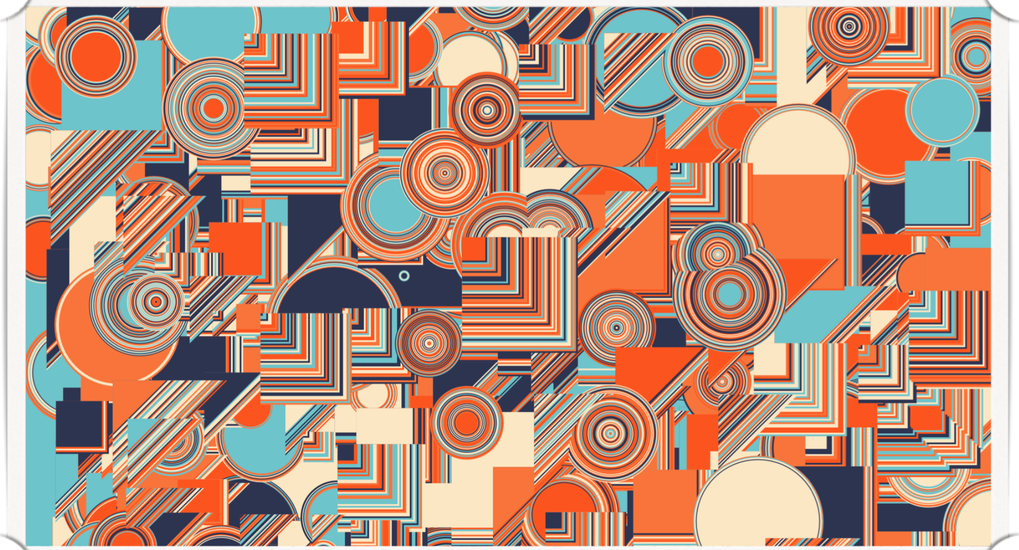
GENERATIVE DESIGN WITH SCALA.JS

Fred CECILIA - @naikyworld

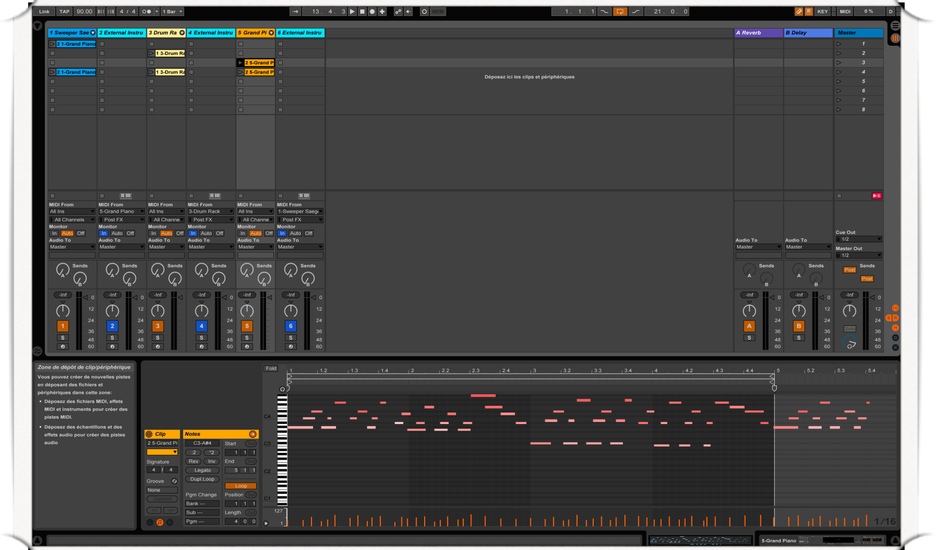
DAY JOB

NIGHT JOB



Web Browser

Midi



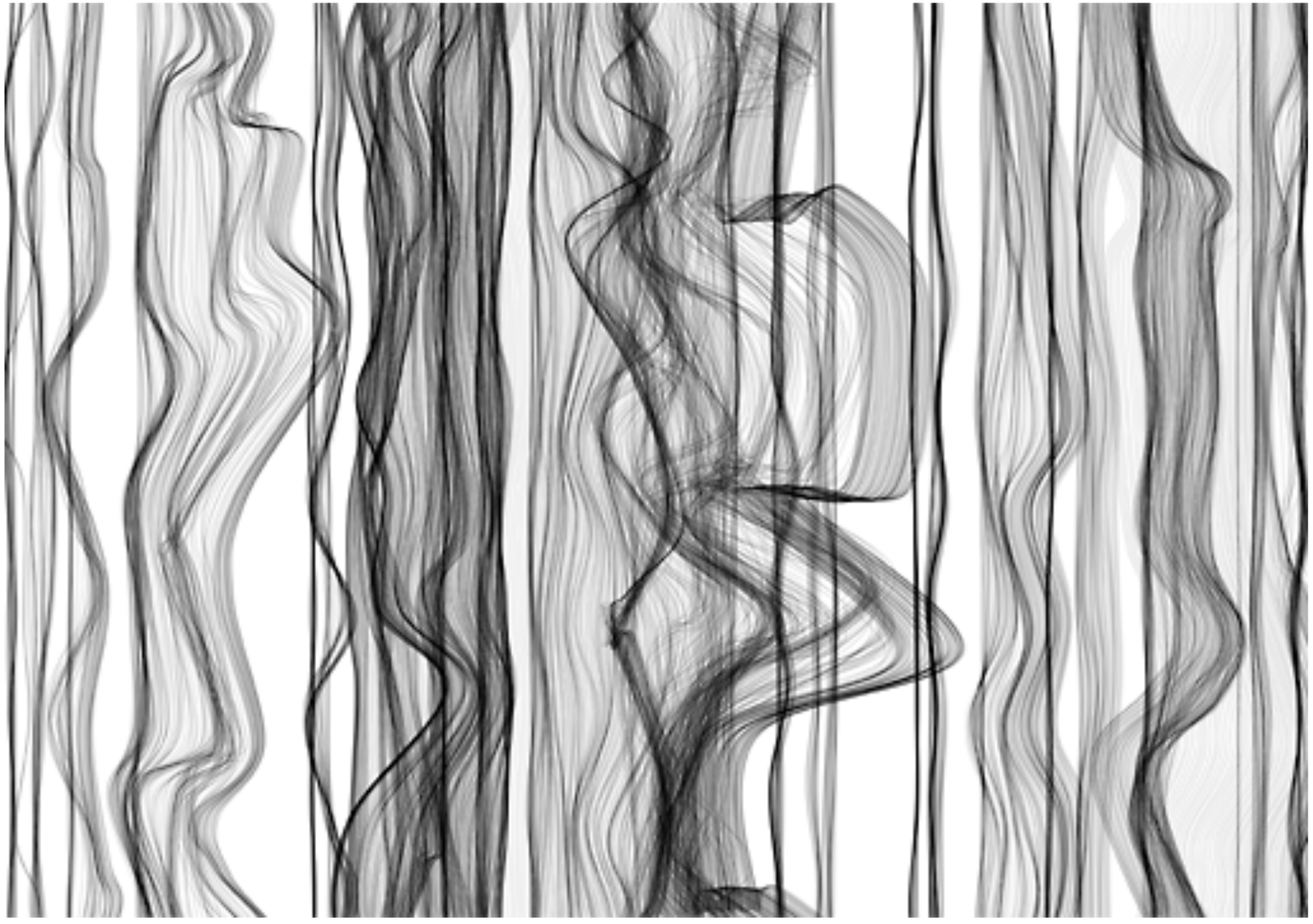
Ableton Live

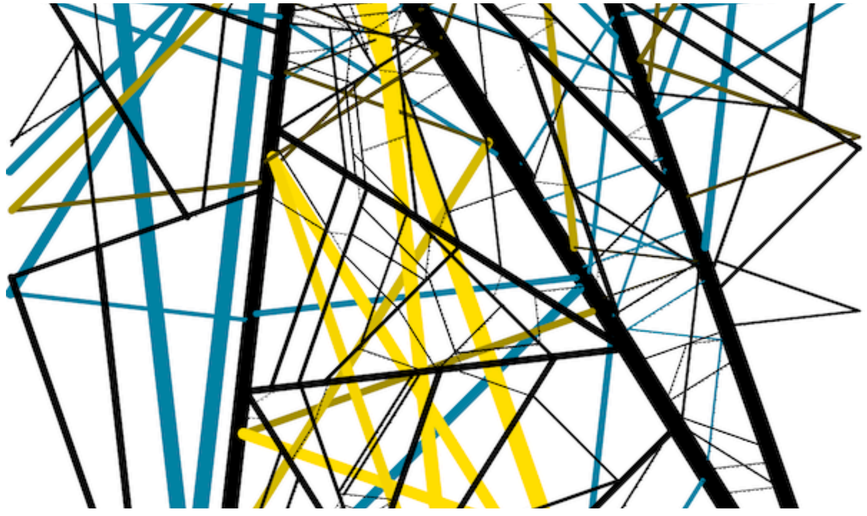
P5.js - Scala.js - Web Midi API

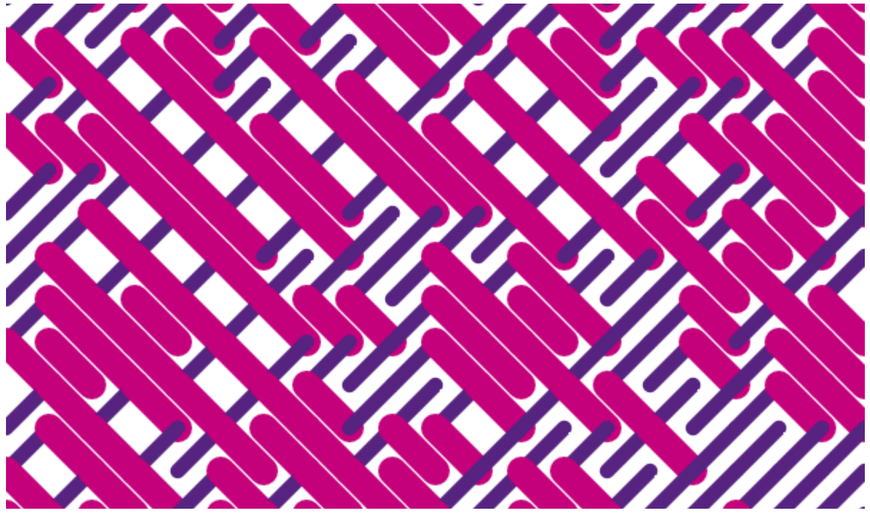
https://www.youtube.com/watch?v=BVl0Jm_P39A

P5.JS

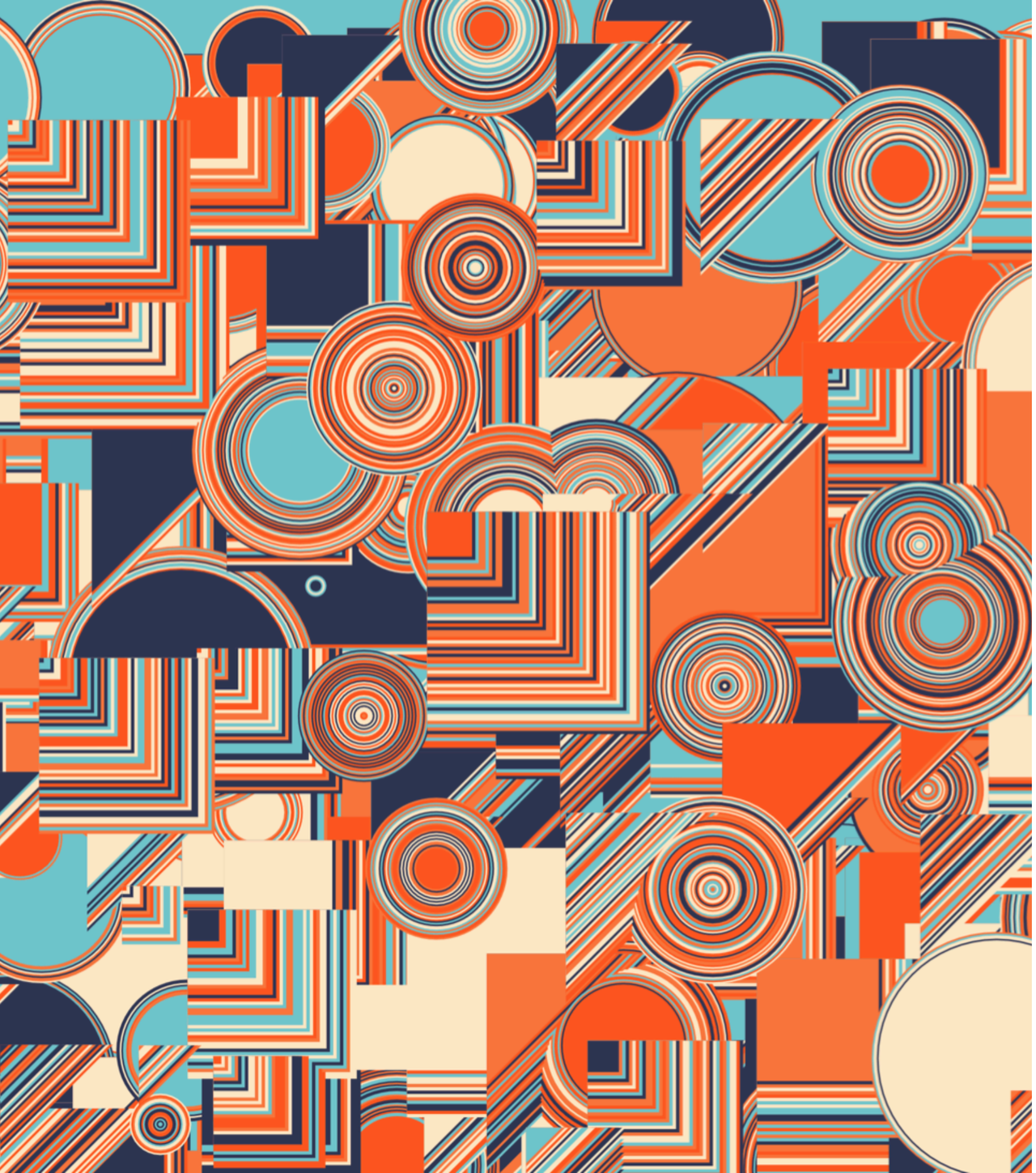
Creative Coding in JS





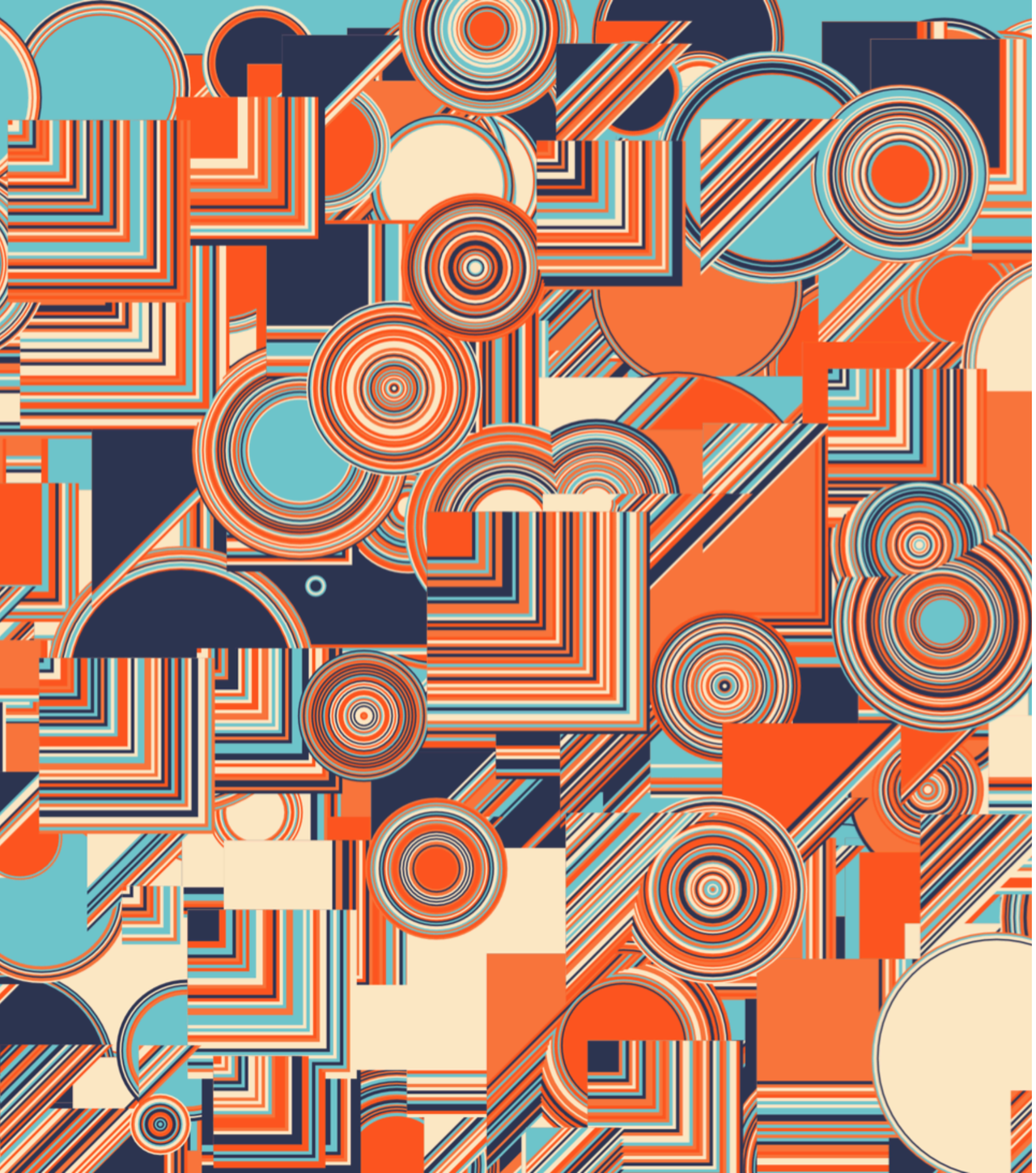


```
var r, g, b;

function setup() {
  createCanvas(640, 480);
  r = random(255);
  g = random(255);
  b = random(255);
}

function draw() {
  fill(r,g,b);
  ellipse(mouseX, mouseY, 80, 80);
}

function mousePressed() {
    r = random(255);
    g = random(255);
    b = random(255);
}
```



PROS

- Simple !
- Good documentation !
- JS ecosystem

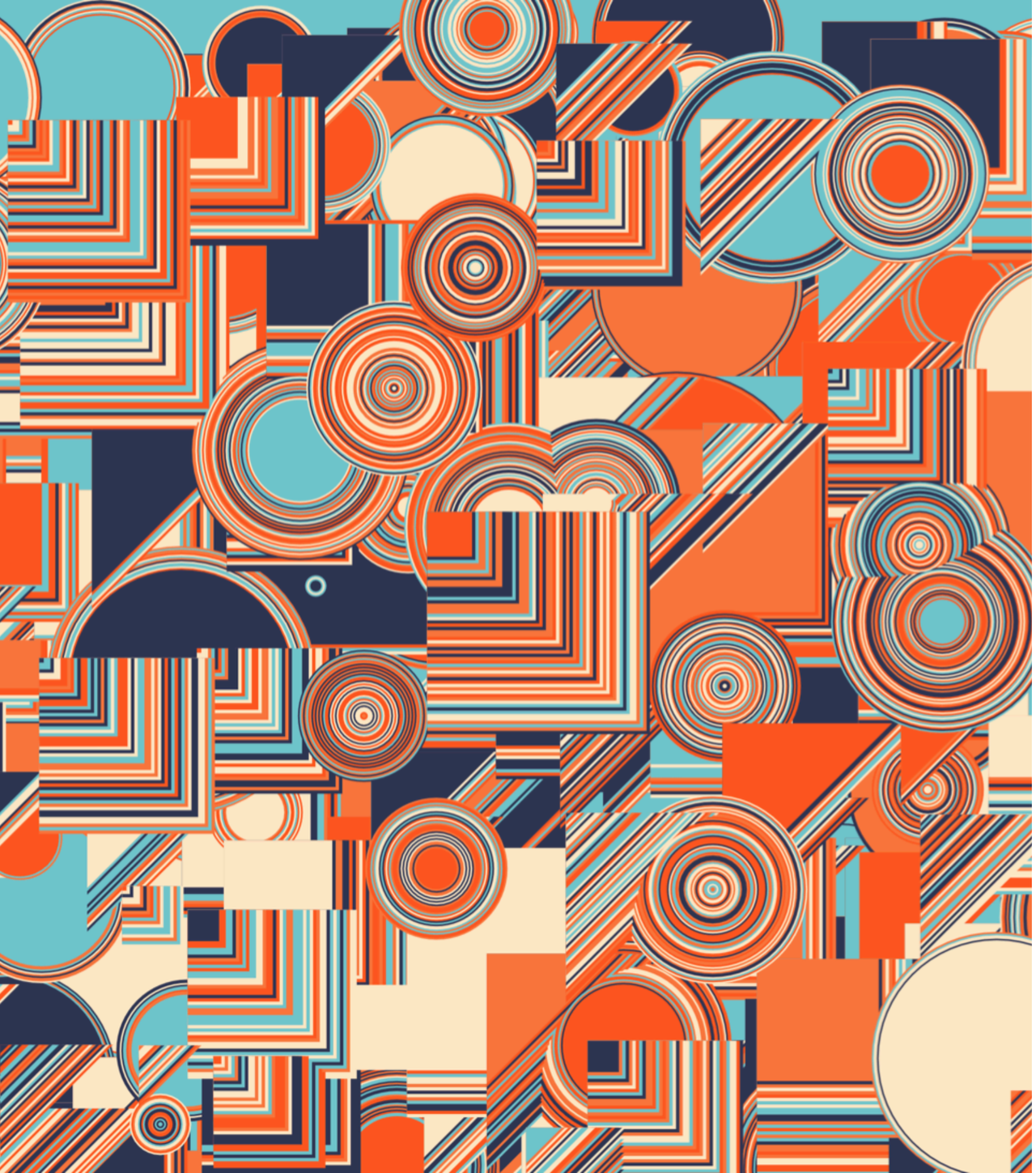


CONS

- Mutable State
- I need my types
- Overall it’s JS

SCALA.JS

Scala-to-Javascript Compiler



WHAT IS SCALA.JS

- Typed & Functional language
- Sharable code between Front and Back
- JS Interoperability (Facade)

```scala
@js.native
trait Window extends js.Object {
  val document: HTMLDocument = js.native
  var location: String = js.native

  def innerWidth : Int = js.native
  def innerHeight: Int = js.native

  def alert(message: String): Unit = js.native

  def open(    url: String,
             target: String,
           features: String = ""): Window = js.native

  def close(): Unit = js.native
}
```

P5.SCALA

Or when P5.js meet Scala.js

STATE

P5.scala

```
case class StateRGB(r: Double, g: Double, b: Double)
```

P5.js

```
var r, g, b;
```

STATE

P5.scala

```
object Demo extends P5Starter[StateRGB] {

  override def setup(): StateRGB = {
    createCanvas(640,480)
    StateRGB(random(255),random(255),random(255))
  }

  override def draw(state: StateRGB): Unit = {
    fill(state.r, state.g, state.b)
    ellipse(mouseX, mouseY,80,80)
  }
}
```

P5.js

```
function setup() {
  createCanvas(640, 480);
  r = random(255);
  g = random(255);
  b = random(255);
}

function draw() {
  fill(r,g,b);
  ellipse(mouseX, mouseY,80, 80);
}
```

P5 EVENTS

P5.scala

```
override def mousePressed(state: StateRGB): StateRGB = {
   StateRGB(random(255),random(255),random(255))
 }
```

P5.js

```
function mousePressed() {
      r = random(255);
      g = random(255);
      b = random(255);
}
```

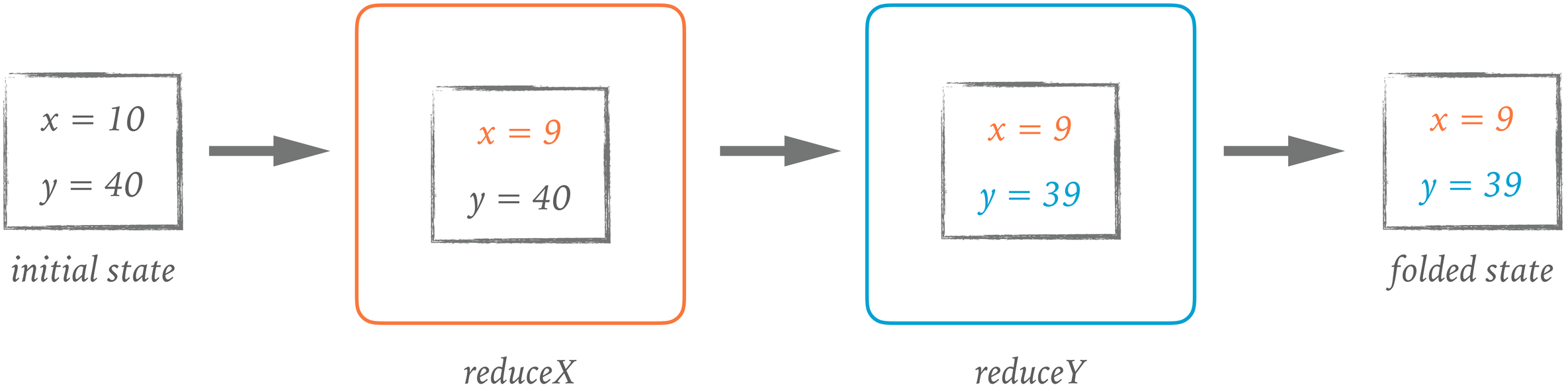
EXTERNAL EVENT

```
override val externalUpdater: List[(State) => State] = List(reduceX, reduceY)
```

EXTERNAL EVENT

```
P5Function.draw = () => {
    val completeState = externalUpdater.foldLeft(state)((s, g) => g(s))
    draw(completeState)
    saveState(completeState)
  }
```

EXTERNAL EVENTS



CONCLUSION

TIME FOR DEMO AGAIN?

THANKS YOU