

**Fisica, tecnologia, scienze applicate:  
*uLAB2, un laboratorio portatile basato su arduino e due  
piccoli robot, LEMU e MOMOLAB, vanno a scuola***

processing per esempio

Daniele Grosso<sup>(1)</sup>

1. Dipartimento di Fisica (Università di Genova)



Scuola Estiva Genova AIF-DIFI  
Piano Lauree Scientifiche, modulo di formazione insegnanti in laboratorio



# Processing

Processing is an open source programming language and environment for people who want to create images, animations, and interactions.

- free to download and open source
- Interactive programs using 2D, 3D or PDF output
- OpenGL integration for accelerated 3D
- For GNU/Linux, Mac OS X, and Windows
- Projects run online or as double-clickable applications
- Over 100 libraries extend the software  
sound, video, computer vision, and more...
- Well documented  
(*processing.org*)

Per info e download: <http://processing.org/>

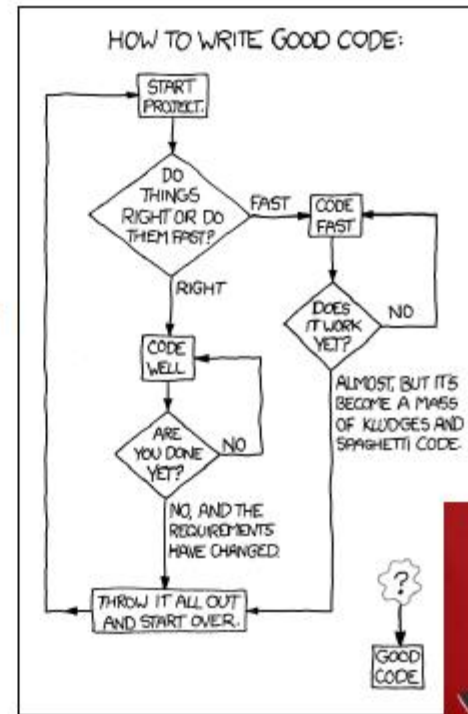
Per l'installazione e' sufficiente decomprimere il programma in una directory

# Codice processing

- **Basic**  
Sequenza di comandi
- **Intermediate**  
Programmazione procedurale  

```
void setup() { ... }  
void draw() { ... }
```
- **Complex**  
Programmazione Orientata agli Oggetti  

```
void setup() { ... }  
void draw() { ... }  
class Puppet {  
  color colore;  
  Puppet(color c_) {  
    colore = c_;  
  }  
}
```



# Esempio conditionals1

- Dimensioni finestra → `size(200, 200);`
  - Colore sfondo → `background(0);`
  - **Struttura di if**
  - Intervalli regolari
  - Cambiare il colore
  - Tracciare linee
- ```
for(int i=10; i<width; i+=10) {  
    // If 'i' divides by 20 with no remainder draw the  
    first line  
    // else draw the second line  
    if(i%20 == 0) {  
        stroke(153);  
        line(i, 40, i, height/2);  
    } else {  
        stroke(102);  
        line(i, 20, i, 180);  
    }  
}
```
-

# Esempio conditionals2

- Dimensioni finestra
  - Colore sfondo
  - Intervalli regolari
  - Cambiare il colore
  - Tracciare linee
  - **else if**
- ```
size(200, 200);
background(0);

for(int i=2; i<width-2; i+=2) {
    // If 'i' divides by 20 with no remainder
    // draw the first line else draw the second line
    if(i%20 == 0) {
        stroke(255);
        line(i, 40, i, height/2);
    } else if (i%10 == 0) {
        stroke(153);
        line(i, 20, i, 180);
    } else {
        stroke(102);
        line(i, height/2, i, height-40);
    }
}
```
-

# Esempio

## EmbeddedIteration

- Disattiva il disegno del bordo
- Cicli for annidati
- Rettangoli e quadrati

```
float box_size = 11;
float box_space = 12;
int margin = 7;

size(200, 200);
background(0);
noStroke();
// Draw gray boxes
for (int i = margin; i < height-margin; i += box_space){
  if(box_size > 0){
    for(int j = margin; j < width-margin; j += box_space){
      fill(255-box_size*10);
      rect(j, i, box_size, box_size);
    }
    box_size = box_size - 0.6;
  }
}
```

# Esempio

## LogicalOperators

- `&&` = AND
- `||` = OR
- `!` = NOT
- Tracciare un punto

```
size(200, 200);
background(126);

boolean op = false;




for(int i=5; i<=195; i+=5) {
  // Logical AND
  stroke(0);
  if((i > 35) && (i < 100)) {
    line(5, i, 95, i);
    op = false;
  }

  // Logical OR
  stroke(76);
  if((i <= 35) || (i >= 100)) {
    line(105, i, 195, i);
    op = true;
  }

  // Testing if a boolean value is "true"
  // The expression "if(op)" is equivalent to "if(op == true)"
  if(op) {
    stroke(0);
    point(width/2, i);
  }

  // Testing if a boolean value is "false"
  // The expression "if(!op)" is equivalent to "if(op == false)"
  if(!op) {
    stroke(255);
    point(width/4, i);
  }
}
```

# Esempio Array

- Array  `size(200, 200);`  
`float[] coswave = new float[width];`
- Map lineare  
`map(var, frommin, frommax, tomin, tomax)`  `for (int i = 0; i < width; i++) {`  
`float amount = map(i, 0, width, 0, PI);`
- `abs` = valore assoluto  `coswave[i] = abs(cos(amount));`  
`sin, cos` f. trigonometriche  
`}`  
`for (int i = 0; i < width; i++) {`  
`stroke(coswave[i]*255);`  
`line(i, 0, i, height/3);`  
`}`  
`for (int i = 0; i < width; i++) {`  
`stroke(coswave[i]*255 / 4);`  
`line(i, height/3, i, height/3*2);`  
`}`  
`for (int i = 0; i < width; i++) {`  
`stroke(255 - coswave[i]*255);`  
`line(i, height/3*2, i, height);`  
`}`



# Esempio simpleRead (lato arduino)

- Imposta il pin su input
- Imposta la velocità  
trasmissione dei dati
- Trasmette **1 byte**  
attraverso la seriale

```
int switchPin = 4;           // Switch connected to
pin 4

void setup() {
  pinMode(switchPin, INPUT); // Set pin 0 as an
input
  Serial.begin(9600);
}

void loop() {
  if (digitalRead(switchPin) == HIGH) {
    Serial.print(1, BYTE);    // send 1 to Processing
  } else {                   // If the switch is not ON,
    Serial.print(0, BYTE);    // send 0 to Processing
  }
  delay(100);                // Wait 100 milliseconds
}
```

# Esempio simpleRead

- Attivare la libreria serial
- Creare un oggetto serial
- Selezionare porta USB da una lista di disponibili
- Impostare la velocità
- **Estrarre un dato dal buffer** dei trasmessi sulla porta

```
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
  }
  background(255);             // Set background to white
  if (val == 0) {               // If the serial value is 0,
    fill(0);                    // set fill to black
  }
  else {                        // If the serial value is not 0,
    fill(204);                  // set fill to light gray
  }
  rect(50, 50, 100, 100);
}
```

setup

draw

# Esempio simpleWrite (lato arduino)

- Imposta il pin su output
- Imposta la velocità trasmissione dei dati
- **Controlla se il dato è disponibile** per la lettura
- **legge 1 byte trasmesso** attraverso la seriale

```
char val; // Data received from the serial port  
int ledPin = 4; // Set the pin to digital I/O 4
```

```
void setup() {  
  pinMode(ledPin, OUTPUT); // Set pin as OUTPUT  
  Serial.begin(9600); // Start serial communication at 9600  
}
```

```
void loop() {  
  if (Serial.available()) { // If data is available to read,  
    val = Serial.read(); // read it and store it in val  
  }  
  if (val == 'H') { // If H was received  
    digitalWrite(ledPin, HIGH); // turn the LED on  
  } else {  
    digitalWrite(ledPin, LOW); // Otherwise turn it OFF  
  }  
  delay(100); // Wait 100 milliseconds for next reading  
}
```

# Esempio simpleWrite

- Attivare la libreria serial
- Dichiarare un oggetto serial
- Selezionare porta USB da una lista di disponibili
- Creare un oggetto serial  
Impostare la velocità
- Cambiare colore (riempimento)
- **Trasmettere un dato**
- Definire una funzione
- Leggere le coordinate del mouse

```
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() {
  background(255);
  if (mouseOverRect() == true) { // If mouse is over
    square,
    fill(204);                // change color and
    myPort.write('H');        // send an H to indicate
  }
  else {                      // If mouse is not over square,
    fill(0);                  // change color and
    myPort.write('L');        // send an L otherwise
  }
  rect(50, 50, 100, 100);    // Draw a square
}

boolean mouseOverRect() { // Test if mouse is over square
  return ((mouseX >= 50) && (mouseX <= 150) &&
    (mouseY >= 50) && (mouseY <= 150));
}
```

# Esempio serialDuplex

- Attivare la libreria serial
- Dichiarare un oggetto serial
- Visualizza un  
**elenco delle porte disponibili**  
Selezionare porta USB  
da una lista di disponibili
- Creare un oggetto serial
- **Ricevere un dato**
- **Trasmettere un dato**

```
import processing.serial.*;

Serial myPort;    // The serial port
int whichKey = -1; // Variable to hold keystroke values
int inByte = -1;  // Incoming serial data

void setup() {
  size(400, 300);
  // create a font with the third font available to the system:
  PFont myFont = createFont(PFont.list()[2], 14);
  textFont(myFont);

  // List all the available serial ports:
  println(Serial.list());

  String portName = Serial.list()[0];
  myPort = new Serial(this, portName, 9600);
}

void draw() {
  background(0);
  text("Last Received: " + inByte, 10, 130);
  text("Last Sent: " + whichKey, 10, 100);
}

void serialEvent(Serial myPort) {
  inByte = myPort.read();
}

void keyPressed() {
  // Send the keystroke out:
  myPort.write(key);
  whichKey = key;
}
```

# Esempio saveFile

- Disattiva il riempimento
- Definisce una forma tramite i suoi vertici
- Aggiunge un elemento in coda ad un array
- Creare un file leggibile da altri programmi
- Salva i dati come stringhe di testo

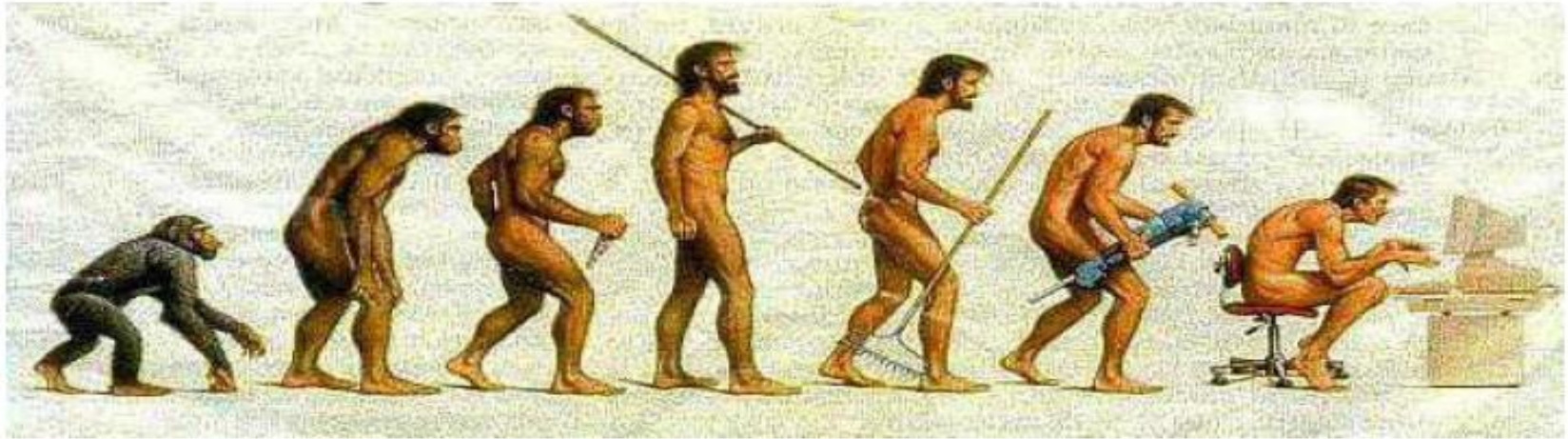
```
int[] x = new int[0];
int[] y = new int[0];
void setup() {
  size(200, 200);
}
void draw() {
  background(204);
  stroke(0);
  noFill();
  beginShape();
  for (int i = 0; i < x.length; i++) {
    vertex(x[i], y[i]);
  }
  endShape();
  // Show the next segment to be added
  if (x.length >= 1) {
    stroke(255);
    line(mouseX, mouseY, x[x.length-1], y[x.length-1]);
  }
}
void mousePressed() { // Click to add a line segment
  x = append(x, mouseX);
  y = append(y, mouseY);
}
void keyPressed() { // Press a key to save the data
  String[] lines = new String[x.length];
  for (int i = 0; i < x.length; i++) {
    lines[i] = x[i] + "\t" + y[i];
  }
  saveStrings("lines.txt", lines);
  exit(); // Stop the program
}
```

# Esempio loadFile

- Dichiarare un array di stringhe
- Imposta il numero di frame/s
- Legge un file
- Estrae i singoli dati usando **tab come separatore**

```
String[] lines;  
int index = 0;  
  
void setup() {  
  size(200, 200);  
  background(0);  
  stroke(255);  
  frameRate(12);  
  lines = loadStrings("positions.txt");  
}  
void draw() {  
  if (index < lines.length) {  
    String[] pieces = split(lines[index], '\t');  
    if (pieces.length == 2) {  
      int x = int(pieces[0]) * 2;  
      int y = int(pieces[1]) * 2;  
      point(x, y);  
    }  
    index = index + 1;  
  }  
}
```

L'evoluzione del software ha portato a Processing  
semplice, flessibile e potente, multiplatforma...



Forse qualcosa è andato storto... ?

... poteva andare anche peggio !!!

ARRGH! MY MAP OF LISTS OF MAPS  
TO STRINGS IS TOO HARD TO  
ITERATE THROUGH! I'LL JUST ASSIGN  
EVERYTHING A NUMBER AND USE  
A \*!\*@ ARRAY

