Agenda

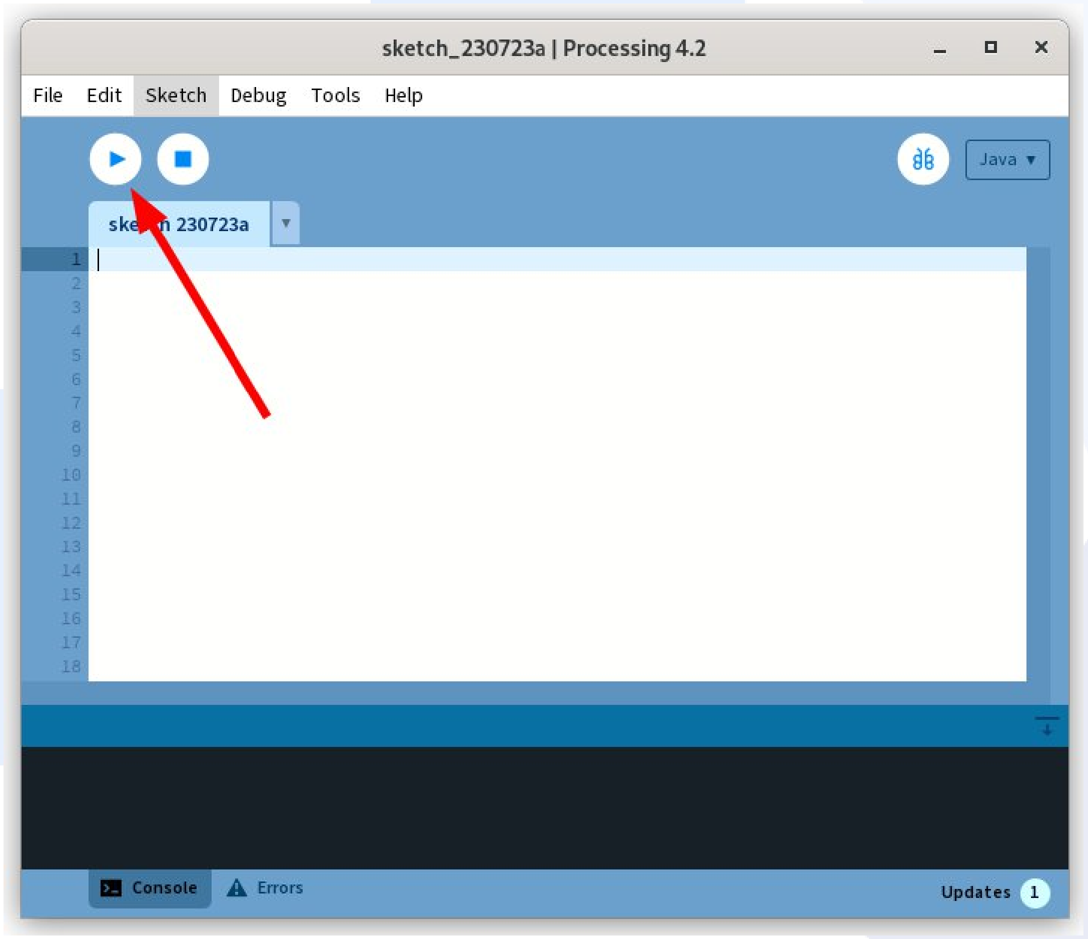
Day 1	Processing, the coding side of Arduino: Functions and Variables
	Processing, the coding side of Arduino: Flow control
Day 2	Arduino: Serial communication between board and PC, I/O ports
	Arduino: Digital Sensors and Digital Actuators
Day 3	Arduino: Analog Sensors
	Arduino: Analog Actuators
Day 4	Arduino: Protocol communication with sensors
	Arduino: Actuator control based on sensor feedback
Day 5	LoRa: Point to point communication
	LoRaWAN: Gateway and Server

Processing
the coding side of Arduino

IDE

Processing IDE

- Setup
- IDE Definition
- Just click RUN
- What is that?



So, what is Processing?

- CAD?
- Let’s write something
- Look at the colors: syntax!
- IDE : Text editor + Compiler + Debugger
- What’s the result?
  - A software application executable by anyone
  - Available exportation for Windows – MacOS ‒ Linux – Android

IDE

- Introduction to the environment
  - An IDE is a text editor? Also. But not only.
  - It helps users with syntax and algorithms, error detection, warning and debug
  - Some quick examples:

```
background(<red>, <green>, <blue>);
background(<grey>);
//background(<grey>);
/*
background(<grey>);
*/
```

  - This was the very first function

Processing
the coding side of Arduino

Functions

Functions pt.1

- We distinguish them by color in the IDE
- They perform a job or return data
- They may require parameters to be executed

Functions pt.2

- The order is relevant
  - If multiple functions have the same “target”, only the latest is relevant

- Just a couple of examples:
  - size(<width>, <height>);
  - fullScreen();

- The same function could require different amount of parameters
  - background(<gray>)
  - background(<red>,<green>,<blue>)

Functions pt.3

Is there a list of functions? Let’s have a look at *Reference*.

Reference

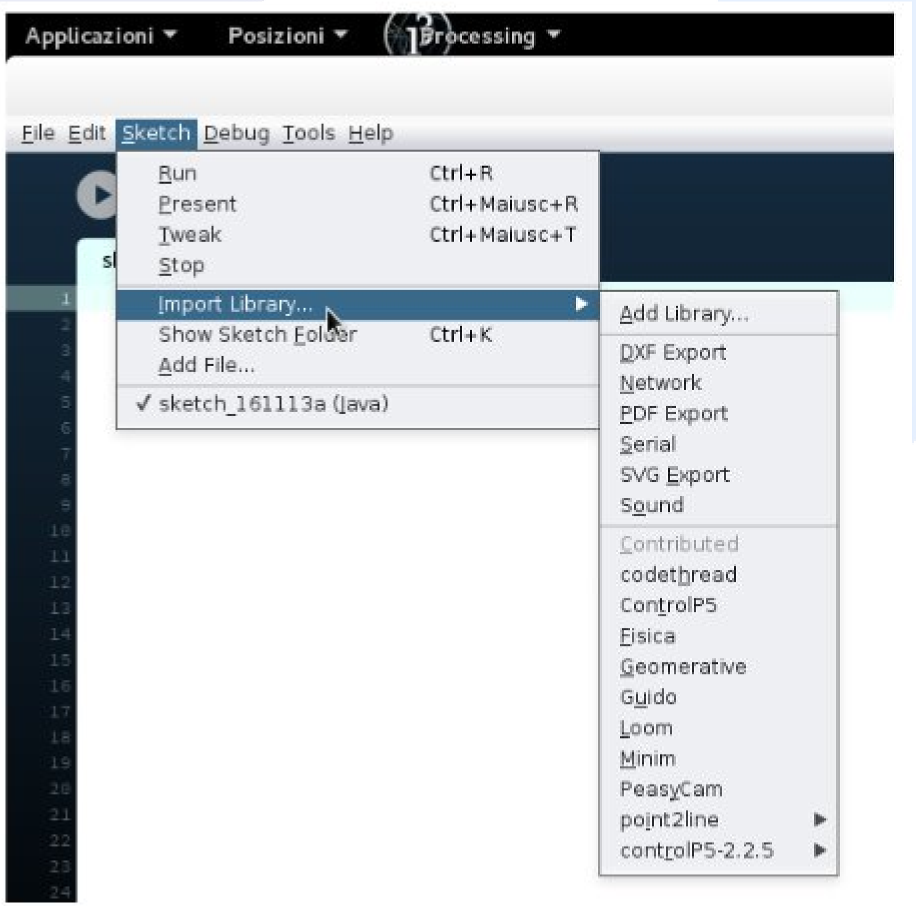
https://processing.org/reference/

- Structure
- **Environment**
- Data
- Control
- Shape
- **Input**
- **Output**
- Transform
- Lights
- Camera
- Color
- Image
- Rendering
- Typography
- **Math**
- Constants

Core Libraries https://processing.org/reference/libraries

- DXF export
- Hardware I/O
- Network
- PDF Export
- Serial
- Sound
- SVG Export
- Video

Contributed libraries
https://processing.org/reference/libraries



Sum up

- There are functions that perform many different tasks
- Some functions are already available by default
- Some are available within libraries, organized by topic, that user has to import into program
- Lots of other functions are available on the web, easily installable
- Many functions → Many features → Many possibilities

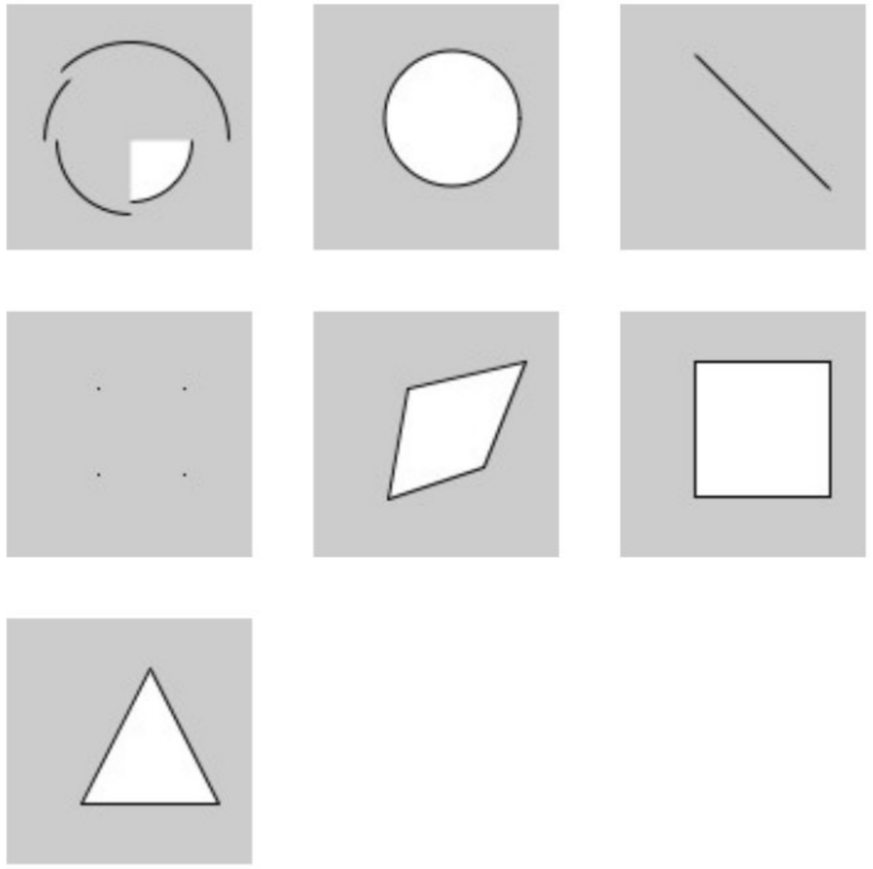
Let’s get started

- Have a look at graphical functions

- Our target is not to create a layout, but understanding the basics of coding

2D Primitives

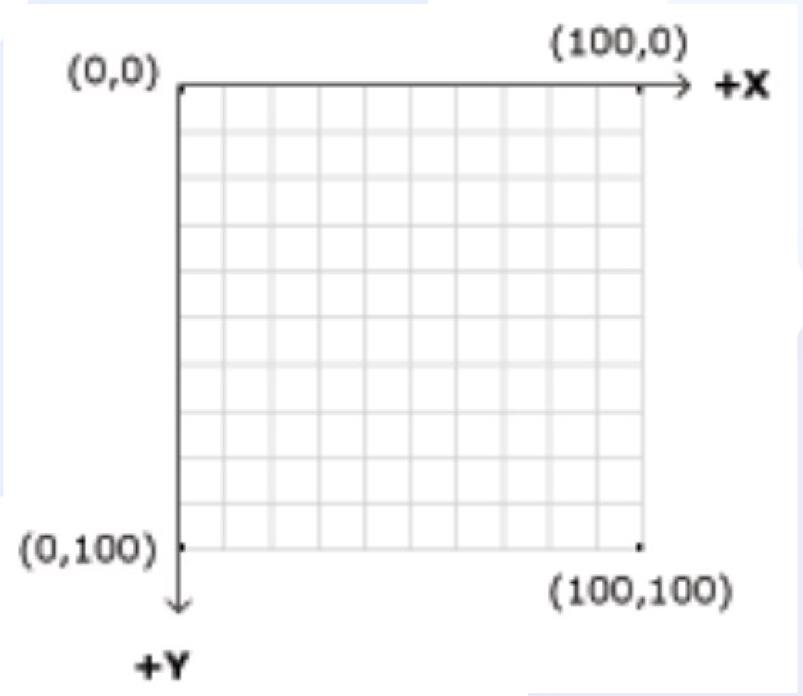
- arc()
- ellipse()
- line()
- point()
- quad()
- rect()
- triangle()
- beginShape()
  vertex()
  endShape()



point(x,y);

- size(400,400);
  point(100,100);

- Axis directions
- Size of lines and points
- By default strokeWeight(1);
- strokeWeight(10);
- Save project
- Auto-format: Ctrl+T



line(x1, y1, x2, y2);

- size(x,y);
  line(0, 0, 300,300);
- Increase strokeWeight
- Size of lines and points
- Change color
- stroke(red, green, blue);
- stroke(gray);
- Tip: Auto-format with *Ctrl+t*

rect(x, y, width, height);

- size(400,400);
- rect(50,100,200,50);

- stroke(R,G,B);
- rect(x,y,b,h);

- noStroke();
- rect(x,y,b,h);

ellipse(x, y, width, height);

- size(400,400);
- rect(50,100,200,50);

- stroke(R,G,B);
- rect(x,y,b,h);

- noStroke();
- rect(x,y,b,h);

Test#1

Draw a picture with a circle in the middle.

Draw a clearly visible point in the middle of the screen.

Also insert the diagonals of the view: don’t overlap the circumference!

Do not use fullScreen() but size().

Size parameters as desired, but less than (300,300).

Tip:

- size() with even and round numbers!
- You can use math into parameter field
  Ex: point(100 / 5 , 100 * 2);

Time:

15 minutes

Processing
the coding side of Arduino

Variables

System variables

Let’s imagine if we change the picture, making it bigger (x2).

Is there any smarter way than change every number? YES.
We have to introduce two system variables: width and height.

Test#2

Re-make test#1 in order to make it work correctly with any size parameters.

Time:

10 minutes

Test#3

Draw an horizontal line and place 3 **visible** points upon it.

Don’t use system variables.

Time:

15 minutes

User variables pt.1

What if we move the line?

What happens to the points?

Is there any way to create a relationship between line and points?

YES.

We have to introduce user variables.

```
int yPosition = 30;
```

User variables pt.2

Variable *definition*	`int position;`
Variable *assignment*	`position = 75;`
Variable *definition* and *assignment*	`int position = 75;`

Test#4

Draw an horizontal line and place 3 **visible** points upon it.

Don’t use system variables, but use a user variable to make the point always be upon the line if moved.

Time:

10 minutes

Test#5

Draw an horizontal line and place 4 points with same distance from each other.

Remember: you can use math inside the parameter field!

Time:

15 minutes

User variables pt.3

- boolean student_present;
  student_present = false;
- char student_mark;
  student_mark= 'b'; // single quote
- float student_height;
  student_height= 1.84;
- int student_age;
  student_age= 22;
- String student_name;
  student_name = "Tom"; // double quote

User variables pt.4

- How to read the content of a variable?
- Write content on screen:

```
text( variable , x , y );
```

- Write content on console:

```
print( variable );
```

Not only variables:

```
print( “Variable value is:” );
println( “example” );
```

Processing
the coding side of Arduino

Flow control

IF condition pt.1

```
if (check) {
        statement;
      }


if (check) {
            statement_1;
   } else {
            statement_2;
       }
```

IF condition pt.2

Check can only give as result **true** or **false.**

• If (pos_x == 80)	equal to
• If (pos_x != 80)	different from
• If (pos_x > 50)	bigger than
• If (pos_x >= 50)	bigger than or equal to

Check can involve more than a variable:

- If (pos_x == pos_y)

IF condition pt.3

- Check can be a logical combination of several simple checks

```
If ( pos_x > 50  && pos_x < 80)
```

- Can include all the typical elements of boolean math

&&	AND
||	OR
!	NOT

Test#6

- Draw an horizontal line and a point
  - fill the background with green if the point is above the line
  - fill the background with red if the point is under the line

Tip: create user variables for the y coordinates

Time 15 minutes

Test#7

- The y position of the point is no more assigned by the user but assigned from a function:

```
random(max_value);
```

Time 5 minutes

Functions pt.4

random() is the first function that returns data instead of executing a job (change color, draw pictures, etc...).

Let’s look at Reference and find other functions that return data

Recap

Where do data come from?

• System variable	width, height,
• User variable	int age, float distance
• System function	random(), sqrt()
• User function	

“Butterfly” programs

All the programs written lived just for a very short amount of time.

Let’s draw on the screen what a new function returns: day()

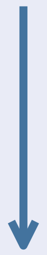
```
size(200,100);
int value;
value = day();
text( value, 10, 10);
```

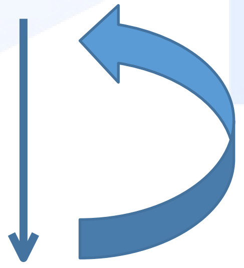
Now change the time function, step by step.

First complete program pt.1

```
void setup(){
        instruction#1;
        instruction#2;
        instruction#...;
        instruction#n;
                }

void draw(){
        instruction#1;
        instruction#2;
        instruction#...;
        instruction#n;
        }
```





First complete program pt.2

Everything we have done so far was as if it had been written in setup().

```
void setup(){
        instruction#1;
        instruction#2;
        instruction#...;
        instruction#n;
        }
```

First complete program pt.3

So, what does it happen in draw()?

Let’s try.

```
void draw(){
                    text( second(), x, y);
          }
```

And also

```
void draw(){
                    ellipse( second(), y, 10, 10);
          }
```

What is happening?

background(255);

More global variables

- mouseX, mouseY
- mousePressed
- key
- keyPressed

Test#8

- Draw an horizontal line and a point in mouse position
- fill the background with green if the point is above the line
- fill the background with red if the point is under the line

Time 10 minutes

Counter

Create a variable that increase its value every time that mouse is clicked.

millis()

Create an ellipse that changes randomly color **every** 10 seconds

Processing
the coding side of Arduino

End of Day 1