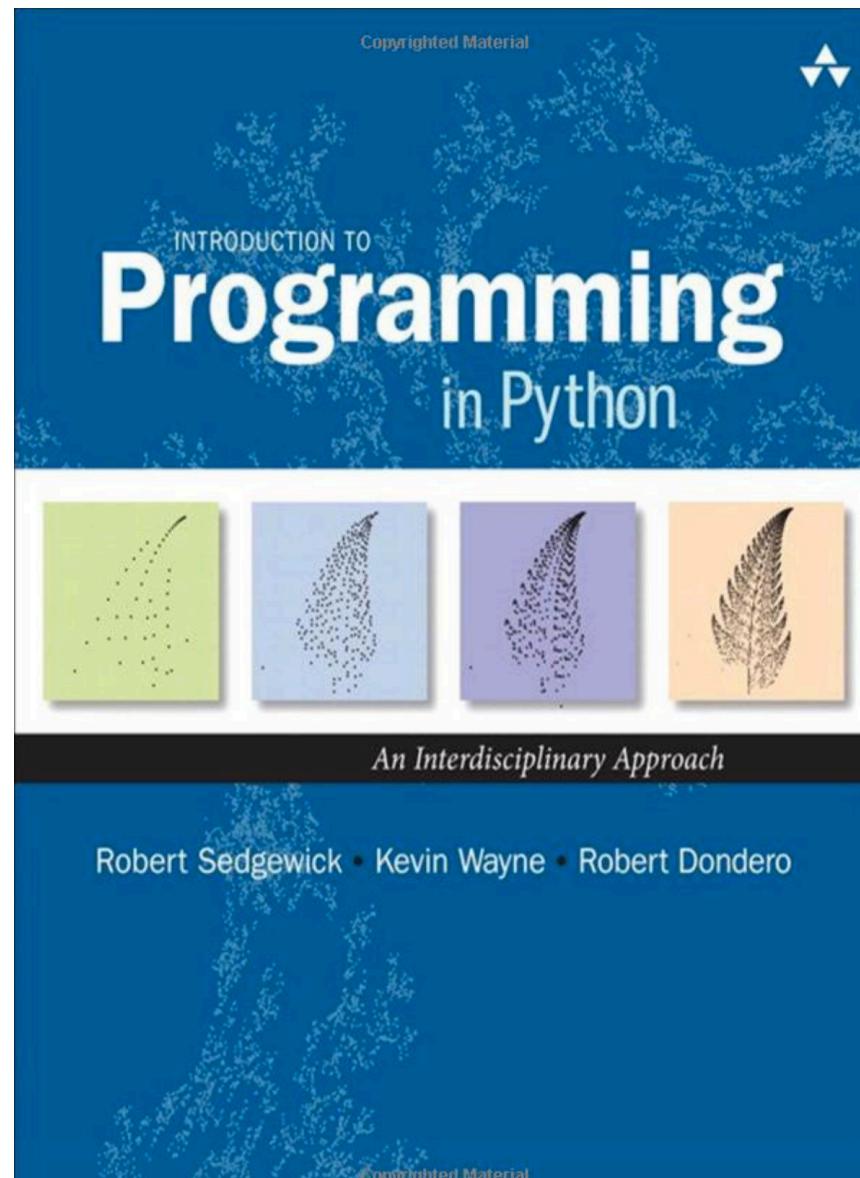


Parte II: Computación científica

Clase 11: Usando tipos de datos 2

Diego Caro
dcaro@udd.cl



Basada en presentaciones oficiales de libro *Introduction to Programming in Python* (Sedgewick, Wayne, Dondero).

Disponible en <https://introcs.cs.princeton.edu/python>

Ejemplo: cuadrados de Albert

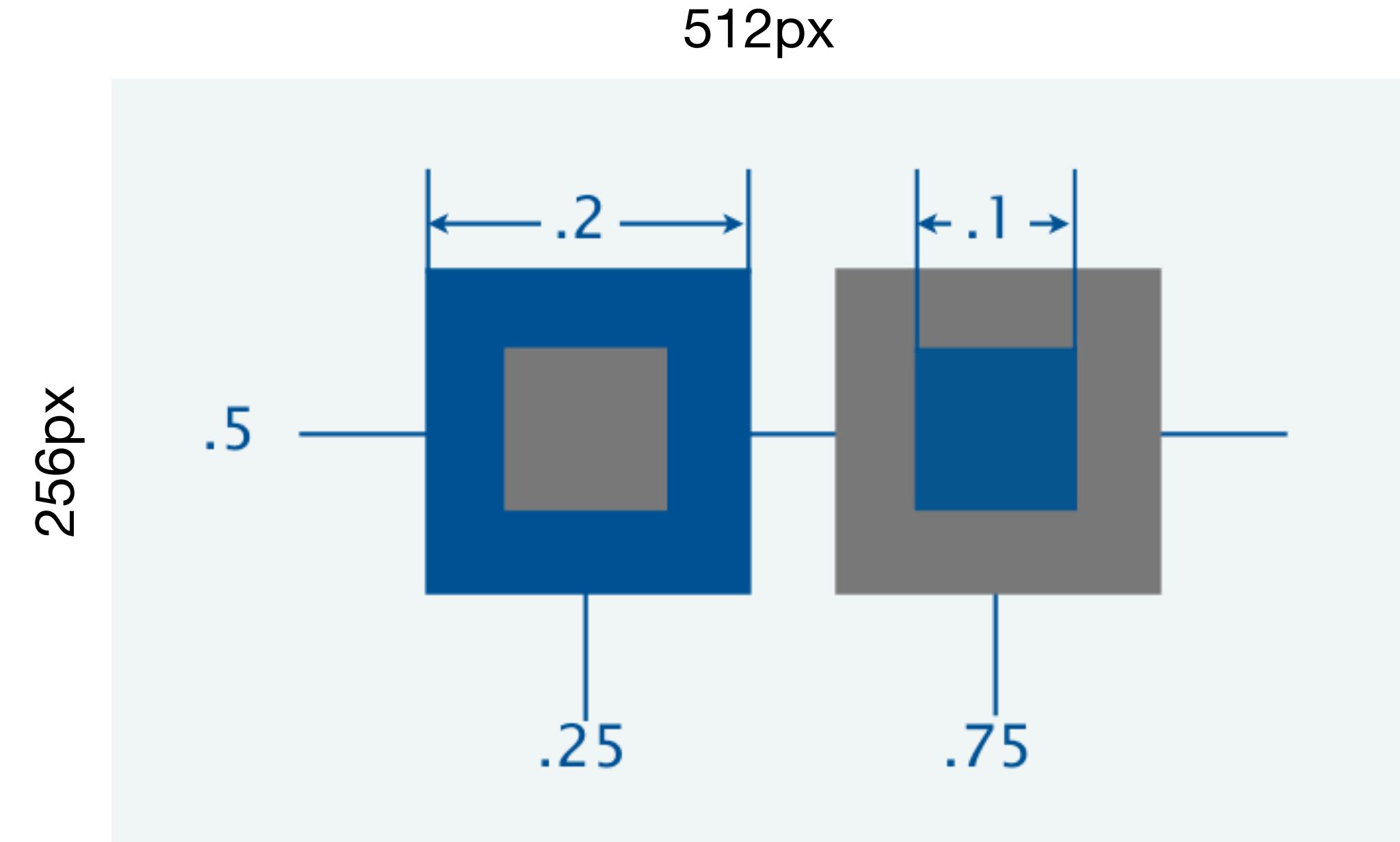
```
1 import stddraw → ¡Módulo para dibujar!
2 from color import Color
3
4 def readint(): return int(input())
5
6 r1 = readint()
7 g1 = readint()
8 b1 = readint()
9 c1 = Color(r1, g1, b1)
10
11 r2 = readint()
12 g2 = readint()
13 b2 = readint()
14 c2 = Color(r2, g2, b2)
15
16 stddraw.setCanvasSize(512, 256)
17 stddraw.setScale(.25, .75)
18
19 stddraw.setPenColor(c1)
20 stddraw.filledSquare(.25, .5, .2)
21
22 stddraw.setPenColor(c2)
23 stddraw.filledSquare(.25, .5, .1)
24
25 stddraw.setPenColor(c2)
26 stddraw.filledSquare(.75, .5, .2)
27
28 stddraw.setPenColor(c1)
29 stddraw.filledSquare(.75, .5, .1)
30
31 stddraw.show()
```

Crea primer color

Crea segundo color

Primer cuadrado

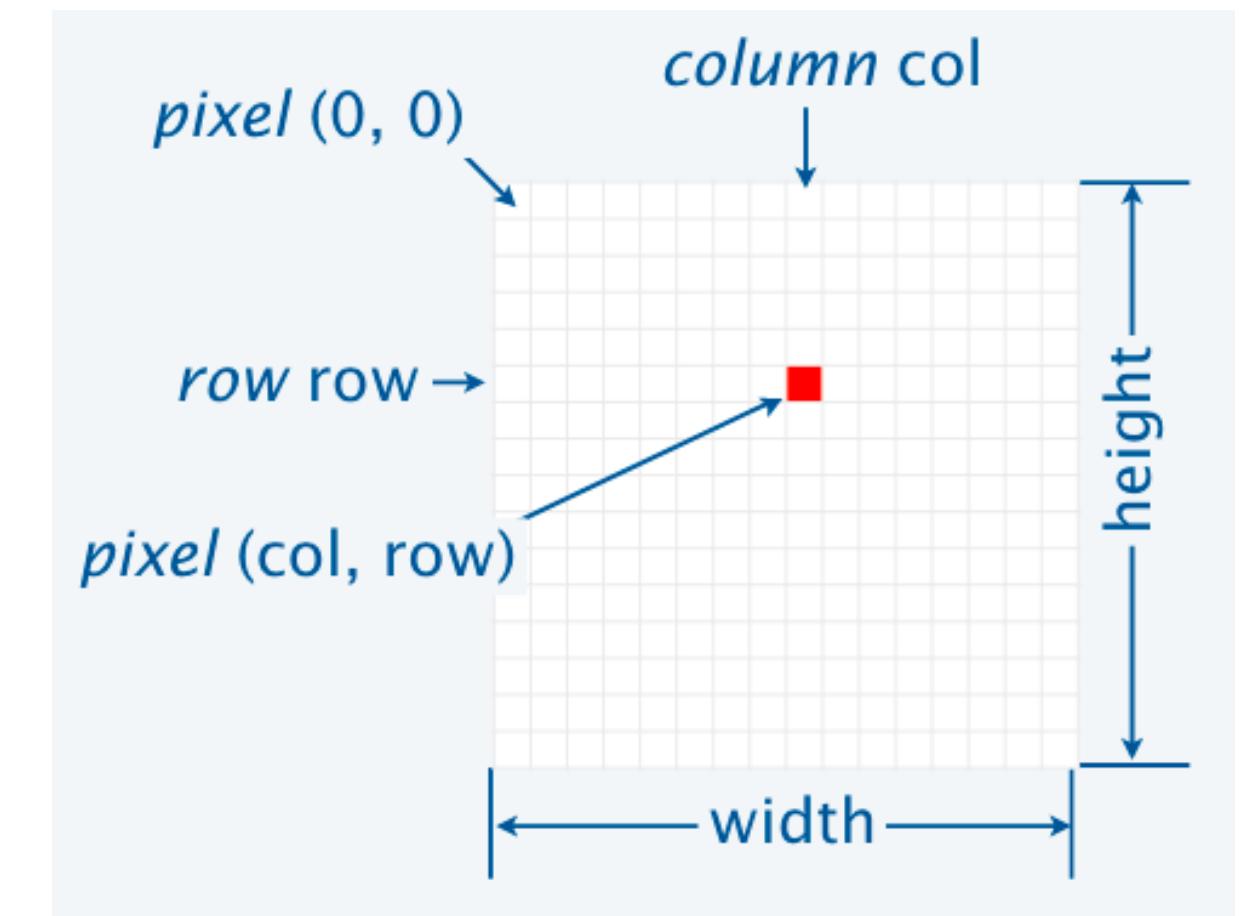
Segundo cuadrado



```
$ python3 alberssquares.py
0
64
128
105
105
105
```

API para imágenes

- Una imagen es un arreglo 2D de pixeles.
 - **Valores:** color de cada pixel.



class Picture

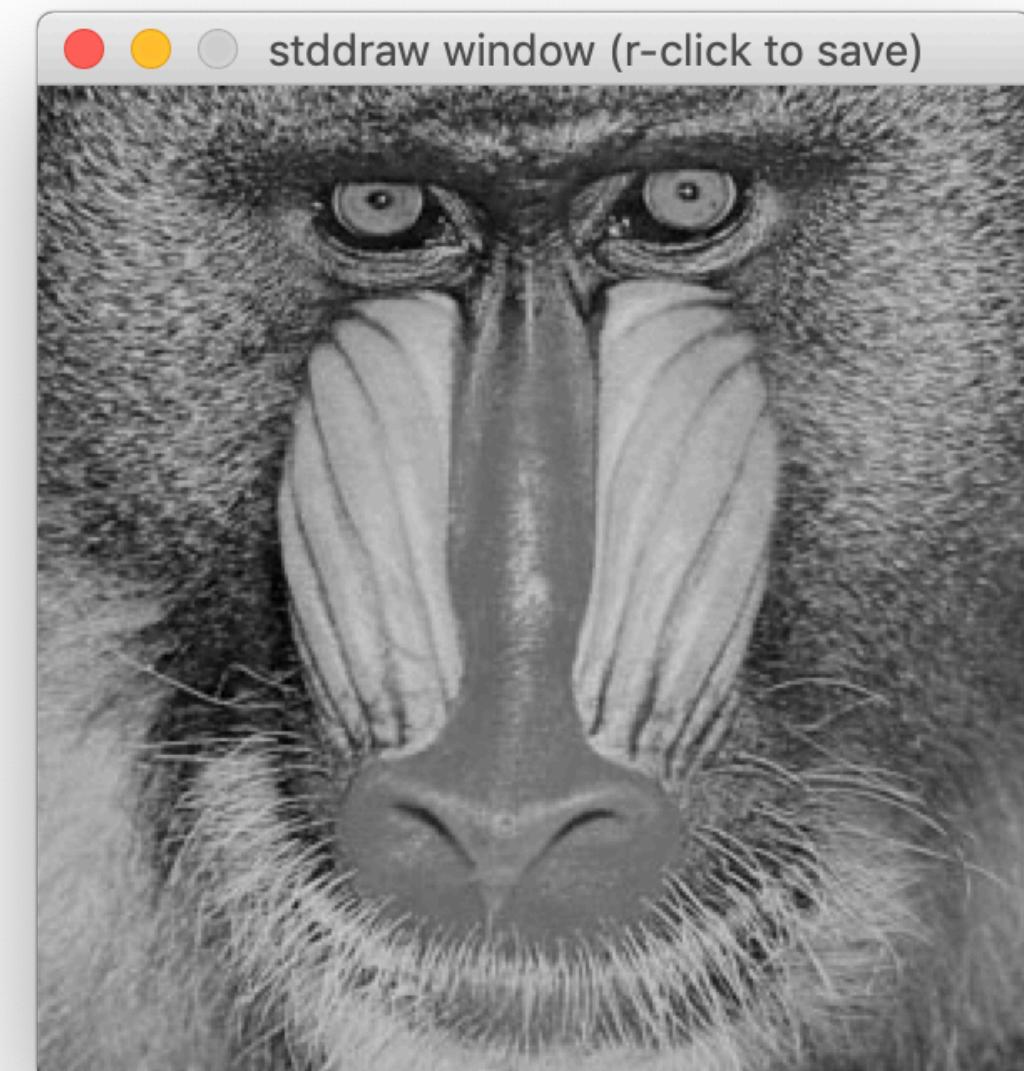
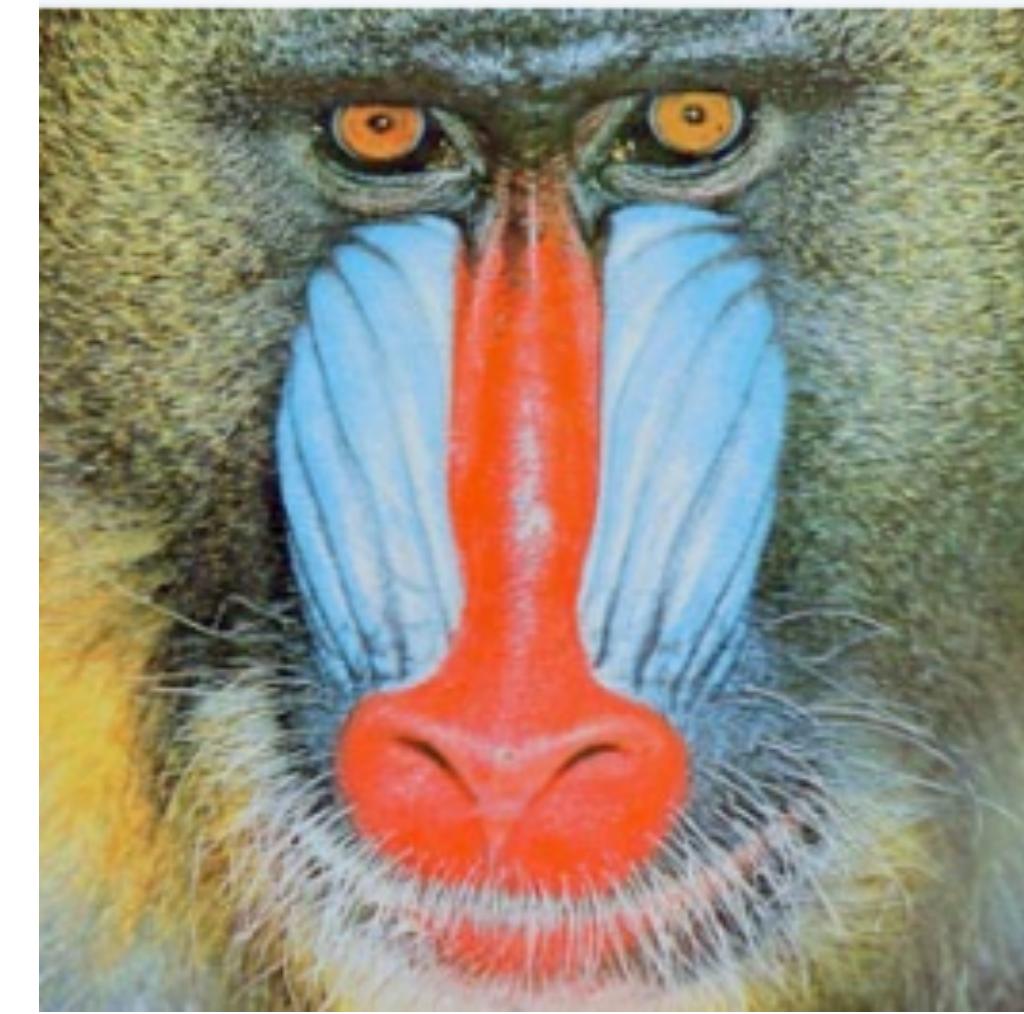
	Picture(filename)	Crea una imagen desde archivo
	Picture(w, h)	Crea una imagen vacía de w por h
int	width()	Ancho de la imagen
int	height()	Altura de imagen
Color	get(col, row)	Color del pixel (col, row)
	set(col, row, c)	Asigna color c a pixel en (col, row)
	show()	Muestra la imagen en una ventana
	save(filename)	Guarda imagen en archivo

Aplicación: convertir a blanco y negro

```
1 import sys  
2 import stddraw  
3 import luminance  
4 from picture import Picture  
5  
6 pic = Picture(sys.argv[1])          Crea imagen desde archivo  
7  
8 for col in range(pic.width()):  
9     for row in range(pic.height()):  
10        pixel = pic.get(col, row)  
11        gray = luminance.toGray(pixel)  
12        pic.set(col, row, gray)  
13  
14 stddraw.setCanvasSize(pic.width(), pic.height())  
15 stddraw.picture(pic)  
16 stddraw.show()
```

Transforma color en blanco y negro

Crea ventana y muestra imagen

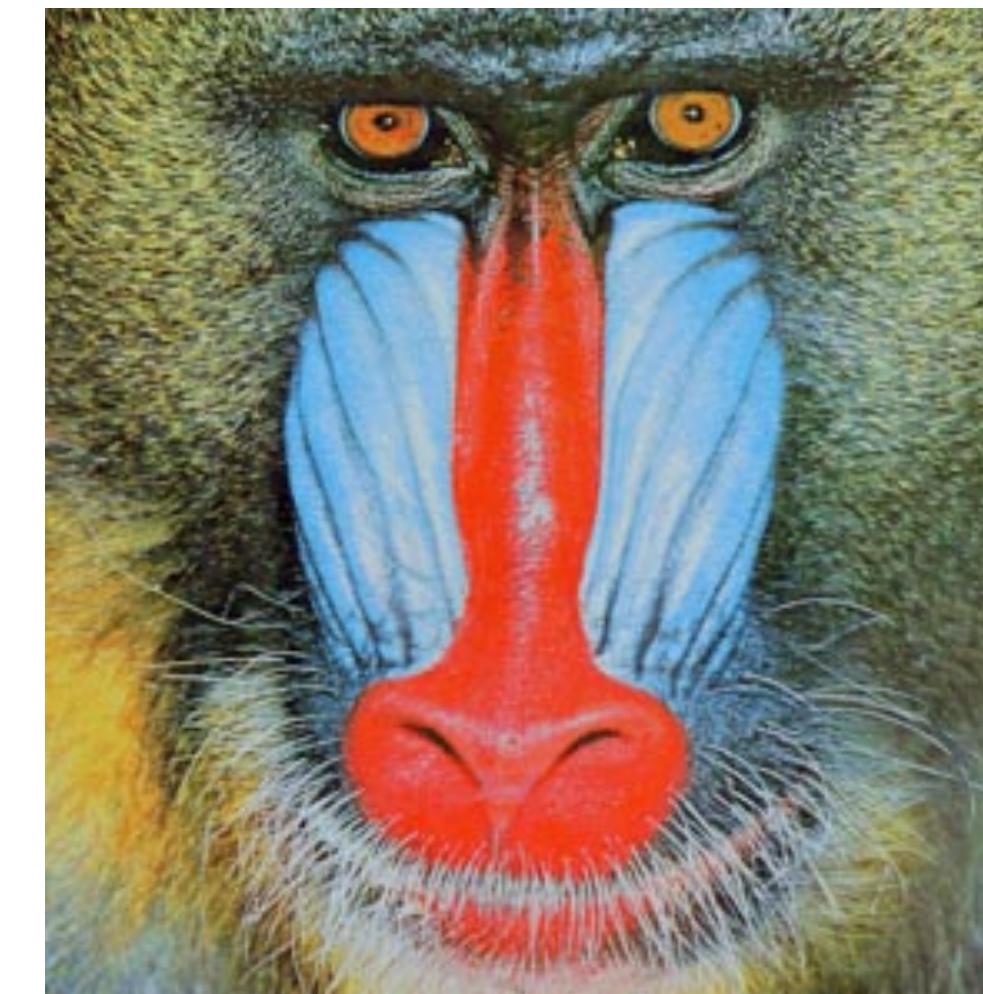
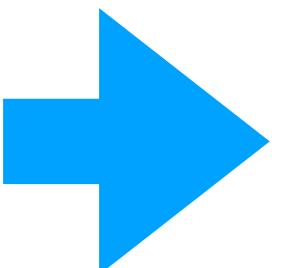


\$ python3 grayscale.py mandrill.jpg

Human-based python interpreter™

- ¿Qué hace el siguiente código?

```
for col in range(pic.width()):  
    for row in range(pic.height()):  
        pic.set(col, row, pic.get(col, row))  
  
stddraw.show()
```

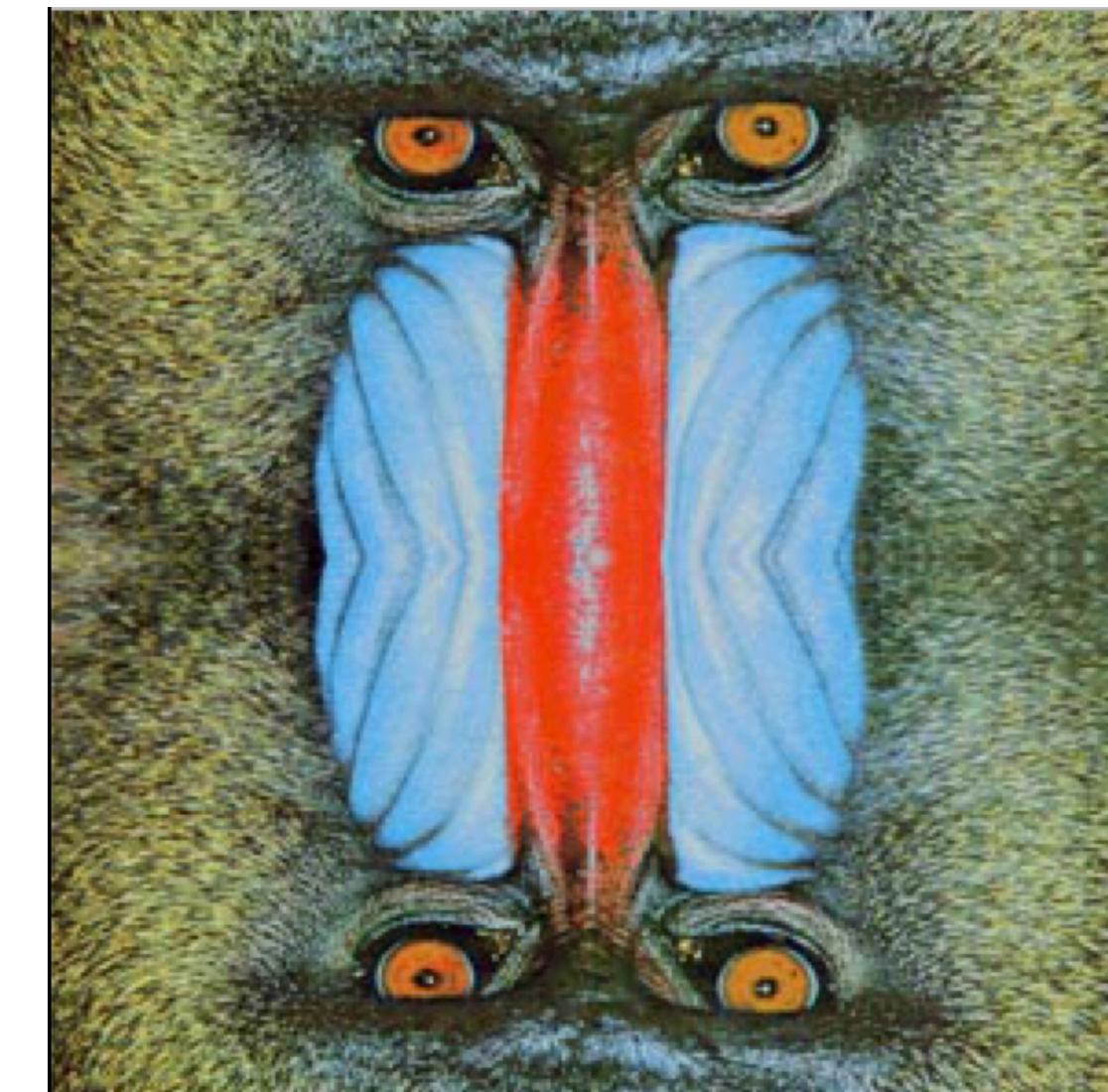
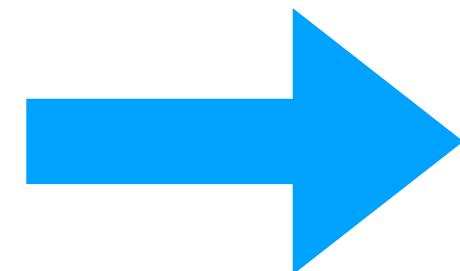
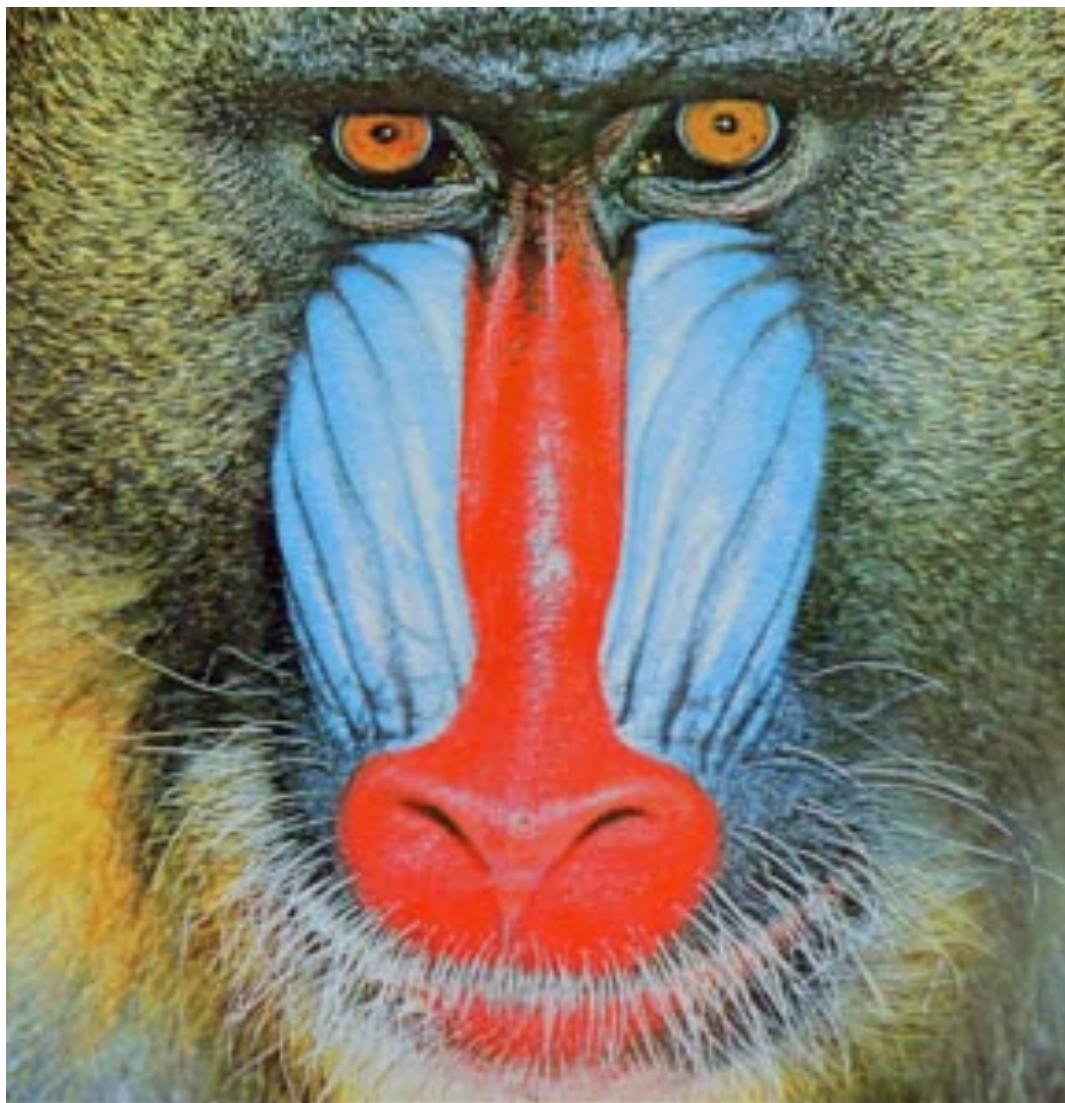


Human-based python interpreter™

- ¿Qué hace el siguiente código? (no tan fácil como el anterior)

```
for col in range(pic.width()):  
    for row in range(pic.height()):  
        pic.set(col, pic.height()-row-1, pic.get(col, row))  
  
stddraw.show()
```

Intenta reflejar verticalmente la imagen...
Pero falla.

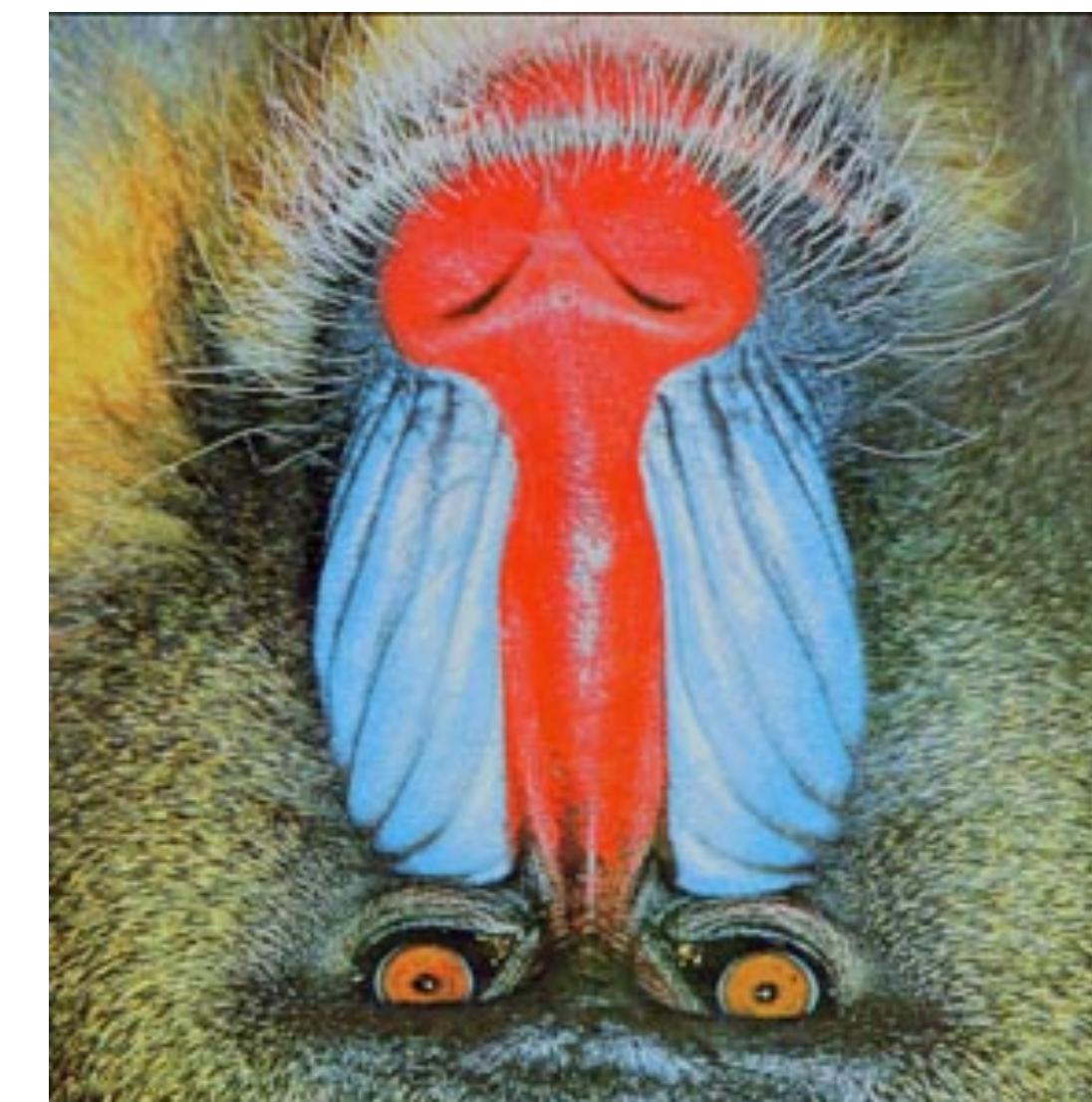
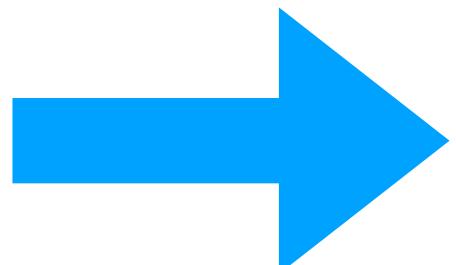


Human-based python interpreter™

- **Solución:** completar una imagen vacía.

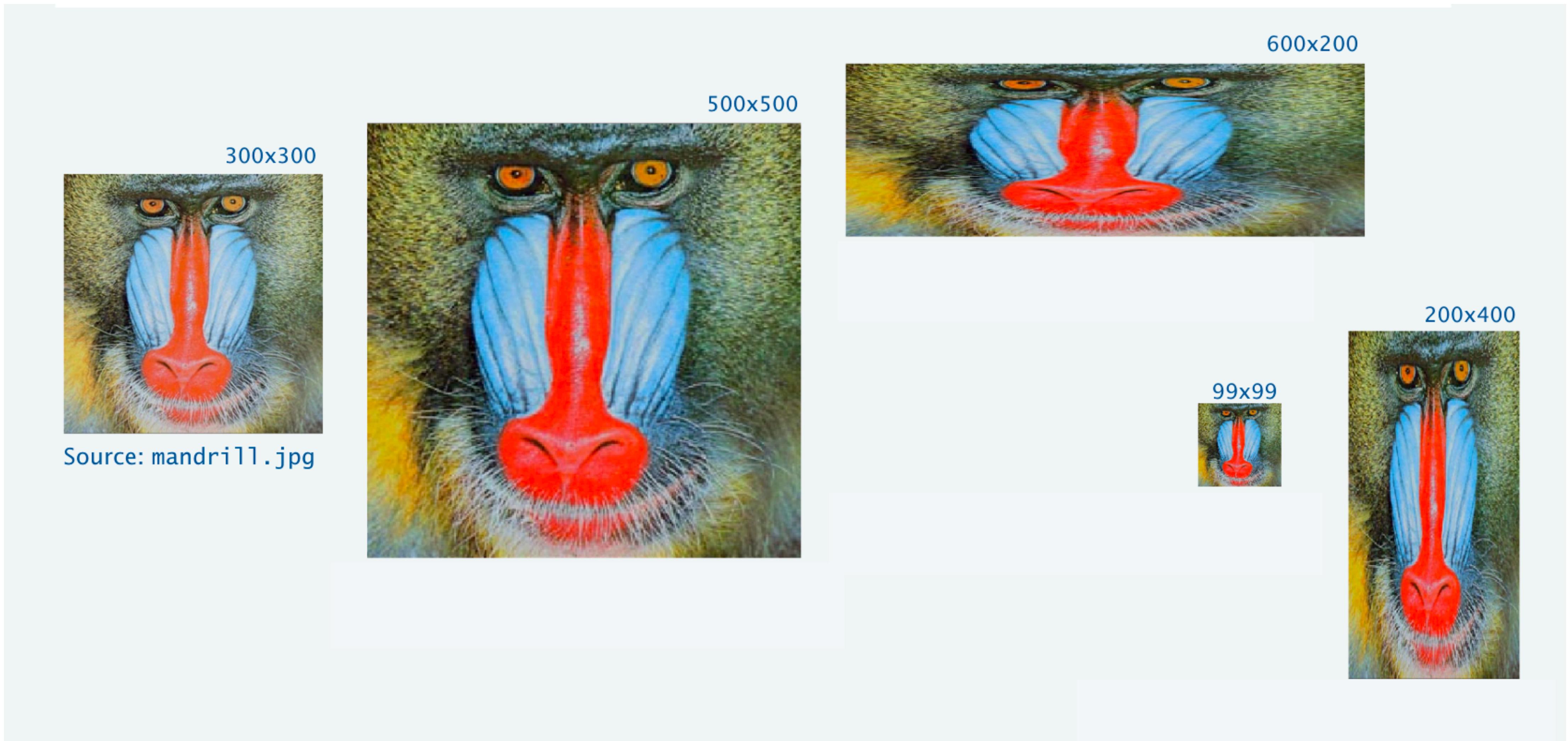
```
source = Picture(sys.argv[1])
target = Picture(source.width(), source.height())
for col in range(source.width()):
    for row in range(source.height()):
        target.set(col, source.height()-row-1, source.get(col, row))

stddraw.setCanvasSize(target.width(), target.height())
stddraw.picture(target)
stddraw.show()
```



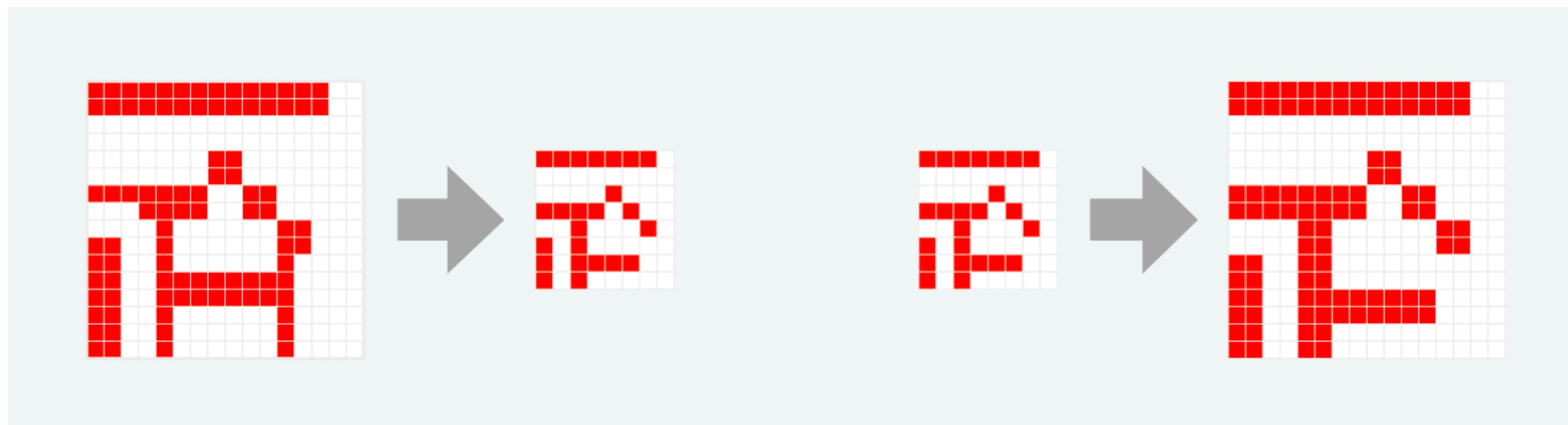
Ejemplo: escalar imagen

- Objetivo: escalar la imagen independiente del ancho y alto destino.



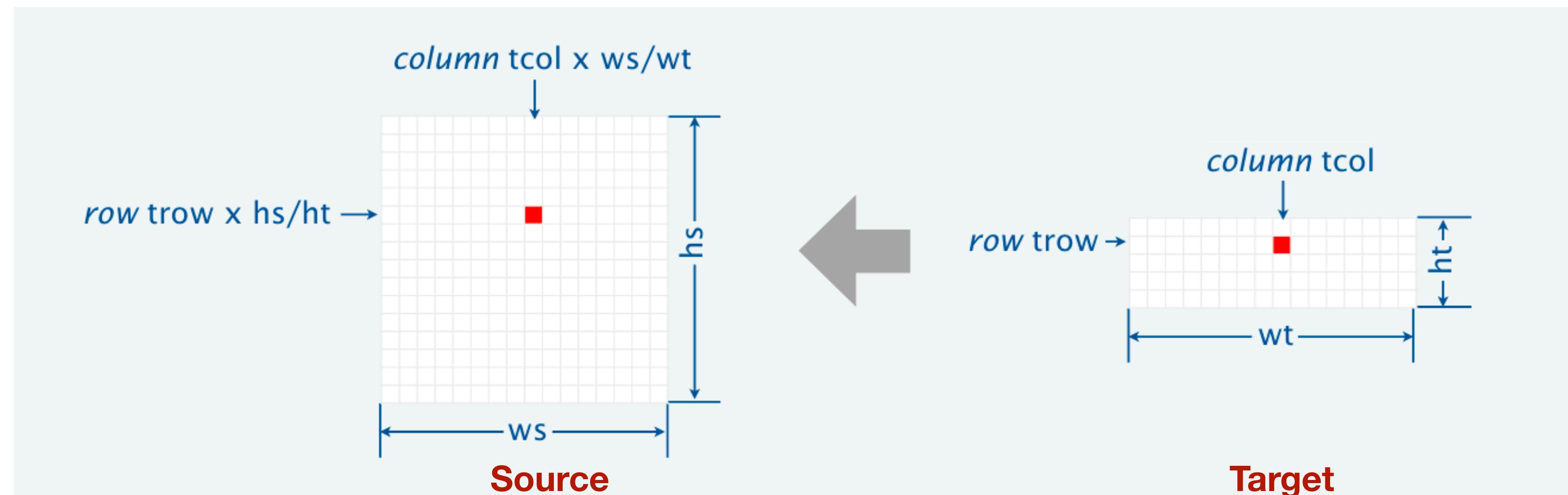
Ejemplo: escalar imagen

- Objetivo: escalar la imagen independiente del ancho y alto destino.
- Ejemplo de estrategia:
 - Achicar imagen eliminando filas y columnas alternadamente.
 - Agrandar imagen cuadruplicando cada pixel.
 - ¿Problema?



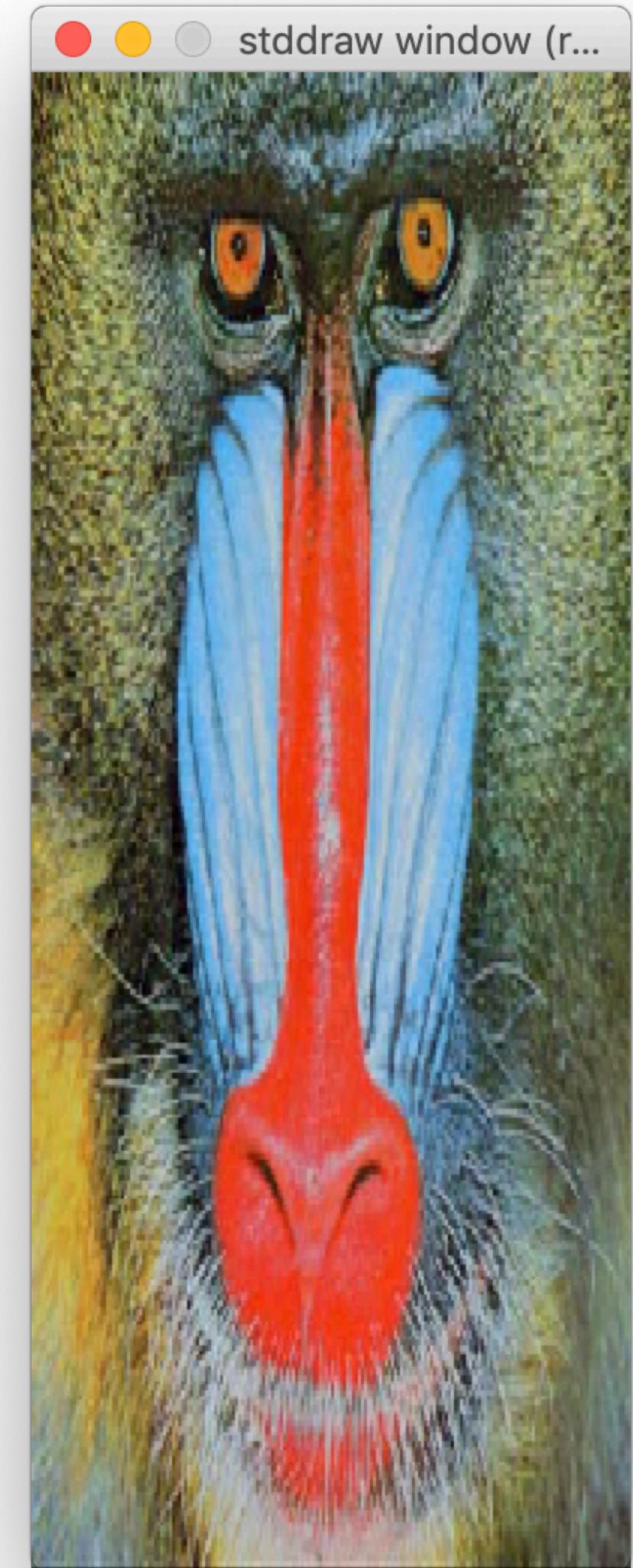
Ejemplo: escalar imagen

- Objetivo: escalar la imagen independiente del ancho y alto destino.
- **Estrategia** uniforme:
 - escalar columnas ubicando el índice en posición $ws//wt$
 - escalar filas ubicando el índice en posición $hs//ht$
- **Enfoque:** computar cada pixel en la imagen destino (del tamaño a escalar).

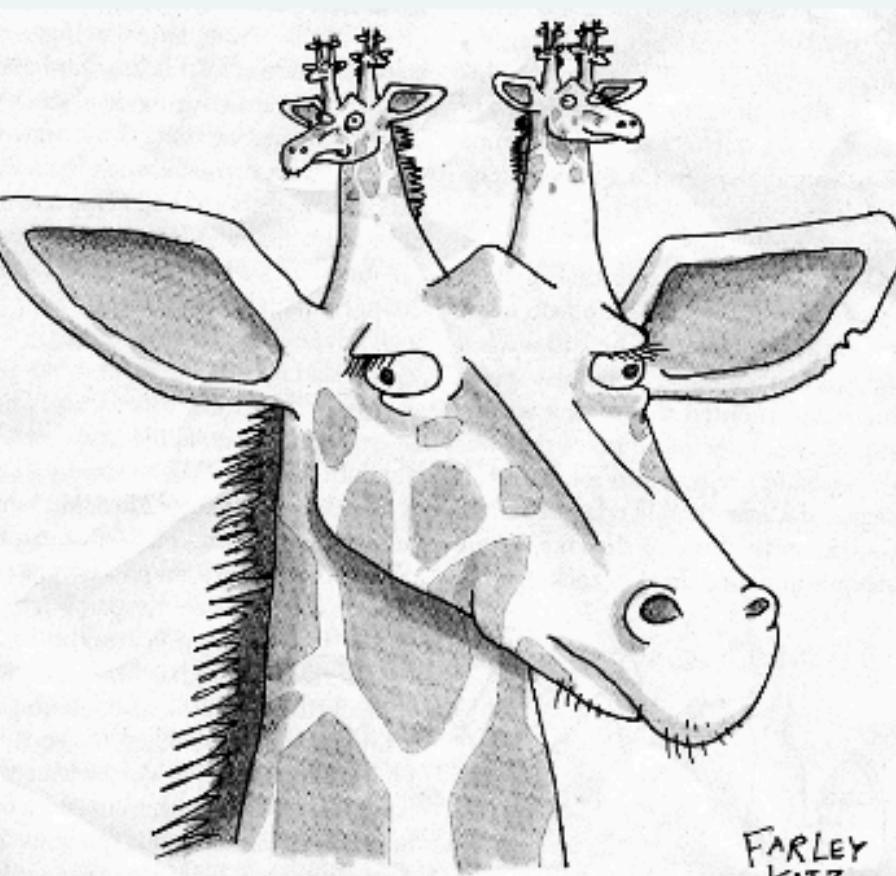
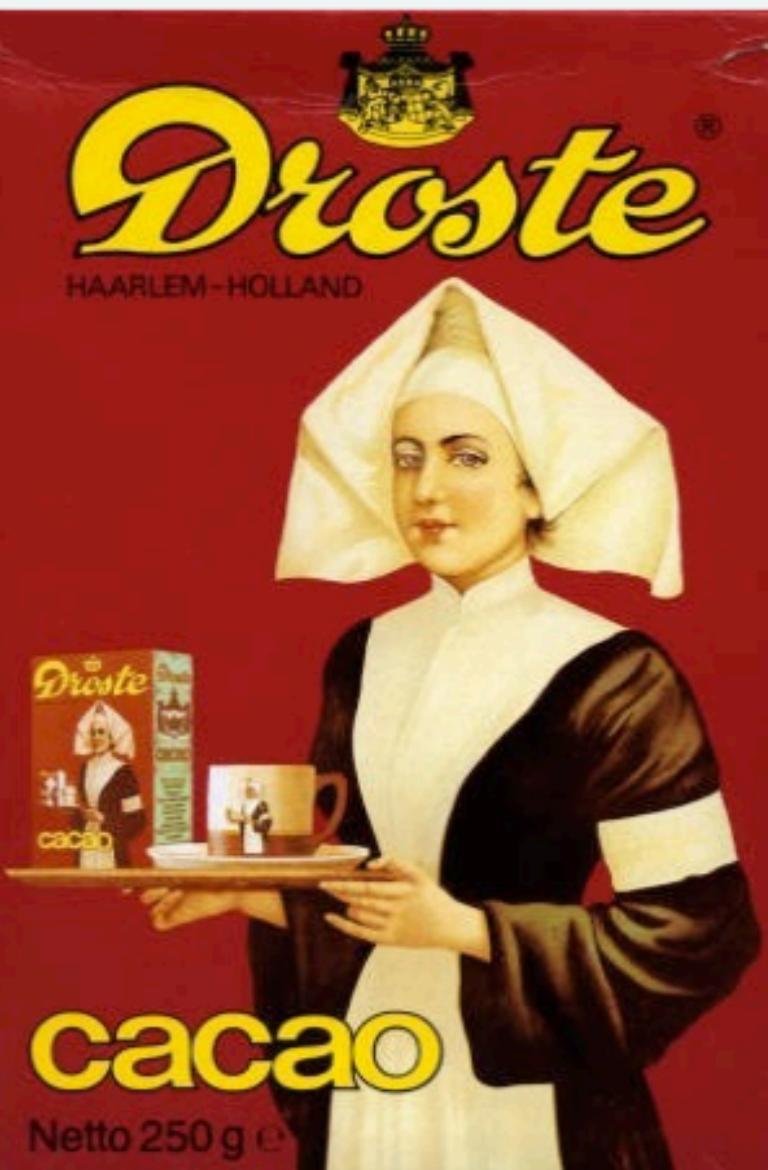


Ejemplo: escalar imagen

```
1 import stddraw
2 import sys
3 from picture import Picture
4
5 fileName = sys.argv[1]
6 w = int(sys.argv[2])
7 h = int(sys.argv[3])
8
9 source = Picture(fileName)
10 target = Picture(w, h)
11
12 for tCol in range(w):
13     for tRow in range(h):
14         sCol = tCol * source.width() // w | Posición en imagen origen
15         sRow = tRow * source.height() // h | Posición en imagen origen
16         target.set(tCol, tRow, source.get(sCol, sRow))
17
18 stddraw.setCanvasSize(w, h)
19 stddraw.picture(target)
20 stddraw.show()
```



Recursividad, imágenes y arte



WEEKEND Arts FINE ARTS LEISURE

N

E41

FRIDAY, DECEMBER 15, 2006

The New York Times

Design Life Now
Cooper-Hewitt
National Design
Museum
Exhibitions
include
"Design
by Hand,"
"Fruit
of Design."
See page A1.

Fruits of Design, Certified Organic

ARTS **DESIGN** **ENTERTAINMENT** **FOOD & DRINK** **HOME & GARDEN** **TRAVEL** **WEDDINGS**

By TONI TISONOWICZ at the Cooper-Hewitt National Design Museum. This is not the former Andrew Carnegie mansion up to its neck in mostly American designs from the last three years. Like its predecessor, "Design Life Now," the museum's third National Design Triennial, is a mixed affair that threatens to be a volatile, contradictory, over-explaining field but fails to call it to order.

The exhibition has been organized by the Cooper-Hewitt curators Harriet Hirshorn, Ellen Lupton and Matilda McQuaid and a guest, Brooke Hodge, a curator at the Museum of Contemporary Art, Los Angeles. Once again the Triennial answers the question "What's design?" with the elusive catchall "What's design?" Covering so many bases as equivalently, it never gets around to tackling the weightier questions of "What is design?" or, more to the point, "What is design for?" It refuses to take sides on the issue of whether design should aim for social or environmental benefit, serve a relatively decorative purpose, etc., the show's skills are many, even if you have to work for them. The displays here range from genius to schlock, decaf to drizzling. They cover life-extending innovation, completely frivolous reinterpretations of received ideas (such as a new take on the hula hoop) and more serious types of reception from your own country court. Photo booths, mosaics, furniture, documents, artworks, and more, all design according to this schema.

The main point comes across loud and clear: design means every aspect of contemporary life. Every-
thing that exists is designed, whether natural or cultural.

While all of nature's designs are intelligent, whether
they're the Bible, the house, food or much

Continued on Page 21

Look Inside the Yale Book of Quotations
From "Postcards From Mars," a selection
of the best holiday books.

The Gifts to Open Again and Again

BOOKS **ARTS** **DESIGN** **ENTERTAINMENT** **FOOD & DRINK** **HOME & GARDEN** **TRAVEL** **WEDDINGS**

I've made my list, and I'm checking it twice. It's a list of qualities that make the ideal holiday book, and I'm really looking forward to this year's Christmas gift. A gift book should be both unique and useful, yet you'll want to be sure to keep it on the shelf for years to come. You'll want to be sure that the one you never know you'll need, it should either be expensive and large, or cheap and small. And no matter what, it should not require a sustained attention, which is impossible during the yuletide season. My gift selections, chosen entirely at random but with exquisite taste, satisfy at least two of these requirements.

Let's open the big present first. The season's whopper, in every way, is "New York 2000," the fifth installment in Robert A. M. Stern's architectural history of New York. The series starts in 1880, when 10 stories defined a skyscraper, and has now caught up to the millennium. Taken together, the volumes make an amazingly fascinating family scrapbook for New Yorkers, who can now view today's pictures of the Twin Towers and look forward to the many more hours of pages and thousands of photographs, in the big, sweeping New York of the Uptown Building, countless projects and the new Tweed Commission. 1,500 pages and 16 pounds 12 ounces. "New York."

Continued on Page 48

Black, White and Read All Over Over

BY RANDY KENNEDY

In one of Jorge Luis Borges's best-known short stories, "Pierre Menard, Author of the Quixote," a 20th-century French writer sets out to compose a verbatim copy of Cervantes's 17th-century masterpiece simply because he thinks he can, originally perhaps not being all that cracked up to be.

He manages two chapters word for word, a spontaneous duplicate that Borges's narrator finds to be "infinitely richer" than the original because it contains all manner of new meanings and inflections, wrenching us it from its proper time and context.

When a young Turkish artist named Serkan Odeyli set out recently to practice his skills as a copyist—a survivor, as he says—his goals were a little less ambitious than channelling Cervantes. He simply wanted to draw and see printed a faithful copy of all the type and pictures planned for a broadsheet page of this newspaper: this very page you are reading right now, which shows his version of the page you are reading right now, which shows his version of his version of the page you are reading right now, which...

Do not be alarmed: There has been no break in the

Black, White and Read All Over Over drawing right now, showing his drawing of the page you are reading right now, showing...

Apart-time-newspaper continues.

Mr. Odeyli, a 33-year-old artist who lives and works in Istanbul, did not propose this exercise in academic surrealism because of a particular love of calligraphy or newspapers or, for that matter, even drawing, which he admits he is not very good at. The

Continued on Page 41

Divine and Devotee Meet Across Hinges

WASHINGTON — For centuries, died

Apollonia ASAP. She'll be ruling in

the West, King St. Matthew, co-bishop, in

April. For Help me, your taxes in

everywhere knows a prayer to

St. Roch, paroxysm friend

play, as good as a flea

shot, and that Rembrandt will

never strike when St. Barbara

is on the job.

Most important, for close

and intimate problems,

real cushion, incomparable art, side-

sides of soul — there's the Virgin. They

right she's on the till-time hot line off-

ing gentle attention and prudent ad-

To European Christians half a millennium ago, the Virgin and a raft of familiar saints were the exalted personae in a kind of celestial software system, available to all believers. And one quick way to access its benefits was through devotional printing of the kind found in "Prayers and Portraits: Unfolding the Netherlandish Print" at the National Gallery of Art.

Probably nothing in Western art

comes closer to formal perfection than

these pictures, produced by the likes of

Jan van Eyck, Roger van der Weyden

and Hugo van der Goes across an area

that now encompasses the Netherlands,

Belgium, Luxembourg and parts of

France. These painters were pictorial magicians, creating visual worlds, centrally abstract and miraculously realistic, of peerless breadth.

You do all of this in one glance at the

40 double-page paintings, oil and grisaille,

here. Then you learn gradually as you move through the show how these paintings

have been remade and remade, broken up and record garbed, over

the centuries, with the result that few survive in their intended form.

"Prayers and Portraits" is an attempt

to restore that form, at least a few of

them. It brings art historians and art

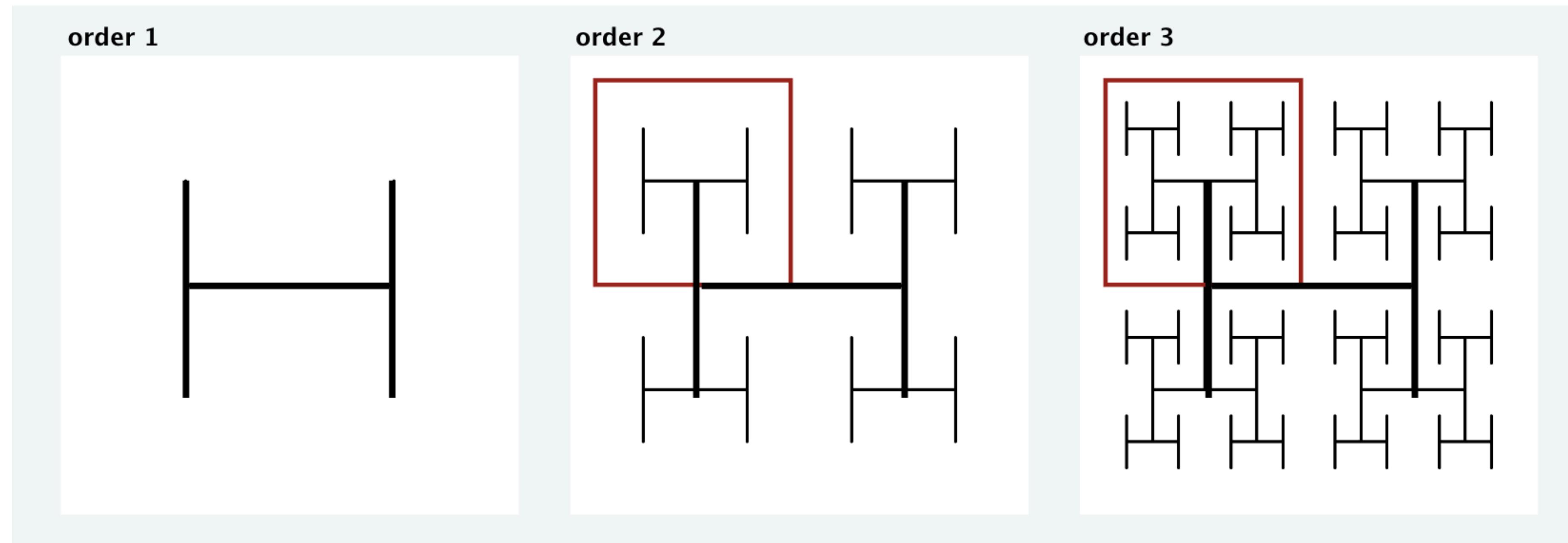
Continued on Page 41

Prayers and Portraits:
Unfolding the Netherlandish
Print

Two panels of an
early Netherlandish
painting by Master
Salomon, left, are
exhibited in an
exhibition on the
National Gallery of
Art in Washington
through Feb. 4.

H-Trees

- Un H-Tree de orden n:
 - Si n es 0, hacer nada.
 - Dibujar una H en el centro
 - Dibujar cuatro H-trees de orden n-1, de la mitad del tamaño y centradas en los bordes de la H



Implementación H-Tree

```
1 import stddraw
2 import sys
3
4 def draw(n, lineLength, x, y):
5     if n == 0:
6         return
7     x0 = x - lineLength/2
8     x1 = x + lineLength/2
9     y0 = y - lineLength/2
10    y1 = y + lineLength/2
11
12    stddraw.line(x0, y, x1, y)
13    stddraw.line(x0, y0, x0, y1)
14    stddraw.line(x1, y0, x1, y1)
15
16    draw(n-1, lineLength/2, x0, y0)
17    draw(n-1, lineLength/2, x0, y1)
18    draw(n-1, lineLength/2, x1, y0)
19    draw(n-1, lineLength/2, x1, y1)
20
21 def main():
22     n = int(sys.argv[1])
23     stddraw.setPenRadius(0.0)
24     draw(n, .5, .5, .5)
25     stddraw.show()
26
27 if __name__ == '__main__':
28     main()
```

