

MEMO PROCESSING

| | Instructions | Résultat affiché sur la console |
|------------|--|------------------------------------|
| Addition 1 | <pre>int a,b,c; a=1; b=2; c=a+b; println("Résultat: "+c);</pre> | Résultat: 3 |
| Addition 2 | <pre>float a,b,c; a=10.30; b=45.90; c = a + b; println("Résultat: " + c);</pre> | Résultat: 56.2 |
| Addition 3 | <pre>int a,b,c; a = 112; a ++; println("Résultat 1: " + a); a --; println("Résultat 2: " + a);</pre> | Résultat 1: 113 Résultat 2: 112 |
| Décalage | <pre>int a,b,c; a = 112; a = a << 2; println("Résultat 1: " + a); a = a >> 1; println("Résultat 2: " + a);</pre> | Résultat 1: 448 Résultat 2: 224 |

| | Instructions | Résultat affiché sur la console |
|--------------|--|---------------------------------|
| Opérateur Ou | <pre>int a,b,c; a = 112; b = 15; c = a b; println("Résultat : " + c);</pre> | Résultat : 127 |
| Opérateur ET | <pre>int a,b,c; a = 117; b = 0xf8; c = a & b; println("Résultat : " + c);</pre> | Résultat : 112 |
| Modulo | <pre>int a,b,c; a = 159; b = 10; c = a % b; println("Résultat : " + c);</pre> | Résultat : 9 |
| Logique | <pre>boolean a,b,c; a = true; b = false; c = a & b; println("Résultat : " + c);</pre> | Résultat : false |


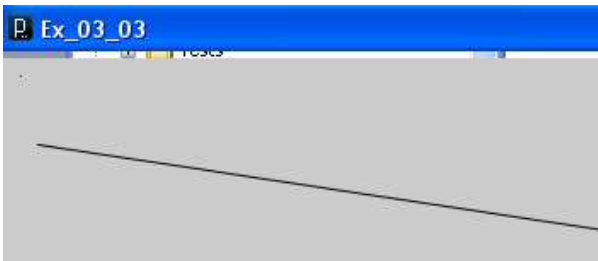
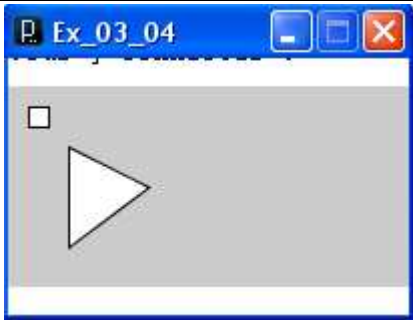

| | Instructions | Résultat affiché sur la console |
|---------------|--|---------------------------------|
| Chaine 1 | <pre>String chaine; chaine="test"; println("votre texte: " + chaine);</pre> | votre texte: test |
| length | <pre>String chaine; int longueur; chaine="bonjour"; longueur=chaine.length(); println("votre texte: " + chaine + " long: " + longueur);</pre> | votre texte: bonjour long: 7 |
| charAt | <pre>String chaine; char car; chaine="bonjour"; car=chaine.charAt(3); println("votre texte: " + chaine + " car : " + car);</pre> | votre texte: bonjour car : j |
| concaténation | <pre>String chaine1, chaine2, chaine3; chaine1="sti2d"; chaine2="sin"; chaine2 = chaine2.toUpperCase(); chaine3 = chaine1 + " " + chaine2; println("votre texte: " + chaine3);</pre> | votre texte: sti2d SIN |
| A voir | <p>indexOf(char c) retourne la position de la première occurrence du caractère c dans la chaîne : par ex. "bonjour".indexOf('o') renvoie 1</p> <p>substring(int beginIndex, int endIndex) renvoie la sous-chaîne allant de beginIndex à endIndex (exclus) : par ex. "bonjour".substring(1, 5) renvoie "onjo"</p> <p>toLowerCase() et toUpperCase() renvoient une nouvelle chaîne contenant la chaîne initiale convertie en minuscules ou majuscules.</p> | |

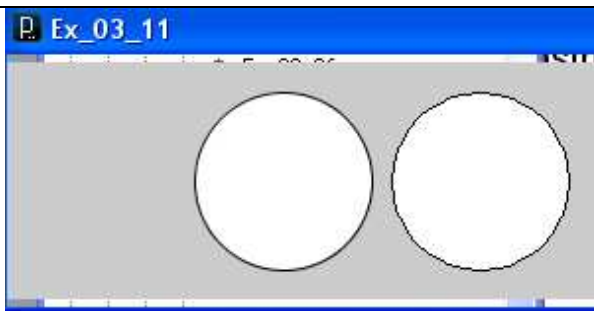
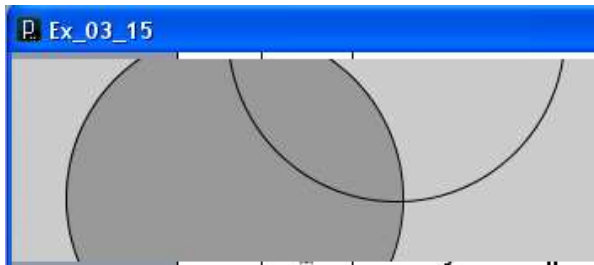
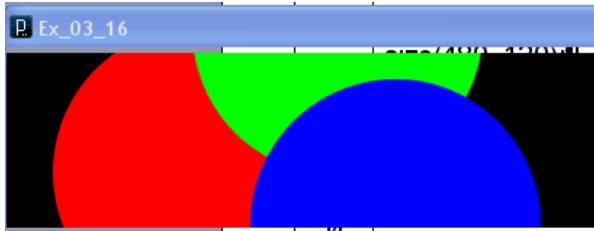
| | Instructions | Résultat affiché sur la console |
|---|---|---|
| Tableau | <pre>int tab[]={1,5,6,7,2}; println("Chiffre " + tab[2]);</pre> | Chiffre 6 |
| Boucle for | <pre>int tab[]={1,5,6,7,2}; int n; for(n = 0; n <=4; n++) println("Chiffre " + tab[n]);</pre> | Chiffre 1 Chiffre 5 Chiffre 6 Chiffre 7 Chiffre 2 |
| Random | <pre>int tab[]= new int[5]; for(int n = 0; n <=4; n++) { tab[n]= (int) random(10); println("Chiffre " + tab[n]); }</pre> | Chiffre 3 Chiffre 1 Chiffre 1 Chiffre 9 Chiffre 3 |
| http://www.processing.org/reference/random_.html | | |
| Condition IF (1) | <pre>int a = 25; if (a > 20) println("chiffre supérieur à 20"); else println("chiffre inférieur ou égal à 20");</pre> | chiffre supérieur à 20 |
| Condition IF (2) | <pre>int a = 1; int b = 3; if ((a == 2) && (b == 3)) println("ok"); else print("non");</pre> | non |

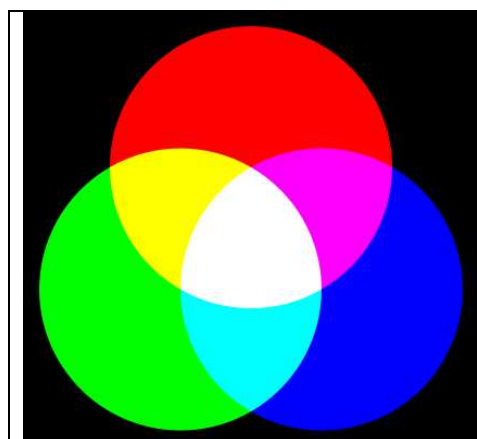
| | Instructions | Résultat affiché sur la console |
|------------------|--|--|
| Condition IF (3) | <pre>int tab[] = new int[100]; int n; int m=0; for (n = 0; n < 100; n ++) { tab[n] = (int) random(10); if (tab[n] == 5) m ++; } println("NB de 5 trouvés " + m);</pre> | NB de 5 trouvés 8 |
| Switch | <pre>int a = 3; switch(a) { case 1 : println("Bonjour"); break; case 2 : println("ça va"); break; case 3 : println("Au revoir"); break; default : println("c'est pas le bon chiffre"); }</pre> | Au revoir |
| While | <pre>int i; i = 0; while (i <= 6) { println("Voici la ligne " + i); i ++; }</pre> | Voici la ligne 0 Voici la ligne 1 Voici la ligne 2 Voici la ligne 3 Voici la ligne 4 Voici la ligne 5 Voici la ligne 6 |

| | Instructions | Résultat affiché sur la console |
|---------------|--|---|
| Do While | <pre>int i; i = 0; do { println("Voici la ligne " + i ++); } while (i <= 6);</pre> | <p>Voici la ligne 0 Voici la ligne 1 Voici la ligne 2 Voici la ligne 3 Voici la ligne 4 Voici la ligne 5 Voici la ligne 6</p> |
| Equal | <pre>String str; str = "toto"; print(str + "\n\r"); if (str.equals("toto") == true) print("oui"); else print("non");</pre> | <p>toto oui</p> |
| String to int | <pre>String str; int n = 0; str = "120"; n = Integer.parseInt(str); print(n + "\n\r");</pre> | <p>120</p> |
| Double to int | <pre>float m; int n; m = 45.6; n = (int) m; print(m + " " + n + "\n\r");</pre> | <p>45.6 45</p> |

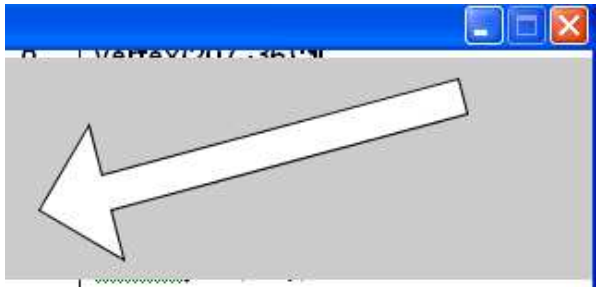

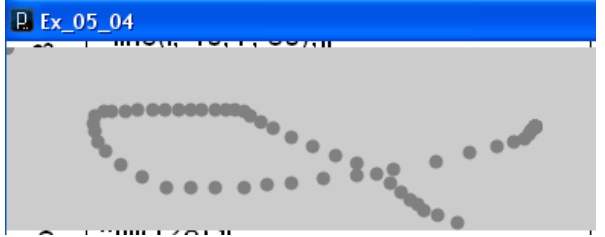
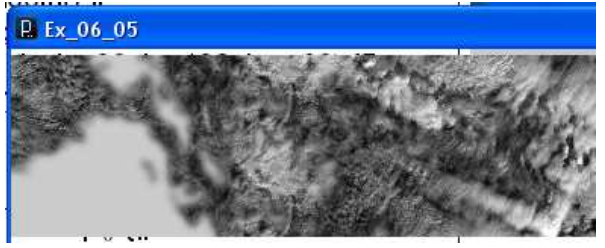
Le graphisme avec Processing

| | Instructions | Résultat affiché |
|--------------------------|--|--|
| La fenêtre graphique | // définit une fenetre de 800 x600 size(800, 600); |  |
| Point et ligne | size(480, 120); line(20, 50, 420, 110); point(10, 10); |  |
| Triangle et quadrilatère | size(200, 100); quad(10, 10, 20, 10, 20, 20, 10, 20); triangle(30, 30, 70, 50, 30, 80) |  |
| Rectangle et ellipse | size(480, 120); rect(10, 20, 50, 20); ellipse(70, 60, 50, 18); |  |


| | Instructions | Résultat affiché |
|-------------|---|--|
| Effet lisse | <pre>size(480, 120); smooth(); // Turns on smoothing ellipse(140, 60, 90, 90); noSmooth(); // Turns off smoothing ellipse(240, 60, 90, 90);</pre> |  |
| Remplissage | <pre>size(480, 120); // Medium gray fill(153); // Gray circle ellipse(132, 82, 200, 200); // Turn off fill noFill(); // Outline circle ellipse(228, -16, 200, 200); // Turn off stroke noStroke(); // Doesn't draw! ellipse(268, 118, 200, 200);</pre> |  |
| Couleur | <pre>size(480, 120); noStroke(); smooth(); background(0, 0, 0); // Dark color fill(255, 0, 0); // Red color ellipse(132, 82, 200, 200); // Red circle fill(0, 255, 0); // Green color ellipse(228, -16, 200, 200); // Green circle fill(0, 0, 255); // Blue color ellipse(268, 118, 200, 200); // Blue circle</pre> |  |



| Couleurs Primaires | Couleur secondaire |
|--------------------|--------------------|
| Rouge + Bleu | magenta |
| Bleu+Vert | cyan |
| Vert + Rouge | jaune |
| Rouge+Vert+Bleu | Blanc |

| | Instructions | Résultat affiché |
|----------------------|---|--|
| Formes | <pre>size(480, 120); beginShape(); vertex(180, 82); vertex(207, 36); vertex(214, 63); vertex(407, 11); vertex(412, 30); vertex(219, 82); vertex(226, 109); vertex(180, 82); endShape();</pre> |  |
| Boucle | <pre>size(480, 120); smooth(); strokeWeight(8); for (int i = 20; i < 400; i += 60) { line(i, 40, i, 80); }</pre> |  |
| Tracé avec la souris | <pre>void setup() { size(480, 120); fill(128); smooth(); noStroke(); } void draw() { ellipse(mouseX, mouseY, 9, 9); }</pre> |  |
| Charger une image | <pre>PImage img; void setup() { size(480, 120); img = loadImage("clouds.png"); } void draw() { background(204); image(img, 0, 0); image(img, 0, mouseY * -1); }</pre> |  |

Texte dans une fenêtre graphique

| | Instructions | Résultat affiché |
|-------|--|---|
| Texte | <pre>PFont font; void setup() { size(480, 120); smooth(); font = loadFont("AndaleMono-36.vlw"); textFont(font); } void draw() { background(102); textSize(36); text("That's one small step for man...", 25, 60); textSize(18); text("That's one small step for man...", 27, 90); }</pre> |  <p>Le fichier AndaleMono-36.vlw doit se trouver dans le répertoire data :</p> <p>Mes documents\Processing\texte\data</p> |

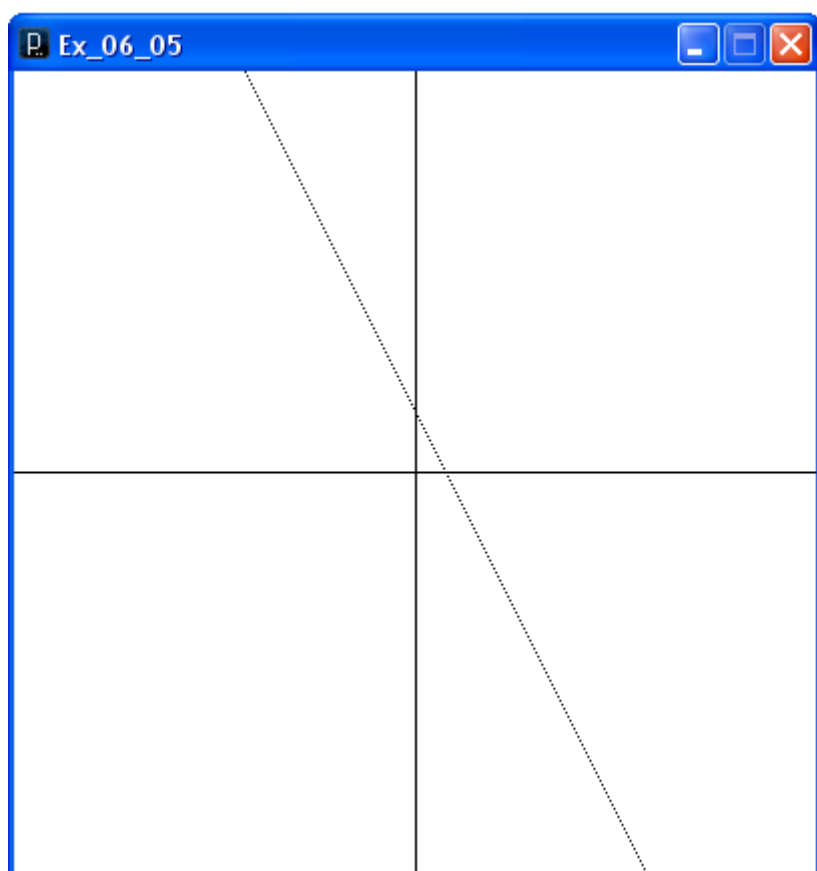
Lecture et écriture dans un fichier texte

| | Instructions | Résultat affiché |
|-------|--|--|
| Read | <pre>import java.io.*; try{ BufferedReader in = new BufferedReader(new FileReader("c:/temp/essai.txt")); String s; while((s = in.readLine()) != null){ println(s); } in.close(); }catch(IOException e){ e.printStackTrace(); }</pre> | <p>ceci est un essai de lecture du fichier essai.txt</p> |
| Write | <pre>import java.io.*; try{ //setting the second parameter in the FileWriter to true //appends the text to what is already there. Otherwise it //will overwrite the file. BufferedWriter out = new BufferedWriter(new FileWriter("c:/temp/essai.txt",true)); out.write("Hello again, "); out.close(); println("ok"); } catch(IOException e) { e.printStackTrace(); }</pre> | <p>ceci est un essai de d'écriture du texte Hello again, dans le fichier essai.txt</p> |

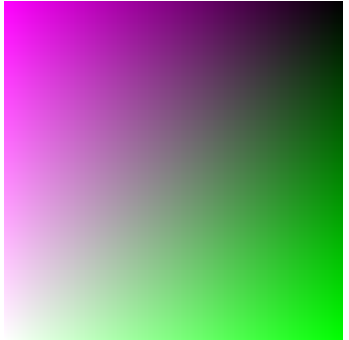
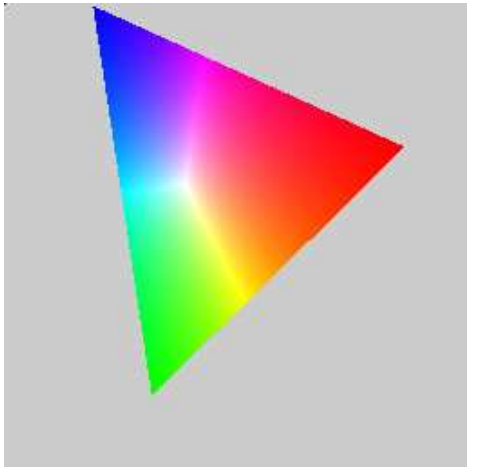
Tracé de fonctions mathématiques $y=2x+30$

```
void setup() {  
  size(400, 400);  
  background(255);  
}  
  
void draw() {  
  line (0,200,399,200);  
  line (200,0,200,399);  
  int x1=0;  
  int y1=0;  
  for (x1=-200;x1<200;x1++){  
    y1=2*x1+30;  
    point(200+x1, 200-y1);  
  }  
}
```

Résultat affiché



Gestion des couleurs

| | Instructions | Résultat affiché |
|---------|--|--|
| Dégradé | <pre> void setup() { size(256, 256); } void draw() { color c; for (int y = 0; y < 256; y++) { for (int x = 0; x < 256; x++) { c = color(255-x, y, 255-x); stroke(c); point (x,y); } } } void keyPressed() { println("sauvegarde image"); save("degrade.png"); } </pre> |  |
| Palette | <pre> void setup() { size(256, 256); float X,Y,Z; int x,y; color c; for (int r = 0; r <= 255; r ++) { println(r); for (int v = 0; v <=255 ; v ++) for (int b = 0; b <= 255; b ++) { X = 0.49 * r + 0.31 * v + 0.20 * b; Y = 0.17697 * r + 0.81240 * v + 0.01063 * b; Z = 0.01 * v + 0.99 * b; x = (int) (X*300 / (X + Y + Z)); y = (int) (Y*300 / (X + Y + Z)); c = color(r, v,b); stroke(c); point (x,y); } } } } void draw() { } void keyPressed() { println("sauvegarde image"); save("palette.png"); } </pre> |  <p>(le temps de traitement est long)</p> |

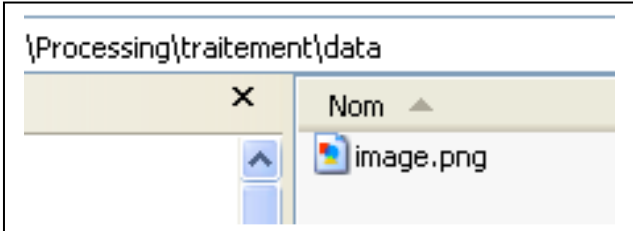
Extraction de vert d'une image

```
PImage img;
```

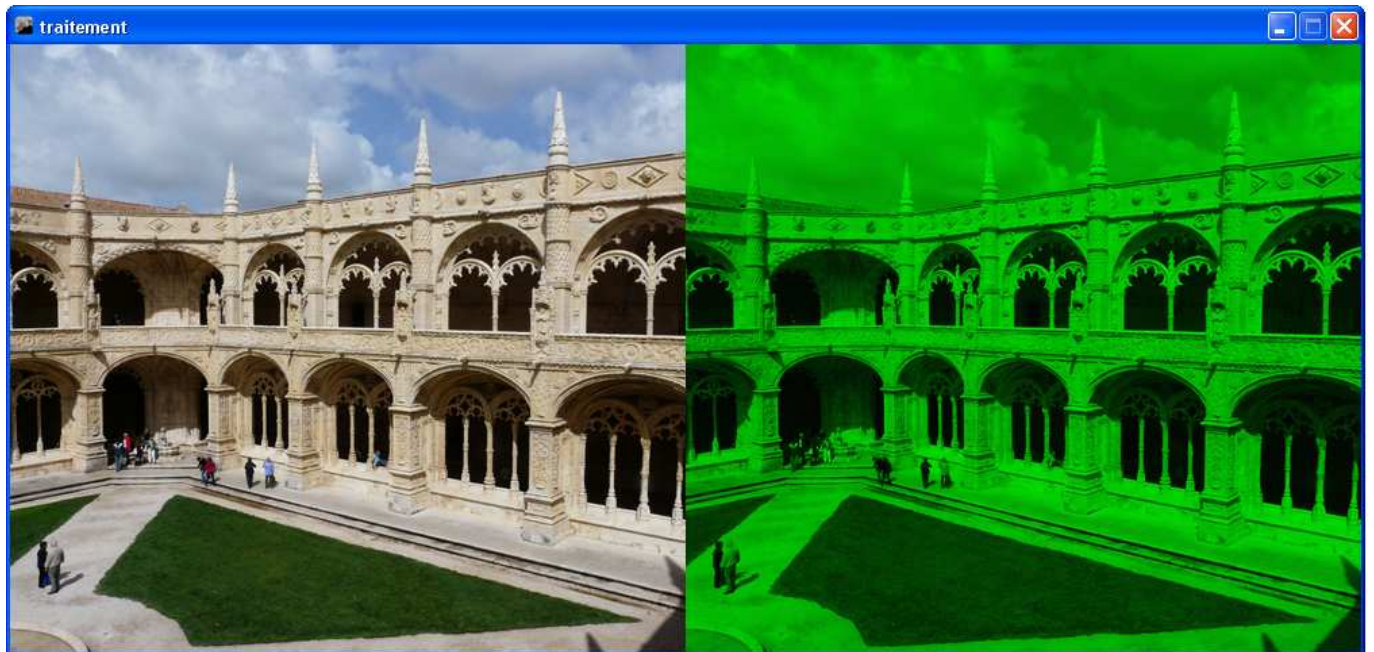
```
void setup() {  
  size(500*2, 450);  
  img = loadImage("image.png"); // image dans répertoire data du projet  
  image(img, 0, 0);  
  img.loadPixels(); // l'image est mise dans un buffer  
}
```

```
void draw() {  
}
```

```
void keyPressed() {  
  int index=0;  
  int vert;  
  color c;  
  for (int y = 0; y < 450; y++) {  
    for (int x = 0; x < 500; x++) {  
      vert = (int) green(img.pixels[index]); // http://processing.org/reference/green\_.html  
      c = color(0, vert, 0);  
      stroke(c);  
      point (x+500,y);  
      index++;  
    }  
  }  
  save("deuximages.png"); //sauve la fenêtre entière  
}
```



Résultat affiché



Ethernet : Client pour serveur industriel type Sollae

Client

```
import processing.net.*;

Client c;
String input;
int data[];

void setup()
{
  size(450, 255);
  background(204);
  stroke(0);
  frameRate(5); // Slow it down a little
  rect(50, 50, 100, 100);
  rect(200, 50, 100, 100);

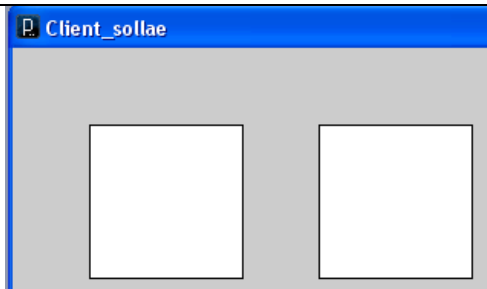
  // Connect to the server's IP address and port
  // Replace with your server's IP and port
  c = new Client(this, "192.168.1.10", 23);

  void draw()
  {
    if (mousePressed == true) {

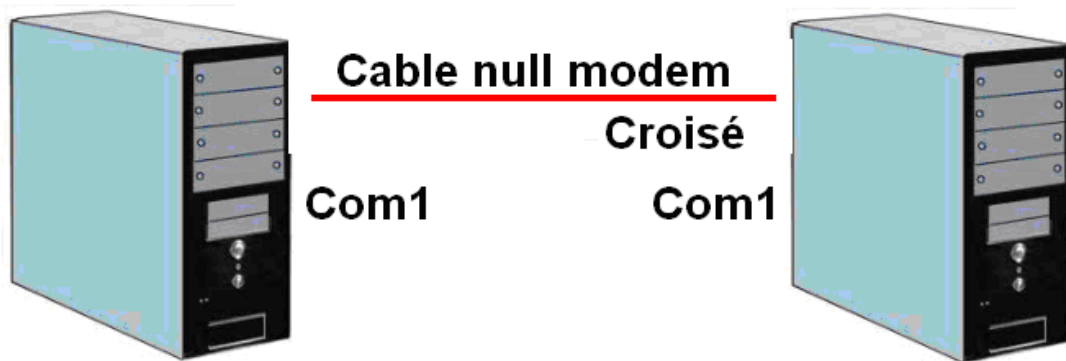
      if ((mouseY>50) && (mouseY<150)){
        if ((mouseX>50) && (mouseX<150)) c.write("1");
        if ((mouseX>200) && (mouseX<300)) c.write("2");
      }
    }
    // Receive data from server
    if (c.available() > 0) {
      input = c.readString();
      print(input);
    }
  }

  void keyPressed() {
    c.write(key);
  }
}
```

Résultat affiché



Liaison série (envoi et réception de caractères)



```
import processing.serial.*;

Serial myPort; // Create object from Serial class
int val;       // Data received from the serial port

void setup()
{
  size(200, 200);
  String portName = Serial.list()[0];
  myPort = new Serial(this, portName, 9600);
}

void keyPressed() {
  myPort.write(key);
}
```

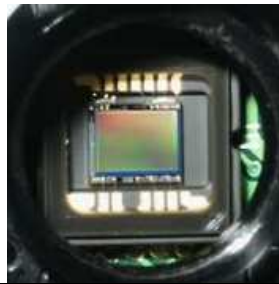
```
import processing.serial.*;

Serial myPort; // Create object from Serial class
int val;       // Data received from the serial port

void setup()
{
  size(200, 200);
  String portName = Serial.list()[0];
  myPort = new Serial(this, portName, 9600);
}

void draw()
{
  if ( myPort.available() > 0) { // If data is available,
    val = myPort.read();        // read it and store it in val
    println(val);
  }
}
```

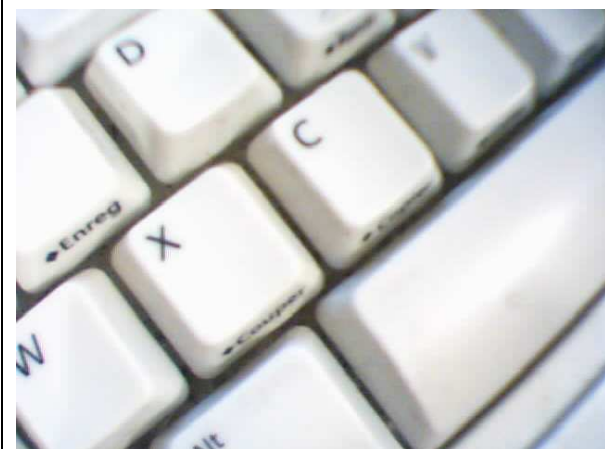

Gestion d'une Webcam



```
import processing.video.*;
```

```
Capture video;
```

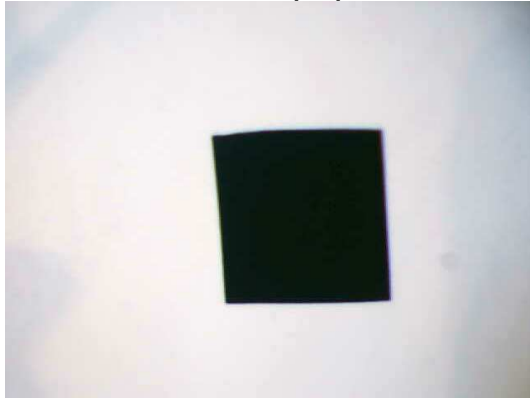
```
void setup() {  
  size(640, 480, P2D);  
  video = new Capture(this, width, height);  
  video.start();  
  noStroke();  
  smooth();  
}  
  
void draw() {  
  if (video.available()) {  
    video.read();           //acquisition vidéo  
    image(video, 0, 0, width, height); // Draw the webcam video onto the screen  
    video.loadPixels();      //chargement des pixels en mémoire pour sauvegarde  
  }  
}  
  
void keyPressed() {  
  println ("sauve image");  
  video.read();  
  PImage cp = video.get(0, 0, width, height);  
  cp.save("capture.png");  
}
```



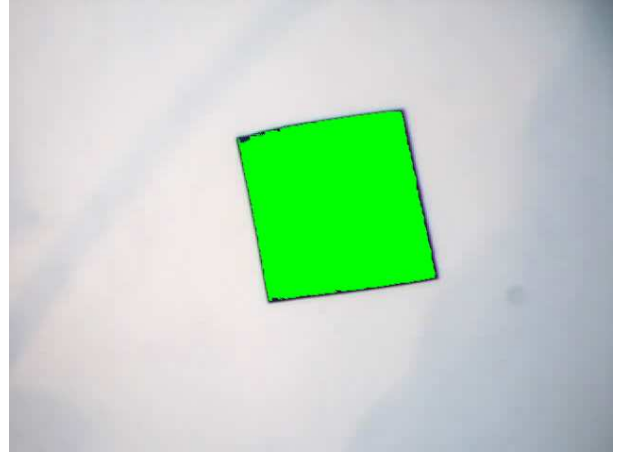
Cliquer sur la fenêtre de prévisualisation, puis appuyer sur la barre d'espace pour sauvegarder l'image en format png.

Reconnaissance de surface temps réel avec d'une Webcam

un adhésif noir sur du papier blanc



La surface noir est maintenant coloriée en vert même si la caméra se déplace.



```
import processing.video.*;
```

```
Capture video;
```

```
void setup() {  
  size(640, 480, P2D); // Change size to 320 x 240 if too slow at 640 x 480  
  video = new Capture(this, width, height);  
  video.start();  
  noStroke();  
  smooth();  
}  
void draw() {  
  if (video.available()) {  
    video.read();  
    image(video, 0, 0, width, height); // Draw the webcam video onto the screen  
    int posX = 0; // X-coordinate of the brightest video pixel  
    int posY = 0; // Y-coordinate of the brightest video pixel  
    float lum = 0; // Brightness of the brightest video pixel  
    video.loadPixels();  
    int index = 0;  
    for (int y = 0; y < video.height; y++) {  
      for (int x = 0; x < video.width; x++) {  
        lum = brightness(video.pixels[index]);  
        if (lum < 50) {  
          posX = x;  
          posY = y;  
          stroke(0, 255, 0);  
          point(x, y);  
        }  
        index++;  
      }  
    }  
  }  
}  
void keyPressed() {  
  println("sauve image");  
  save("image.png");  
}
```

Décoder un QR code avec Processing

Descriptif : http://fr.wikipedia.org/wiki/Code_QR

Comment créer un QR code ? <http://qrcode.fr/encoder>

Message :

bonjour

caractères : 7 (250 max. conseillé)

Paramètres avancés

redondance ECC :

☐ L (7% de récupération de données)

☒ M (15% de récupération de données) par défaut

☐ Q (25% de récupération de données)

☐ H (30% de récupération de données)

type de fichier :

☐ JPEG

☒ PNG (par défaut)

taille du QR : auto (auto par défaut)

remettre à zéro créer le QR_code



Votre QR Code est prêt!

Propriétés de l'image à sauvegarder au format png.



Vérifier le décodage du QR code en ligne : <http://zxing.org/w/decode.aspx>



ZXing Decoder Online

Decode a 1D or 2D barcode from an image on the web. Supported formats include:

- UPC-A and UPC-E
- EAN-8 and EAN-13
- Code 39
- Code 93
- Code 128
- ITF
- Codabar
- RSS-14 (all variants)
- RSS Expanded (most variants)
- QR Code
- Data Matrix
- Aztec ('beta' quality)
- PDF 417 ('alpha' quality)

Enter an image
URL:

Envoyer

Or upload a file
(<2MB):

C:\Documents and Settings\... Parcourir...

Envoyer



Decode Succeeded

| | |
|--------------------|---|
| Raw text | bonjour |
| Raw bytes | 40 76 26 f6 e6 a6 f7 57 20 ec 11 ec 11 ec 11 ec 20 ec 11 ec 11 ec 11 ec |
| Barcode format | QR_CODE |
| Parsed Result Type | TEXT |
| Parsed Result | bonjour |

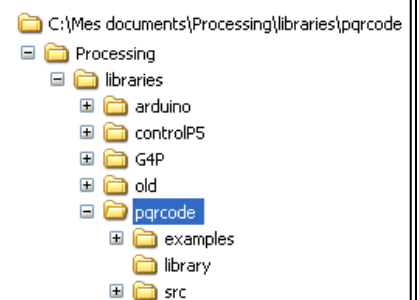
Décodage de l'image avec processing :

Installation de la librairie de décodage :

<http://www.shiffman.net/p5/pqrcode/>

<http://www.shiffman.net/p5/libraries/pqrcode/pqrcode.zip>

copier le répertoire pqrcode dans : Mes documents/processing/libraries



Enfin le programme de décodage du QR code avec Processing

```
import pqrcode.*;

Decoder decoder;
PImage img;
PFont font;

void setup() {
  size(640, 480);
  decoder = new Decoder(this);
  img = loadImage("qrcode1.png");      // ne pas oublier dans Mes documents\Processing\testqr\data
  font = loadFont("AndaleMono-36.vlw"); //les fichiers images png et de police de caractères
  textFont(font);
}

void draw() {
  image(img, 0, 0);
}

void keyPressed() {      //taper sur une touche pour lancer le décodage
  decoder.decodeImage(img);
  String statusMsg = decoder.getDecodedString();
  text(statusMsg, 50, 300);
  println(statusMsg);
}
```

Résultat affiché

