Processing
P04

1- Boucles

Cédric MASSART

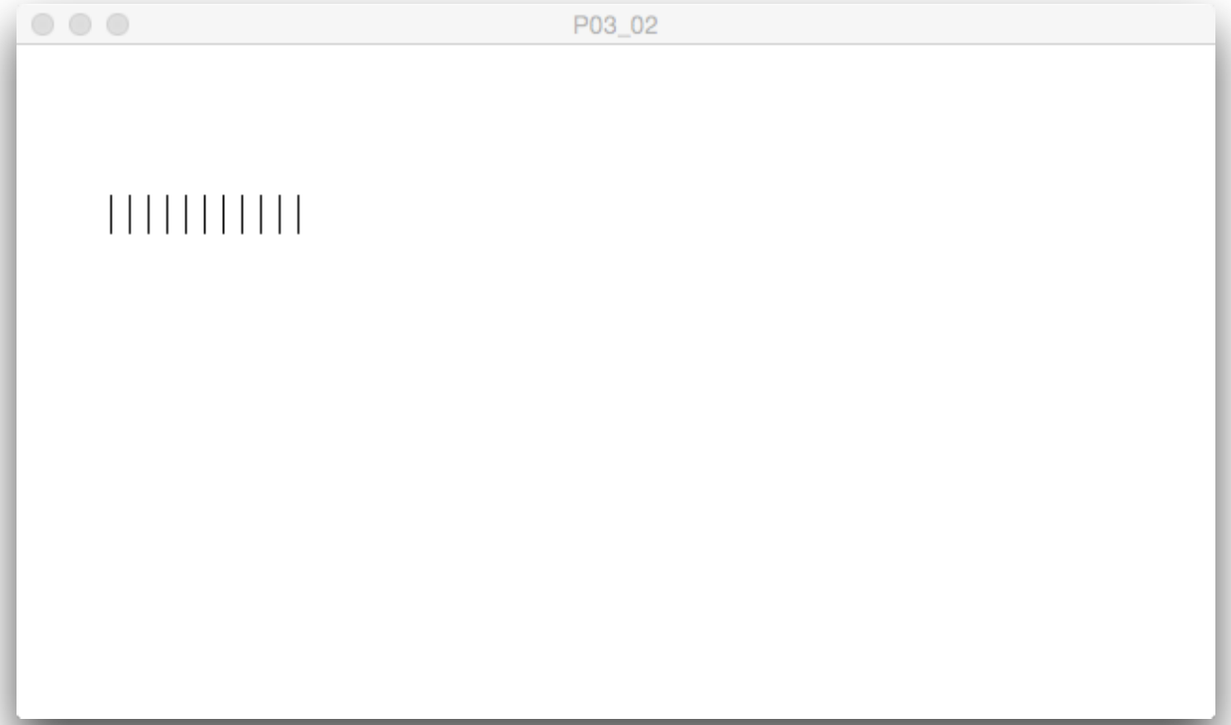
Boucles

Principe (1/5)

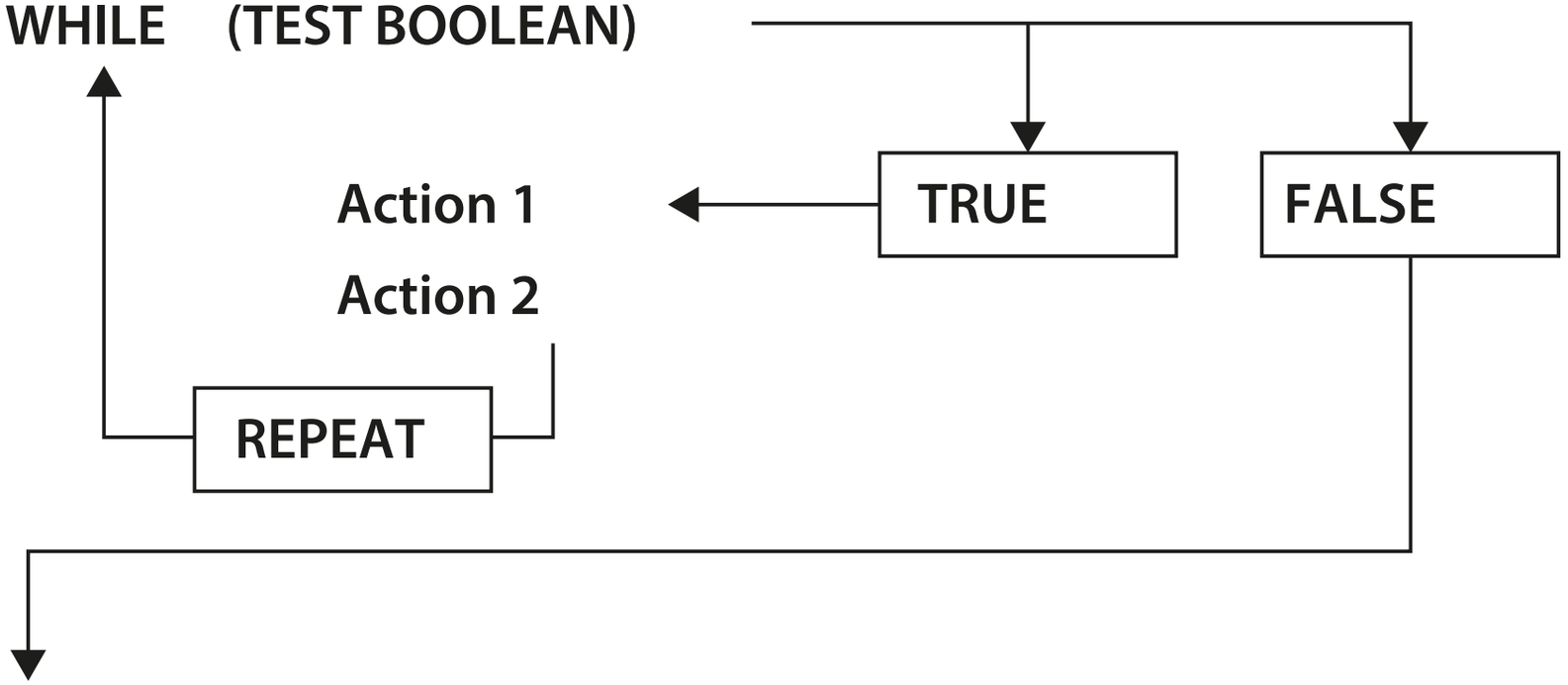
```
size(640,360);
background(255);

stroke(0);
line(50,60,50,80);
line(60,60,60,80);
line(70,60,70,80);
line(80,60,80,80);
line(90,60,90,80);
line(100,60,100,80);
line(110,60,110,80);
line(120,60,120,80);
line(130,60,130,80);
line(140,60,140,80);
line(150,60,150,80);
```

Principe (2/5)

```
size(640,360);
background(255);

stroke(0);
int y = 80;
int x = 50;
int spacing = 10;
int len = 20;

// 1
line(x,y,x,y + len);
x = x + spacing;

// Le reste
line(x,y,x,y + len);
x = x + spacing;
line(x,y,x,y + len);
x = x + spacing;
line(x,y,x,y + len);
x = x + spacing;
line(x,y,x,y + len);
x = x + spacing;
line(x,y,x,y + len);
x = x + spacing;
line(x,y,x,y + len);
x = x + spacing;
line(x,y,x,y + len);
x = x + spacing;
line(x,y,x,y + len);
x = x + spacing;
line(x,y,x,y + len);
x = x + spacing;
line(x,y,x,y + len);
x = x + spacing;
line(x,y,x,y + len);
```

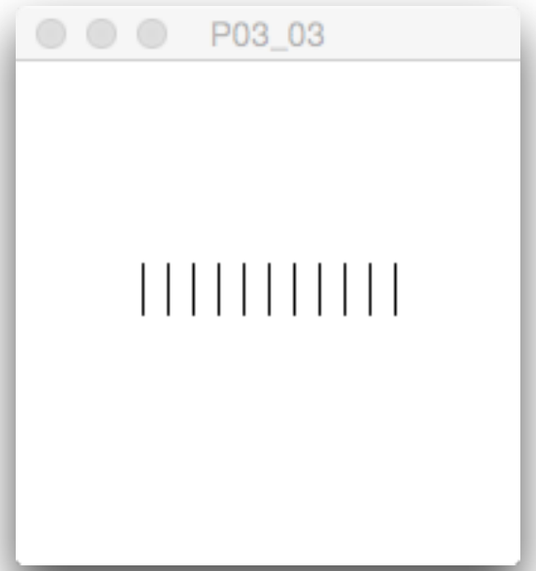


Principe (3/5)



Principe (4/5)

```
size(200, 200);
background(255);

int y = 80;
int x = 50;
int spacing = 10;
int len = 20;

int endLegs = 150;  1
stroke(0);

while (x <= endLegs) {
  line (x, y, x, y + len);
  x = x + spacing;
}
```



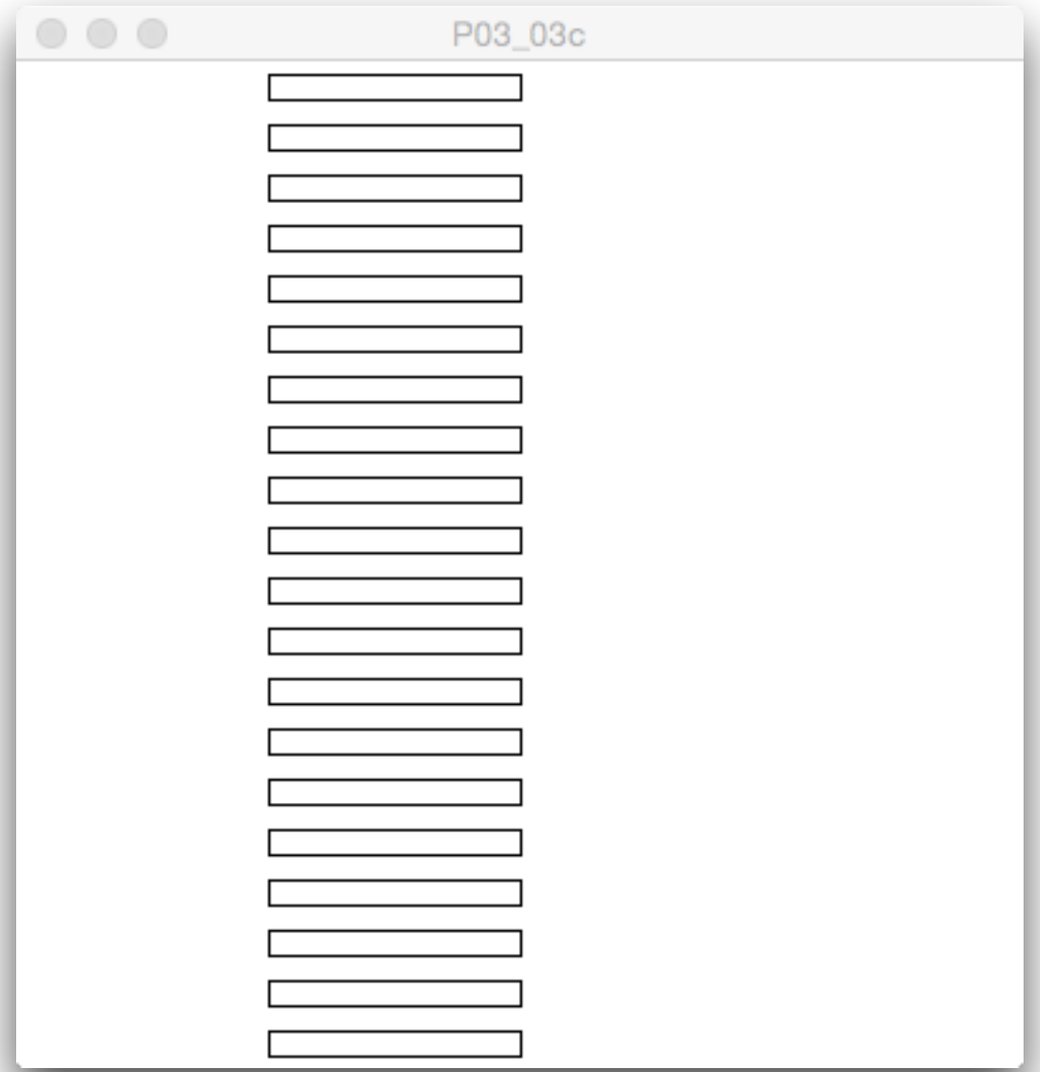
1: variable contenant la valeur de fin

Principe (5/5)

```
size(200, 200);
background(255);

int y = 80;
int x = 50;
int spacing = 4;
int len = 20;

int endLegs = 150;
stroke(0);

while (x <= endLegs) {
  line (x, y, x, y + len);
  x = x + spacing;
}
```

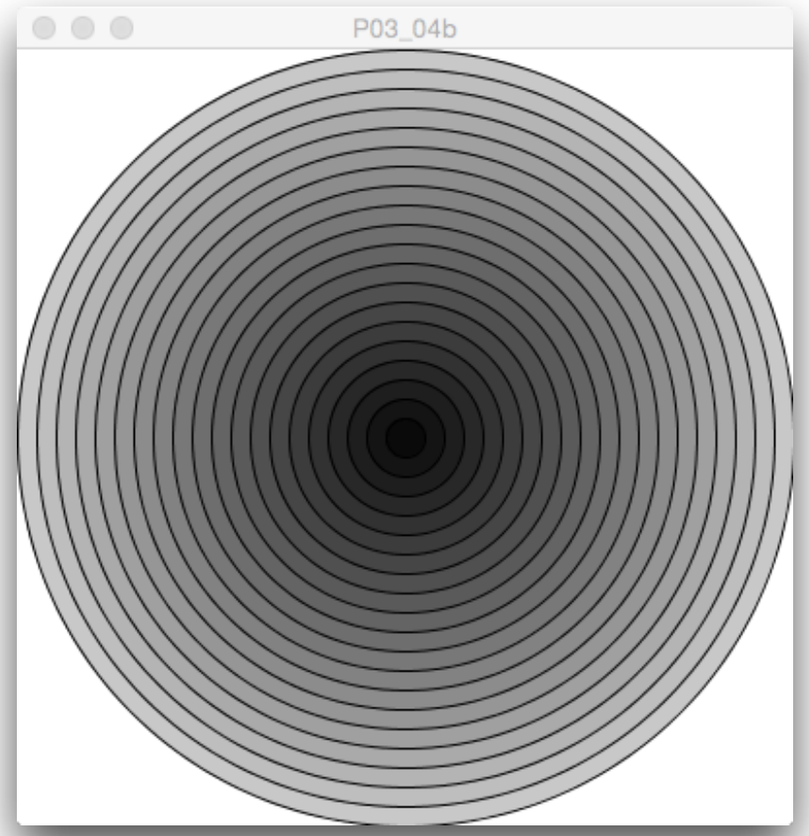
Principe (5/6)

```
size(400, 400);
background(255);

int y = 5;

stroke(0);

while (y <= height) {
  rect (100, y, 100, 10);
  y = y + 20;
}
```



P03_03c.pde

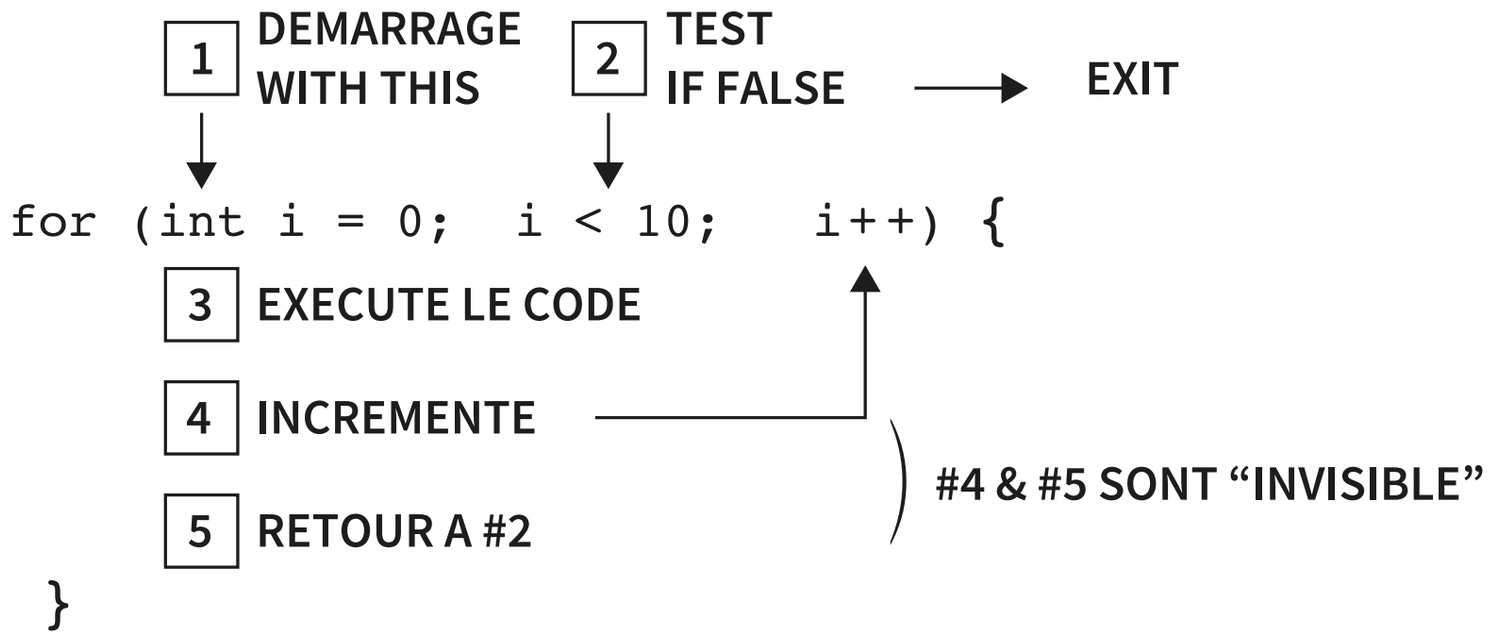
Exercices



P03_04b.pde

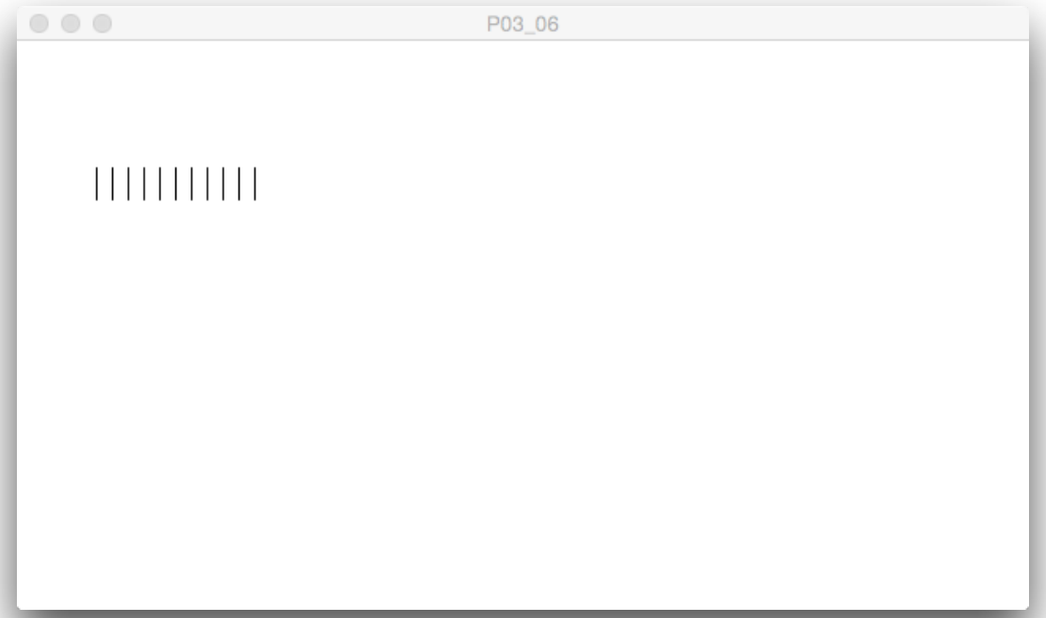
La boucle ‘for’

- initialisation
- condition
- incrémentation

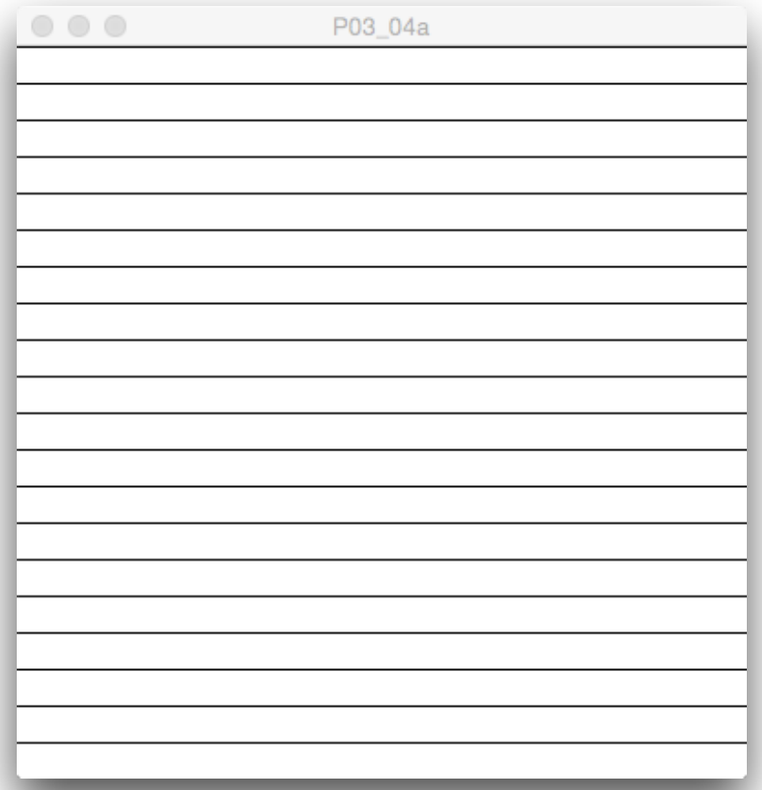


La boucle ‘for’

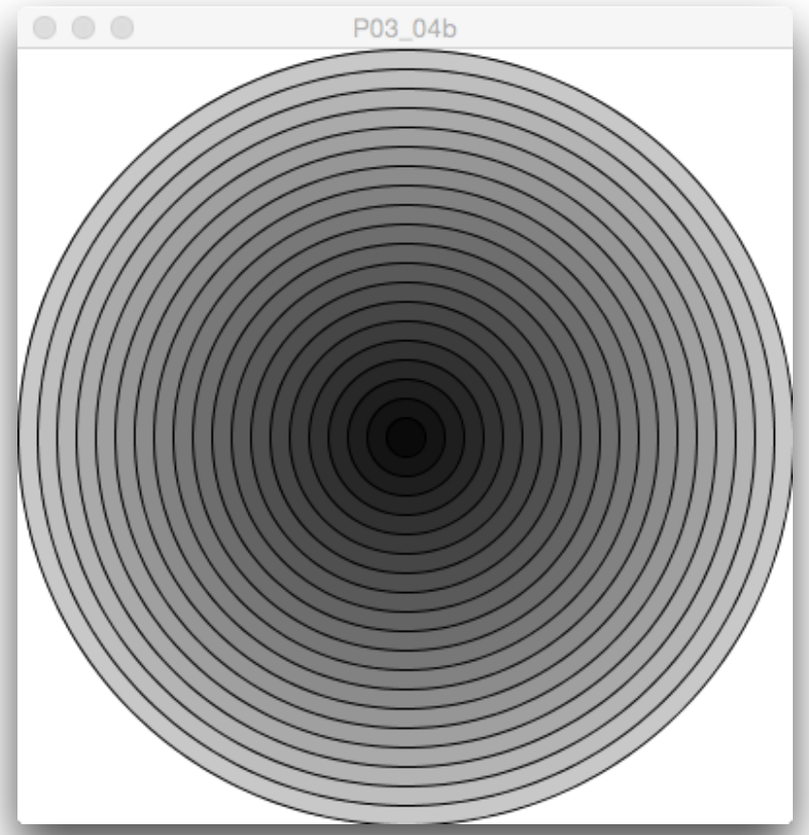
```
size(640,360);
background(255);

int y = 80;
int spacing = 10;
int len = 20;

for (int x = 50; x <= 150; x += spacing) {
  line(x,y,x,y + len);
}
```



Exercice



P03_07a.pde



P03_07b.pde