PostScript Language

서울대학교 컴퓨터공학부
김명수
http://cse.snu.ac.kr/mskim
http://3map.snu.ac.kr

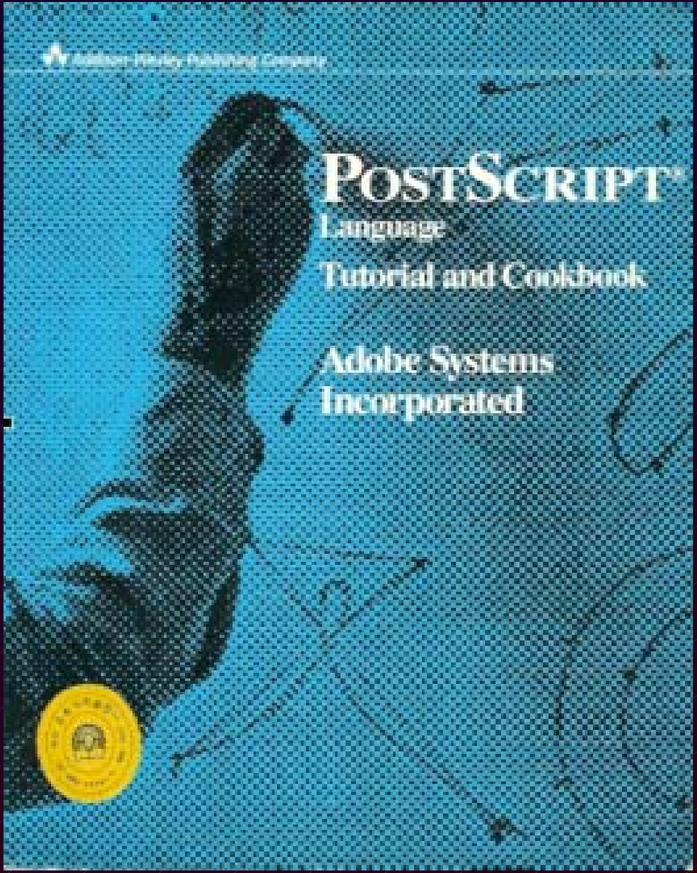
PostScript Language

- Adobe사에서 개발한 페이지 서술 언어
- 2차원 정적 그래픽스 라이브러리
- 주로 레이저 프린터 출력용으로 많이 사용
- 스택기반의 언어
- “2 3 +”과 같은 형태의 Postfix 문법을 사용

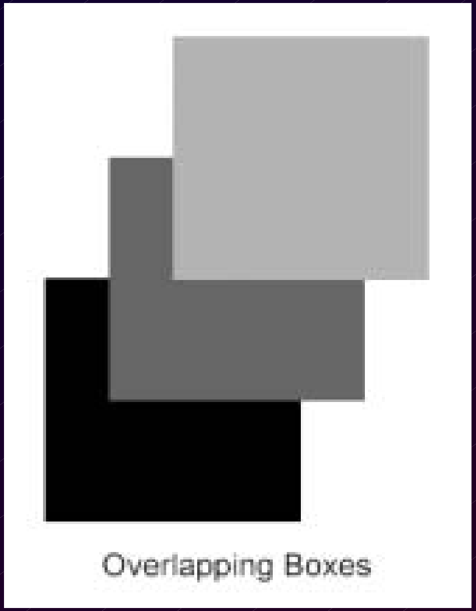
PostScript 관련 사이트

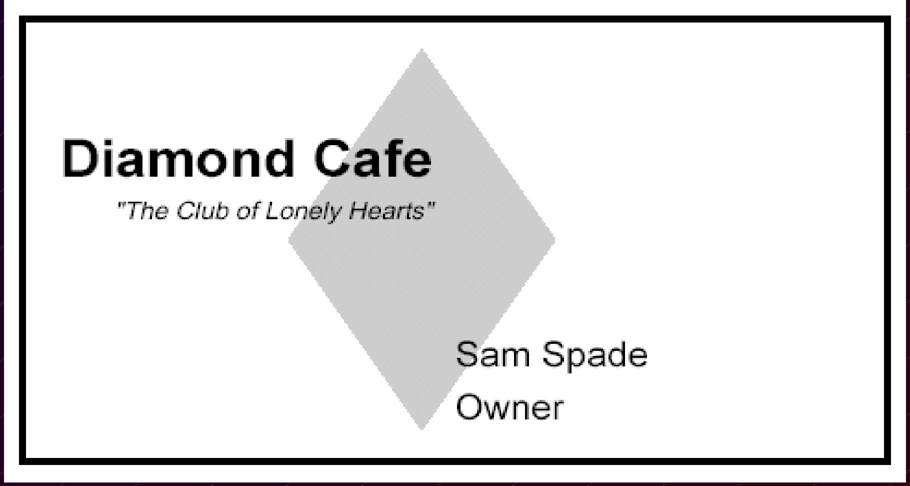
- BLUE BOOK을 얻는 사이트
  http://www-cdf.fnal.gov/offline/PostScript/BLUEBOOK.PDF
- BLUE BOOK의 예제들을 얻는 사이트
  http://www.science.uva.nl/~robbert/ps/bluebook/
- AFPL Ghostscript와 GSview를 얻는 사이트
  http://www.cs.wisc.edu/~ghost

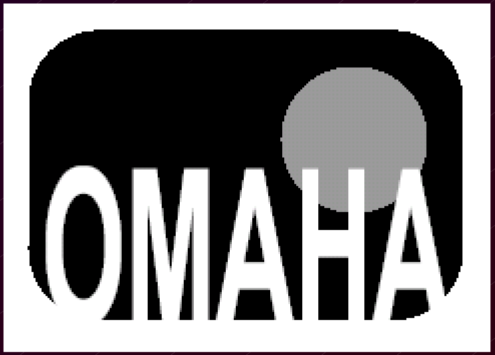
BLUE BOOK


Addison-Wesley Publishing Company
POSTSCRIPT®
Language
Tutorial and Cookbook
Adobe Systems
Incorporated

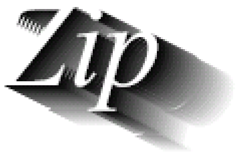
PostScript 예제





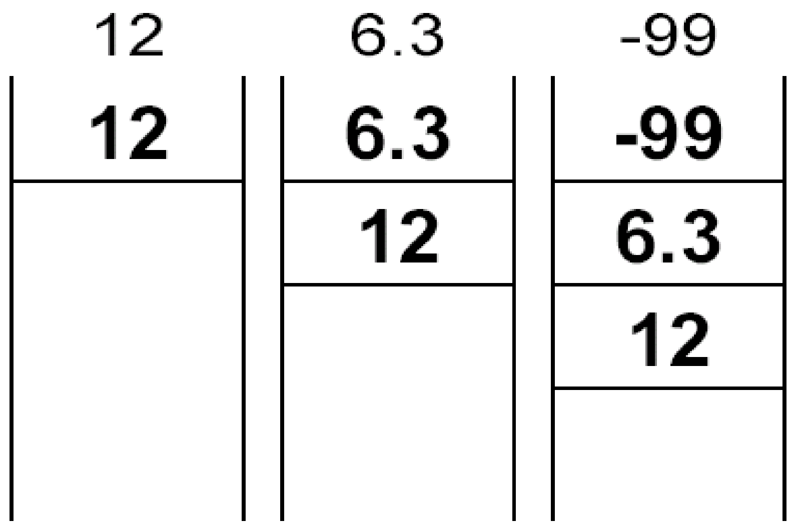


PostScript 프로그램 예제



```
/Times-Italic findfont  30 scalefont setfont

/printZip

 { 0 0 moveto  (Zip) show} def
320  400  translate

.95  –.05  0        % start  incr.  end
{setgray  printZip –1 .5 translate } for

1 setgray printZip

showpage
```

PostScript 스택



12 6.3 -99

12

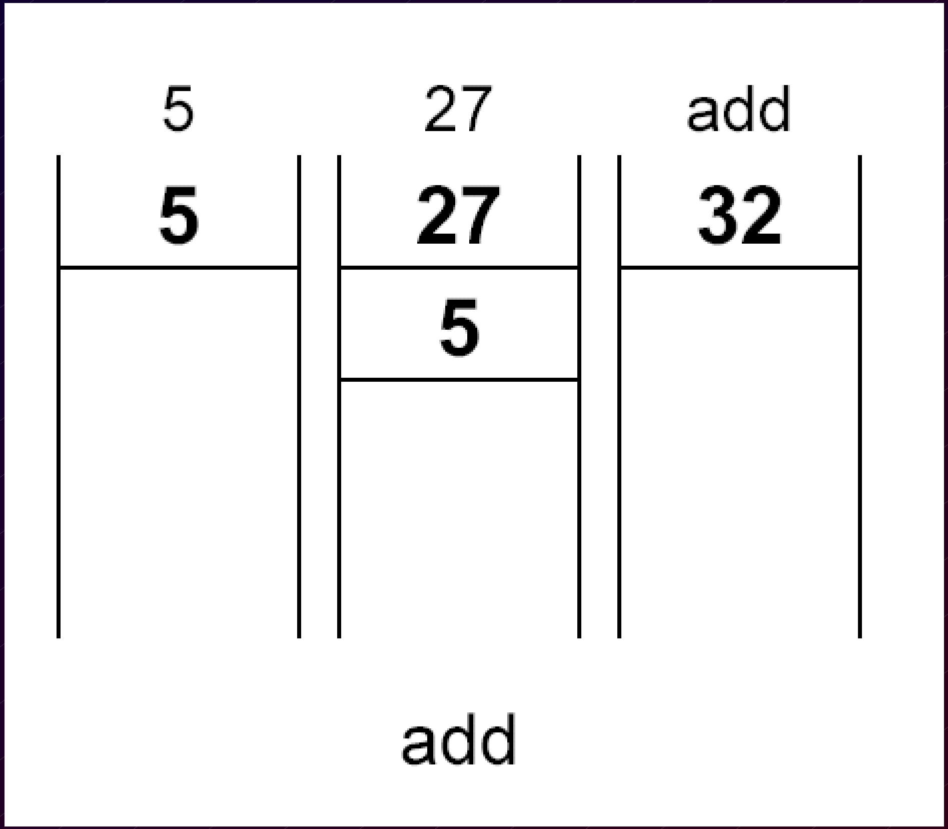
6.3
12

-99
6.3
12

POSTSCRIPT Stack

12 6.3 –99

PostScript 연산



5 27 add

5

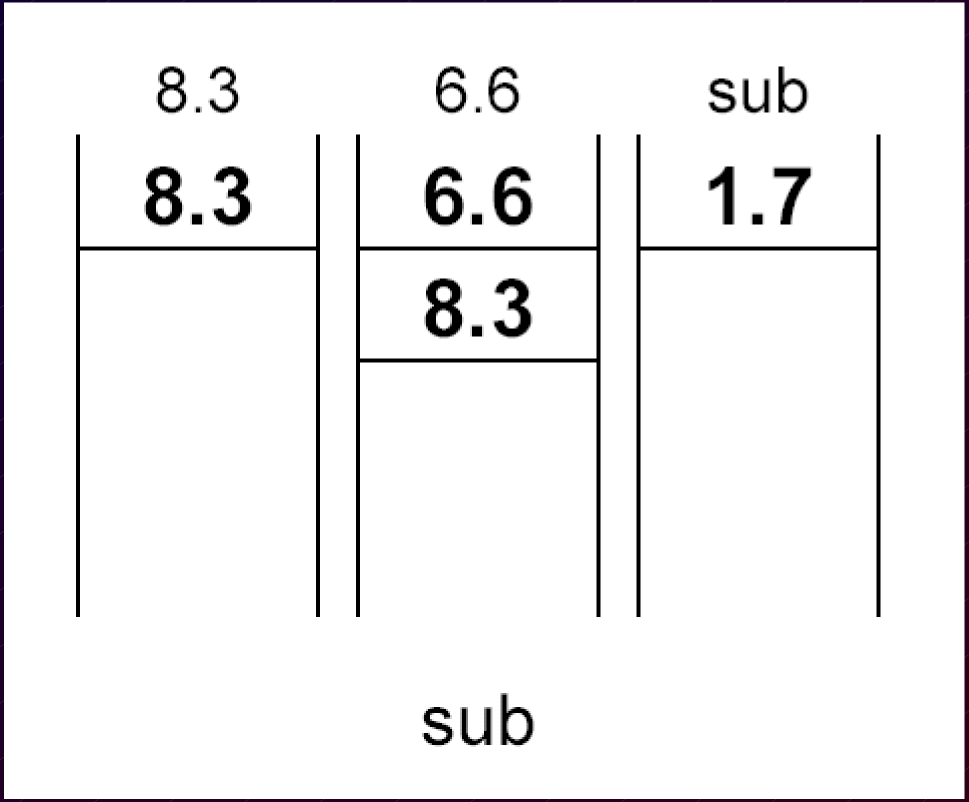
27 5

32

add

5 27 add

PostScript 연산



8.3 6.6 sub

8.3

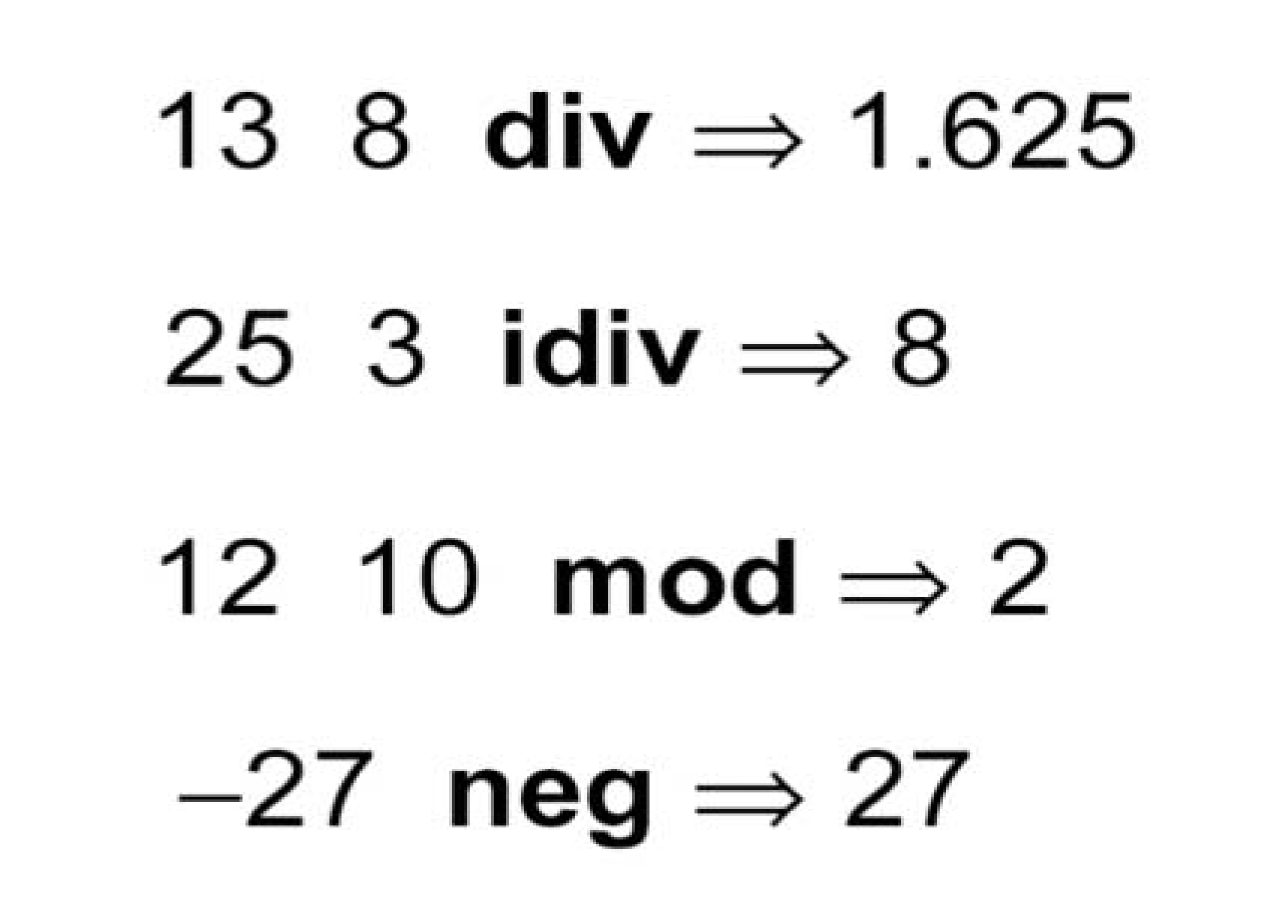
6.6
8.3

1.7

sub

8.3 6.6 sub

PostScript 연산



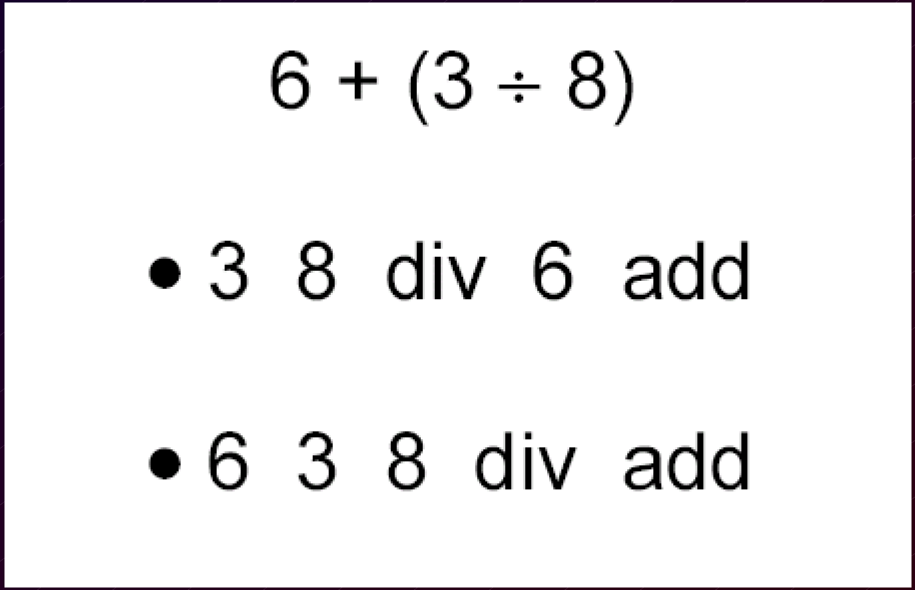
13 8 **div** ⇒ 1.625

25 3 **idiv** ⇒ 8

12 10 **mod** ⇒ 2

−27 **neg** ⇒ 27

PostScript 연산

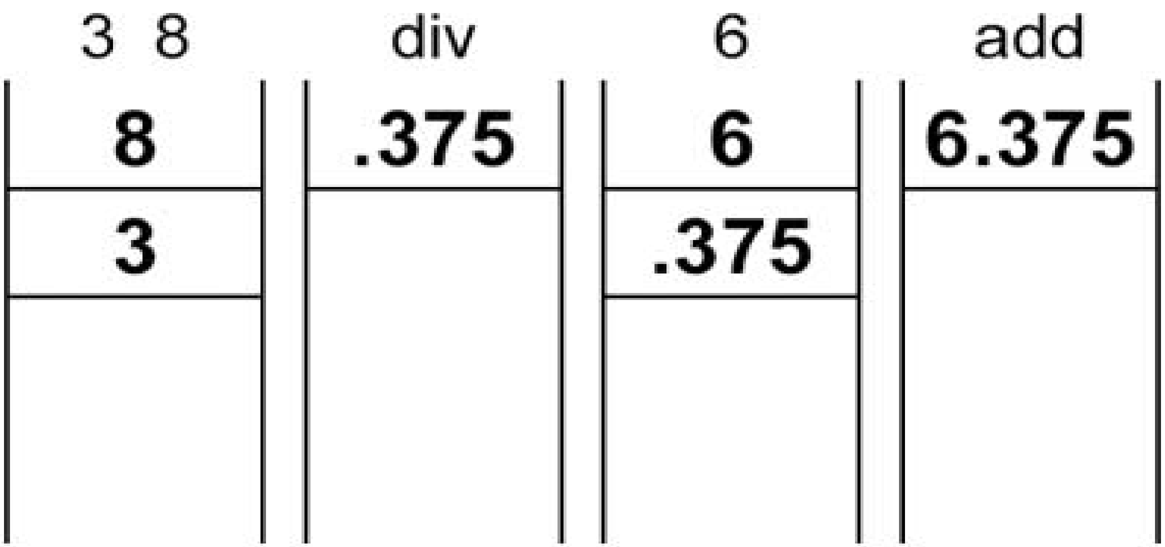


6 + (3 ÷ 8)

- 3 8 div 6 add
- 6 3 8 div add

PostScript 연산

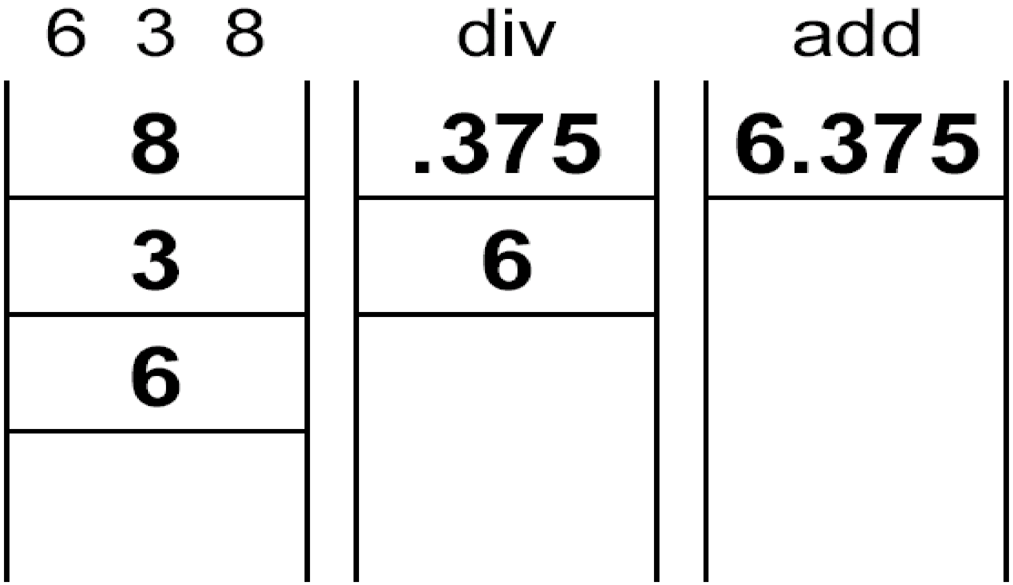
- 3 8 div 6 add



6+3/8, Example 1

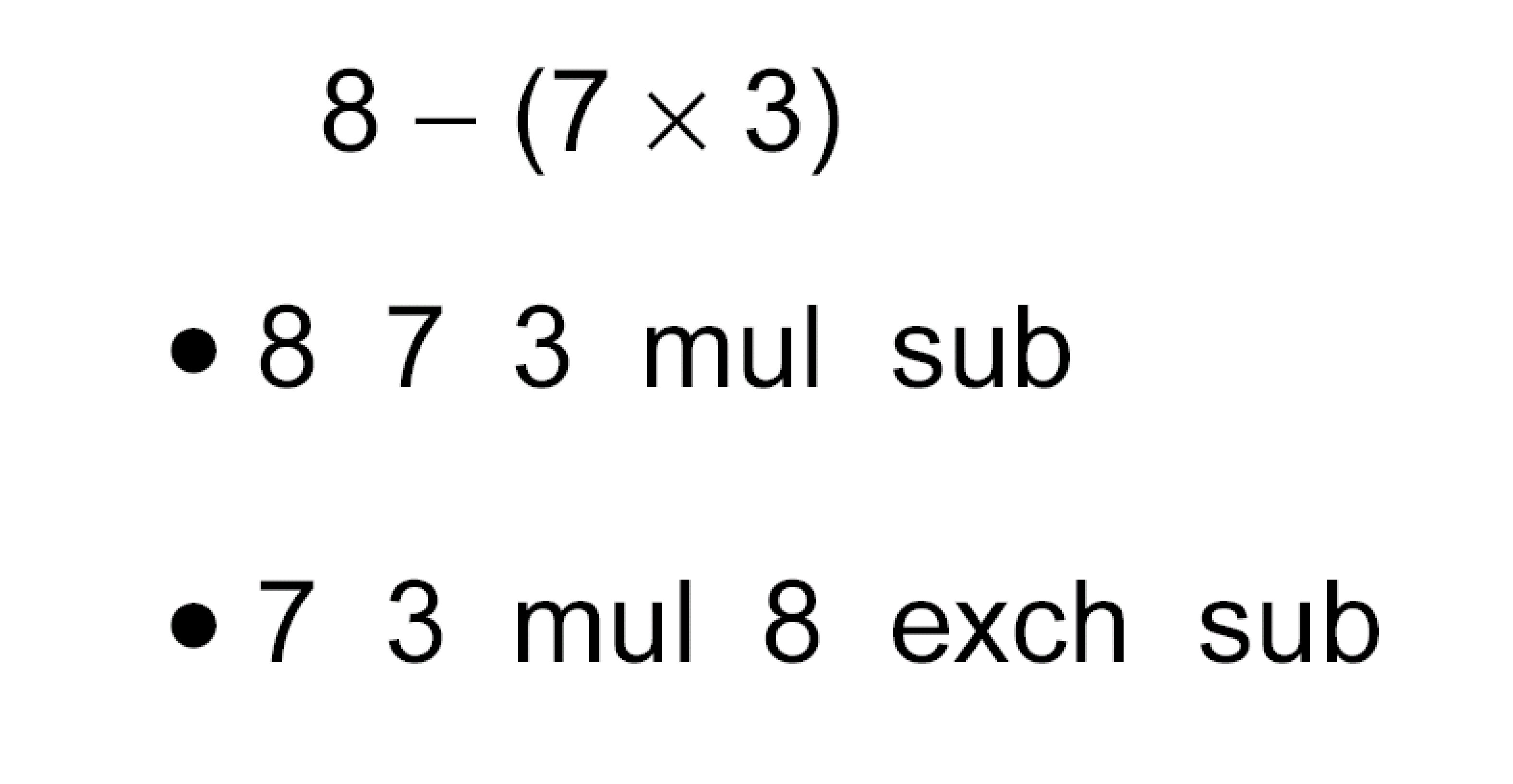
PostScript 연산

- 6 3 8 div add



6+3/8, Example 2

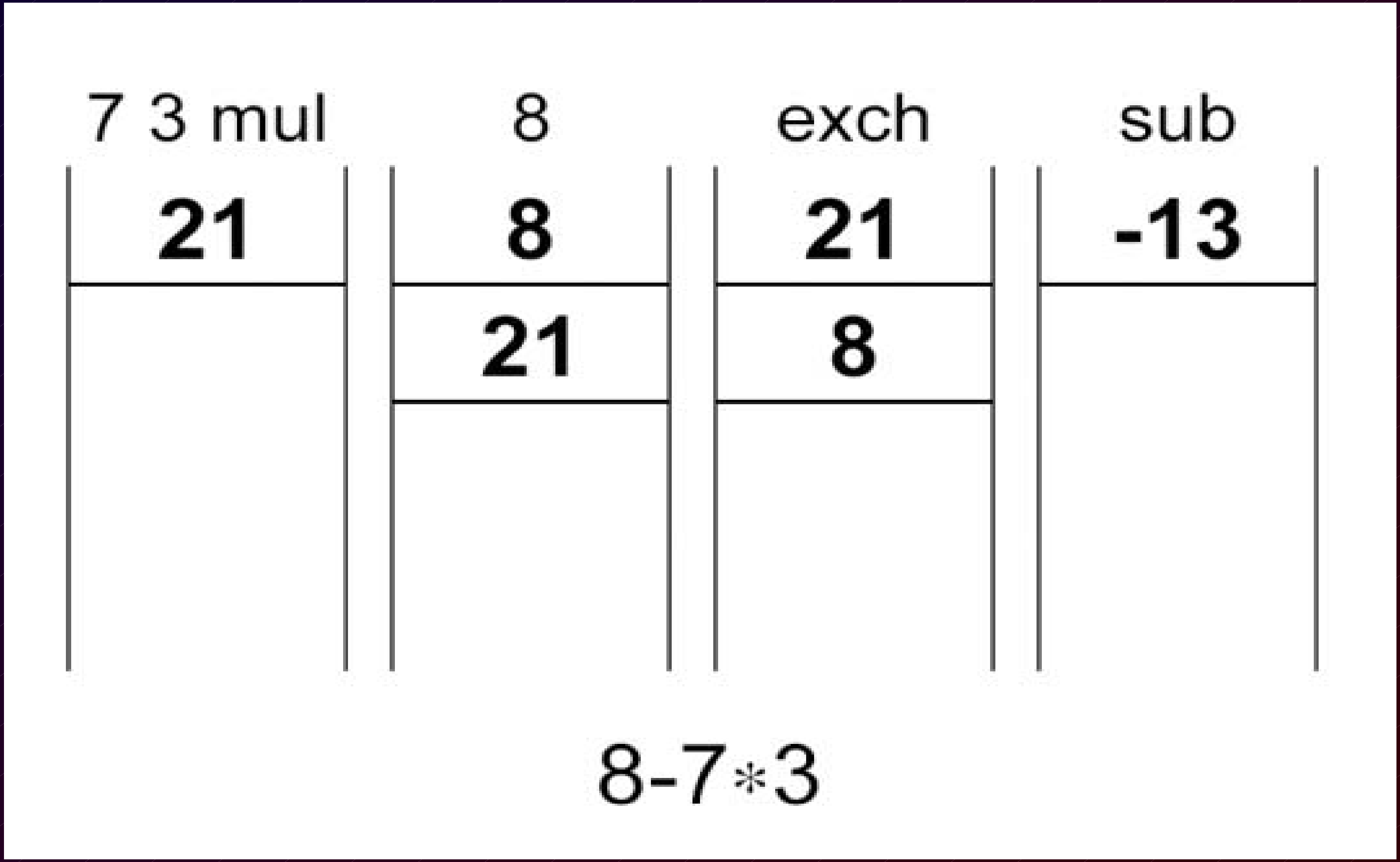
PostScript 연산



$8 - (7 \times 3)$

- 8 7 3 mul sub
- 7 3 mul 8 exch sub

PostScript 연산



7 3 mul

21

8

8

21

exch

21

8

sub

-13

8-7*3

스택 처리

6 8 12 **clear** ⇒ —

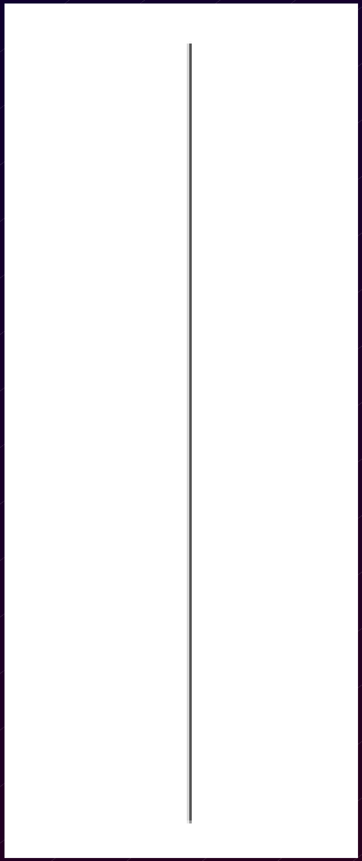
6 **dup** ⇒ 6 6

17 8 **pop** ⇒ 17

7 8 9 3 1 **roll** ⇒ 9 7 8

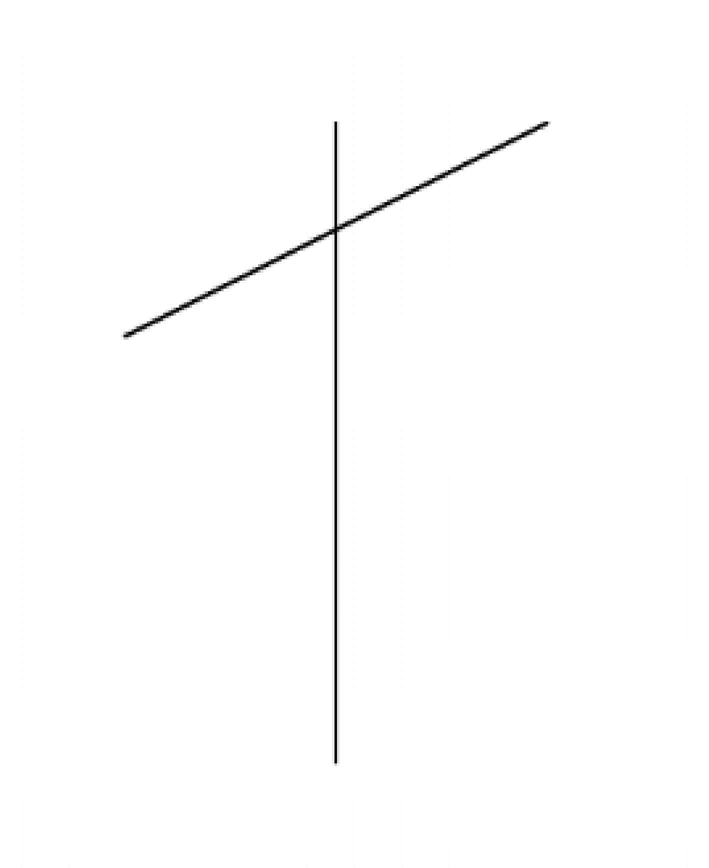
7 8 9 3 –1 **roll** ⇒ 8 9 7

PostScript 그래픽스



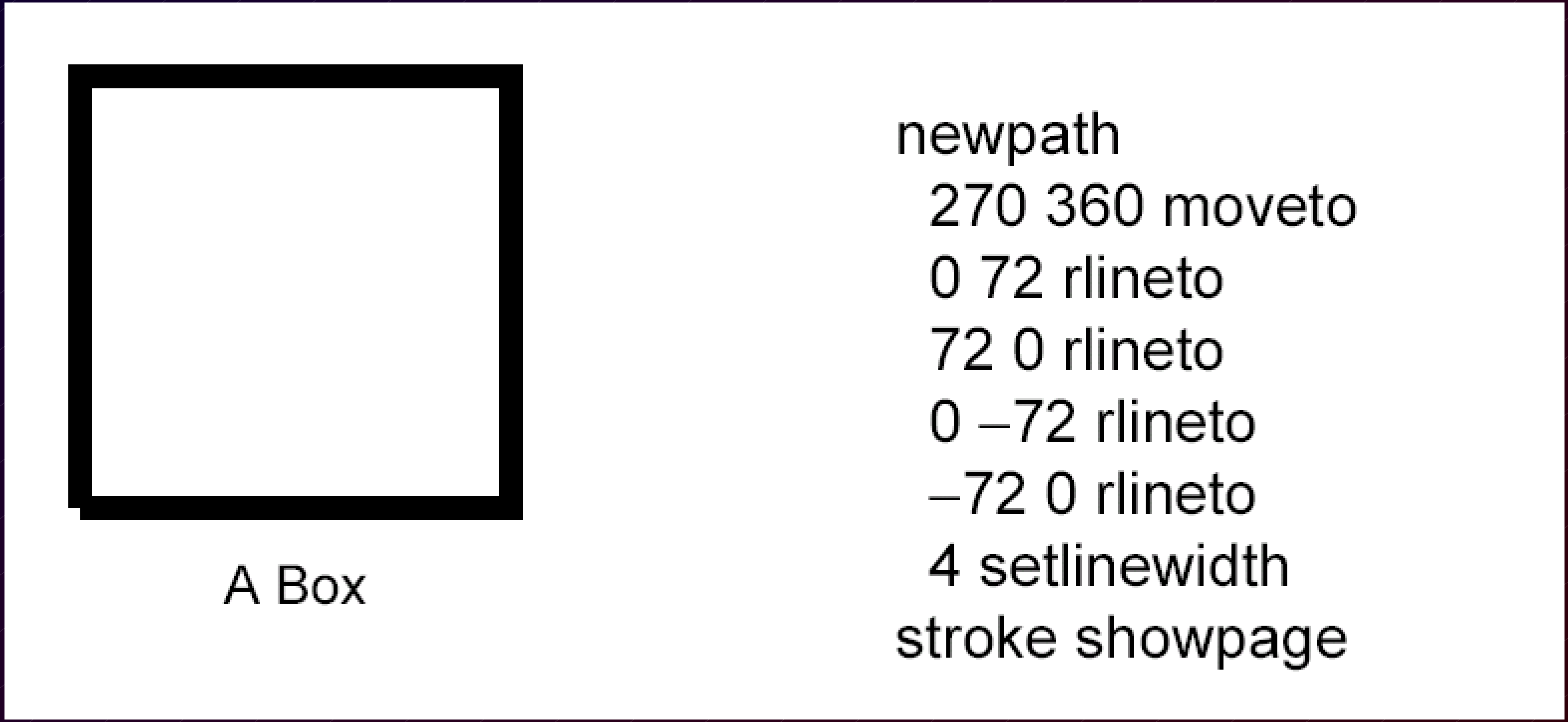
```
newpath
  144 72 moveto
  144 432 lineto
stroke
showpage
```

PostScript 그래픽스



```
newpath
  72 360 moveto
  144 72 rlineto
  144 432 moveto
  0 –216 rlineto
stroke
showpage
```

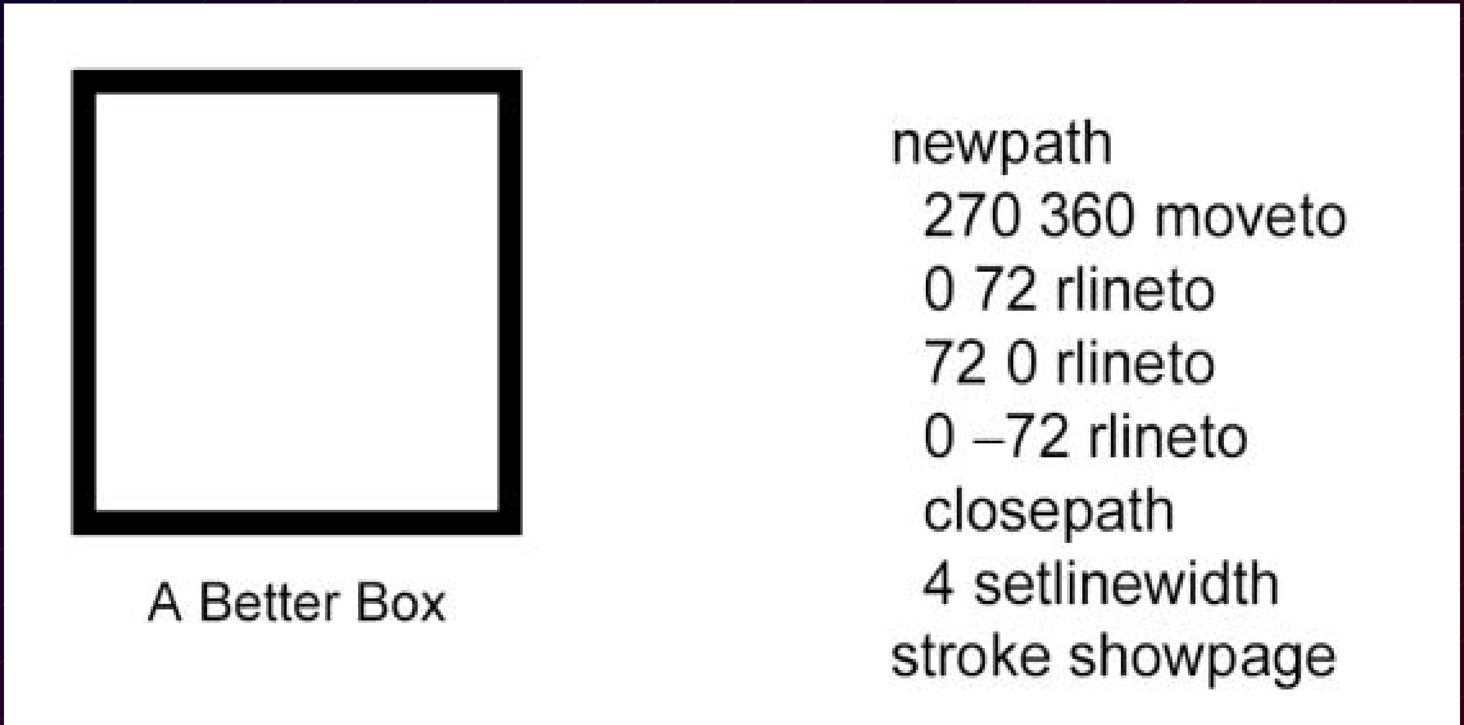
PostScript 그래픽스



A Box

```
newpath
  270 360 moveto
  0 72 rlineto
  72 0 rlineto
  0 –72 rlineto
  –72 0 rlineto
  4 setlinewidth
stroke showpage
```

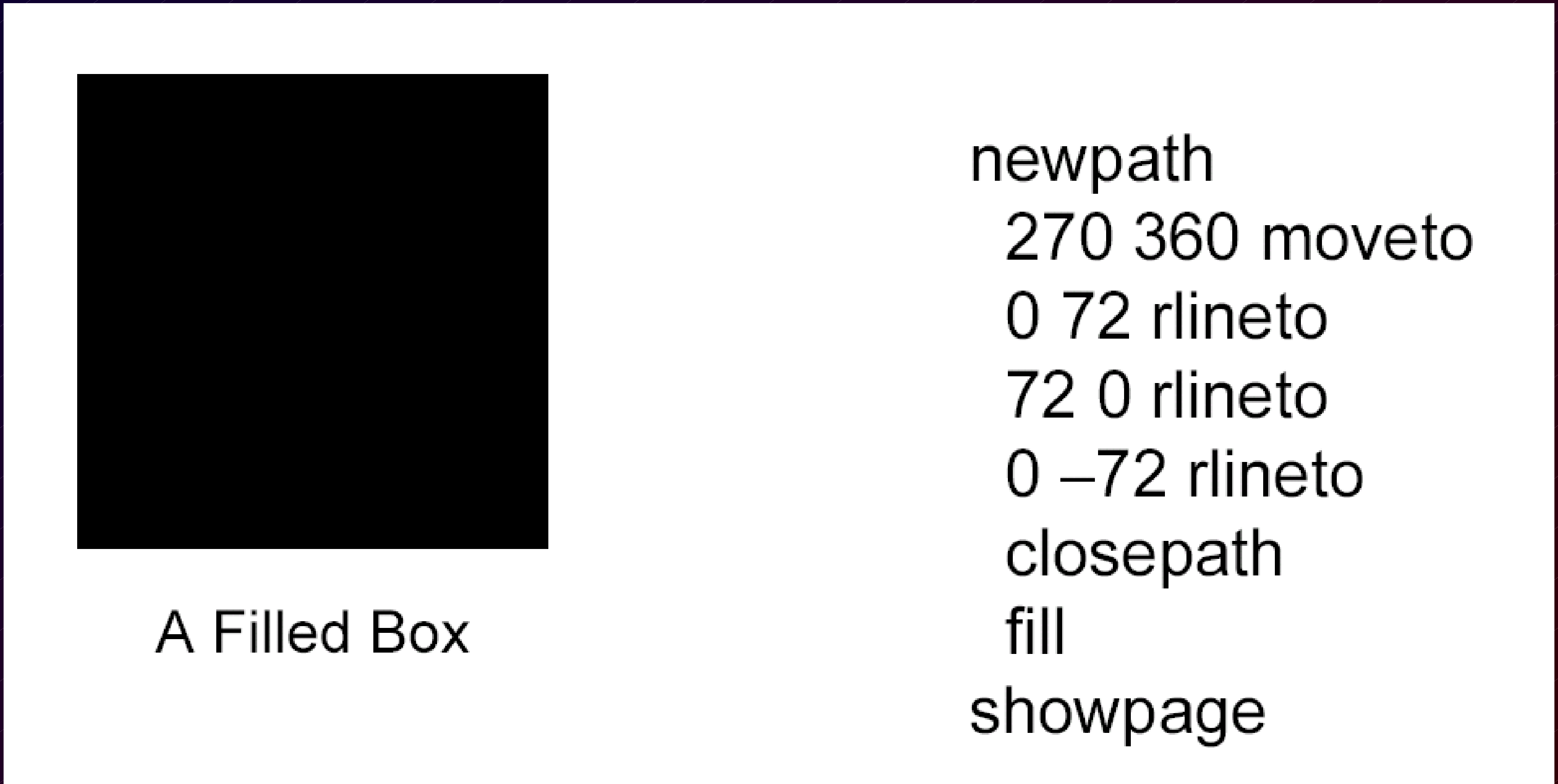
PostScript 그래픽스



A Better Box

```
newpath
  270 360 moveto
  0 72 rlineto
  72 0 rlineto
  0 –72 rlineto
  closepath
  4 setlinewidth
stroke showpage
```

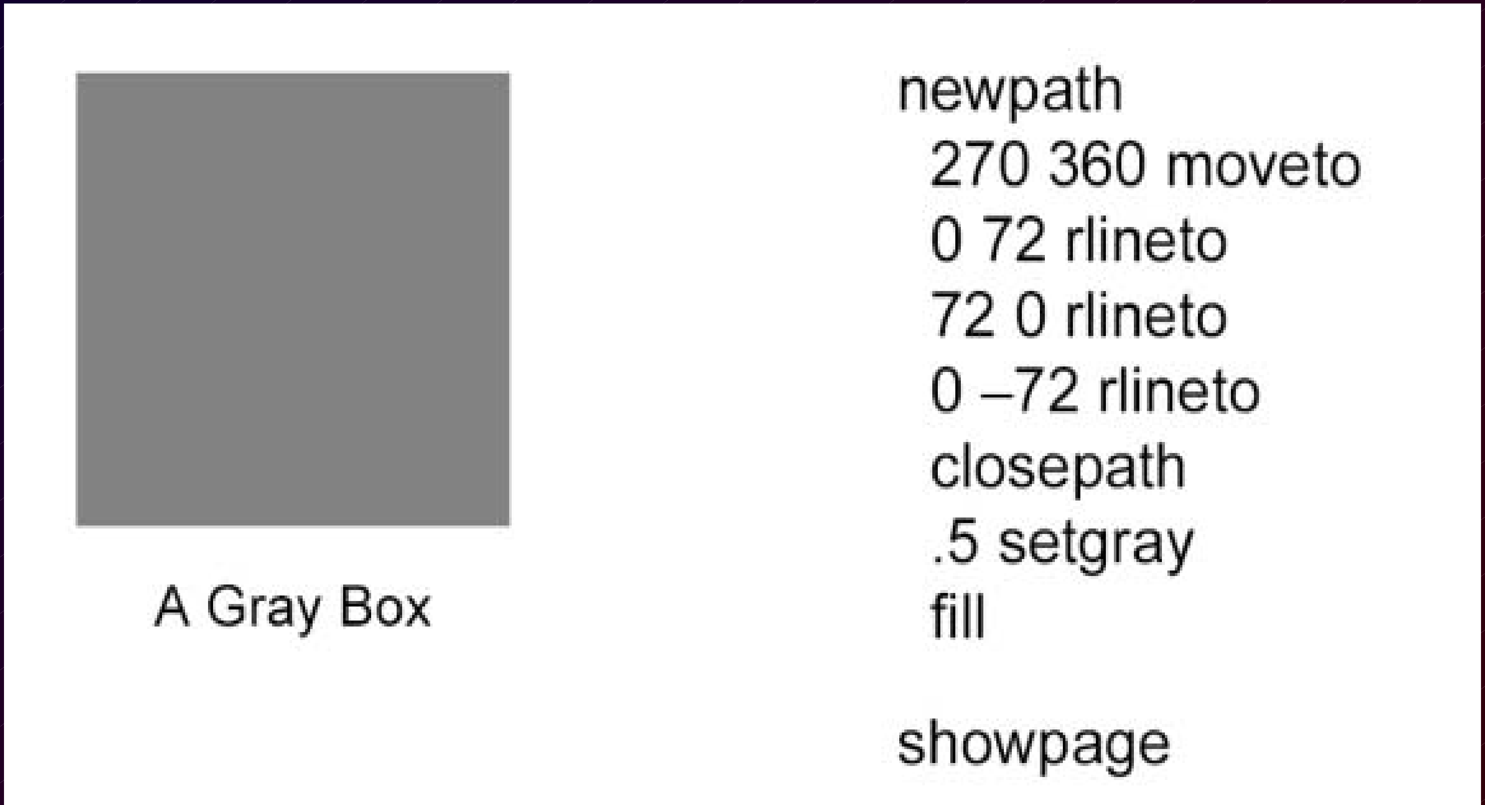
PostScript 그래픽스



A Filled Box

```
newpath
  270 360 moveto
  0 72 rlineto
  72 0 rlineto
  0 –72 rlineto
  closepath
  fill
showpage
```

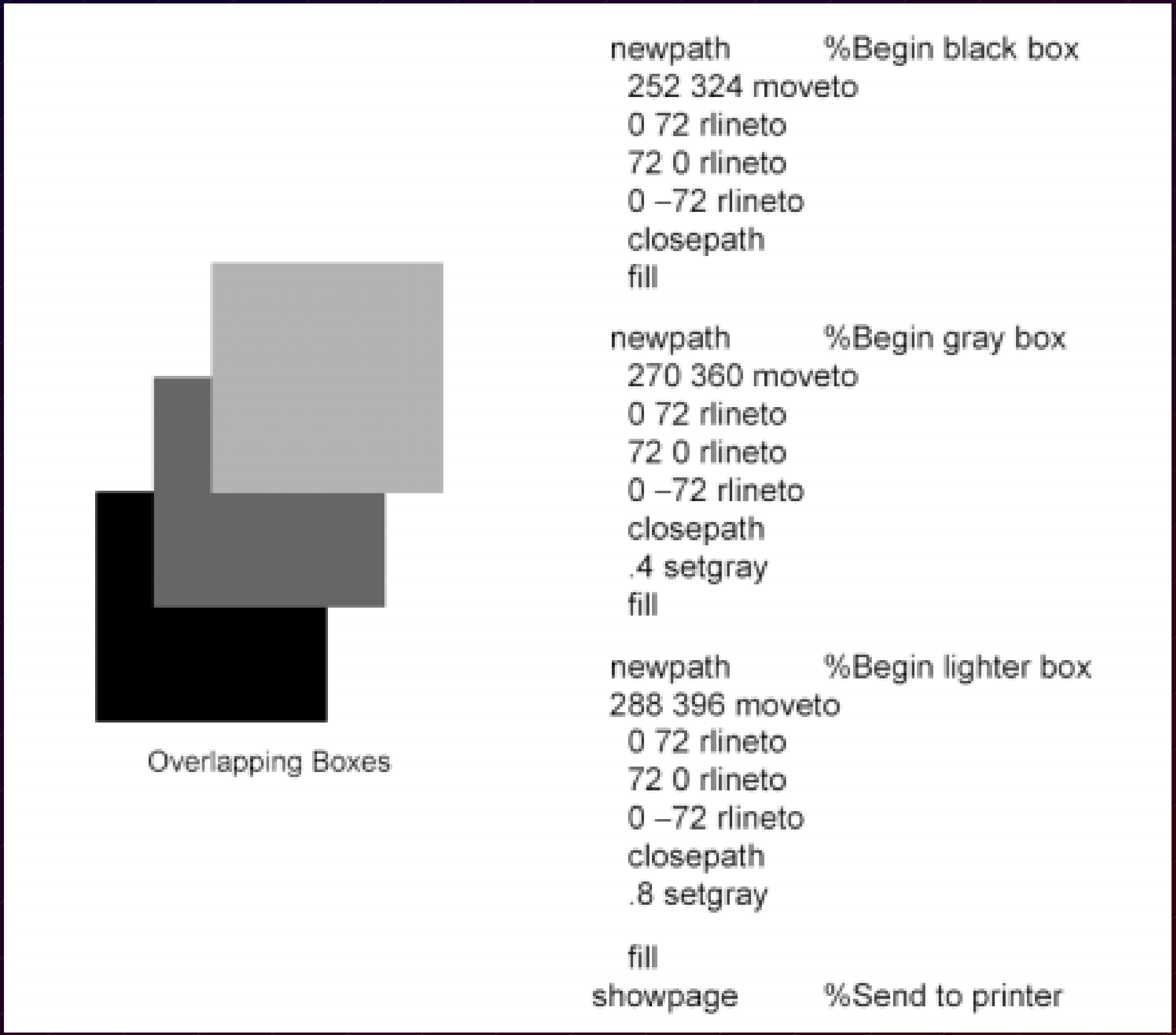
PostScript 그래픽스



A Gray Box

```
newpath
  270 360 moveto
  0 72 rlineto
  72 0 rlineto
  0 –72 rlineto
  closepath
  .5 setgray
  fill

showpage
```

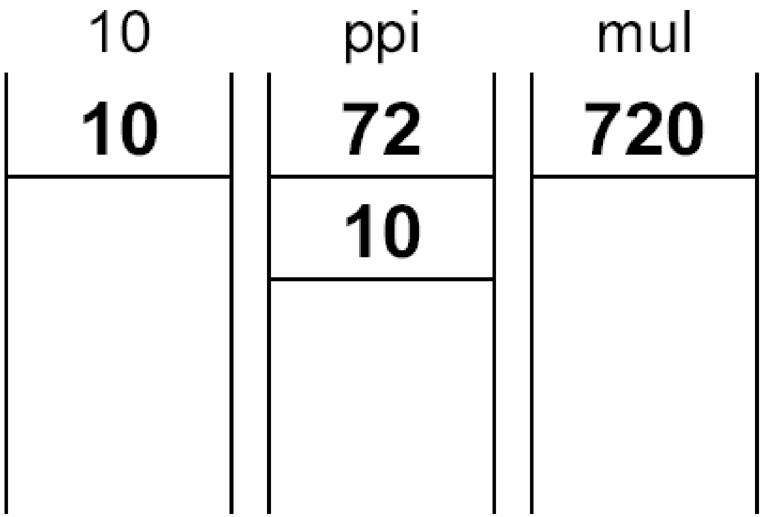
PostScript 그래픽스



Overlapping Boxes

```
 newpath        %Begin black box
  252 324 moveto
  0 72 rlineto
  72 0 rlineto
  0 –72 rlineto
  closepath
  fill

 newpath        %Begin gray box
  270 360 moveto
  0 72 rlineto
  72 0 rlineto
  0 –72 rlineto
  closepath
  .4 setgray
  fill

 newpath        %Begin lighter box
 288 396 moveto
  0 72 rlineto
  72 0 rlineto
  0 –72 rlineto
  closepath
  .8 setgray

  fill
showpage        %Send to printer
```

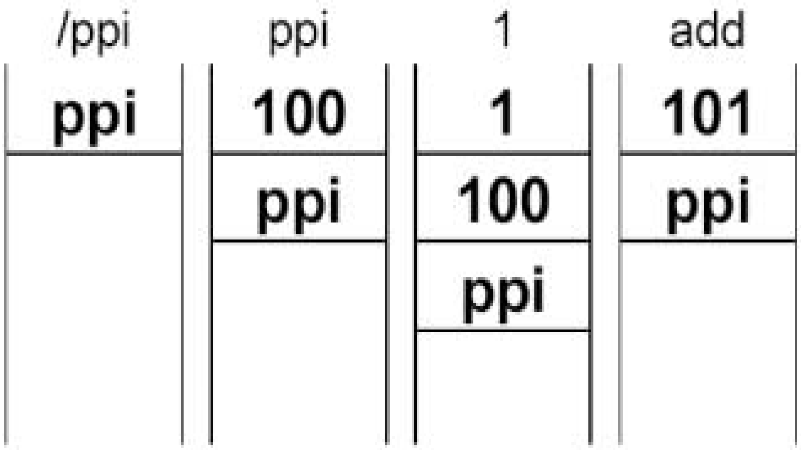
PostScript 변수



Using a Variable

```
/ppi  72  def

10  ppi  mul
```

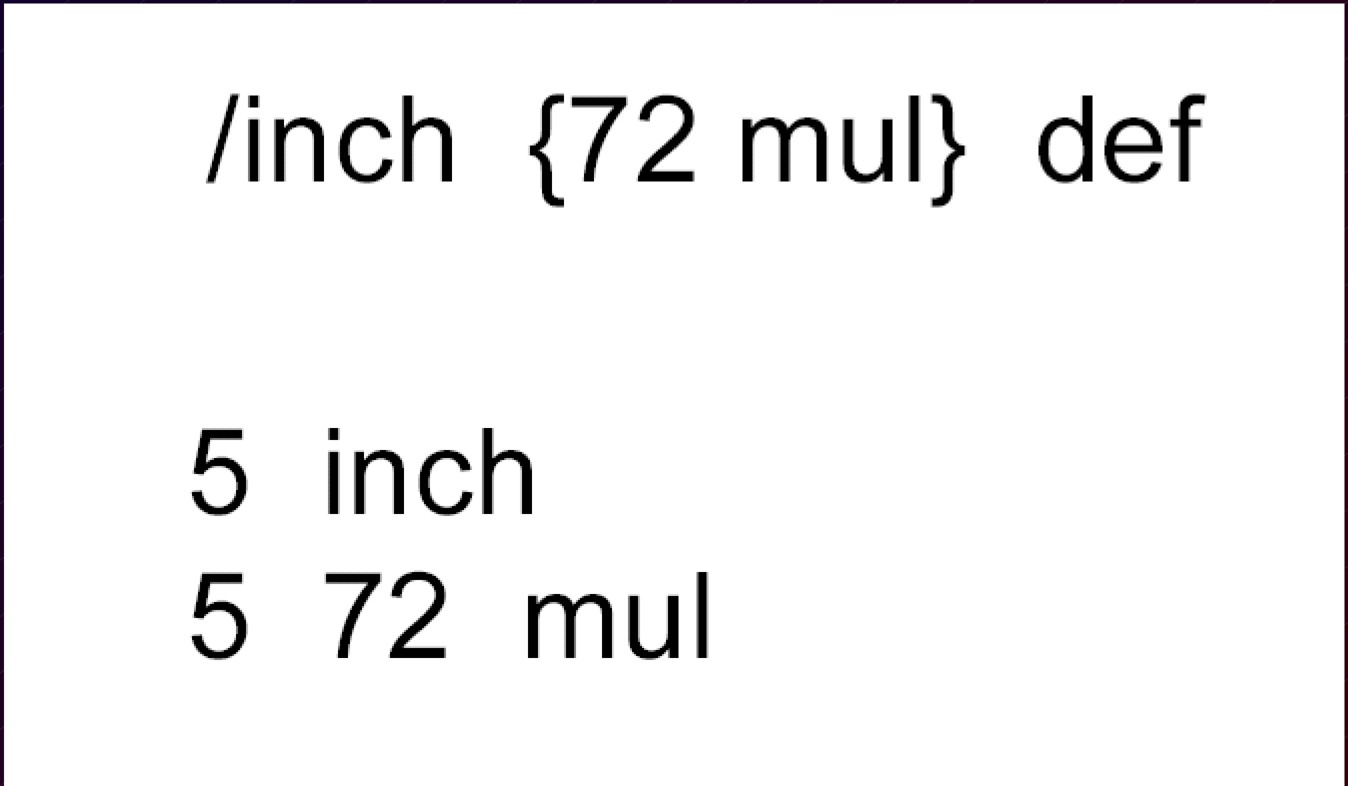
PostScript 변수



Incrementing a Variable

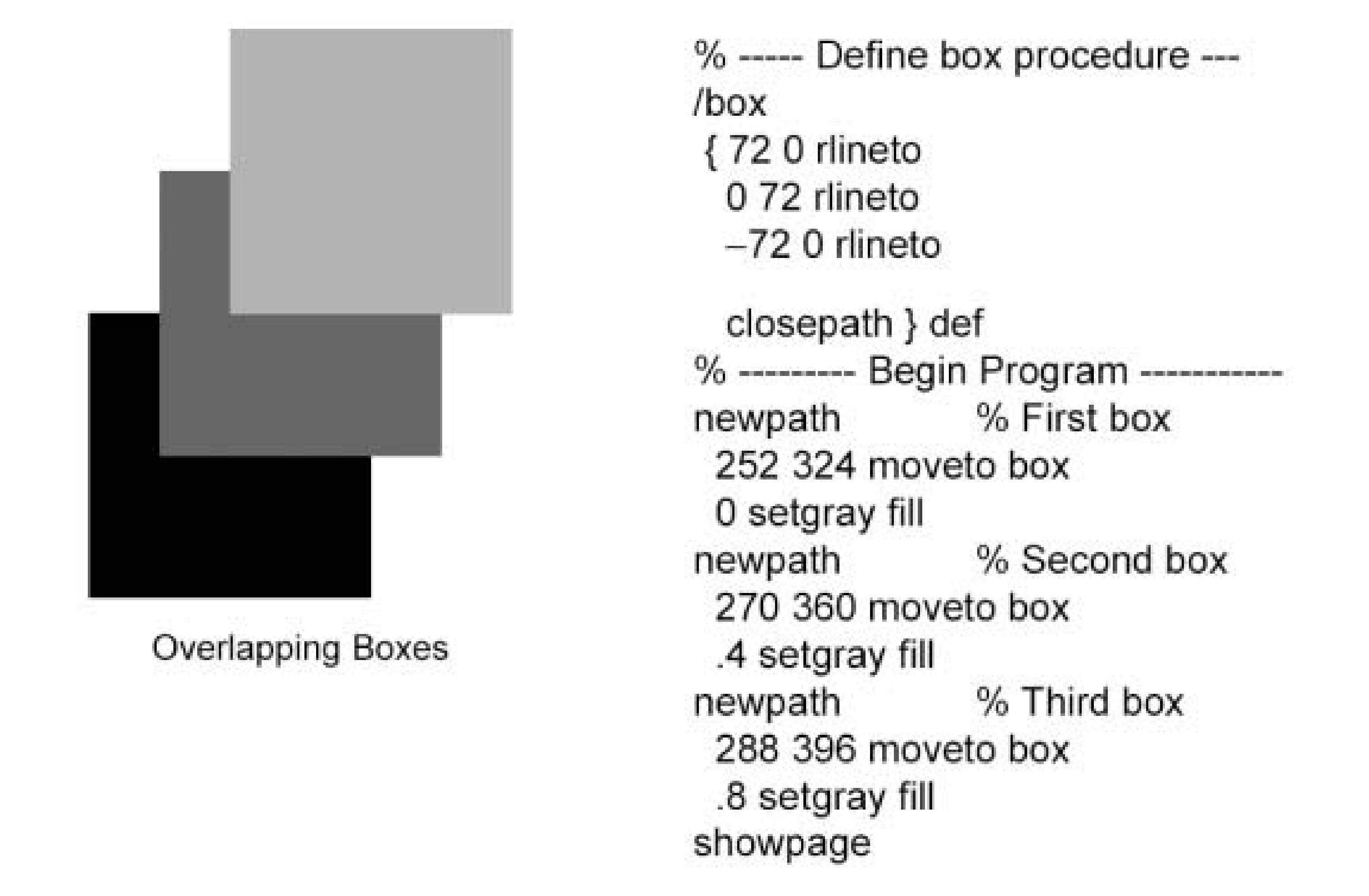
```
/ppi  100  def

/ppi  ppi 1 add  def
```

PostScript 프러시저



```
/inch  {72 mul}  def


5  inch
5  72  mul
```

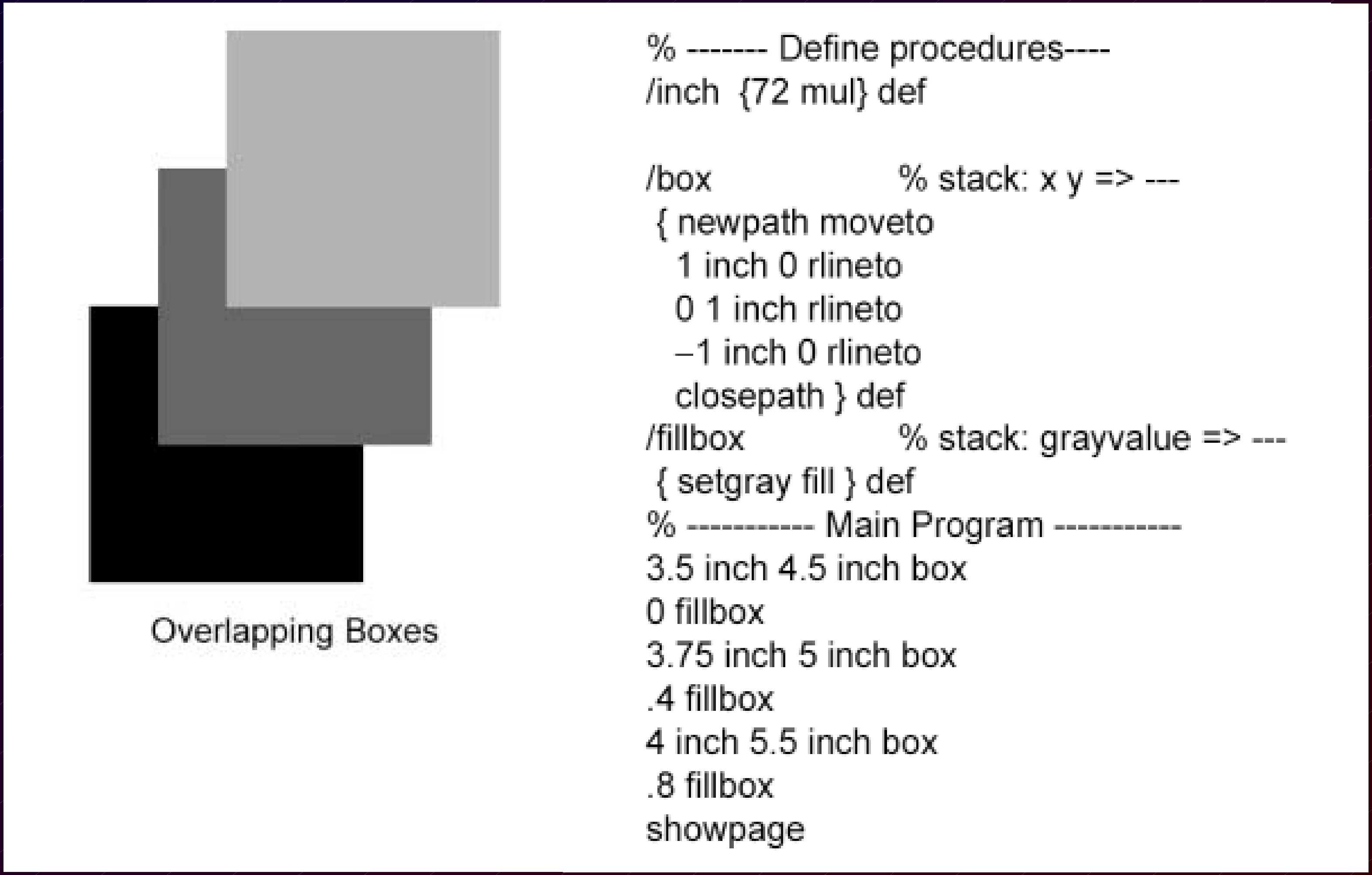
PostScript 프러시저



Overlapping Boxes

```
% ----- Define box procedure ---
/box
 { 72 0 rlineto
   0 72 rlineto
   –72 0 rlineto

   closepath } def
% --------- Begin Program -----------
newpath              % First box
  252 324 moveto box
  0 setgray fill
newpath              % Second box
  270 360 moveto box
  .4 setgray fill
newpath              % Third box
  288 396 moveto box
  .8 setgray fill
showpage
```

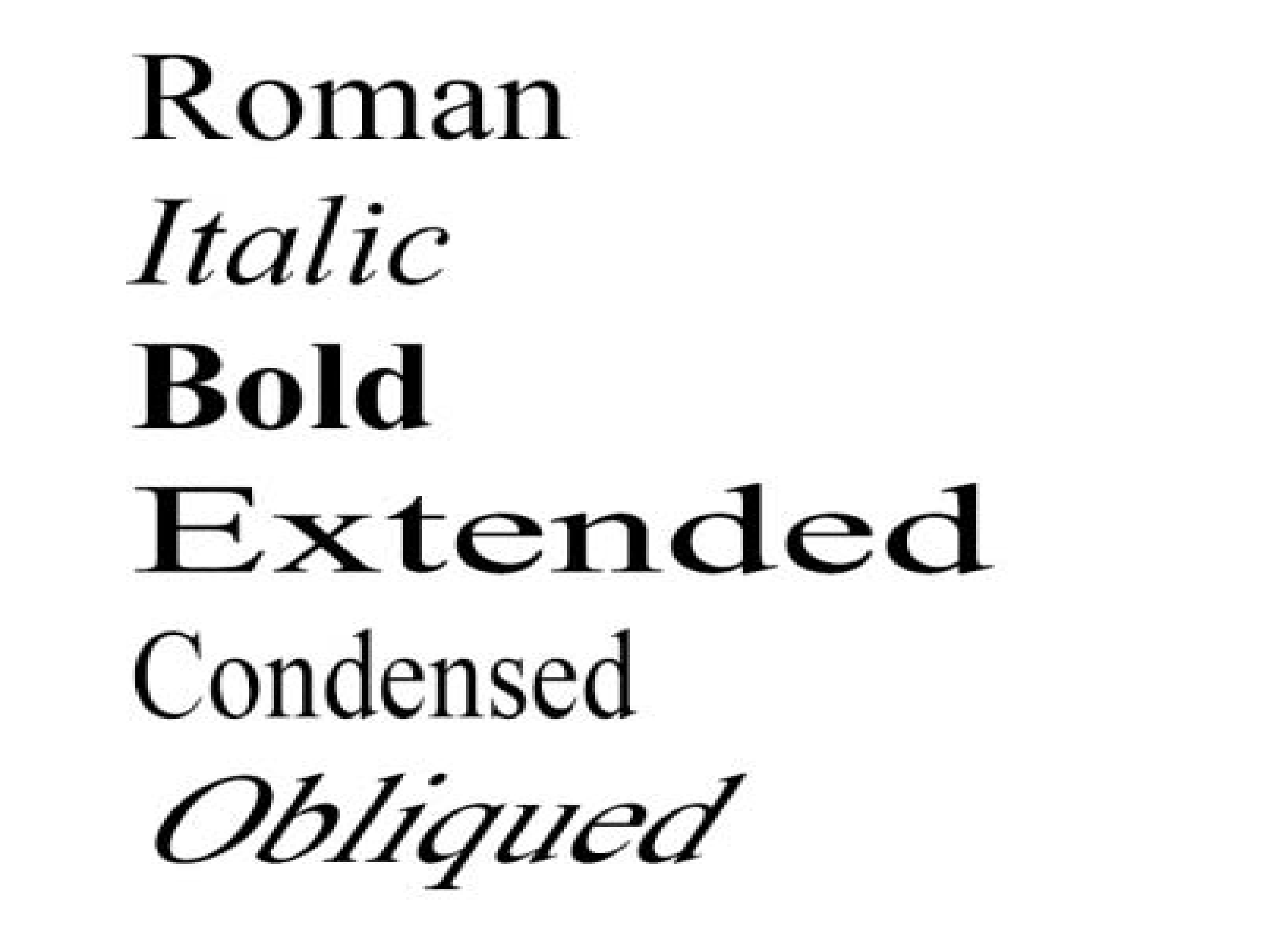
PostScript 프러시저



Overlapping Boxes

```
% ------- Define procedures----
/inch  {72 mul} def

/box                % stack: x y => ---
 { newpath moveto
   1 inch 0 rlineto
   0 1 inch rlineto
   –1 inch 0 rlineto
   closepath } def
/fillbox            % stack: grayvalue => ---
 { setgray fill } def
% ----------- Main Program -----------
3.5 inch 4.5 inch box
0 fillbox
3.75 inch 5 inch box
.4 fillbox
4 inch 5.5 inch box
.8 fillbox
showpage
```

PostScript 폰트



Roman

Italic

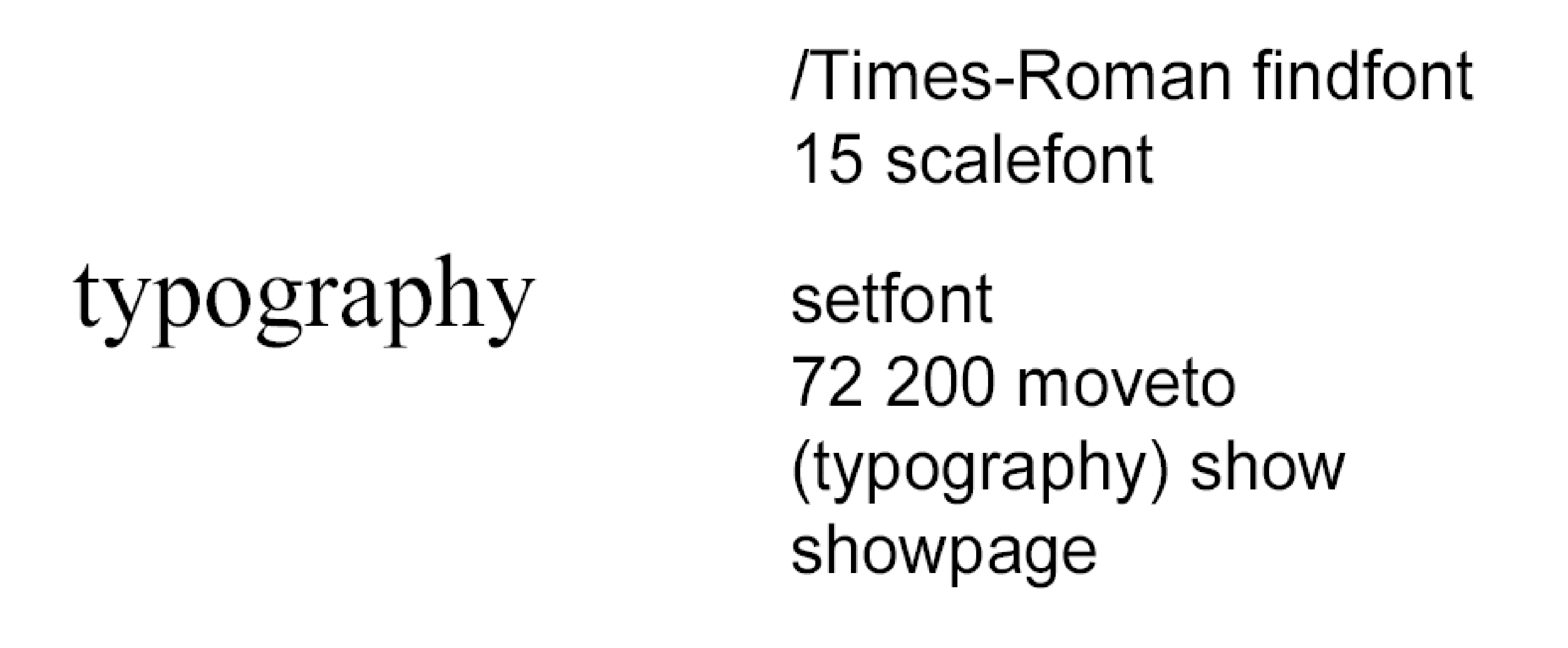
Bold

Extended

Condensed

Obliqued

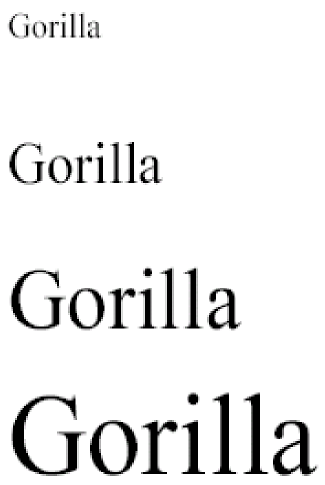
PostScript 문자열



typography

```
/Times-Roman findfont
15 scalefont
setfont
72 200 moveto
(typography) show
showpage
```

PostScript 문자열



Gorilla

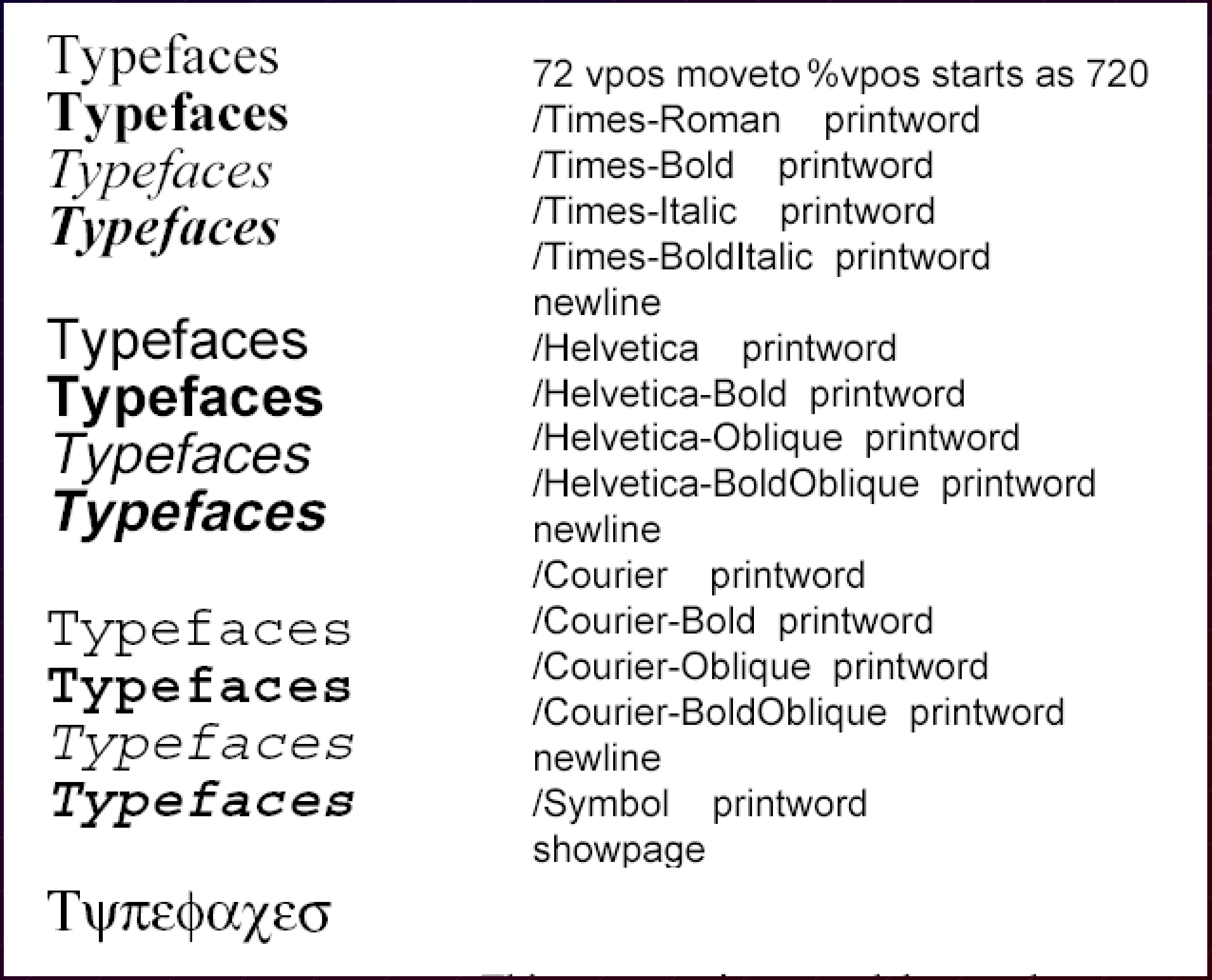
Gorilla

Gorilla

Gorilla

```
/showGorilla      % stack: x y ---

 { moveto (Gorilla) show }def
/Times-Roman findfont  6 scalefont  setfont
72 300 showGorilla
/Times-Roman findfont  10 scalefont  setfont
72 275  showGorilla
/Times-Roman findfont  15 scalefont  setfont
72 250  showGorilla
/Times-Roman findfont  20 scalefont  setfont
72 225 showGorilla

showpage
```

PostScript 폰트



Typefaces
Typefaces
Typefaces
Typefaces

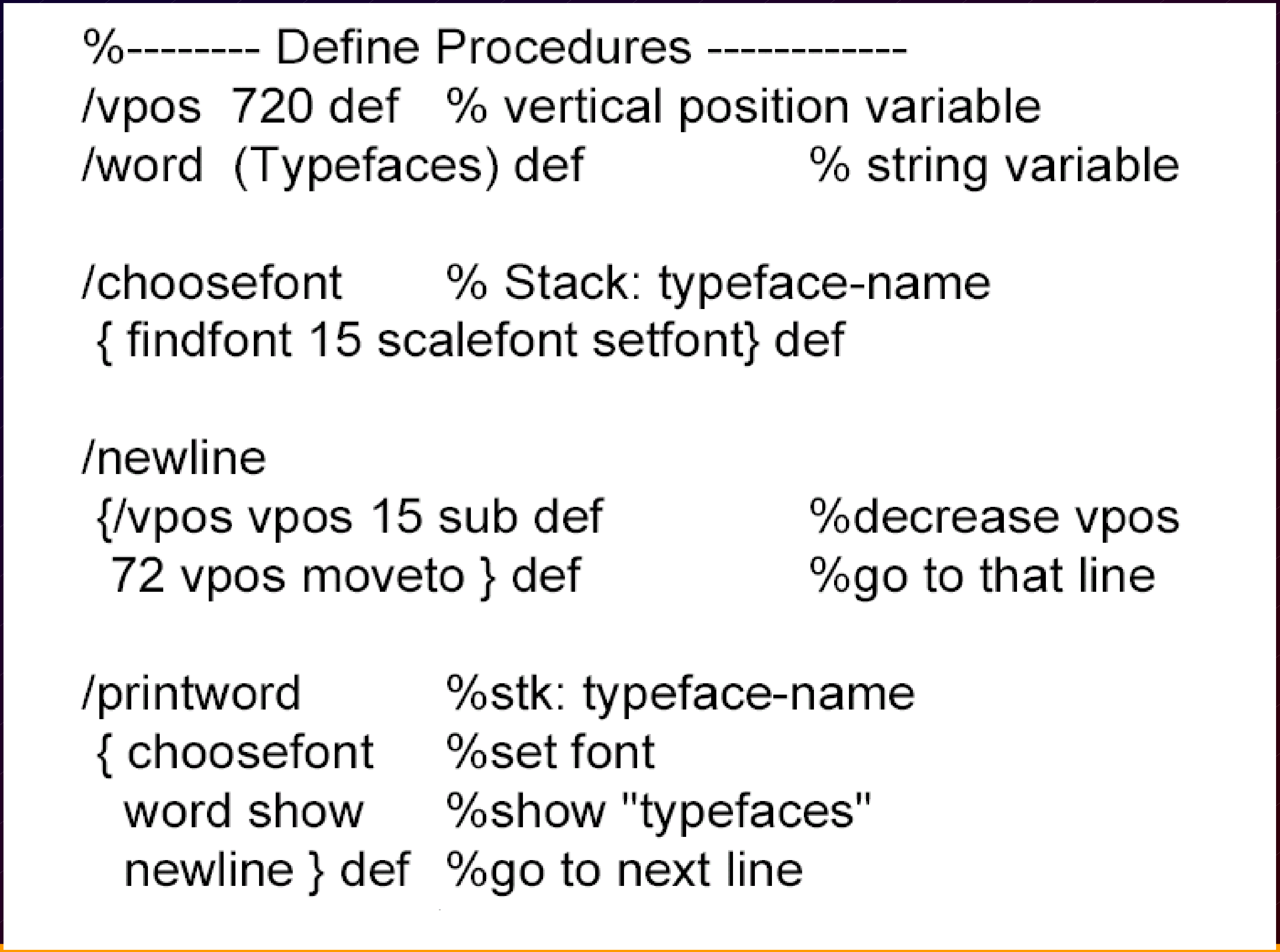
Typefaces
Typefaces
Typefaces
Typefaces

Typefaces
Typefaces
Typefaces
Typefaces

Τψπεφαχεσ

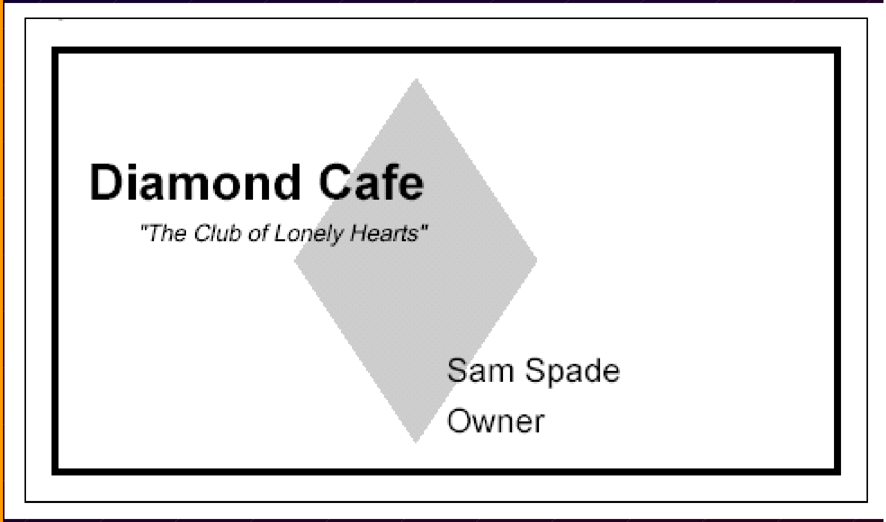
```
72 vpos moveto %vpos starts as 720
/Times-Roman    printword
/Times-Bold    printword
/Times-Italic    printword
/Times-BoldItalic  printword
newline
/Helvetica    printword
/Helvetica-Bold  printword
/Helvetica-Oblique  printword
/Helvetica-BoldOblique  printword
newline
/Courier    printword
/Courier-Bold  printword
/Courier-Oblique  printword
/Courier-BoldOblique  printword
newline
/Symbol    printword
showpage
```

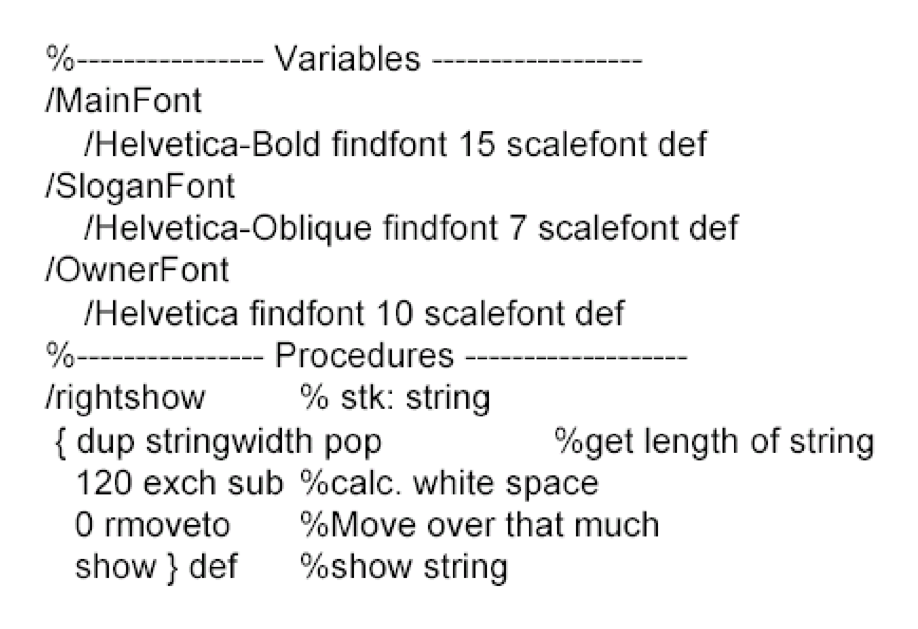
사용된 프러시저들



```
%-------- Define Procedures ------------
/vpos  720 def   % vertical position variable
/word  (Typefaces) def              % string variable

/choosefont        % Stack: typeface-name
 { findfont 15 scalefont setfont} def

/newline
 {/vpos vpos 15 sub def             %decrease vpos
  72 vpos moveto } def              %go to that line

/printword         %stk: typeface-name
 { choosefont      %set font
   word show       %show "typefaces"
   newline } def   %go to next line
```

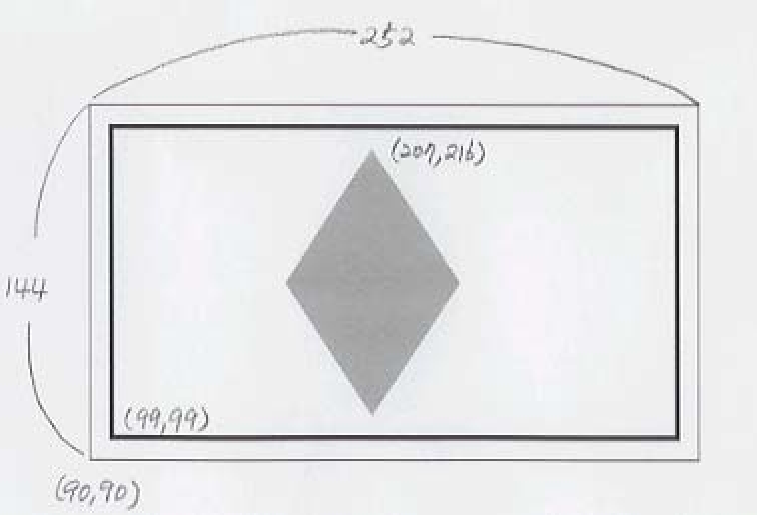
명함 글자의 폰트정의

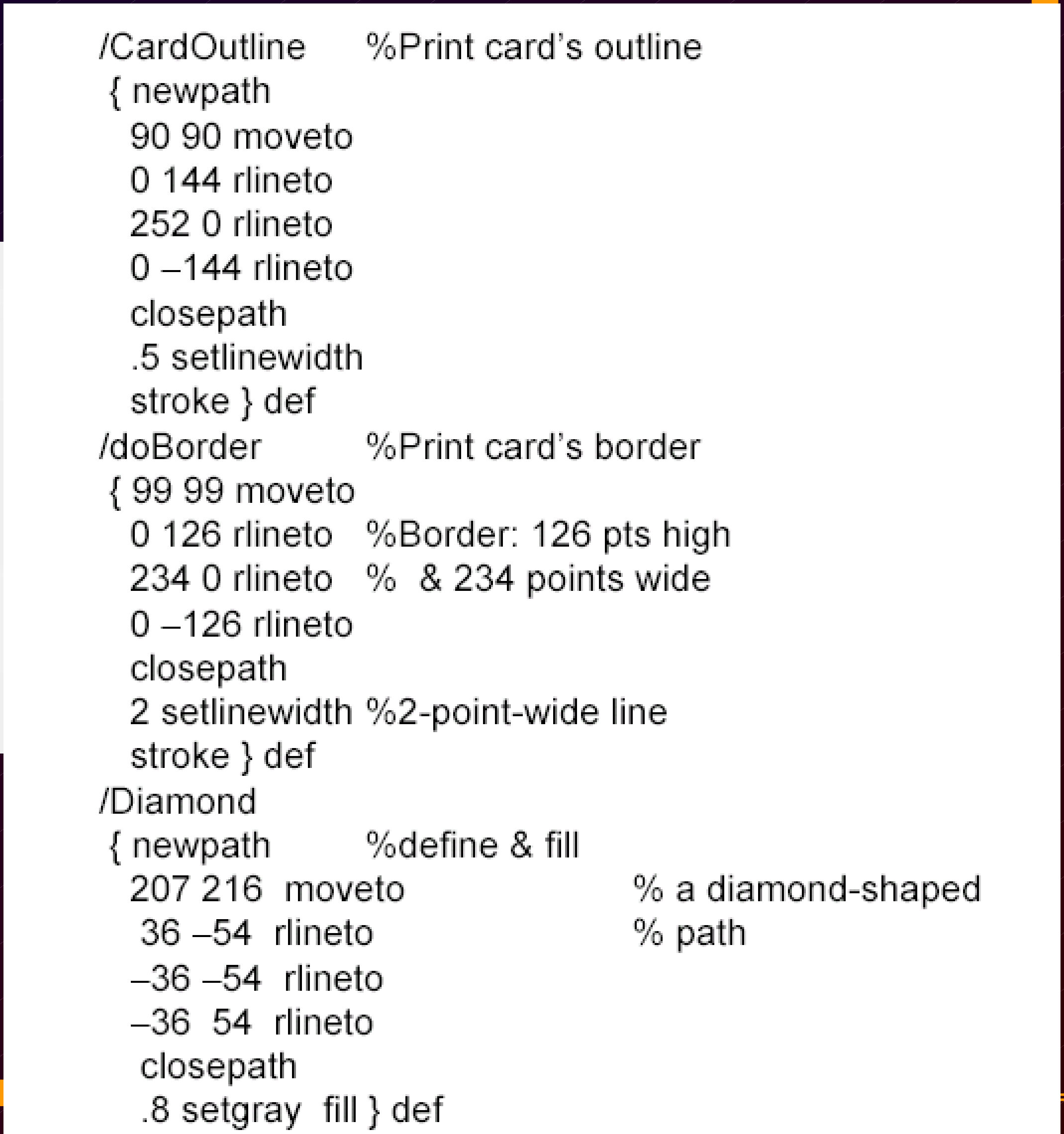




```
%---------------- Variables ------------------
/MainFont
   /Helvetica-Bold findfont 15 scalefont def
/SloganFont
   /Helvetica-Oblique findfont 7 scalefont def
/OwnerFont
   /Helvetica findfont 10 scalefont def
%---------------- Procedures -------------------
/rightshow        % stk: string
 { dup stringwidth pop                %get length of string
  120 exch sub  %calc. white space
  0 rmoveto       %Move over that much
  show } def      %show string
```

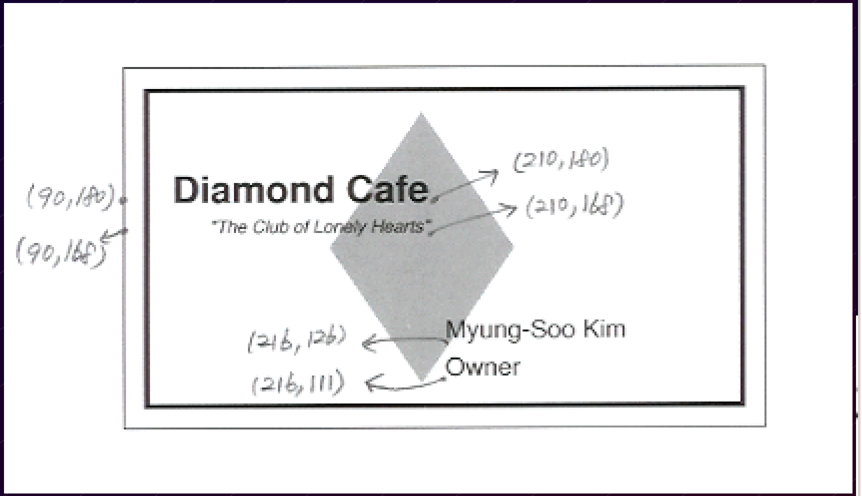
명함 테두리

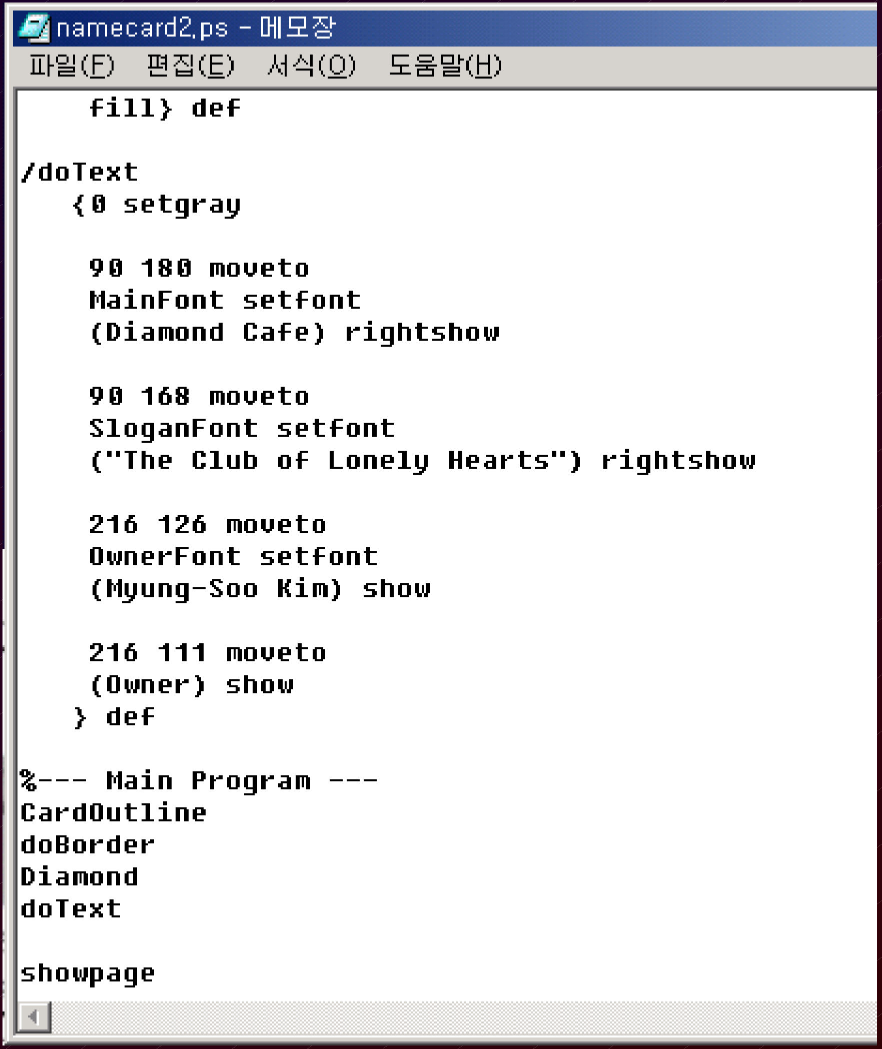




```
/CardOutline      %Print card's outline
 { newpath
   90 90 moveto
   0 144 rlineto
   252 0 rlineto
   0 –144 rlineto
   closepath
   .5 setlinewidth
   stroke } def
/doBorder         %Print card's border
 { 99 99 moveto
   0 126 rlineto   %Border: 126 pts high
   234 0 rlineto   %  & 234 points wide
   0 –126 rlineto
   closepath
   2 setlinewidth %2-point-wide line
   stroke } def
/Diamond
 { newpath         %define & fill
   207 216  moveto                % a diamond-shaped
    36 –54  rlineto               % path
   –36 –54  rlineto
   –36  54  rlineto
    closepath
    .8 setgray  fill } def
```

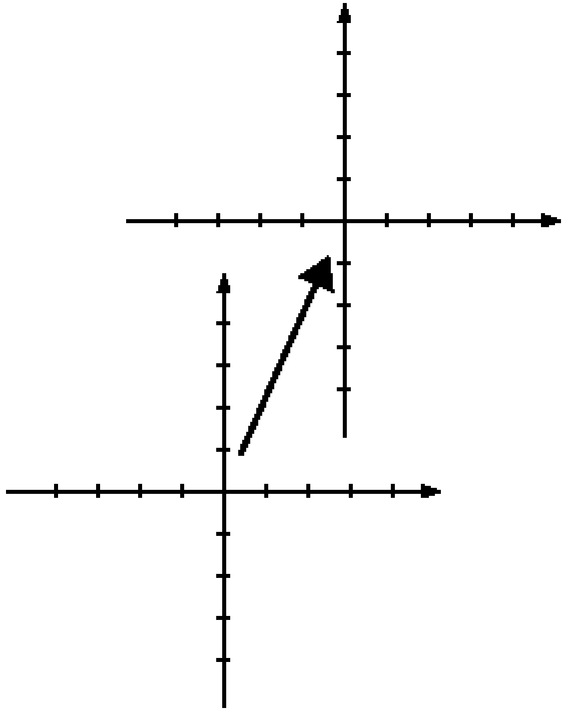
명함 내용





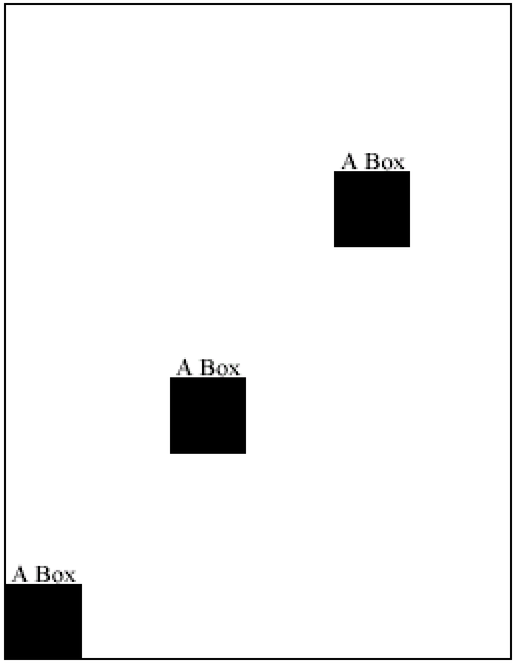
```
    fill} def

/doText
   {0 setgray

    90 180 moveto
    MainFont setfont
    (Diamond Cafe) rightshow

    90 168 moveto
    SloganFont setfont
    ("The Club of Lonely Hearts") rightshow

    216 126 moveto
    OwnerFont setfont
    (Myung-Soo Kim) show

    216 111 moveto
    (Owner) show
   } def

%--- Main Program ---
CardOutline
doBorder
Diamond
doText

showpage
```

좌표변환 (평행이동)



Translation

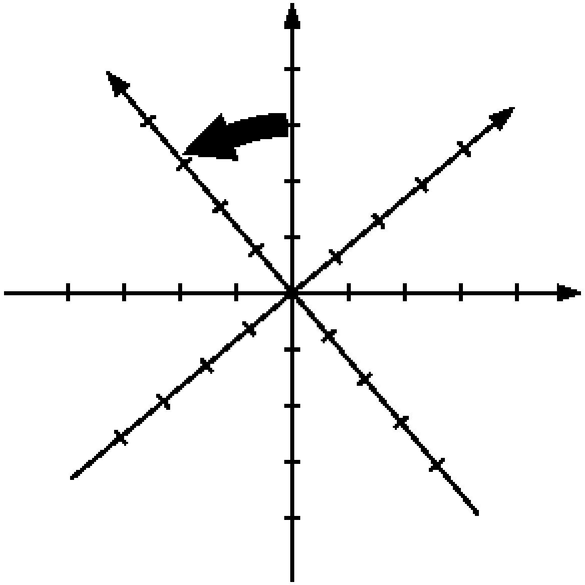
좌표변환 (평행이동)



Translated Squares

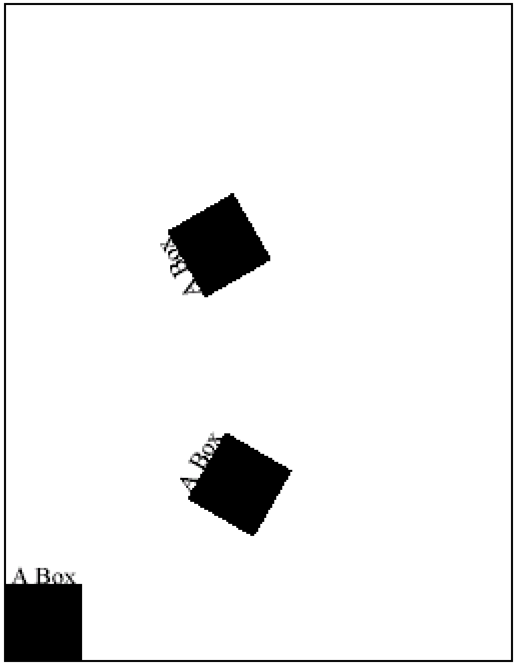
```
/Times-Roman findfont 30 scalefont setfont


/square              %procedure to draw a
 { newpath           %  filled square
   0 0 moveto
   90 0 lineto       %define a square path
   90 90 lineto
   0 90 lineto
   closepath  fill   %fill it
   6 92 moveto       % & label it
   (A Box) show   } def

square               %do a square
200 250 translate                        %move coord. sys.
square               %do another square
200 250 translate                        %and move again
square               %do a third square

showpage
```

좌표변환 (회전이동)



Rotation

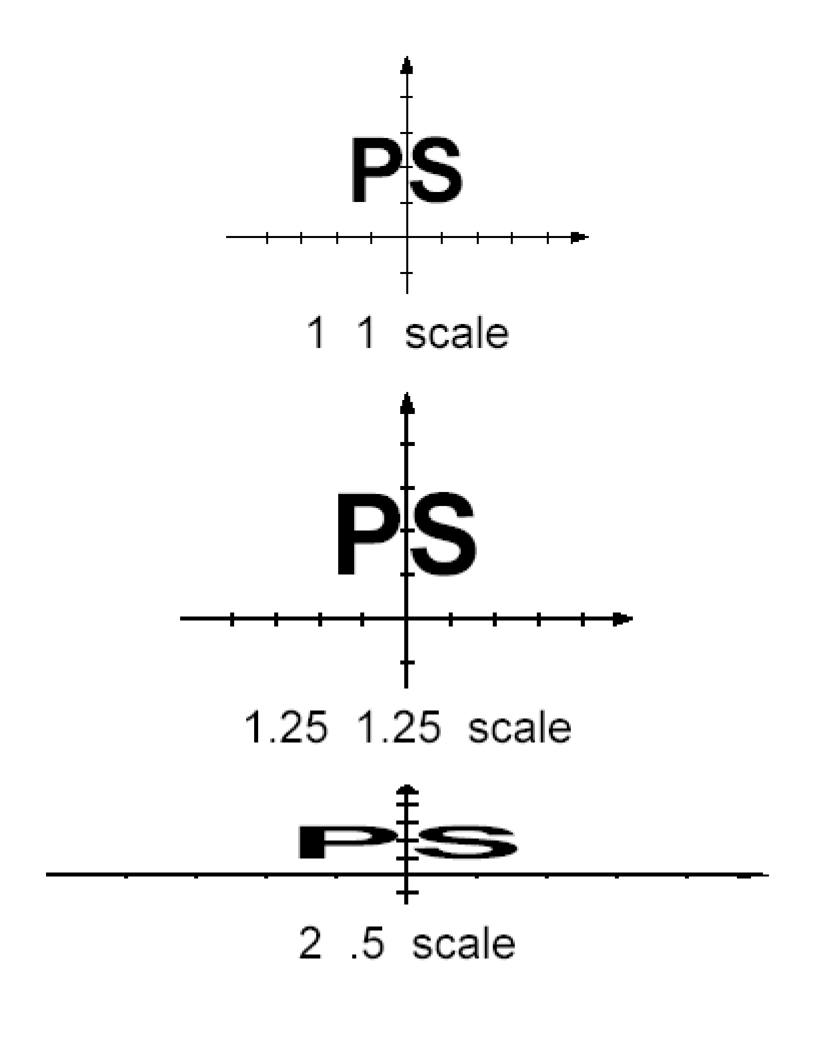
좌표변환 (회전이동)



Rotated Squares

```
/Times-Roman findfont 30 scalefont setfont

/square           %procedure from
 { newpath        % previous program
   0 0 moveto
   90 0 lineto
   90 90 lineto
   0 90 lineto
   closepath  fill
   6 92 moveto    %Label the box
   (A Box) show  } def

square            %do a square
300 150 translate                %move coord. sys.
60 rotate         %and rotate it
square            %do it again...
300 150 translate
60 rotate
square            %do a third square

showpage
```

좌표변환 (크기변환)

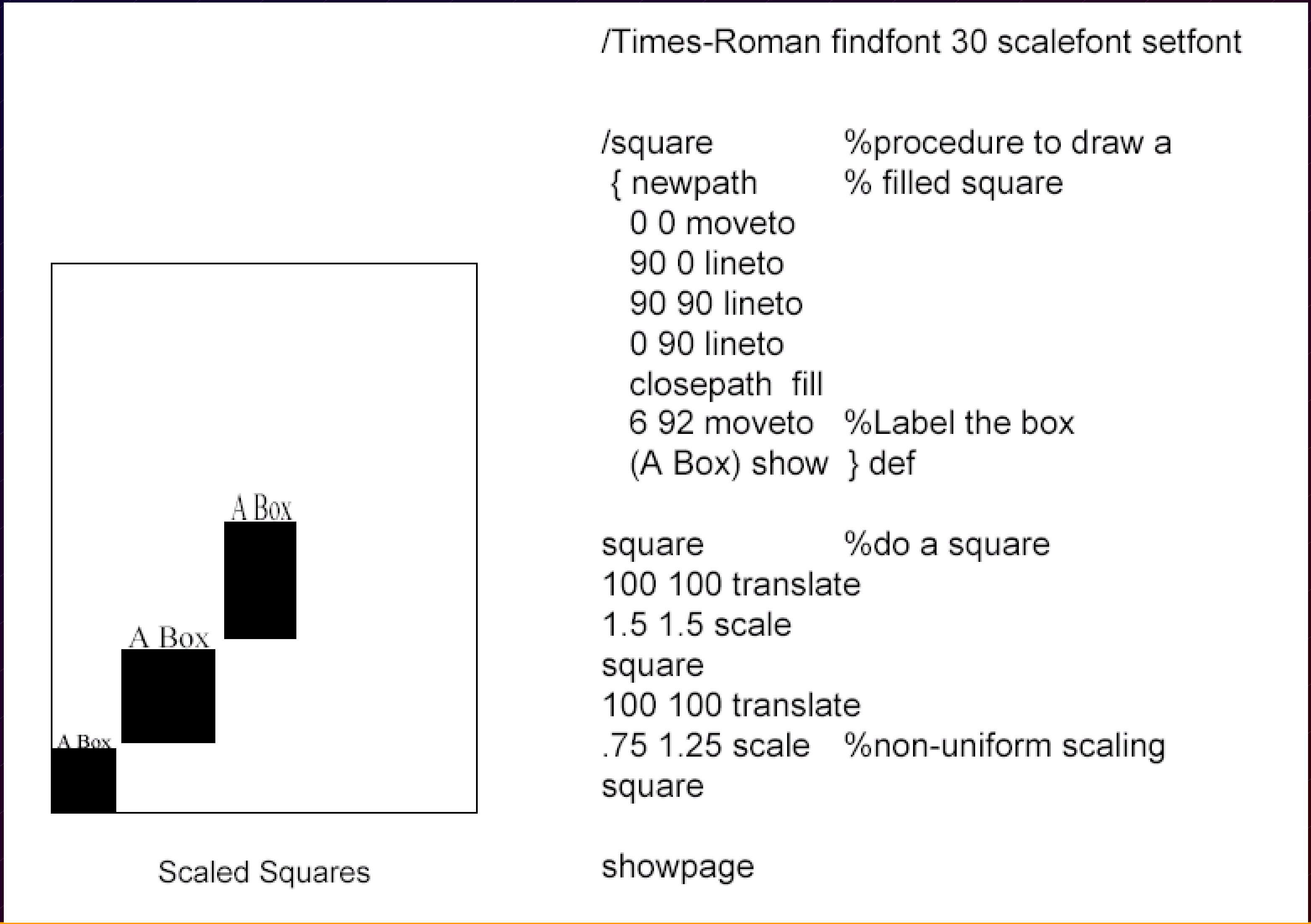


1 1 scale

1.25 1.25 scale

2 .5 scale

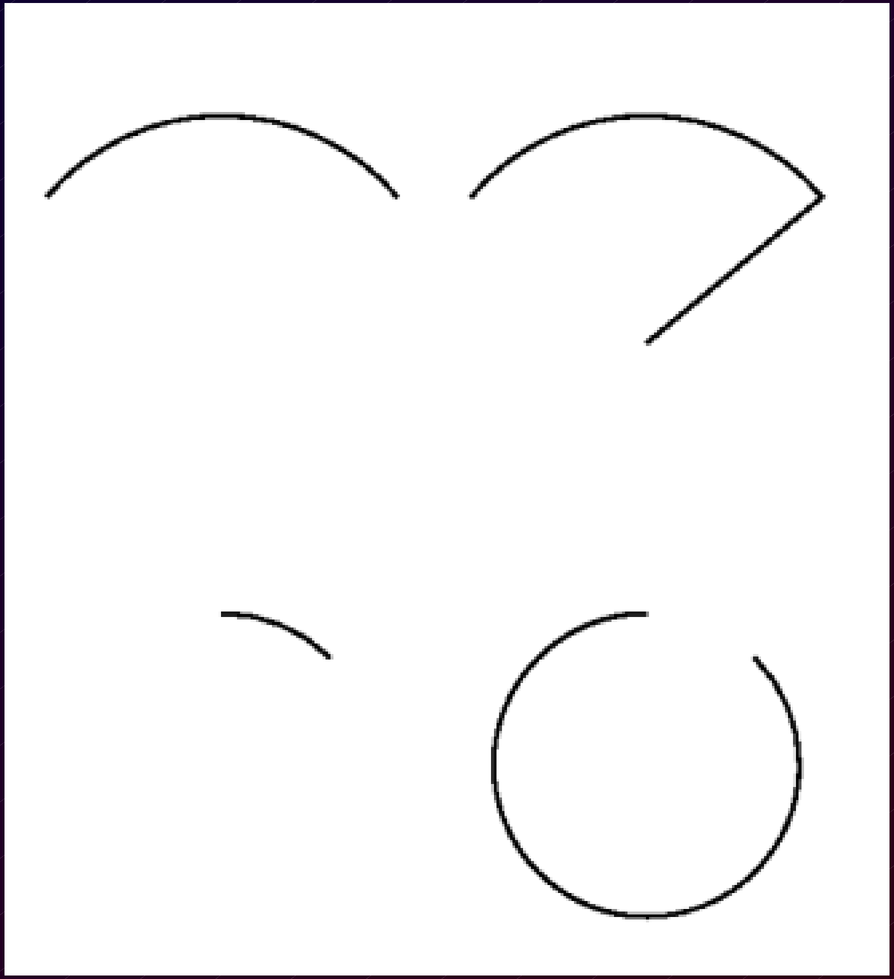
좌표변환 (크기변환)

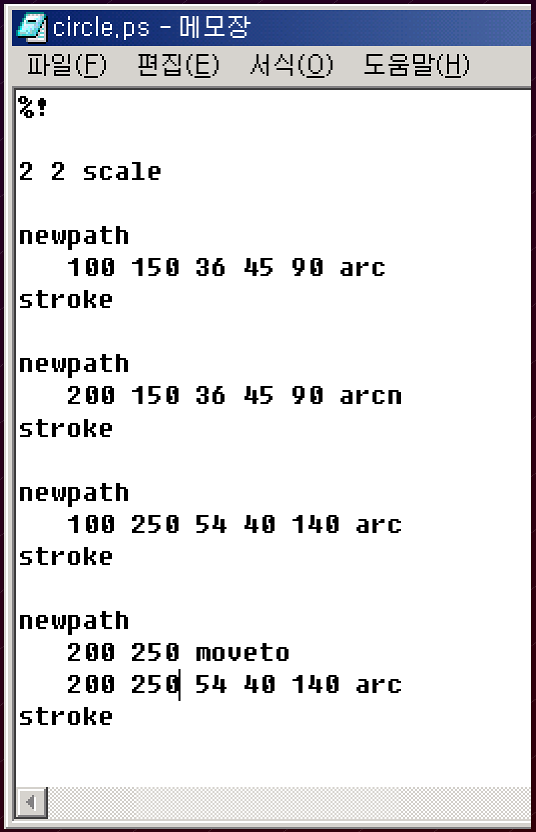


```
/Times-Roman findfont 30 scalefont setfont

/square              %procedure to draw a
 { newpath           % filled square
   0 0 moveto
   90 0 lineto
   90 90 lineto
   0 90 lineto
   closepath  fill
   6 92 moveto       %Label the box
   (A Box) show  } def

square               %do a square
100 100 translate
1.5 1.5 scale
square
100 100 translate
.75 1.25 scale       %non-uniform scaling
square

showpage
```

Scaled Squares

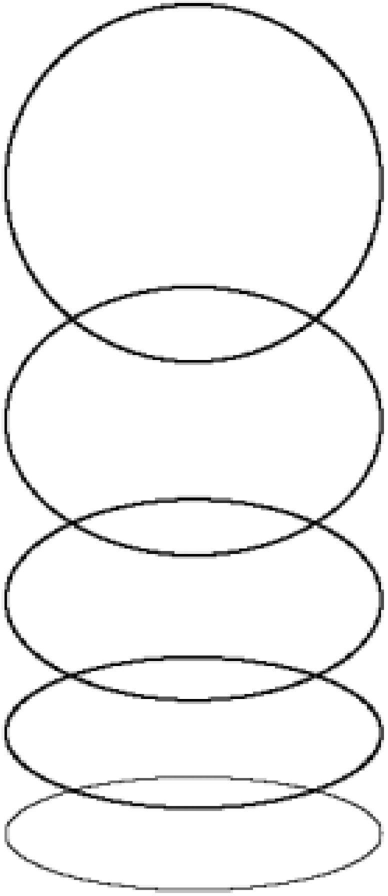
원호그리기





```
%!

2 2 scale

newpath
   100 150 36 45 90 arc
stroke

newpath
   200 150 36 45 90 arcn
stroke

newpath
   100 250 54 40 140 arc
stroke

newpath
   200 250 moveto
   200 250 54 40 140 arc
stroke
```

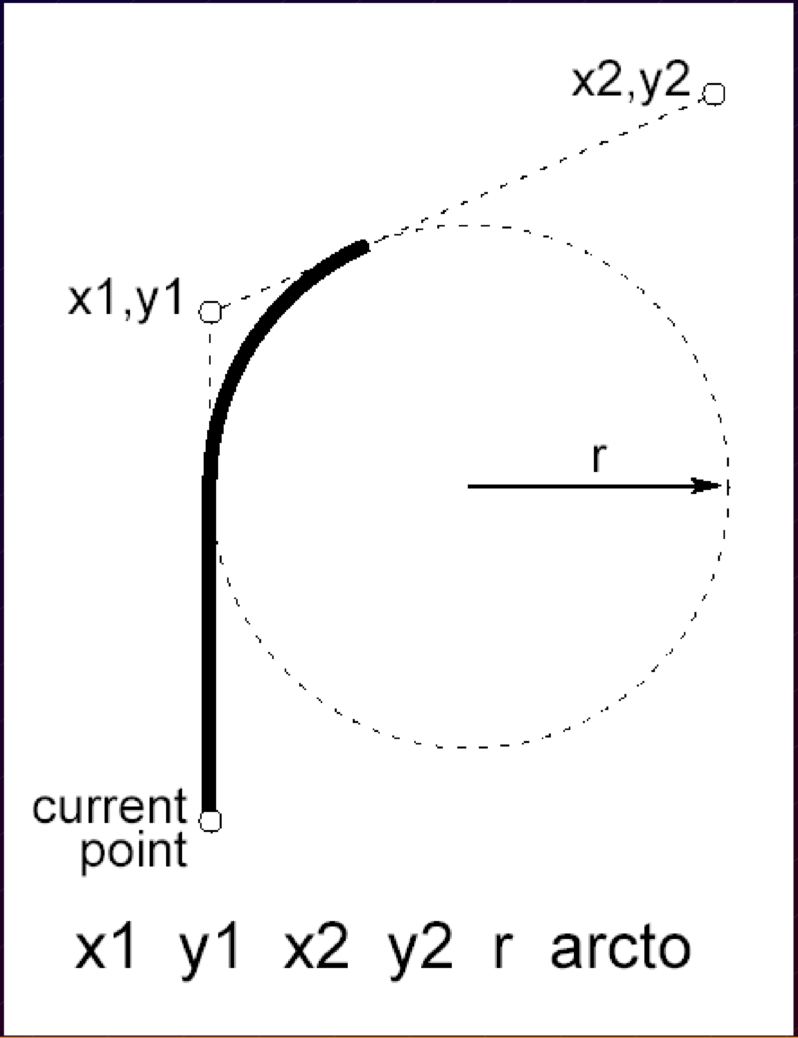
원과 타원



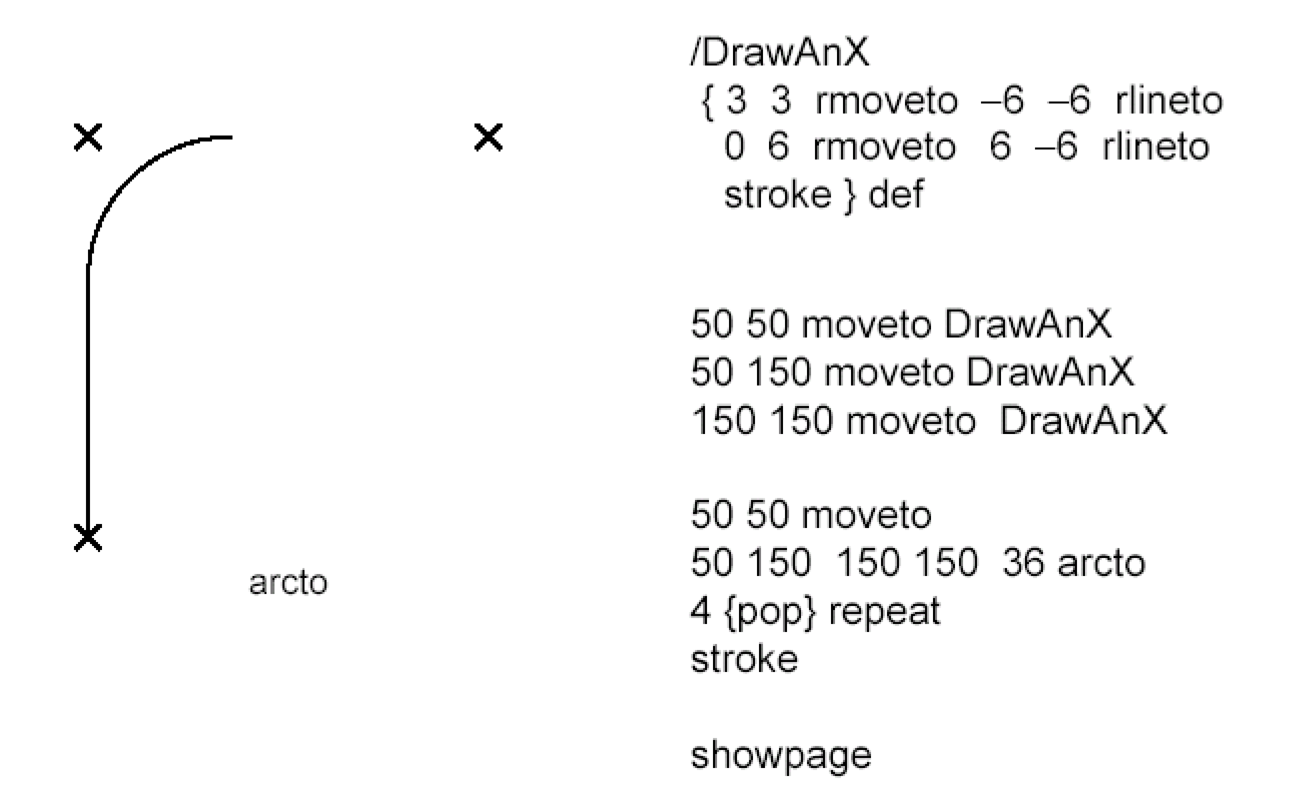
Ellipses

```
/doACircle
 { 0 0 54 0 360 arc stroke } def

/doAnEllipse
 { 1 .75 scale
   doACircle
   stroke } def

300 500 translate  doACircle

4 {0 –72 translate
   doAnEllipse} repeat

showpage
```

원호로 모서리 깍기

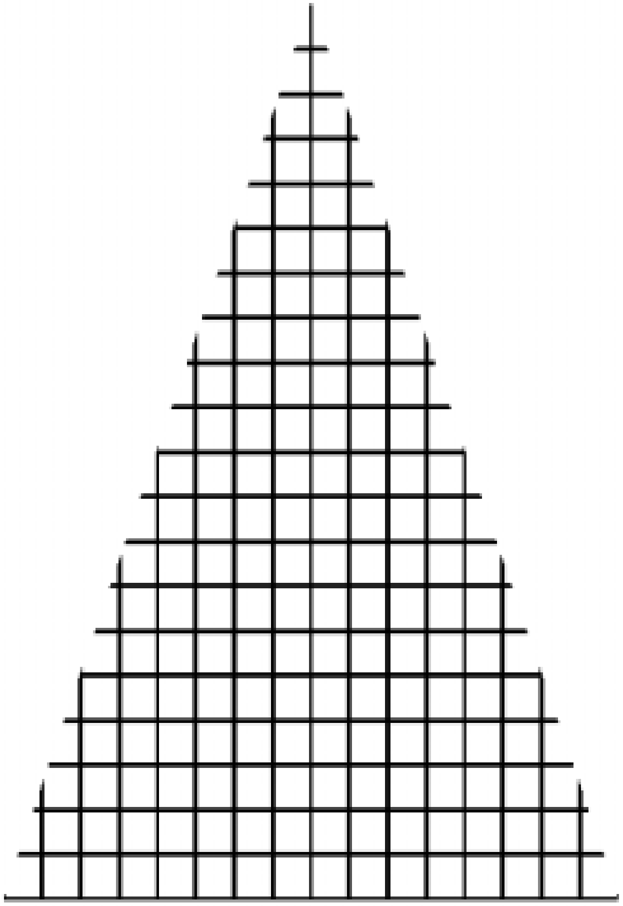


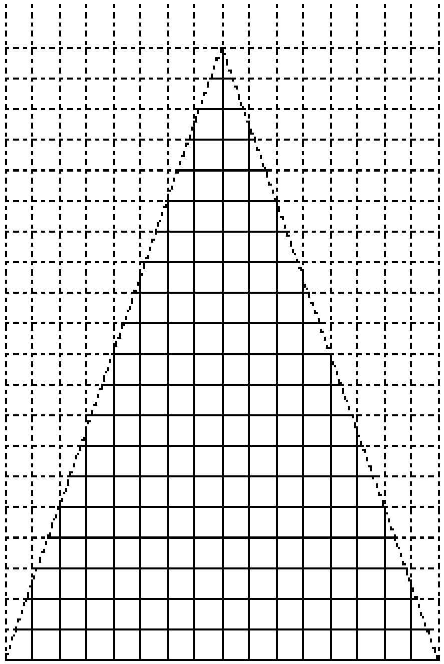
원호로 모서리 깎기



```
/DrawAnX
 { 3  3  rmoveto  –6  –6  rlineto
   0  6  rmoveto   6  –6  rlineto
   stroke } def


50 50 moveto DrawAnX
50 150 moveto DrawAnX
150 150 moveto  DrawAnX

50 50 moveto
50 150  150 150  36 arcto
4 {pop} repeat
stroke

showpage
```

클리핑





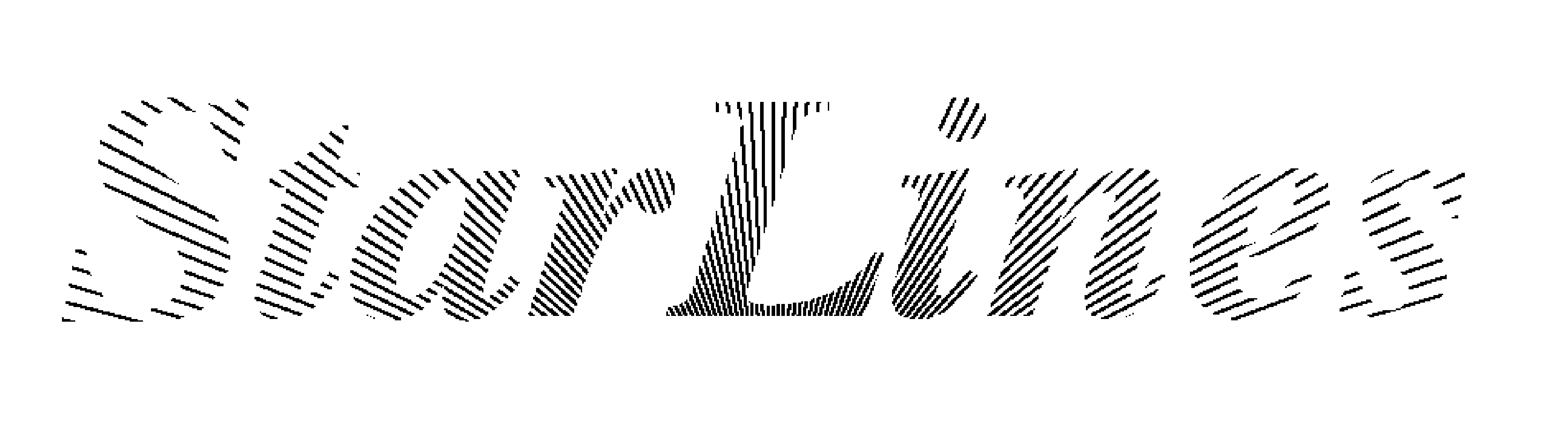
Only the part of the grid that falls within the triangular clipping path reaches the current page.

클리핑 경로설정

```
% ---- Procedures ----
/trianglepath
 { newpath
   0 0 moveto
   144 0 lineto
   72 200 lineto

   closepath } def
```

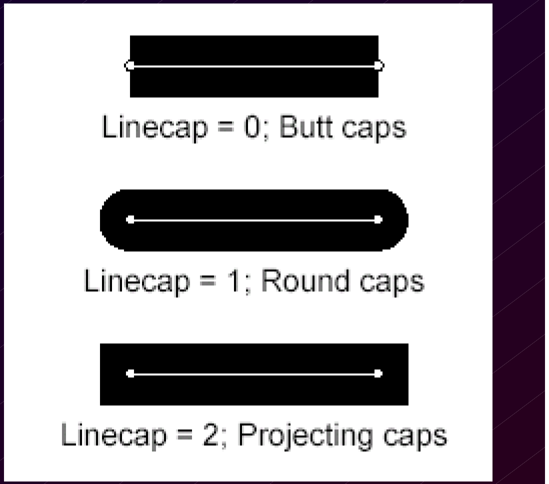
```
/verticals
 { newpath
   0 9 144
   { 0 moveto
     0 216 rlineto } for
   stroke } def
/horizontals
 { newpath
   0  10  200
   { 0 exch moveto
     144 0 rlineto } for
   stroke } def
% ---- Begin Program ---
230 300 translate
trianglepath  clip
verticals             %Do grid
horizontals
```

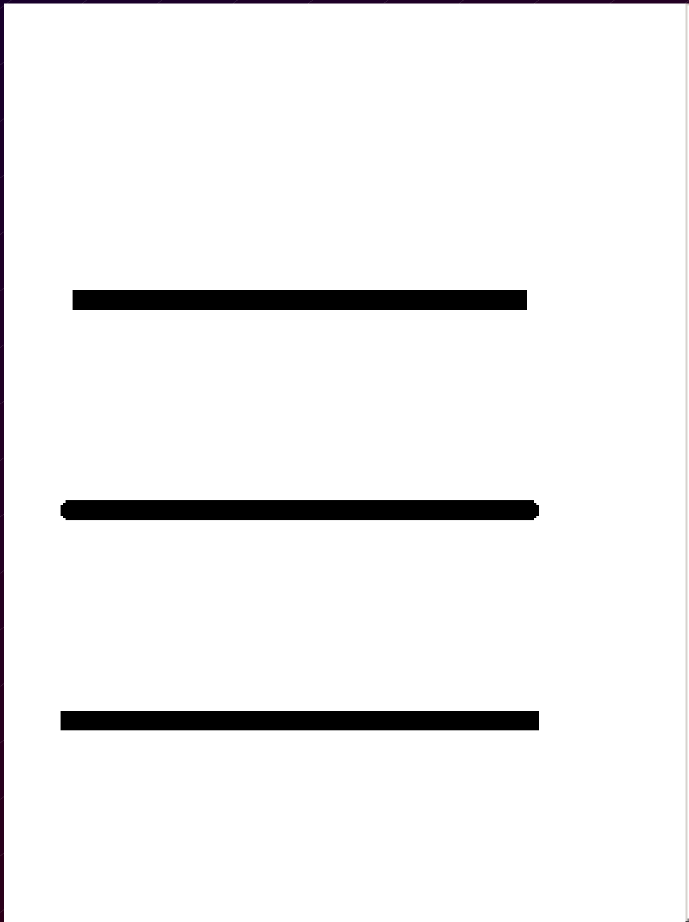
문자열로 클리핑한 경우

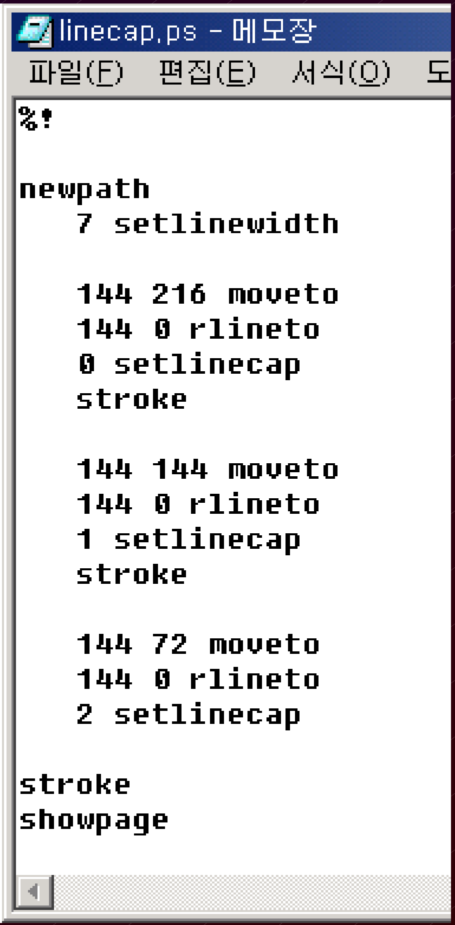


```
newpath
0 0 moveto
(StarLines) true
charpath clip
```

선분의 끝부분 마무리

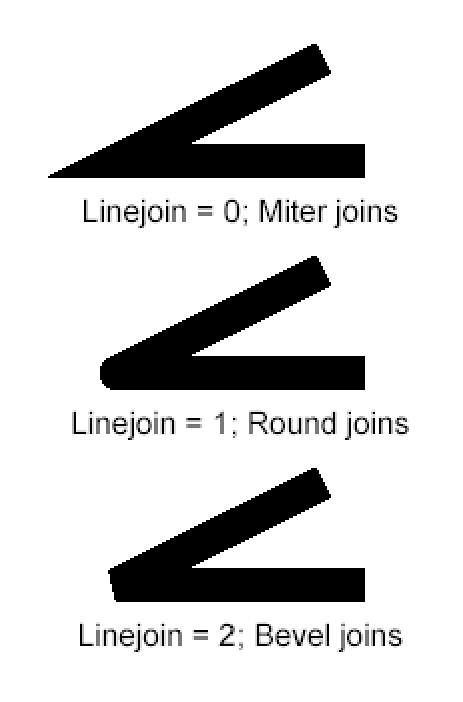






```
%!

newpath
   7 setlinewidth

   144 216 moveto
   144 0 rlineto
   0 setlinecap
   stroke

   144 144 moveto
   144 0 rlineto
   1 setlinecap
   stroke

   144 72 moveto
   144 0 rlineto
   2 setlinecap

stroke
showpage
```

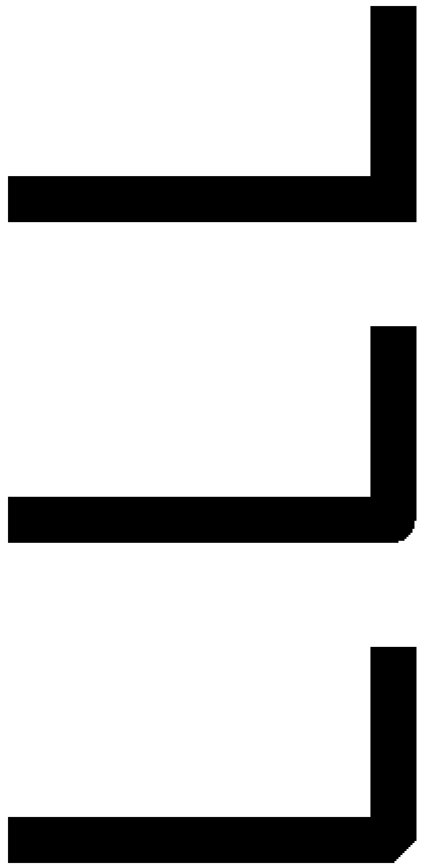
선분의 연결부분 마무리

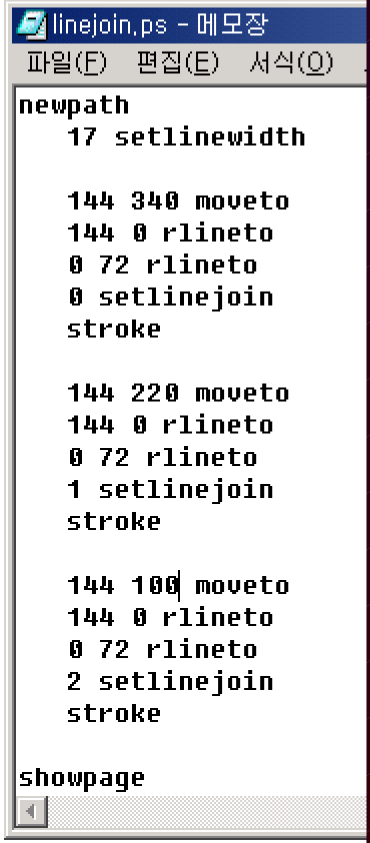


Linejoin = 0; Miter joins

Linejoin = 1; Round joins

Linejoin = 2; Bevel joins

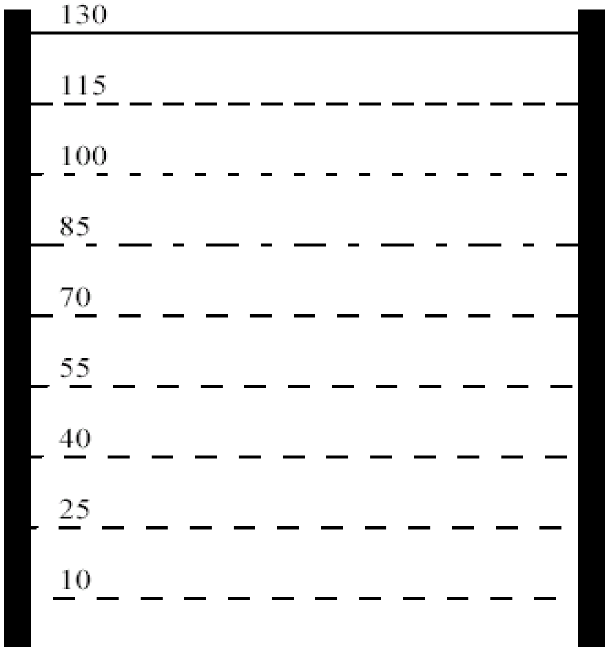




```
newpath
   17 setlinewidth

   144 340 moveto
   144 0 rlineto
   0 72 rlineto
   0 setlinejoin
   stroke

   144 220 moveto
   144 0 rlineto
   0 72 rlineto
   1 setlinejoin
   stroke

   144 100 moveto
   144 0 rlineto
   0 72 rlineto
   2 setlinejoin
   stroke

showpage
```

점선 그리기

```
[3  5  1  5]  0  setdash
```



```
.5 setlinewidth
[ ] 0 setdash doLine
[4 2] 0 setdash doLine
[2 4] 0 setdash doLine
[6 4 2 4] 0 setdash doLine
[4 4] 0 setdash doLine
[4 4] 1 setdash doLine
[4 4] 2 setdash doLine
[4 4] 3 setdash doLine
[4 4] 4 setdash doLine

showpage
```