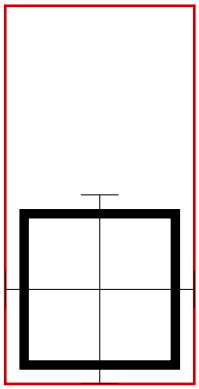
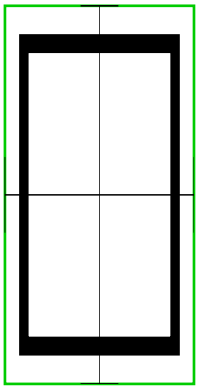
PostScript::Simple example file: EPS import functions (from a file)

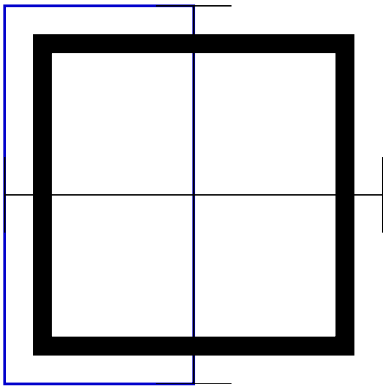
```
$ps->importepsfile("demo-square.eps", 20, 210, 45, 260);
```

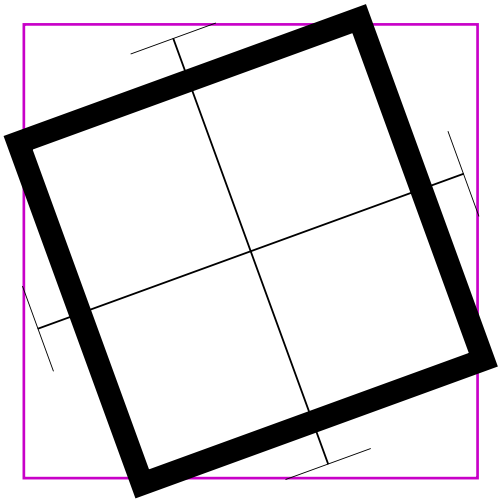


```
$ps->importepsfile({stretch => 1}, "demo-square.eps", 80, 210, 105, 260);
```



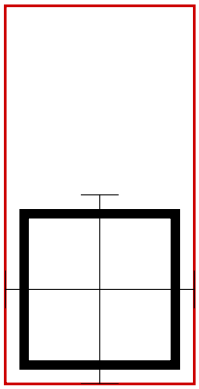
```
$ps->importepsfile({overlap => 1}, "demo-square.eps", 140, 210, 165, 260);
```



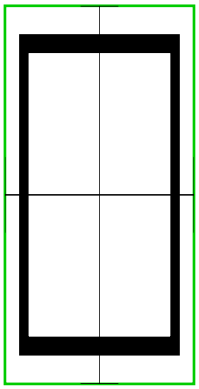


```
$eps = new PostScript::Simple::EPS
     (file => "demo-square.eps");
$eps->scale(60/100);
$eps->translate(50, 50);
$eps->rotate(20);
$eps->translate(-50, -50);
$ps->importeps($eps, 30, 30);
```

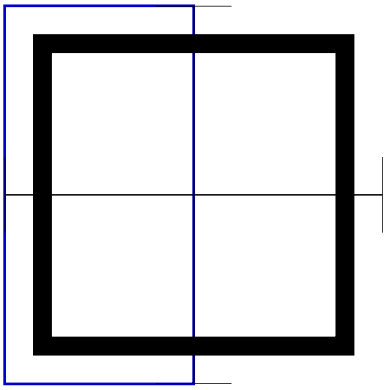
PostScript::Simple example file: EPS import functions (using internal EPS object)



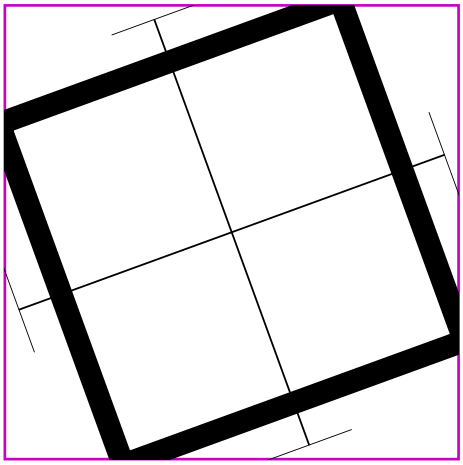
```
$directeps->reset();
$directeps->scale(25/$directeps->width());
$ps->importeps($directeps, 20, 210);
```



```
$directeps->reset();
$directeps->scale(25/$directeps->width(), 50/$directeps->height());
$ps->importeps($directeps, 80, 210);
```



```
$directeps->reset();
$directeps->scale(50/$directeps->height());
$ps->importeps($directeps, 140, 210);
```



```
$directeps->reset();
$directeps->translate(50, 50);
$directeps->rotate(20);
$directeps->translate(-50, -50);
# round-about way to set clipping path
$eps = new PostScript::Simple(eps => 1,
     xsize => 100, ysize => 100);
$eps->importeps($directeps, 0, 0);
$directeps = $eps->geteps();
$directeps->scale(60/100);
$ps->importeps($directeps, 30, 30);
```

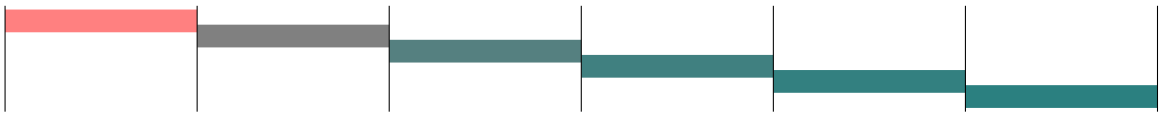
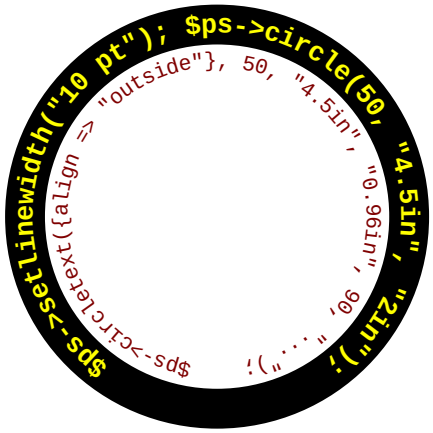
PostScript::Simple example file: Using different units

```
$ps->line(20,265, 190,265);  # default units is mm
```

```
$ps->setlinewidth("5 pt");
$ps->line(20,255,  190,255);
```

```
$ps->setlinewidth([0.25, "in");
$ps->line(20,232,  190,232);
```

```
$ps->setlinewidth("thin");  # thin is 0.4 pt
$ps->line(20,216,  190,216);
```



```
$y = 210;
for (my $x = 1; $x < 7; $x++) {
  $ps->setlinewidth([8, "dd"]);
  $ps->line([$x, "in"],$y, [$x+1, "in"],$y, 255/$x, 128, 128);  # also set colour
  $ps->setlinewidth("thin");
  $ps->line([$x, "in"],212, [$x, "in"],198, 0,0,0);  # set black
  $y -= 2;
}
$ps->line([7, "in"],212, [7, "in"],198, 0,0,0);
```