

PV259

Generative Design Programming

Week 2

Transformations

Marko Řeháček & Megi Kejstová
rehacek@mail.muni.cz

Recap

program structure

```
// runs once when program starts
function setup(){
  createCanvas(800, 600); // width, height in pixels
  // can use windowWidth, windowHeight

  // switch to full screen anytime (also in draw())
  fullscreen(booleann)
}

// run continuously after setup
function draw(){
  // rendering loop
}
```

globals

```
windowWidth / windowHeight // of browser window
width / height // of canvas

mouseX / mouseY
// current horizontal / vertical mouse position
```

non-visual feedback

```
print();
console.log(msg, ...);
```

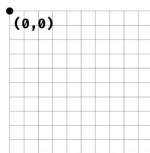
JS

```
let empty_arr = [];
const empty_object = {};
```

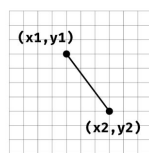
geometry

```
point(x, y)
line(x1, y1, x2, y2)
circle(x, y, radius)
ellipse(x, y, width, height)
square(x, y, side_length)
rect(x, y, width, height)
rectMode(MODE) // CENTER, CORNERS
arc(x, y, width, height, start, stop)
triangle(x1, y1, x2, y2, x3, y3)
```

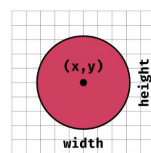
grid system



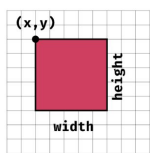
line()



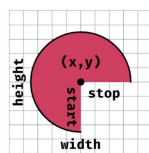
ellipse()



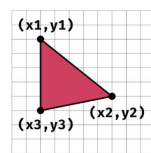
rect()



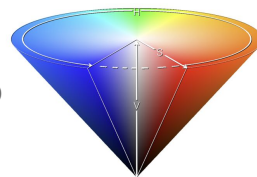
arc()



vertex()



colors



```
background(color)
clear()
```

```
fill(color)
noFill()
```

```
fill(120); //gray: 0-255
fill(100,125,255); //r, g, b: 0-255
fill(255, 0, 0, 50); //r, g, b, alpha
fill('red'); //color string
fill('#ccc'); //3-digit hex
fill('#222222'); //6-digit hex fill
```

```
const mycolor = color(0, 0, 255);
//p5.Color object
```

```
stroke(color)
strokeWeight(weight: number)
noStroke()
```

```
colorMode(MODE) // HSB, RGB
colorMode(MODE, maxValue: number)
// default for HSB
colorMode(HSB, 360, 100, 100, 1.0)
// change default for RGB
colorMode(RGB, 100)
// HSB is preferred
// RGB used for specific calculations
```

<https://coolors.co/>

Map()

Convert (/transform/map) number from one range to another.

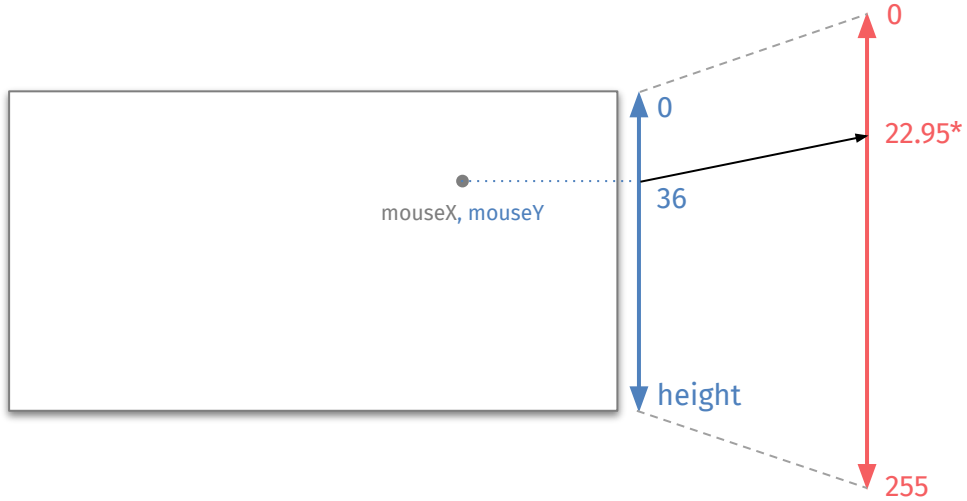
Example

Use **height of the mouse** to control the **color of a shape**.

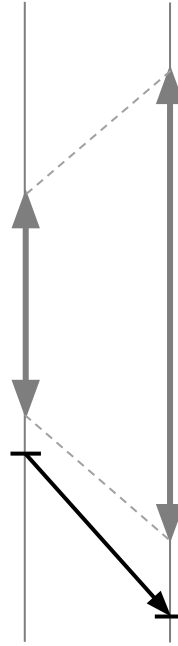


```
new_color = map( mouseY, 0, height, 0, 255 );
```

Map() cont.

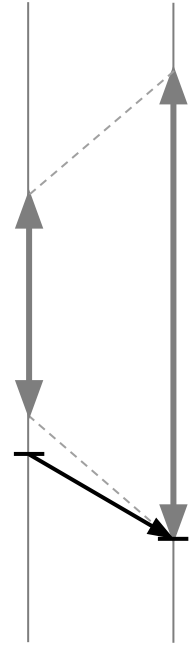


*where height == 400



withinBounds: **false**

(Default)



withinBounds: **true**

Map() recap.

Take any number and scale it to a new number that is more useful.

```
map(value, start1, stop1, start2, stop2,[withinBounds])
```

value:	number; the number to be converted
start1:	number; lower bound of the number's current range
stop1:	number; upper bound of the number's current range
start2:	number; lower bound of the number's target range
stop2:	number; upper bound of the number's target range
withinBounds:	boolean; constrain the number to the newly mapped range (Optional)

Example

Use mouse position to control the size or color of a shape.

Set HSB color mode, and map mouseX values in interval 0-width to 0-360 (hue).

```
fill( map( mouseX, 0, width, 0, 360 ) );
```

Simple animation

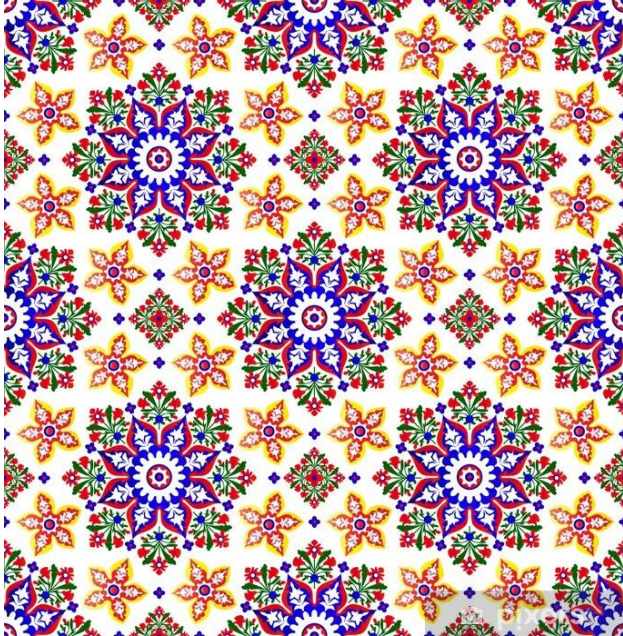
Returns the number of milliseconds since a sketch started running.

```
millis()
```

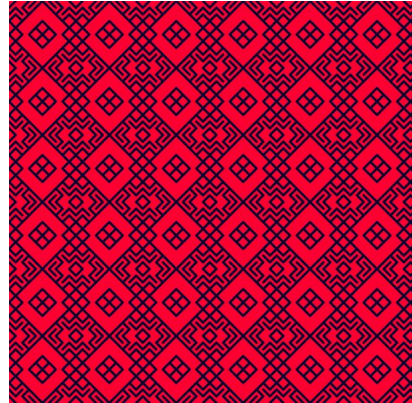
Can also try to combine it with sinus for constrained motion:

```
sin(x)
```

How tf I draw this?



and this??



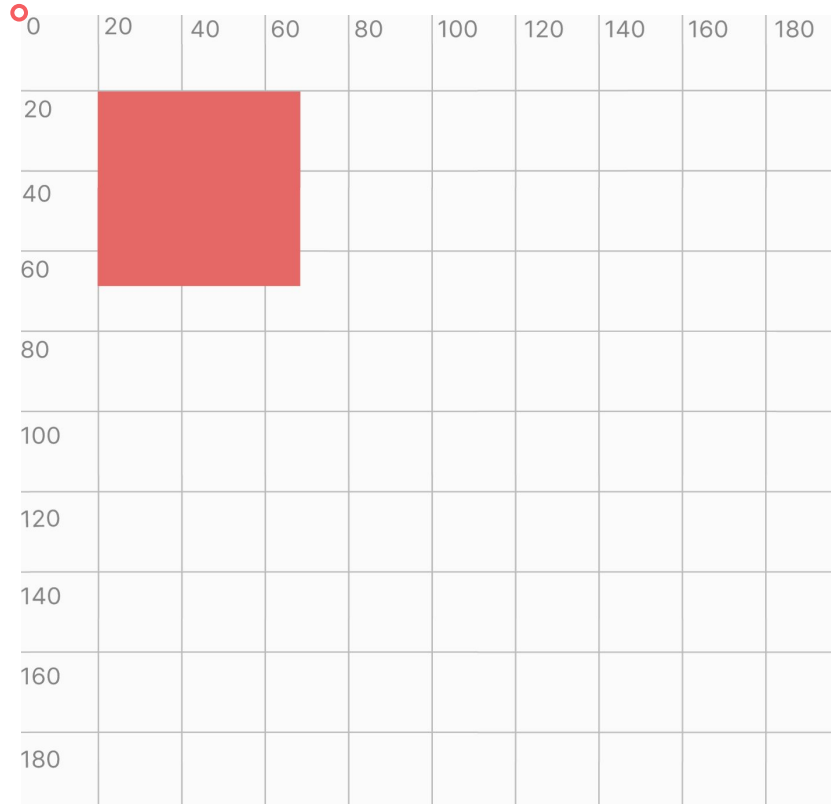
→ we decompose
using functions



Transformations

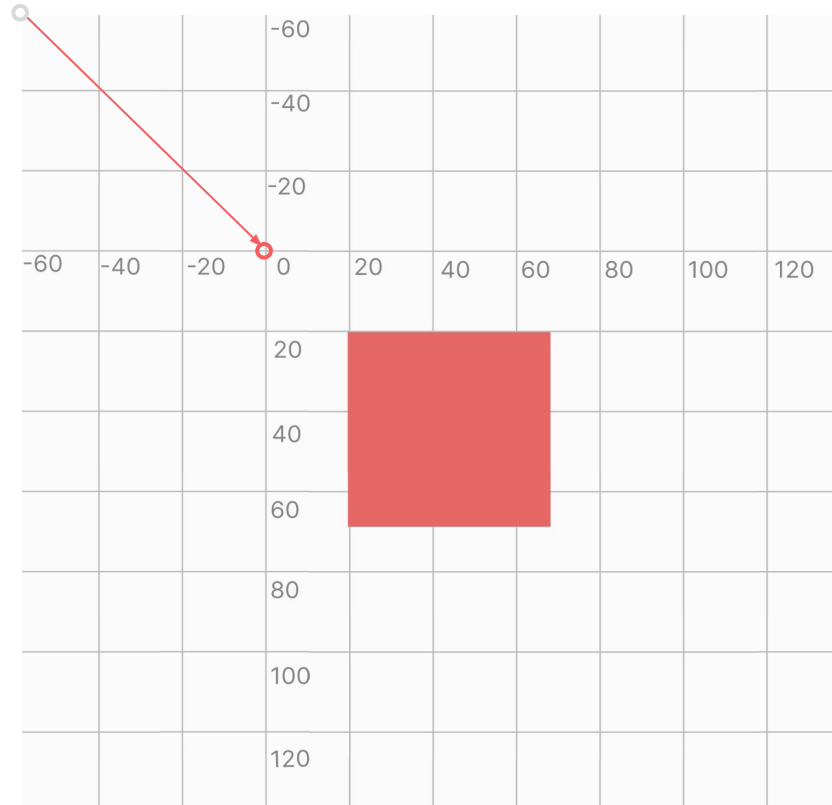
Translate

Move square 60 pixels
down and right?



Translate

Move square 60 pixels
down and right?



Translate

Built in function that shifts origin (point 0,0) of coordinate system

translate(number, number)

It is cumulative:

Resets each draw cycle.

```
translate(10, -10);  
translate(0, 50);
```

is the same as

```
translate(10, 40);
```

Rotate

Rotate coordinate system by an angle

- **always rotates around origin (0,0)**
- expects angle in radians
- positive numbers rotate objects in a clockwise direction
- accumulates as translate()

```
rotate( angle_radians )
```

```
rotate( PI ) // HALF_PI
```

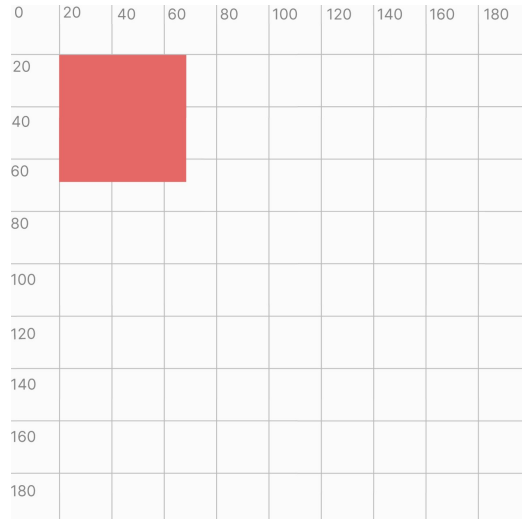
the angle of rotation, specified in radians or degrees, depending on current angleMode

```
angleMode( DEGREES ) // RADIANS, DEGREES
```

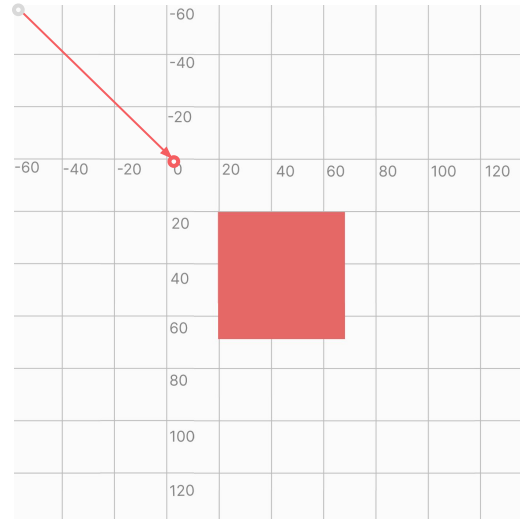
```
rotate( 180 )
```

```
rotate( radians( degrees ) ) // also possibility
```

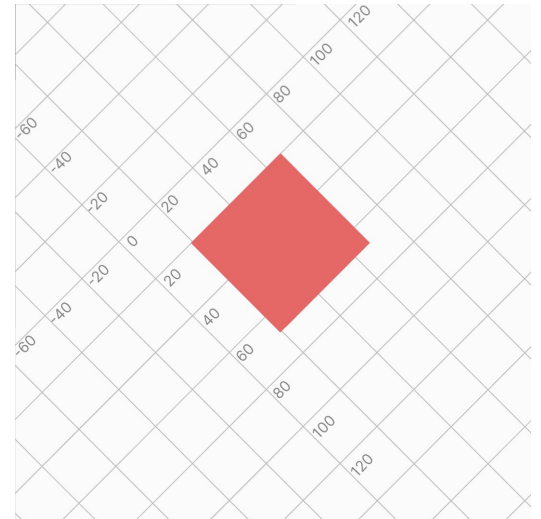
Order matters



Translation (60, 60)



Rotation 90°



#6 Move it

1. Draw a square in the middle of canvas, by drawing it at origin (point 0,0) and moving it towards center using transformation.
2. Make it rotate. Try function **millis()** which returns number of milliseconds from start of sketch.
3. Switch: first rotate then translate, or first translate then rotate.
4. Make it slower, make it faster, make it stronger, make it better.
5. Play with **framerate()**.
6. Try **sin(x)** together with **map()**.

[Finished example](#)

Scale

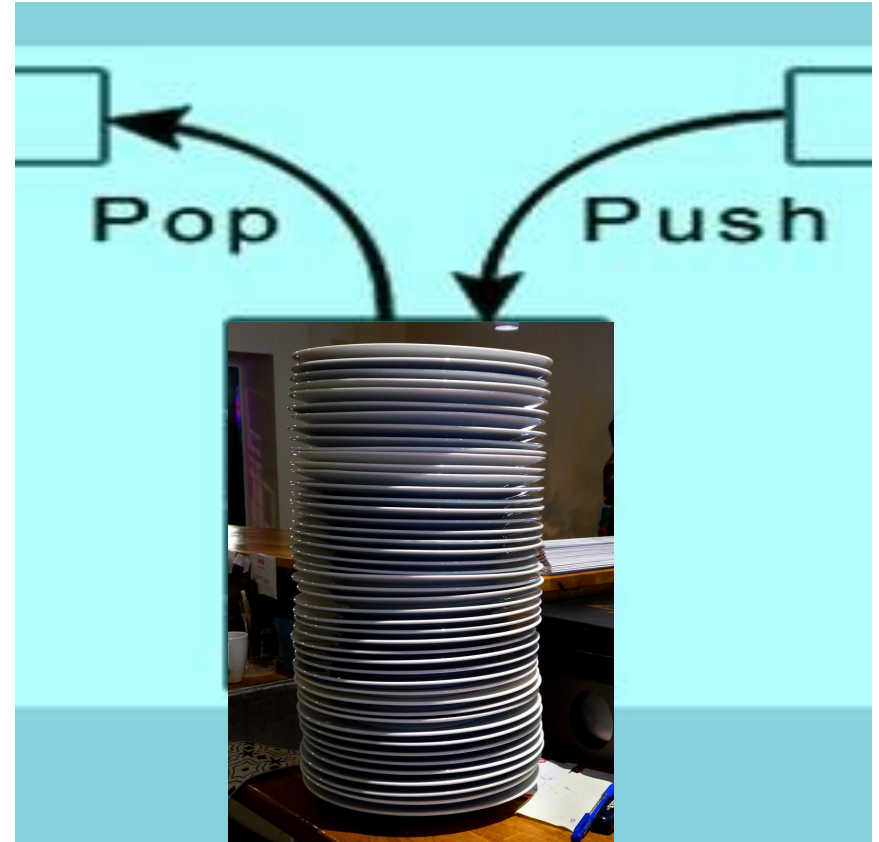
Stretches or shrinks coordinate system

scale(float)

Push & pop

The **push()** function saves the current drawing style settings and transformations, while **pop()** restores these settings.

When a new state is started with **push()**, it builds on the current style and transform information.



#7: Moroccan night

Draw an object consisting of circles, recursing. Use function to draw the object.

Choose it nice color palette (colors), I will use:

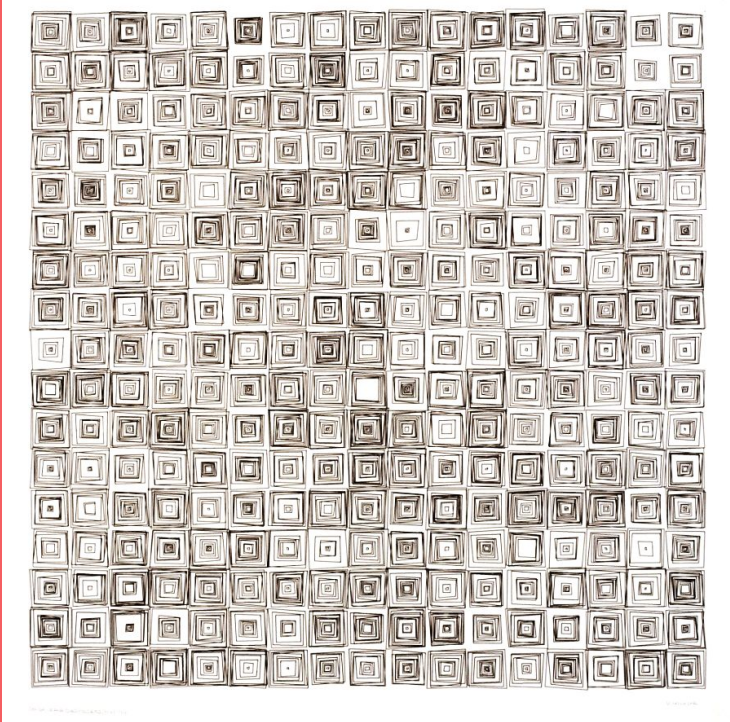
'#924234', '#A64839', '#D37E5C', '#DFB281', '#DEE5CC', '#2F3170'

Position the objects in grid to repeat as pattern. Use transformations for positioning (function should start drawing in origin).

Sprinkle in some stars.

(Des)Ordres

Recreate Vera Molnar's *(Des)Ordres* (1974) using transformations and **recursion**.



Hints

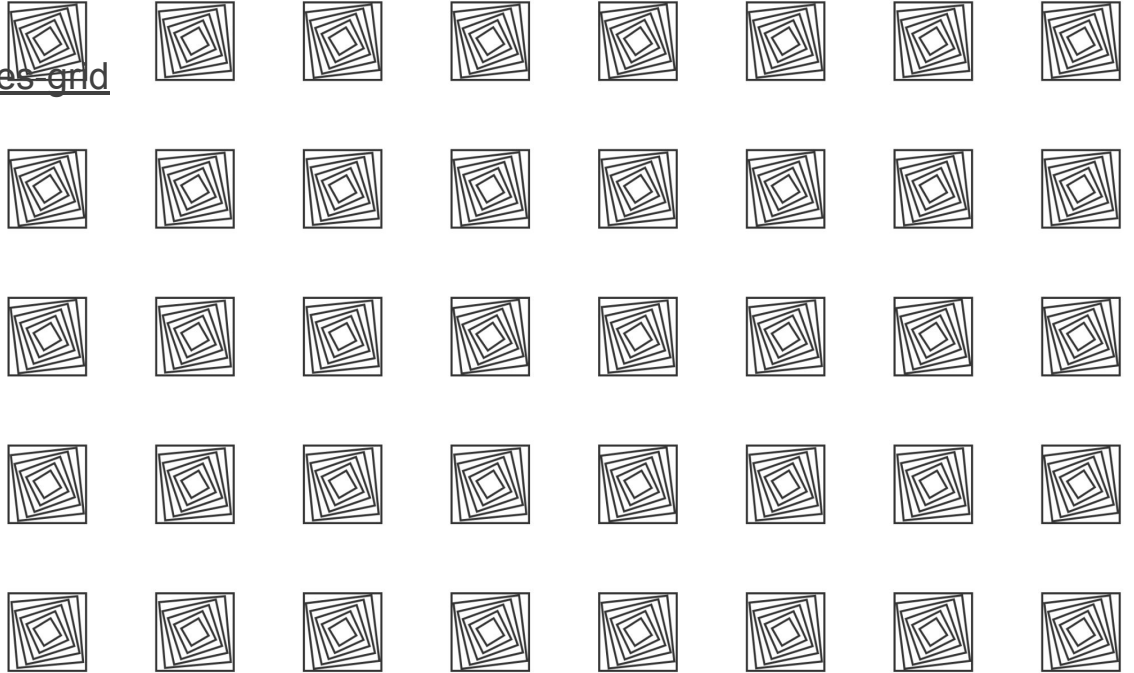
1. create a recursive function to draw a single element from the grid
2. use *for* loops to create a grid, then use transformations to draw the element where desired using the function; do not use `push()`, `pop()` or `translate()` more than once

When done

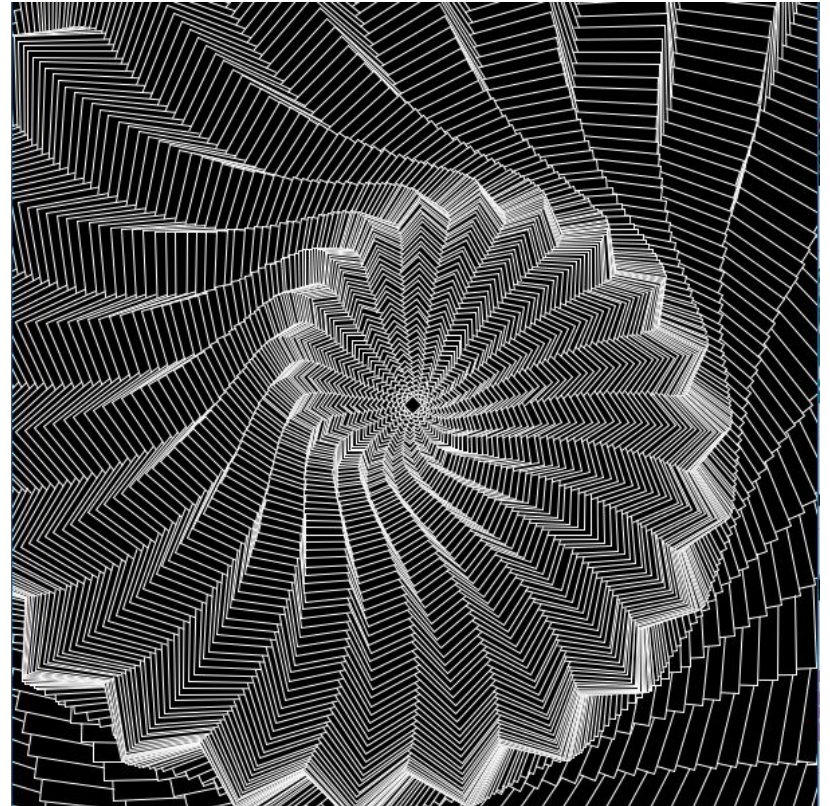
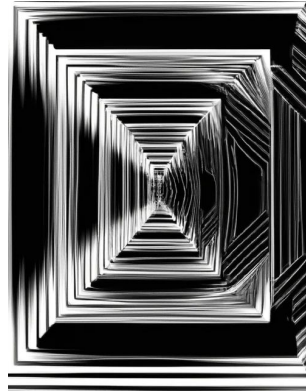
- try different shapes
- experiment with color – handpick a nice color scheme or generate it
- use `random()` for shape positions, rotations, colors, and more
- don't stop here omg!

Interactive example

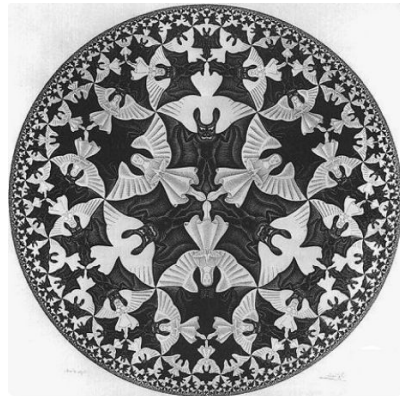
<https://github.com/mrehacek/p5-examples-grid>



Recursion art



Bloom, Holger Lippmann



Sketch as HTML/JS file



```
Sketch Files + < index.html •
index.html ▾
JS sketch.js
style.css
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <script src="https://cdnjs.cloudflare.com/ajax/libs/p5.js/1.10.0/p5.js"></script>
5   <script src="https://cdnjs.cloudflare.com/ajax/libs/p5.js/1.10.0/addons/p5.sound.min.js"></script>
6   <link rel="stylesheet" type="text/css" href="style.css">
7   <meta charset="utf-8" />
8
9 </head>
10 <body>
11 <main>
12 </main>
13 <script src="sketch.js"></script>
14 </body>
15 </html>
```

Using VS Code

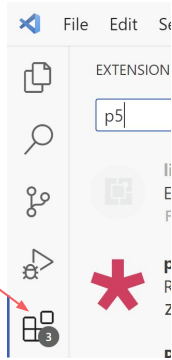
Install extensions:

1. [samplavigne.p5-vscode](#)
2. [zsakowitz.p5-resources](#)

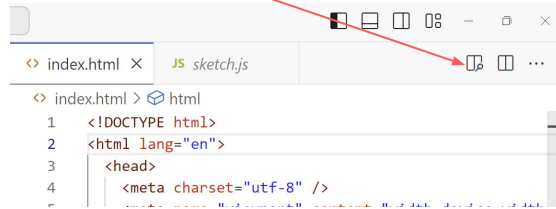
Create project:

Command palette: CTRL+Shift+P

→ search for “p5 create project”



Open preview (must select *index.html*):



Useful extension: Github Copilot (free for education, but need apply)

Using libraries

Look at options: <https://p5js.org/libraries/>.

Let's try <https://github.com/bitcraftlab/p5.gui>!

UI

Add UI elements using p5.gui:

1. download library source code to your project files, add in index.html
2. use



<https://mrehacek.github.io/p5-examples-grid/>

Save to image

```
keyPressed ( ) {  
  if (key === "s") {  
    save();  
  }  
  ...  
}
```

Check **keyCode** for special keys (LEFT_ARROW, BACKSPACE, ESC, ... [see more](#)):
if (keyCode === RETURN) ...

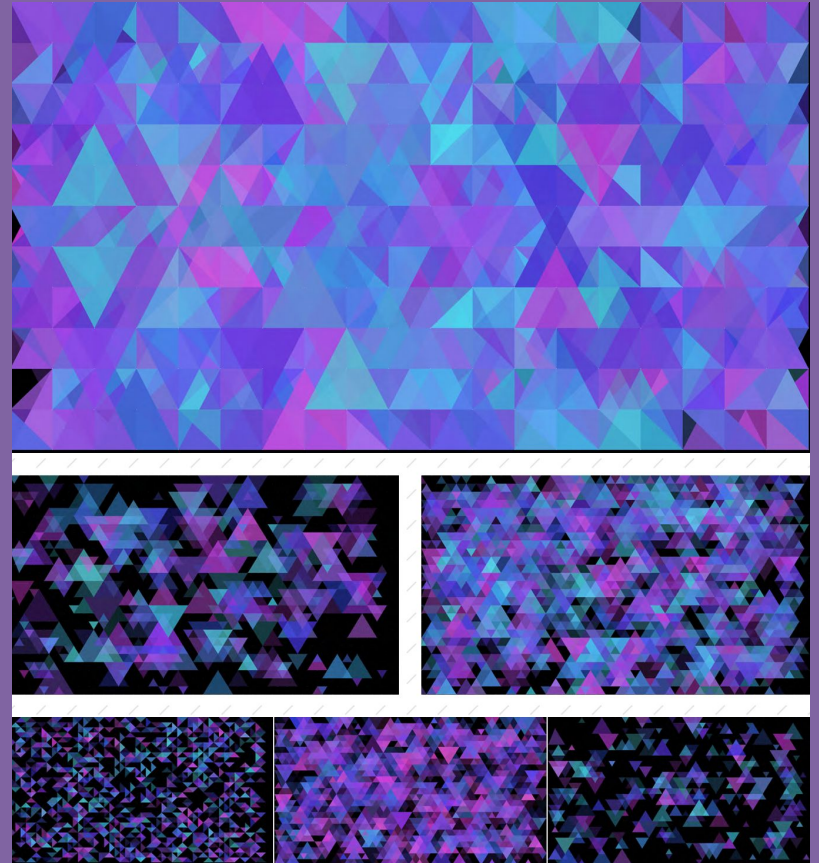
Use **keyTyped()** if you want to distinguish between lower and uppercase letters...

Homework 1: Geometric patterns

Your task is to create a sketch that generates a set of artworks of geometric pattern (for example, for a new desktop background). The emphasis of this project lies on the generative creation, i.e. your sketch is expected to create multiple artworks. This can be achieved by using randomness, input data, mathematical functions or user interaction such as mouse movement. There is no limitation to your ideas. Do not use libraries specialized for patterns.

To export your artwork, use function `save()`. In case you are interested in artwork postprocessing, you can try vector format (SVG).

Technical information regarding the submission and deadline are in interactive syllabus in IS.



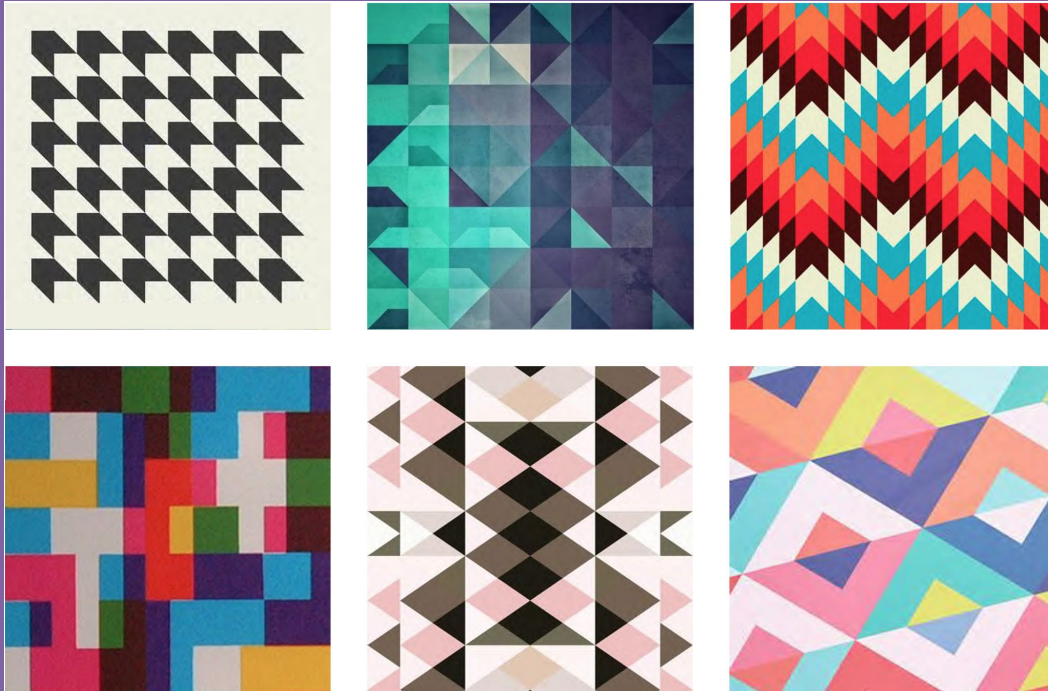
Examples

Tell me, what to do.



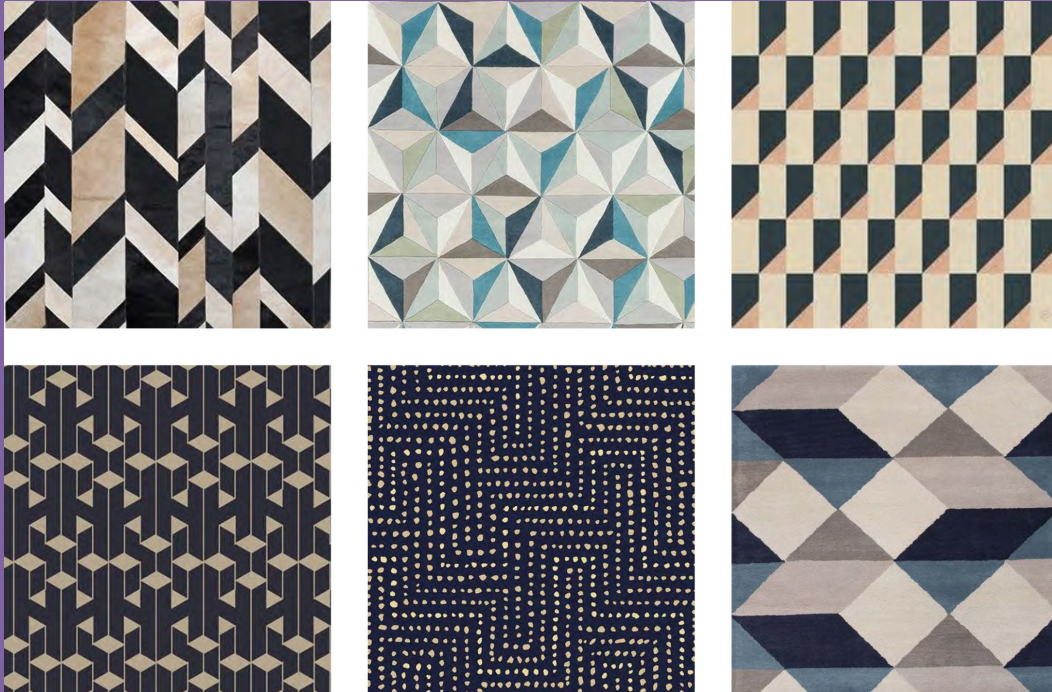
Examples

Give me nice colors.



Examples

You can be a little bit fuzzy about it. I do like randomness.



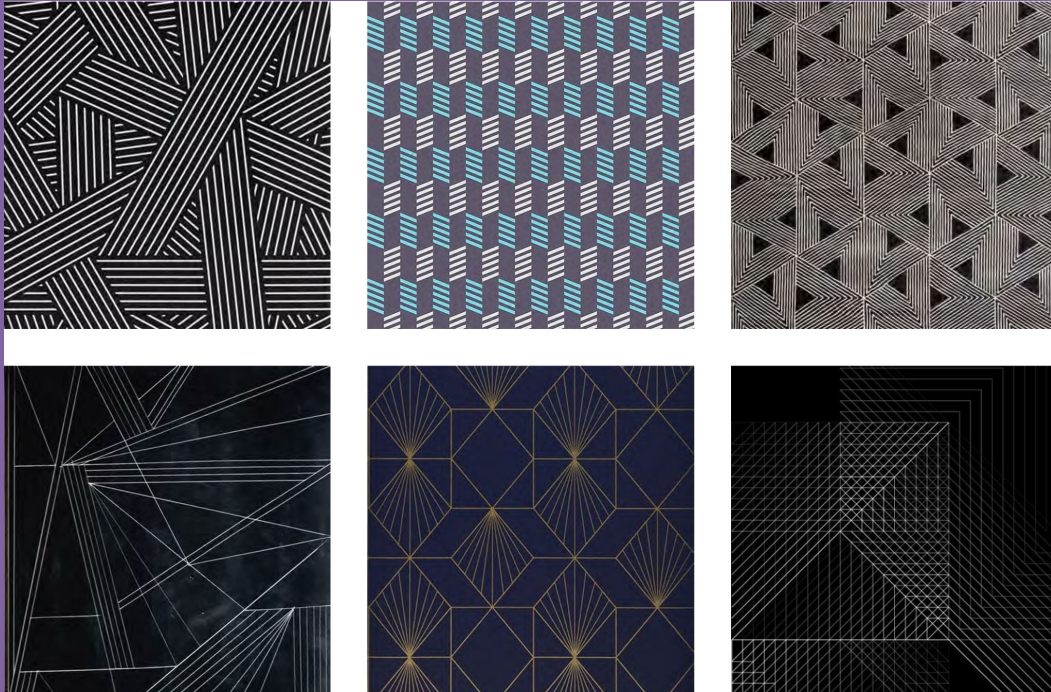
Examples

Or you can try just lines.



Examples

You can do a lot with me. A lot.



Examples

Draw a shape.

Maybe try it multiple times.

In a grid.

Or randomly?

Assign it color, or multiple of them.

Again randomly?

Or parametrically?

You say based on position?

Sure. Also the size sounds cool!

Sure, they can overlap.

Yeah, go along and tweak it.

Maybe redraw it with a new shape?

Or just copy from your classmate.

But you can also just ask anyone to give you a tip.

And now it's you and geometric patterns

PS

We will have a competition for the best homework!!!

The winner gets a large print of his artwork.

Voting will be anonymous and held at the beginning of the following class through discord. You will have 2 votes, and each teacher will have 4.

Have fun creating your sketches!

