## **Premier** « Sketch »

•	Environnement   Processing 3.0.1	
		Java 🔻
	Environnement <b>v</b>	
1	<pre>print("Hello !!");</pre>	
2		
3		
5		
6		_
7		_
8		
9		
10		
12		
13		
	Hello !!	
	Console	

Processing-P01

#### **Premier « Sketch »**

size(200, 200);
background(102);



# Règles de mise en forme

• Processing est sensible à la casse

ENSA LIMOGES

- Processing n'est pas troublé lorsque l'on place plusieurs espaces
- Dans Processing les sauts de ligne sont cosmétiques
- Ce qui est ouvert doit être fermé :
  - les doubles guillemets et les simples guillemets,
  - parenthèses (), accolades {} et cochet []
- Les commentaires /\* \*/ et //

## La documentation

Processing p5.js P Process	sing		
Language Libraries Tools	Reference. Processi sketchbook.	ng was designed to b	oe a flexible software
Environment	Structure	Shape	Color
	<pre>() (parentheses) , (comma) . (dot) /**/ (multiline comment) /***/ (doc comment) // (comment) ; (semicolon) = (assign) [] (array access) {} (curly braces) catch class</pre>	createShape() loadShape() PShape 2D Primitives arc() ellipse() line() point() quad() rect() triangle()	Setting background() clear() colorMode() fill() noFill() noStroke() stroke() stroke() Creating & Reading alpha() blue()
	draw() exit() extends false final implements import loop() new noLoop()	Curves bezier() bezierDetail() bezierPoint() bezierTangent() curve() curveDetail() curvePoint() curveTangent()	brightness() color() green() hue() lerpColor() red() saturation()
	null popStyle() private rublic	curveTightness() 3D Primitives	createImage() PImage

Processing-P01

#### **Pixels**

# Système de coordonnées



#### Ordinateur

# Les formes



#### Afficher un point



- en Processing:
  - point(x, y);

#### Afficher une ligne



- en Processing:
  - line(x1, y1, x2, y2);

#### Afficher un rectangle



- en Processing:
  - rect(x, y, width, height);
  - NOTE: le mode par défaut est CORNER

#### Afficher un rectangle (mode CENTER)



- en Processing:
  - rectMode(CENTER);
  - rect(x, y, width, height);

### Afficher un rectangle (mode CORNERS)



- en Processing:
  - rectMode(CORNERS);
  - rect(x1, y1, x2, y2);

## Afficher une ellipse (mode CENTER)



- en Processing:
  - ellipseMode (CENTER) ;
  - ellipse(x, y, width, height);
  - NOTE: le mode par défaut est CENTER

## Afficher une ellipse (mode CORNER)



- en Processing:
  - ellipseMode(CORNER);
  - ellipse(x, y, width, height);

### Afficher une ellipse (mode CORNERS)



- en Processing:
  - ellipseMode(CORNERS);
  - ellipse(x1, y1, x2, y2);

#### Les autres formes



On reviendra dessus un peu plus tard.



# Niveau de gris

0	50	87	162	209	255	

#### Mise en place



#### stroke () et fill()



# noFill()





### **Deux formes**





# Exercice



# Couleur RVB (RGB)

Pixels Codage de la teinte saumon

- rouge = 100%, vert = 80%, bleu = 60% ou
- rouge = 255, vert = 204, bleu = 153





# Couleur RVB (RGB)

dans Processing :

Menu -> Outils -> Sélecteurs de couleurs

	Sélecteur de couleurs		
P			
		н	°
		s	0 %
		В	99 %
		R	255
		G	255
		В	255
			#FFFFFF
			Copier
			Annuler
		_	

## Exemple

```
background(255);
noStroke();
// Rouge
fill(255, 0, 0);
ellipse(20, 20, 16, 16);
// Rouge foncé
fill(127, 0, 0);
ellipse(40, 20, 16, 16);
// Rose
fill(255, 200, 200);
ellipse(60, 20, 16, 16);
```



#### Transparence

```
size(480, 270);
background(0);
noStroke();
// trois arguments donc 100% d'opacité.
fill(0, 0, 255);
rect(0, 0, 240, 200);
// 255 donc 100% d'opacité.
fill(255, 0, 0, 255);
rect(0, 0, 480, 40);
// 75% d'opacité.
fill(255, 0, 0, 191);
rect(0, 50, 480, 40);
// 55% d'opacité.
fill(255, 0, 0, 127);
rect(0, 100, 480, 40);
// 25% d'opacité.
fill(255, 0, 0, 63);
rect(0, 150, 480, 40);
```

000	Environnement		
_			

# Exercice

