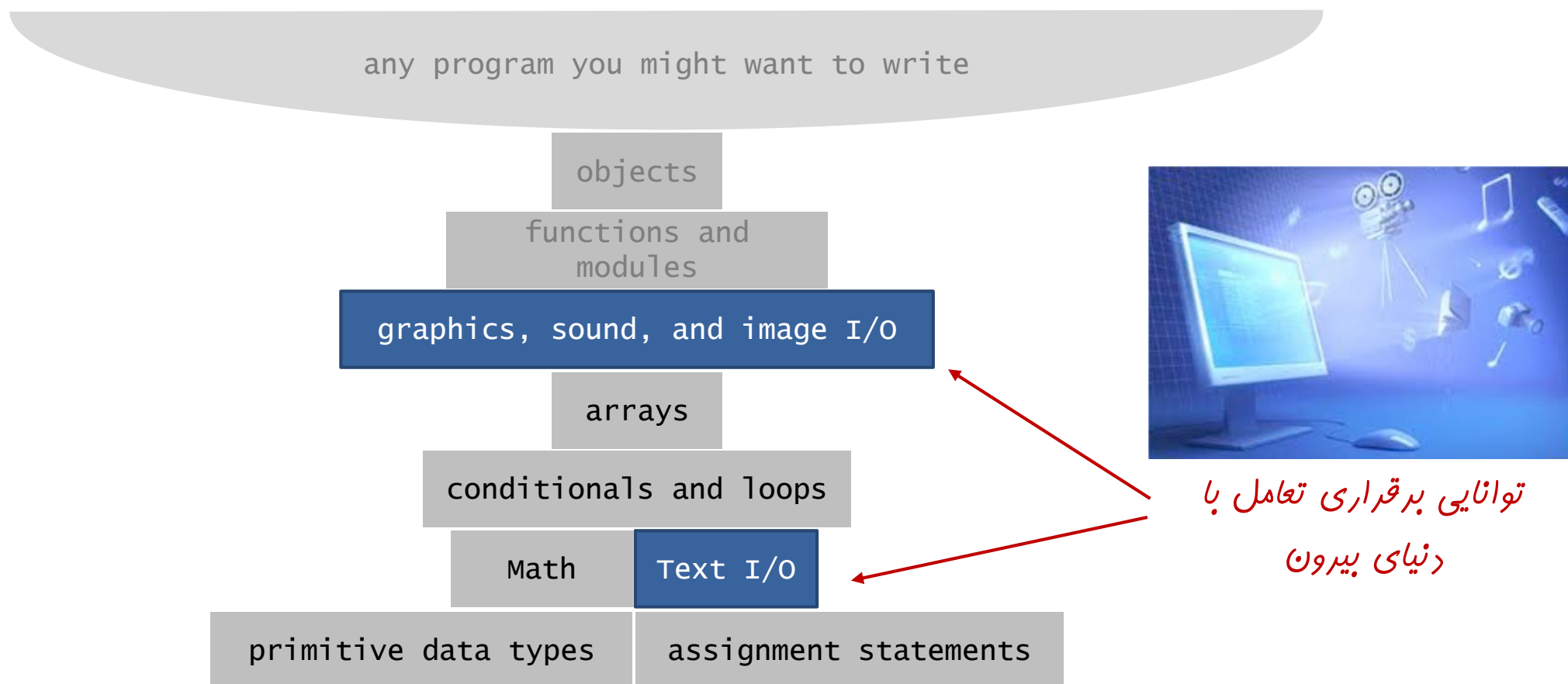
۱-۵ ورودی و خروجی

۲



عناصر برنامه‌نویسی

۳



ورودی و خروجی

۴

□ وسایل ورودی.



□ وسایل خروجی.



□ هدف. برنامه‌هایی که با دنیای خارج تعامل دارند.

ورودی و خروجی

۵

□ وسایل ورودی.



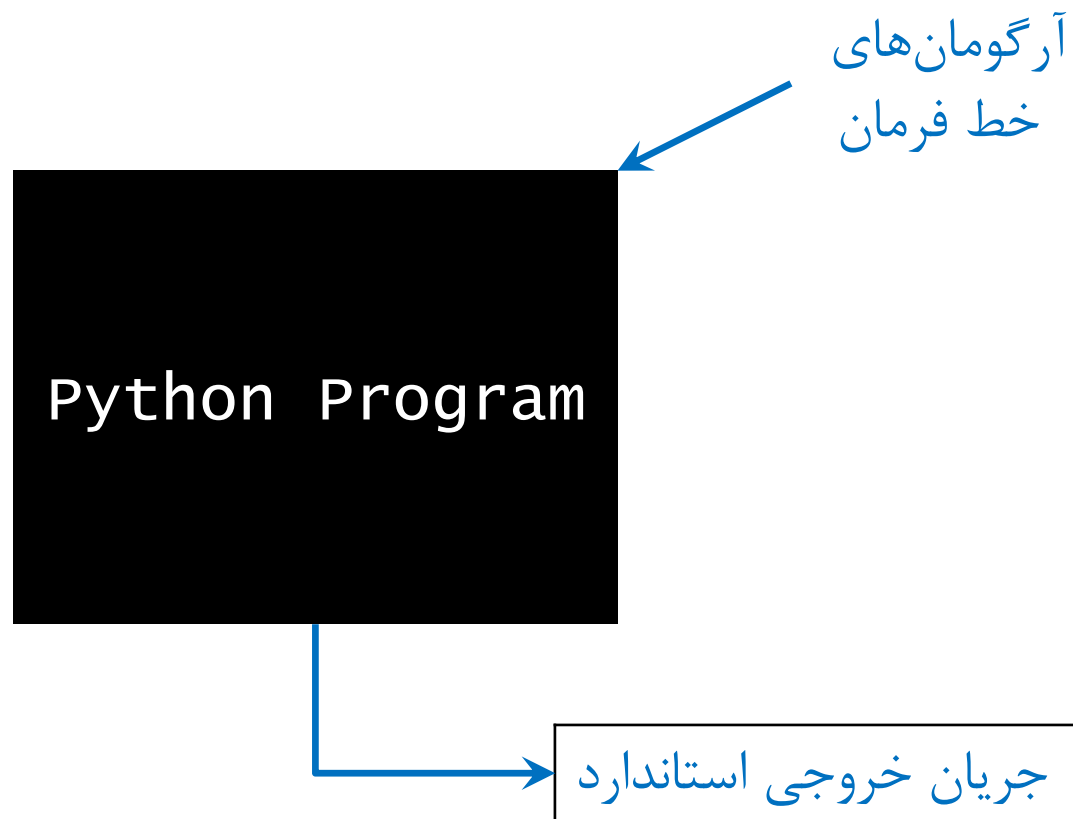
□ وسایل خروجی.



□ رویکرد ما. تعریف کتابخانه‌هایی از توابع برای انجام عملیات ورودی و خروجی.

ورودی و خروجی (تا کنون)

۶



ورودی و خروجی (تا کنون)

۷

□ ورودی خط فرمان. خواندن عدد صحیح n به عنوان آرگومان خط فرمان.

□ خروجی استاندارد.

□ در پایتون، خروجی دستور `print()` به خروجی استاندارد می‌رود.

□ به طور پیش فرض، خروجی استاندارد به ترمینال فرستاده می‌شود.

```
import sys
import random

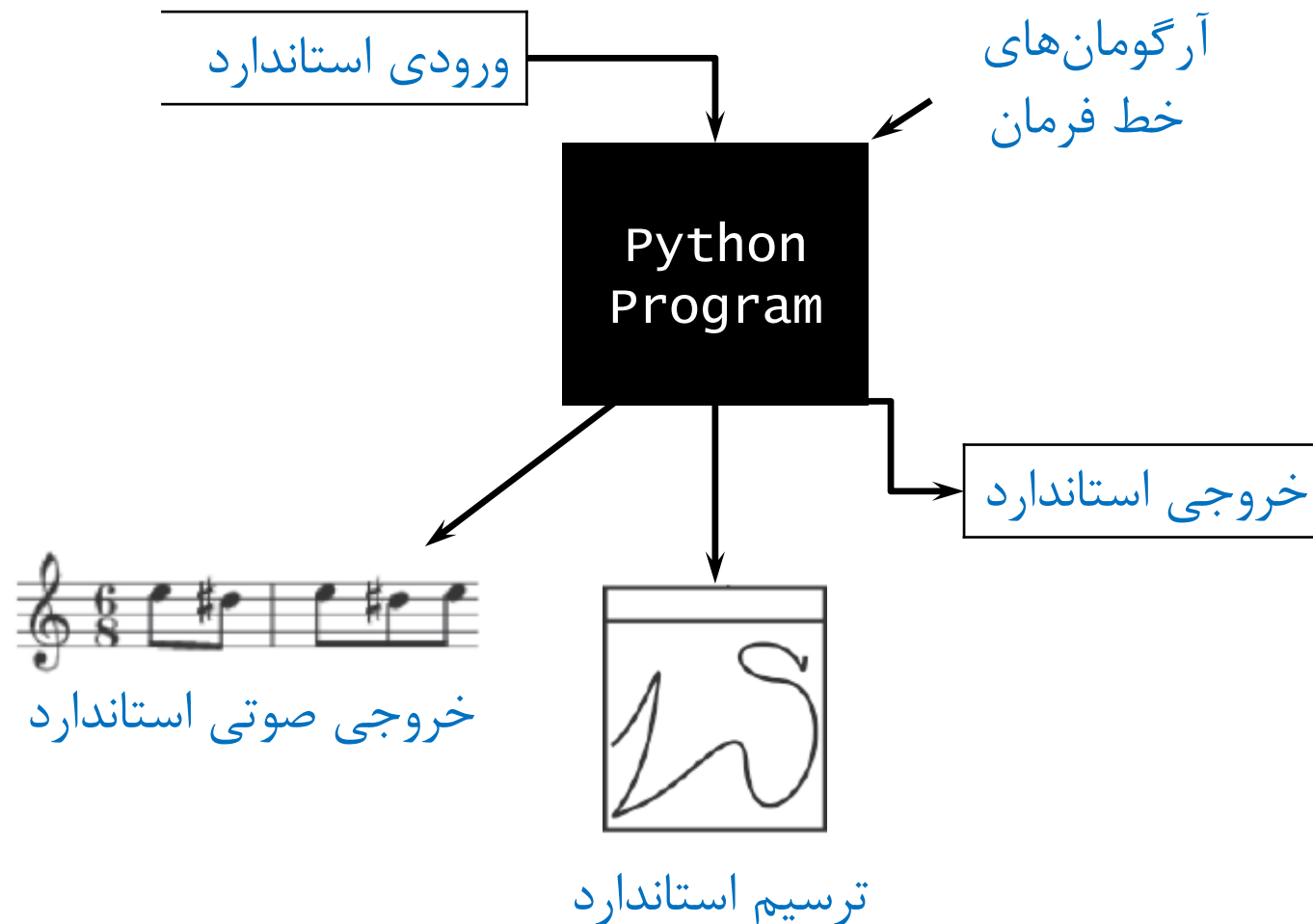
n = int(sys.argv[1])

for i in range(n):
    print(random.random())
```

```
% python randomseq.py 4
0.9320744627218469
0.4279508713950715
0.0899461507116099
0.6579792663546435
```

ورودی و خروجی

۸



ورودی خط فرمان و ورودی استاندارد

۹

□ ورودی‌های خط فرمان.

- برای دریافت تعداد **کمی** از ورودی‌های کاربر مناسب است.
- برای تعداد زیاد ورودی‌ها قابل استفاده نیست.
- دادن ورودی به برنامه، **پیش از** شروع اجرای برنامه.

```
% python randomseq.py 4
```

```
0.9320744627218469  
0.4279508713950715  
0.0899461507116099  
0.6579792663546435
```

```
% python addints.py 4
```

```
10 20 30 40  
sum is 100
```

□ ورودی‌های استاندارد.

- به طور پیش فرض از طریق پنجره ترمینال دریافت می‌شود.
- دادن ورودی به برنامه، زمانی که برنامه **در حال اجرا** است.

ورودی و خروجی استاندارد

□ ورودی استاندارد. `stdio.py` یک کتابخانه برای خواندن ورودی‌های متنی است.

تابع

توضیحات

`stdio.isEmpty()`

اگر مقدار دیگری در ورودی وجود نداشته باشد، `true` است

`stdio.readInt()`

خواندن یک مقدار از نوع `int`

`stdio.readFloat()`

خواندن یک مقدار از نوع `float`

`stdio.readBool()`

خواندن یک مقدار از نوع `bool`

`stdio.readString()`

خواندن یک مقدار از نوع `string`

`stdio.hasNextLine()`

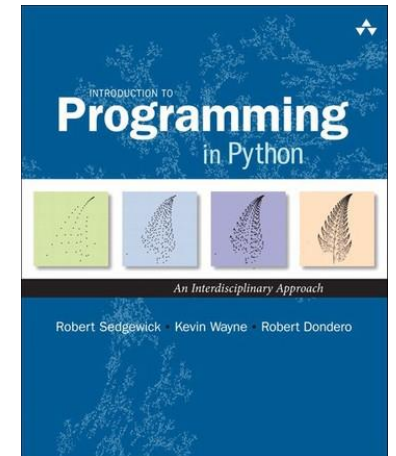
آیا خط دیگری در ورودی استاندارد وجود دارد

`stdio.readLine()`

خواندن عناصر باقیمانده از یک سطر و برگرداندن آنها به صورت رشته

`stdio.readAll()`

خواندن عناصر باقیمانده از متن و برگرداندن آنها به صورت رشته



خروجی استاندارد: قالب‌بندی خروجی

۱۱

<i>type</i>	<i>code</i>	<i>typical literal</i>	<i>sample format string</i>	<i>converted string values for output</i>
int	d	512	'%14d' '%-14d'	'512' '512'
float	f e	1595.1680010754388	'%14.2f' '%.7f' '%14.4e'	'1595.17' '1595.1680011' '1.5952e+03'
String	s	'Hello, world'	'%14s' '%-14s' '%-14.5s'	'Hello, world' 'Hello, world' 'Hello'

```
>>> import math
>>> print('pi is approximately %.2f' %math.pi)
pi is approximately 3.14
>>>
```

ورودی و خروجی استاندارد

۱۲

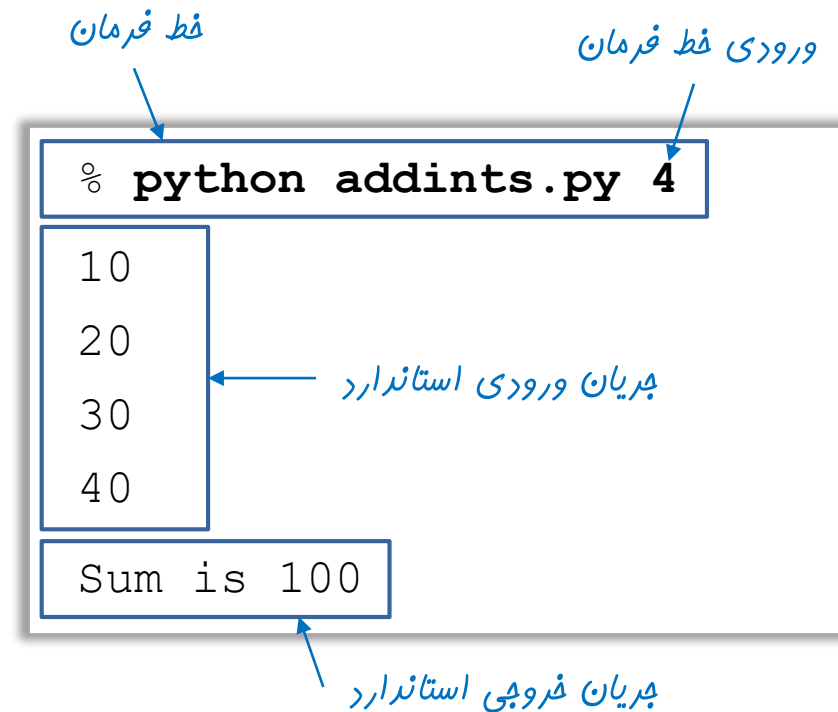
□ روش استفاده. فایل `stdio.py` را در مسیر دایرکتوری کاری قرار دهید.

```
import sys
import stdio

n = int(sys.argv[1])

total = 0
for i in range(n):
    total += stdio.readInt()

print('Sum is %s' %total)
```



ورودی و خروجی استاندارد

۱۳

□ روش استفاده. فایل `stdio.py` را در مسیر دایرکتوری کاری قرار دهید.

```
import sys
import stdio

n = int(sys.argv[1])

total = 0
for i in range(n):
    total += stdio.readInt()

print('Sum is %s' %total)
```

```
% python addints.py 4
10 20 30 40
Sum is 100
```

ورودی و خروجی استاندارد: ورودی تعاملی

۱۴

```
import stdio, random

RANGE = 1000000

secret = random.randrange(1, RANGE + 1)
print('Guess a number between 1 and %d' %RANGE)

guess = 0
while guess != secret:

    print('what is your guess?', end=' ')
    guess = stdio.readInt()

    if guess < secret: print('Too low')
    elif guess > secret: print('Too high')
    else                : print('You win!')
```

```
% python twentyquestions.py
Guess a number between 1 and 1000000
What is your guess? 500000
Too low
What is your guess? 750000
Too high
What is your guess? 625000
Too high
.
.
.
What is your guess? 501686
You win!
```

محاسبه میانگین دنباله‌ای از اعداد

۱۵

□ مثال. دنباله‌ای از اعداد را بخوان و میانگین آنها را در خروجی چاپ کن.

```
import stdio

total = 0.0
count = 0

while not stdio.isEmpty():
    value = stdio.readFloat()
    total += value
    count += 1

print('Average is %.2f' % (total / count))
```

```
% python average.py
10.0  5.0  6.0
 3.0  7.0 32.0
<ctrl-z>
Average is 10.50
```

لینوکس: <ctrl-d>

تغییر مسیر خروجی استاندارد

۱۶

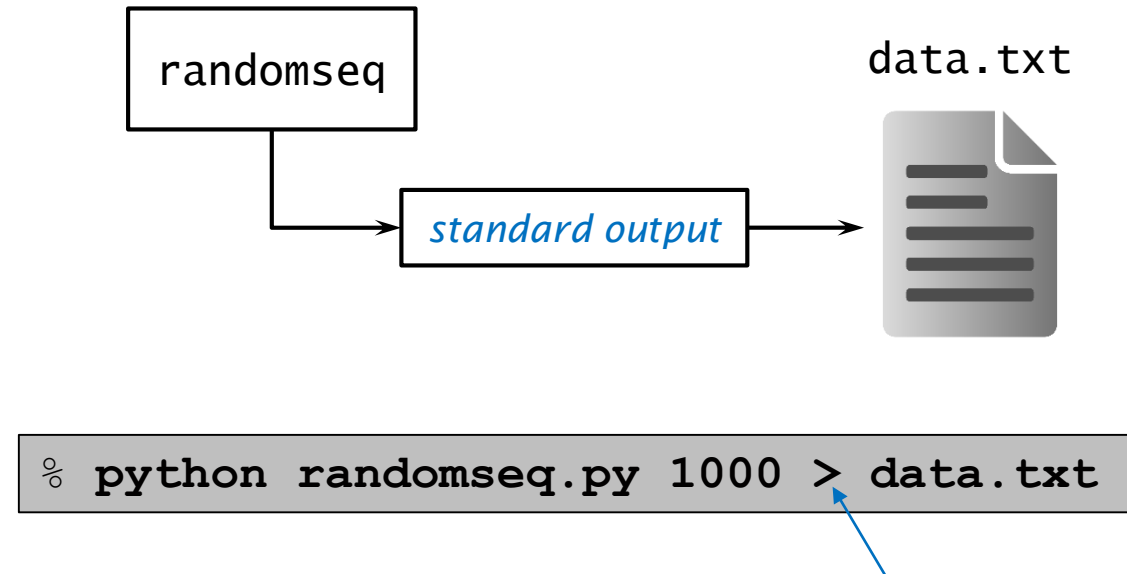
□ تغییر مسیر خروجی استاندارد. استفاده از راهنماهای سیستم عامل برای فرستادن خروجی استاندارد به یک فایل به منظور ذخیره دائمی (به جای فرستادن به پنجره ترمینال)

```
import sys
import random

n = int(sys.argv[1])

for i in range(n):
    print(random.random())
```

تولید تعدادی عدد تصادفی و چاپ آنها در خروجی استاندارد



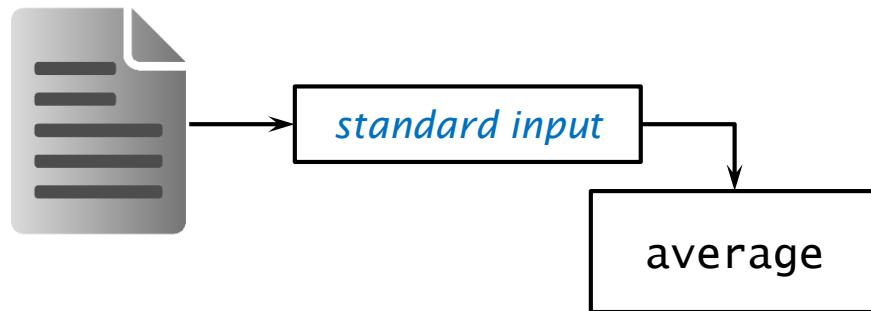
تغییر مسیر خروجی استاندارد

تغییر مسیر ورودی استاندارد

۱۷

□ تغییر مسیر ورودی استاندارد. استفاده از راهنماهای سیستم عامل برای خواندن ورودی استاندارد از یک فایل (به جای خواندن از پنجره ترمینال)

data.txt



```
% more data.txt
0.5475375782884312
0.4971087292684019
0.23123808041753813
...

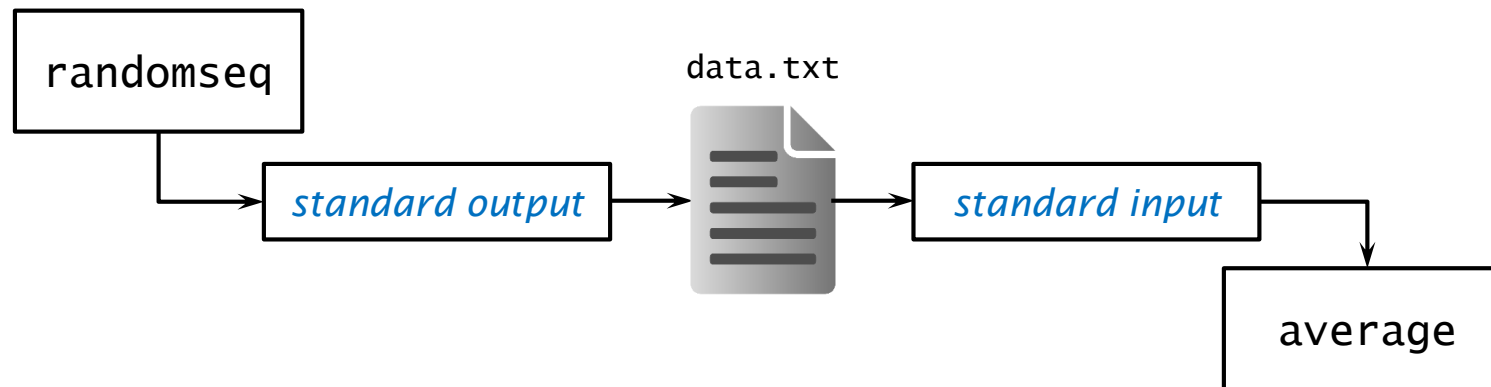
% python average.py < data.txt
0.4947655567740991
```

تغییر مسیر ورودی استاندارد

اتصال برنامه‌ها

۱۸

□ **لوله‌کشی.** استفاده از راهنماهای سیستم عامل برای فرستادن خروجی استاندارد یک برنامه به ورودی استاندارد یک برنامه دیگر.

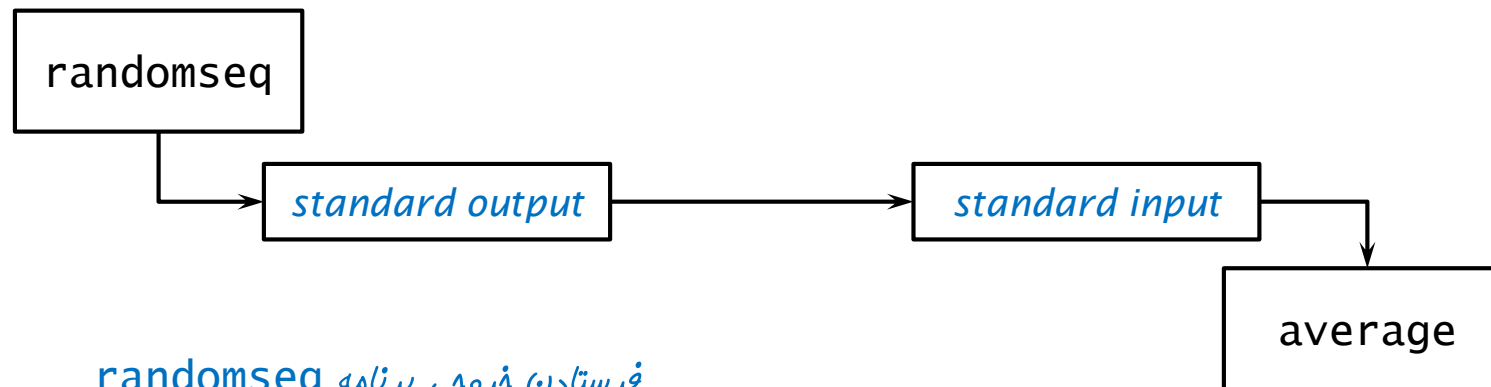


```
% python randomseq.py 1000000 > data.txt  
  
% python average.py < data.txt  
0.5002071875644842
```

اتصال برنامه‌ها

۱۹

□ **لوله‌کشی.** استفاده از راهنماهای سیستم عامل برای فرستادن خروجی استاندارد یک برنامه به ورودی استاندارد یک برنامه دیگر.



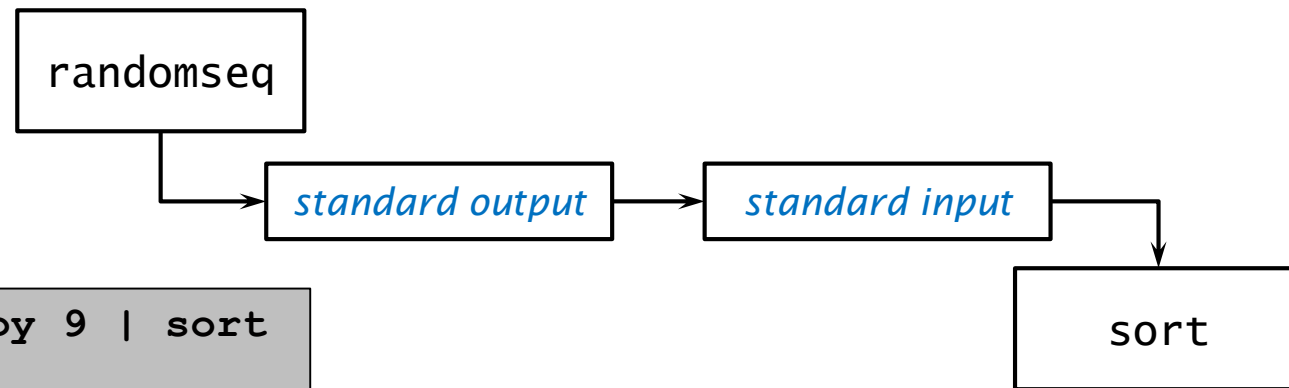
فرستادن خروجی برنامه randomseq
به عنوان ورودی به برنامه average

```
% python randomseq.py 1000000 | python average.py  
0.4997970473016028  
  
% python randomseq.py 1000000 | python average.py  
0.5002071875644842
```

اتصال برنامه‌ها: فیلترها

۲۰

□ **لوله‌کشی.** استفاده از راهنماهای سیستم عامل برای فرستادن خروجی استاندارد یک برنامه به ورودی استاندارد یک برنامه دیگر.



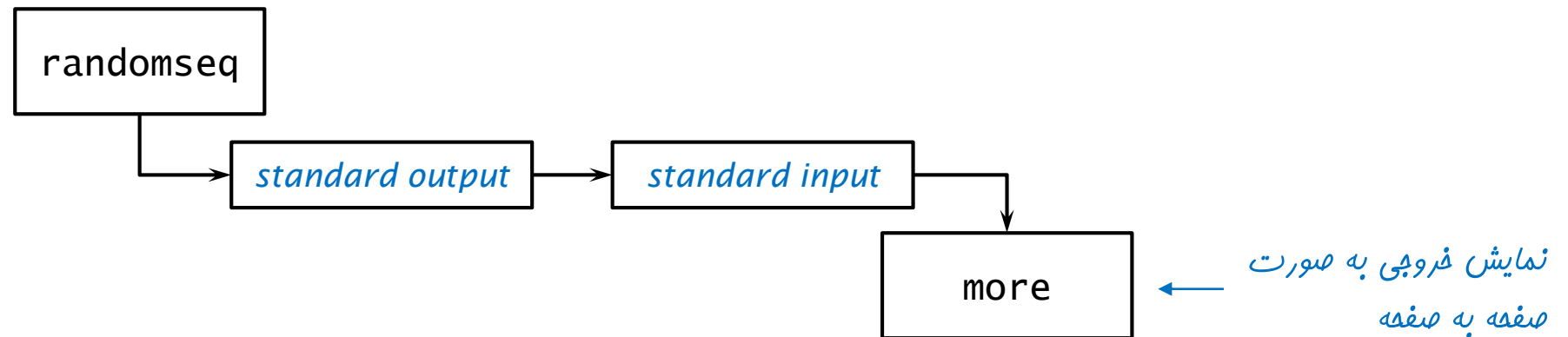
```
% python randomseq.py 9 | sort
```

```
0.19349705468566114
0.19913632461426878
0.2548930267381452
0.438576726344272
0.5662068930742593
0.6240702774550413
0.7056219487952354
0.9140872908136548
0.950131428032779
```

اتصال برنامه‌ها: فیلترها

۲۱

□ **لوله‌کشی.** استفاده از راهنماهای سیستم عامل برای فرستادن خروجی استاندارد یک برنامه به ورودی استاندارد یک برنامه دیگر.



```
% python randomseq.py 1000 | more
```

ترسیم استاندارد

۲۲

□ ترسیم استاندارد. استفاده از کتابخانه `stdraw.py` برای تولید خروجی گرافیکی.

تابع

`stddraw.point(x, y)`

`stddraw.line(x0, y0, x1, y1)`

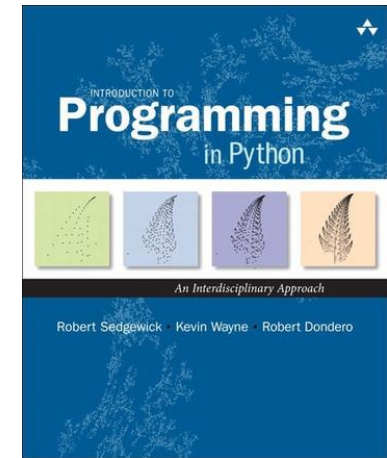
`stddraw.show()`

توضیحات

ترسیم یک نقطه در (x, y)

ترسیم یک پاره‌خط از $(x0, y0)$ به $(x1, y1)$

کپی کردن از بوم پس‌زمینه و نمایش در پنجره استاندارد ترسیم



ترسیم استاندارد

۲۳

□ روش استفاده. فایل `stddraw.py` را در مسیر دایرکتوری کاری قرار دهید.

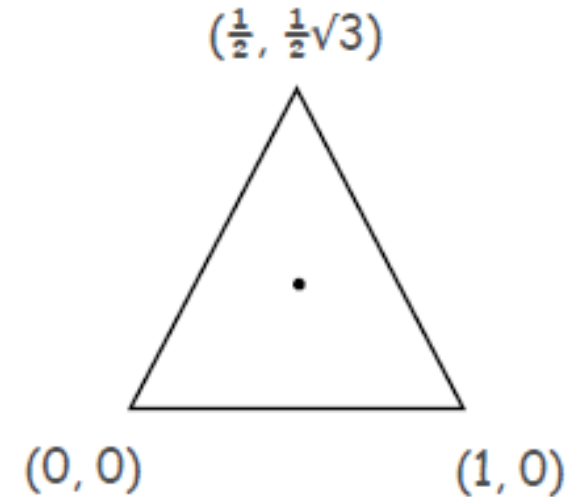
```
import stddraw
import math

t = math.sqrt(3.0) / 2.0

stddraw.line(0.0, 0.0, 1.0, 0.0)
stddraw.line(1.0, 0.0, 0.5, t)
stddraw.line(0.5, t, 0.0, 0.0)
stddraw.point(0.5, t/3.0)

stdraw.show()
```

```
% python triangle.py
```



نصب کتابخانه‌ها در پایتون

۲۴

□ کتابخانه `stdraw.py` از کتابخانه `pygame` استفاده می‌کند. بنابراین ابتدا باید این کتابخانه را نصب نمایید.

□ روش اول. استفاده از دستور `pip` در خط فرمان



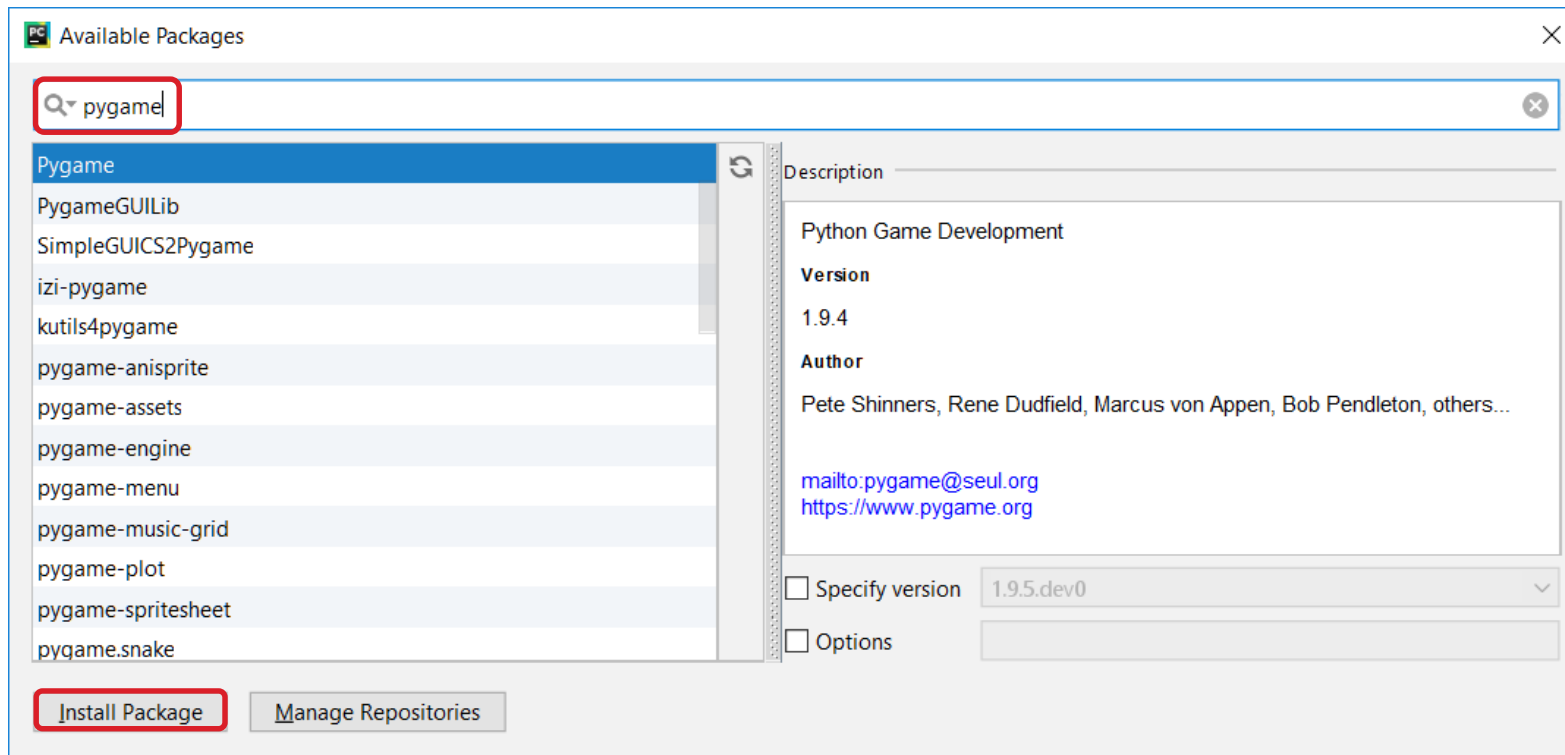
```
Command Prompt
C:\Users\Razavi>pip install pygame
Collecting pygame
  Using cached https://files.pythonhosted.org/packages/b0/30/4df62b0eb0f09a8ad1c329ff5479b3b115e8dce25475fd0c2f73cb790aa4/pygame-1.9.4-cp37-cp37m-win32.whl
Installing collected packages: pygame
Successfully installed pygame-1.9.4
C:\Users\Razavi>
```


نصب کتابخانه‌ها در پایتون

۲۵

□ کتابخانه `stdraw.py` از کتابخانه `pygame` استفاده می‌کند. بنابراین ابتدا باید این کتابخانه را نصب نمایید.

□ روش دوم. استفاده از نصب کننده پکیج در پای چارم



کتابخانه stdDraw: دستورات کنترلی

۲۶

تابع

توضیحات

`stdDraw.setCanvasSize(w, h)`

تغییر اندازه پنجره ترسیم استاندارد به w در h پیکسل
(مقادیر پیش فرض w و h برابر با ۵۱۲)

`stdDraw.setXscale(x0, x1)`

تنظیم مقیاس محور x به $(x0, x1)$
(مقدار پیش فرض $x0$ برابر با ۰ و مقدار پیش فرض $x1$ برابر با ۱)

`stdDraw.setYscale(y0, y1)`

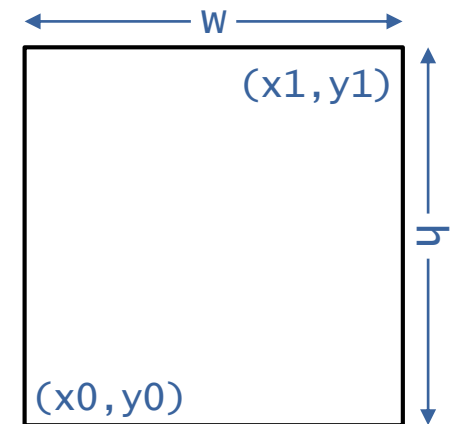
تنظیم مقیاس محور y به $(y0, y1)$
(مقدار پیش فرض $y0$ برابر با ۰ و مقدار پیش فرض $y1$ برابر با ۱)

`stdDraw.setPenRadius(r)`

تغییر ضخامت قلم به مقدار r
(مقدار پیش فرض r برابر با ۰.۰۰۵)

`stdDraw.setPenColor(c)`

تنظیم رنگ قلم به رنگ c
(مقدار پیش فرض رنگ قلم برابر با سیاه)



پنجره ترسیم استاندارد

ترسیم داده‌ها

۲۷

```
import stdio, stddraw
```

```
x0 = stdio.readFloat()
```

```
y0 = stdio.readFloat()
```

```
x1 = stdio.readFloat()
```

```
y1 = stdio.readFloat()
```

خواندن مفتصات
بجبهه مفاطی

```
stddraw.setXscale(x0, x1)
```

```
stddraw.setYscale(y0, y1)
```

تغییر مقیاس

```
while not stdio.isEmpty():
```

```
    x = stdio.readFloat()
```

```
    y = stdio.readFloat()
```

```
    stddraw.point(x, y)
```

خواندن و ترسیم
یک نقطه

```
stdraw.show()
```

مفتصات بجبهه مفاطی

```
% more usa.txt
```

```
669905.0 247205.0 1244962.0 490000.0
```

```
1097038.8890 245552.7780
```

```
1103961.1110 247133.3330
```

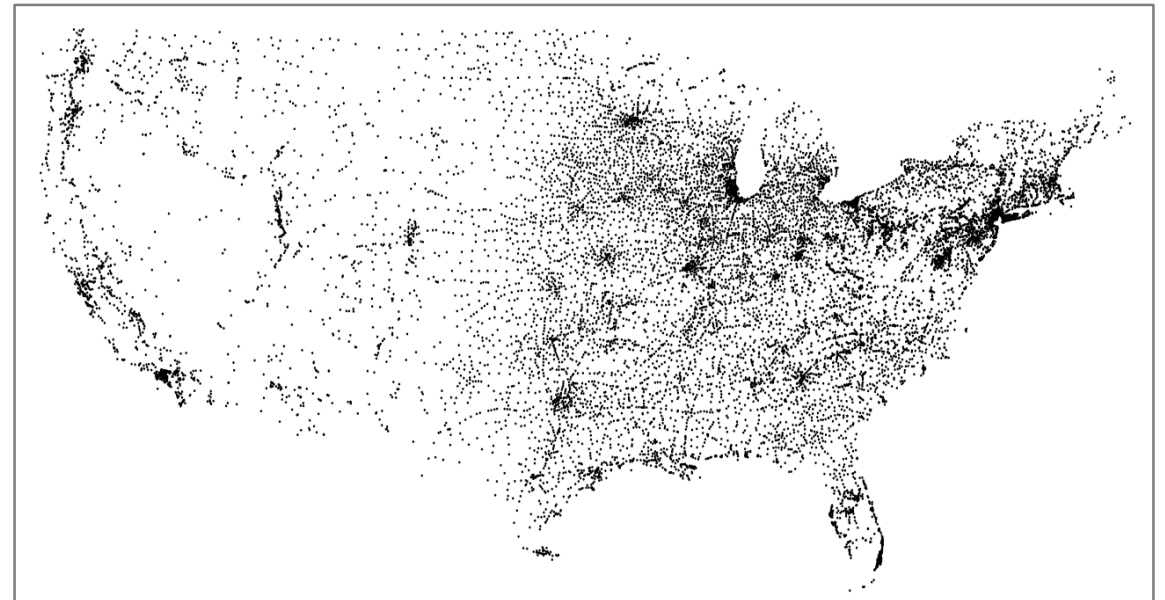
```
1104677.7780 247205.5560
```

```
...
```

دنباله نقاط

(۱۳,۵۰۹ شهر)

```
% python plotfilter.py < usa.txt
```



```
import sys, math, stddraw

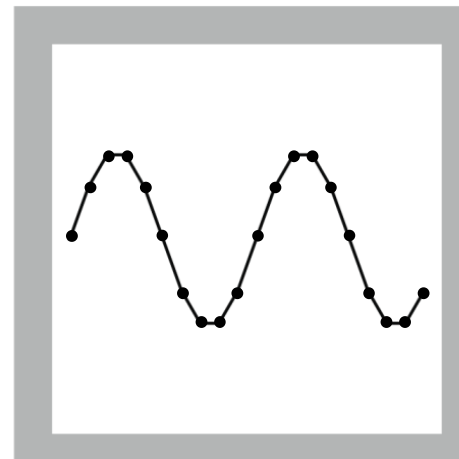
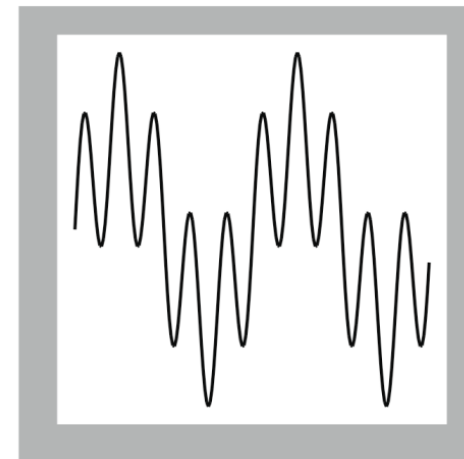
n = int(sys.argv[1])
x, y = [], []

for i in range(n + 1):
    x += [math.pi * i / n]
    y += [math.sin(4*x[i]) + math.sin(20*x[i])]

stddraw.setXscale(0, math.pi)
stddraw.setYscale(-2.0, 2.0)

for i in range(n):
    stddraw.line(x[i], y[i], x[i+1], y[i+1])

stdraw.show()
```

 $n = 20$  $n = 200$ 

$$y = \sin 4x + \sin 20x, \quad \forall x \in [0, \pi]$$

بازی آشوب

۲۹

□ بازی آشوب. بر روی یک مثلث متساوی الاضلاع با رئوس R، G و B بازی کنید.

□ از رأس R شروع کنید.

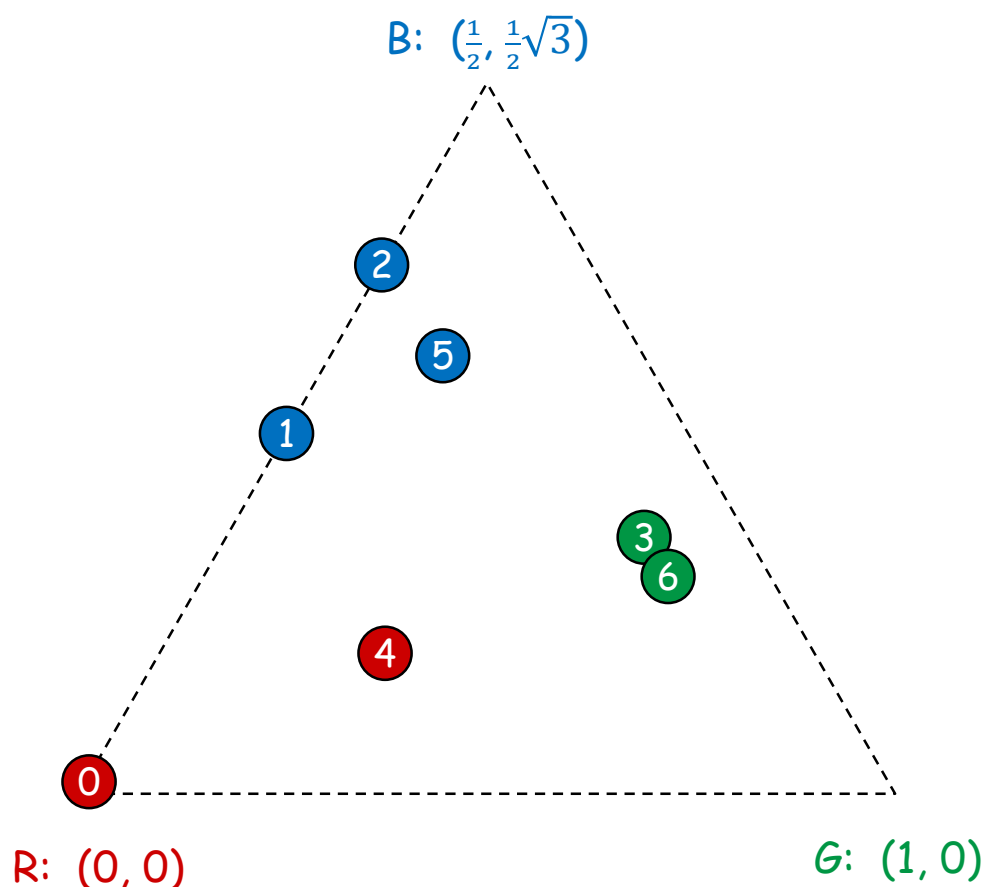
□ مراحل زیر را n مرتبه تکرار کنید:

■ یک رأس به صورت تصادفی انتخاب کنید.

■ از نقطه فعلی به سمت رأس انتخابی نصف مسیر را بپیمایید.

■ در این مکان یک نقطه به رنگ رأس انتخابی ترسیم کنید.

□ پرسش. چه تصویری مشاهده خواهید نمود؟



بازی آشوب

۳۰

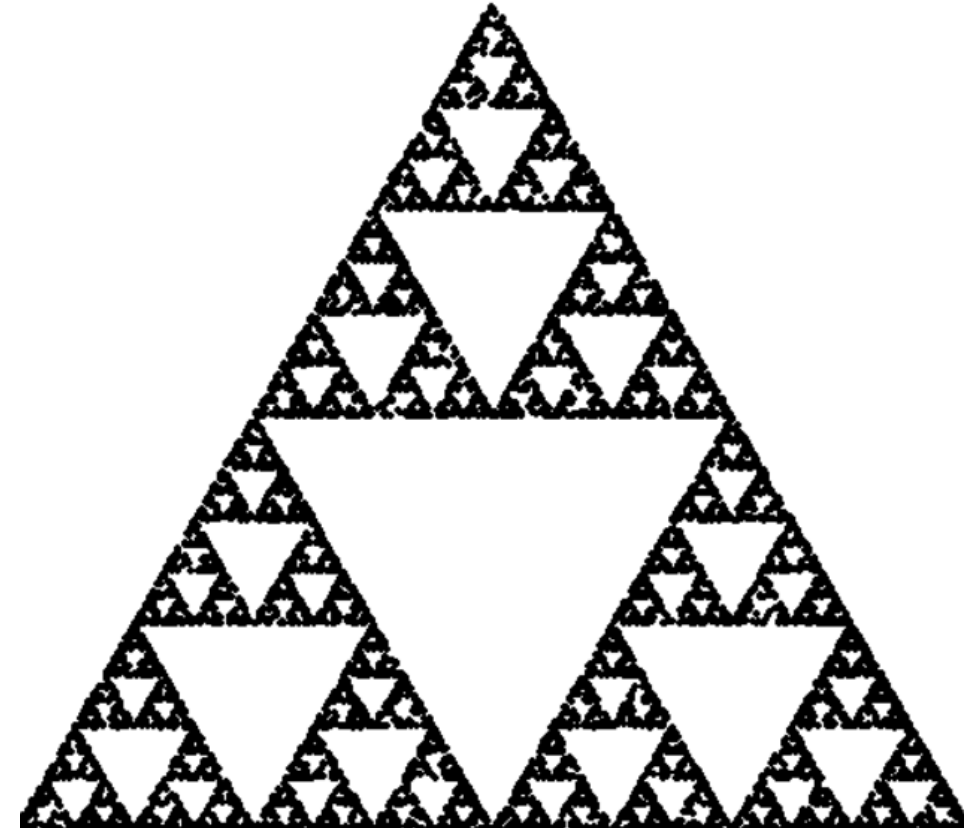
```
import sys, random, stddraw

n = int(sys.argv[1])
cx = [0.0, 1.0, 0.500]
cy = [0.0, 0.0, 0.866]

x, y = 0, 0

for i in range(n):
    r = random.randrange(0, 3)
    x = (x + cx[r]) / 2.0
    y = (y + cy[r]) / 2.0
    stddraw.point(x, y)

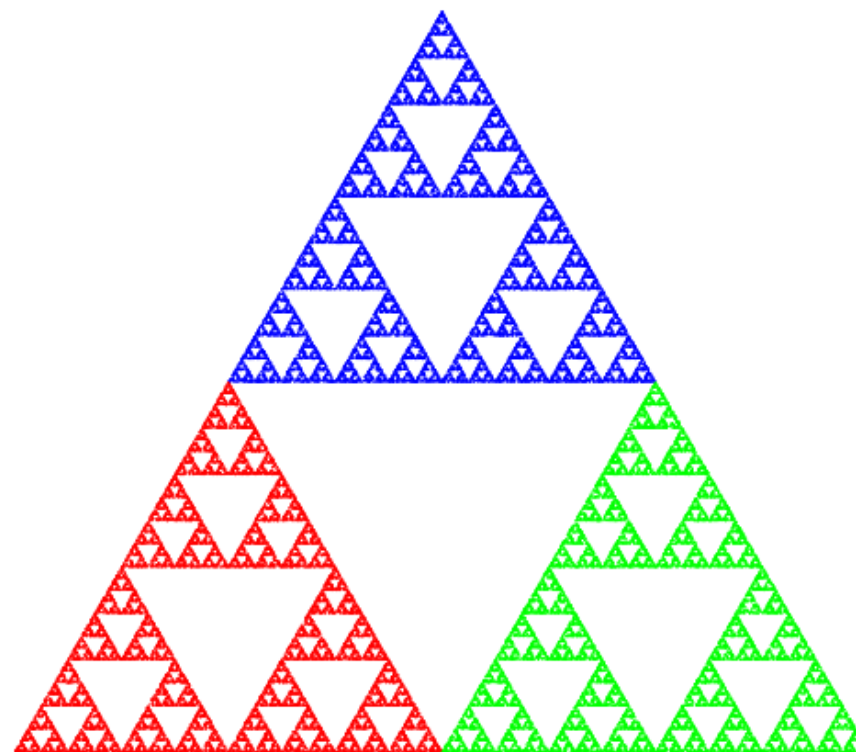
stdraw.show()
```



```
% python chaos.py 10000
```

□ یک اصلاح ساده. نقطه محاسبه شده را همرنگ با رأس تصادفی انتخاب شده ترسیم کن.

```
% python chaos.py 10000
```



مثلث سیرپینسکی

سرخس بارنزی

۳۲

□ سرخس بارنزی. انجام بازی آشوب با استفاده از قواعد مختلف.



probability	new x	new y
2%	.50	.27y
15%	$-.14x + .26y + .57$	$.25x + .22y - .04$
13%	$.17x - .21y + .41$	$.22x + .18y + .09$
70%	$.78x + .03y + .11$	$-.03x + .74y + .27$

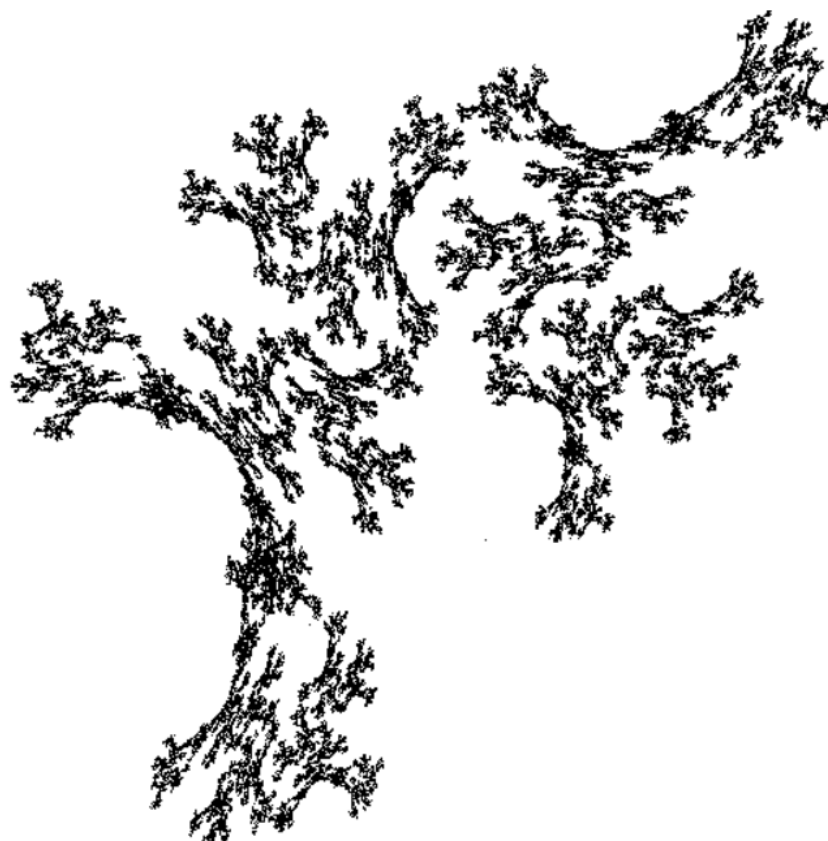
□ پرسش. محاسبات چه چیزی درباره طبیعت به ما می گوید؟

□ پرسش. طبیعت چه چیزی درباره محاسبات به ما می گوید؟

صخره مرجانی

۳۳

□ صخره مرجانی. انجام بازی آشوب با استفاده از قواعد مختلف.



probability	new x	new y
40%	$.31x - .53y + .89$	$-.46x - .29y + 1.04$
15%	$.31x - .08y + .22$	$.15x - .45y + 0.34$
45%	$.55y + .01$	$.69x - .20y + 0.38$

□ پرسش. محاسبات چه چیزی درباره طبیعت به ما می گوید؟

□ پرسش. طبیعت چه چیزی درباره محاسبات به ما می گوید؟

کتابخانه stdDraw: دستورات دیگر

۳۴

تابع

`stdDraw.circle(x, y, r)`

`stdDraw.square(x, y, r)`

`stdDraw.rectangle(x, y, w, h)`

`stdDraw.polygon(x, y)`

`stdDraw.filledCircle(x, y, r)`

`stdDraw.filledSquare(x, y, r)`

`stdDraw.filledPolygon(x, y)`

توضیحات

ترسیم یک دایره به مرکز (x, y) و به شعاع r

ترسیم یک مربع به مرکز (x, y) و اضلاع $2r$

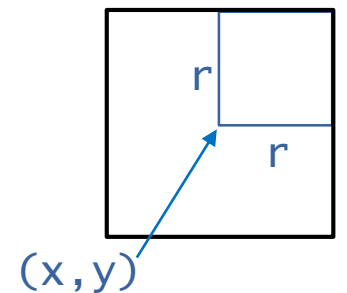
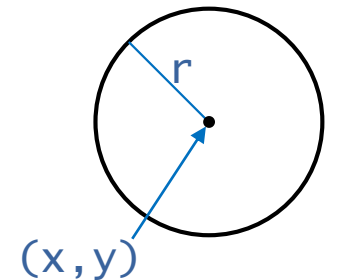
ترسیم یک مستطیل با ابعاد w در h و گوشه پپ پایین (x, y)

ترسیم یک چندضلعی با وصل کردن نقاط آرایه x و y

ترسیم یک دایره پر به مرکز (x, y) و به شعاع r

ترسیم یک مربع پر به مرکز (x, y) و اضلاع $2r$

ترسیم یک چندضلعی پر با وصل کردن نقاط آرایه x و y



پویانمایی

۳۵

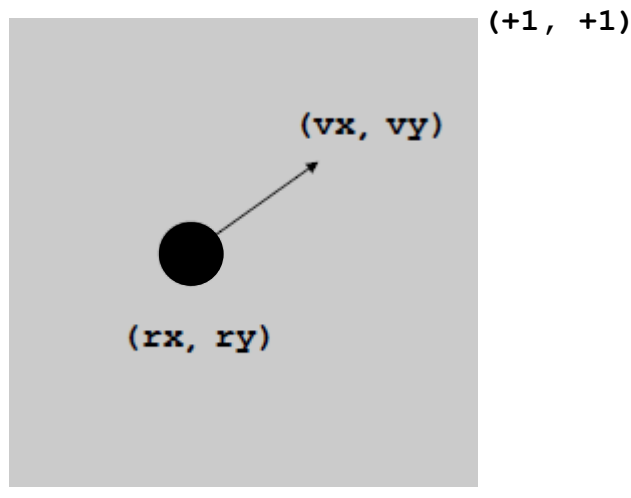
□ حلقه پویانمایی. تکرار مراحل زیر:

□ صفحه را پاک کن.

□ شی مورد نظر را حرکت بده.

□ شی مورد نظر را ترسیم کن.

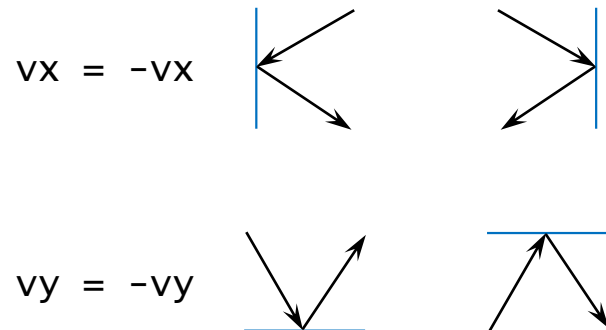
□ تصویر جدید را نمایش بده و چند لحظه مکث کن.



□ مثال. توپ متحرک.

□ توپ دارای مکان (rx, ry) و سرعت ثابت (vx, vy) است.

□ در صورت برخورد با دیواره‌ها، سرعت را معکوس کن.



```
import stddraw

RADIUS, DT = 0.05, 15.0

stdraw.setXscale(-1.0, +1.0)
stdraw.setYscale(-1.0, +1.0)

rx, ry = .480, .860
vx, vy = .015, .023

while True:
    if abs(rx + vx) + RADIUS > 1.0: vx = -vx
    if abs(ry + vy) + RADIUS > 1.0: vy = -vy

    rx += vx
    ry += vy

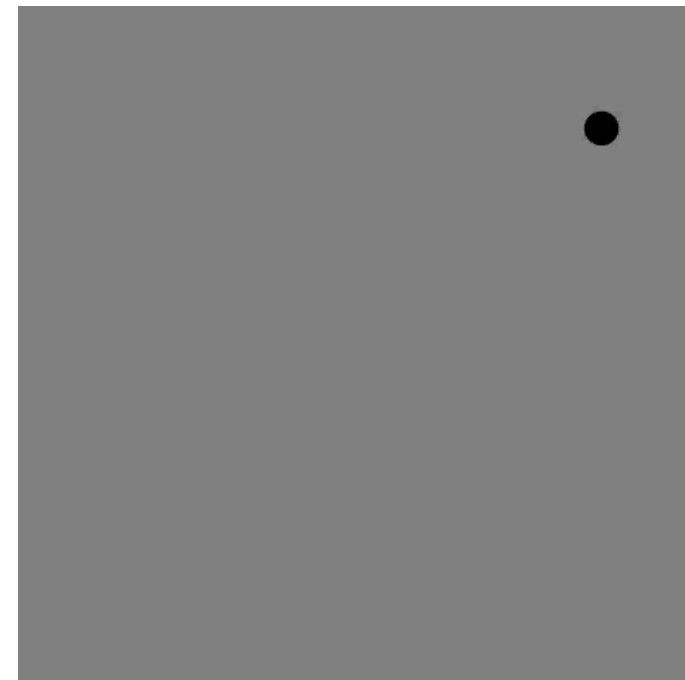
    stddraw.clear(stddraw.GRAY)
    stddraw.filledCircle(rx, ry, RADIUS)
    stddraw.show(DT)
```

به روز رسانی مکان توپ

پاک کردن پس زمینه

ترسیم توپ

% python bouncingball.py



جلوہای ویژه

۳۷

- **تصاویر.** یک فایل تصویری با یکی از پسوندهای `.gif` یا `.png` یا `.jpg` را در دایرکتوری کاری قرار داده و از دستور `std::draw.picture()` برای ترسیم آن استفاده کنید.
- **صدا.** یک فایل صوتی با یکی از پسوندهای `.wav` یا `.mid` یا `.au` را در دایرکتوری کاری قرار داده و از دستور `std::audio.play()` برای پخش آن استفاده کنید.
- **مثال.** برنامه قبل را برای نشان دادن تصویر کره زمین و پخش یک صدا در صورت برخورد توپ با دیواره اصلاح کنید.
 - به جای `std::draw.filledCircle()` از دستور زیر استفاده کنید:

```
std::draw.picture('earth.gif', rx, ry)
```

□ تکه کد زیر را به برنامه اضافه کنید:

```
std::audio.play('boing.wav')
```

اجرای نمایشی

۳۸

□ جلوه‌های ویژه. صدا و تصویر.

```
% python deluxebouncingball.py
```

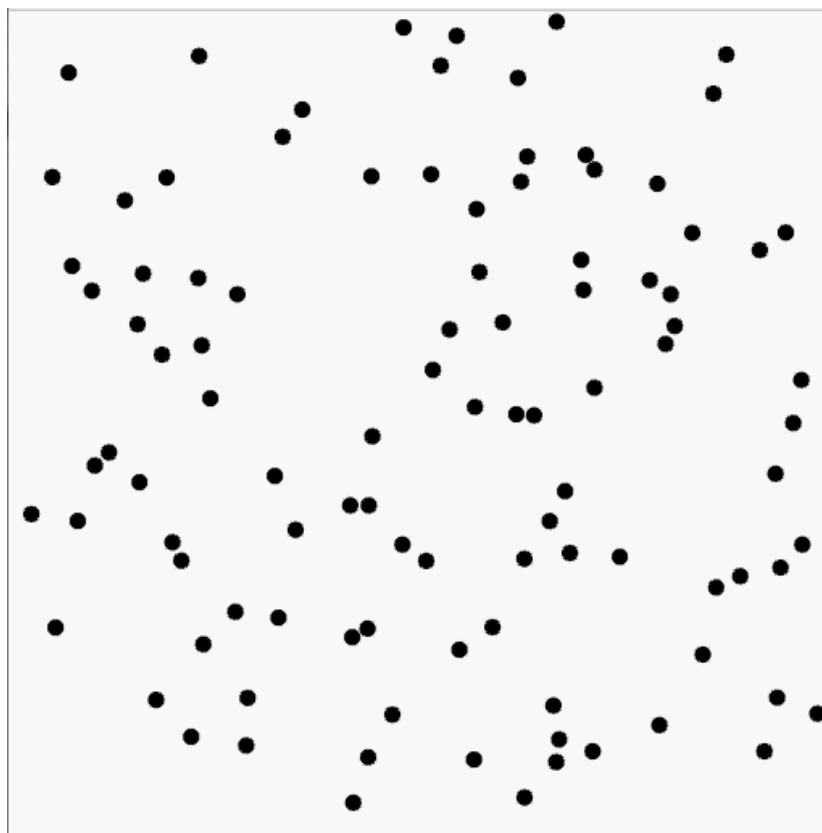


توپ‌های برخورد کننده

۳۹

□ چالش. افزودن برخورد.

```
% python collisionsystem.py 100
```



شبیه‌سازی چند جسمی

۴۰

□ چالش. افزودن نیروی جاذبه.

T dt
% python nbody.py 157788000.0 25000.0 < planets.txt

