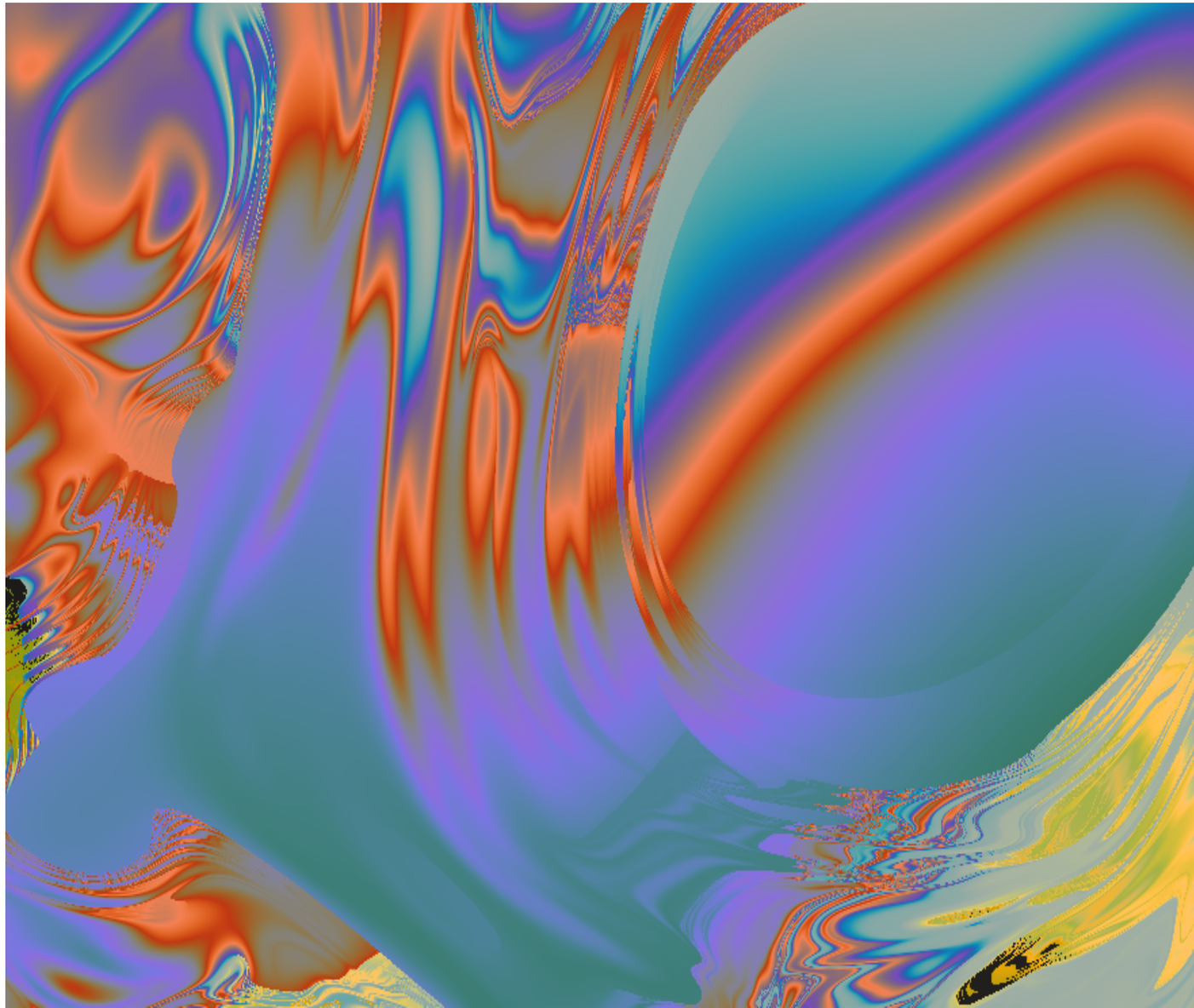
Andreas Schlegel
andreas.schlegel@lasalle.edu.sg

Coding Basics 1



Andreas Schlegel
andreas.schlegel@lasalle.edu.sg

Environment
Text Editor
Local Server

Variables
Functions
Conditionals
Loops
Arrays
Objects
Shapes

Andreas Schlegel
andreas.schlegel@lasalle.edu.sg

Environment

In this class we will use the programming environment p5js, a javascript based coding platform.

p5js p5js.org
web editor editor.p5js.org
reference p5js.org/reference
learn p5js.org/learn

Andreas Schlegel
andreas.schlegel@lasalle.edu.sg

Text Editor

Coding p5js sketches can be done inside the web based editor or locally on your laptop using a Text Editor. Please choose and download one of the following.

Sublime Text sublimetext.com
Atom atom.io

Local Server

p5js sketches will run inside a browser. In some cases you will need to serve the sketch from a server. The following link will help you to setup node.js on your laptop to run a local server.

Setting up a local server with node.js link

Andreas Schlegel
andreas.schlegel@lasalle.edu.sg

Variables
Functions
Conditionals
Loops
Arrays
Objects

Andreas Schlegel
andreas.schlegel@lasalle.edu.sg

Variables

Variables are the backbone of many computer programs. In computer programming, a variable is a location and a symbolic name containing a value. This means that information can be stored in a variable, which can be accessed at other points in the program.

→ **MDN Variables (let)**

Variables

You can use the let or var statement when declaring a variable. Using const instead declares the value of the variable as constant and it can't be changed afterwards.

```
var x = 0

function setup() {
  x = 1
}

function draw() {
  x = x + 1  // increment x by 1
}
```

prefix *assignment*

```
var x = 0
```
value

Andreas Schlegel
andreas.schlegel@lasalle.edu.sg

Variables

Variables are useful to store data and change that data while the program is running.

Andreas Schlegel
andreas.schlegel@lasalle.edu.sg

Functions

A function in programming is a block of code that performs some operation and may or may not return a value. Functions have inputs and outputs. An output is present when the function returns a value. Inputs are called arguments or parameters, they are used to pass external values to a function.

→ **MDN Functions**

Functions

```
function setup() {
  createCanvas(600, 400)
}
```

```
function myOwnFunction() {
  console.log(“Hello.”)
}
```

prefix *function name*

```
function myOwnFunction() {
  console.log(“Hello.”)
}
```

function body

function call *parameter*

Andreas Schlegel
andreas.schlegel@lasalle.edu.sg

Functions

Functions are useful in organising code and for reusability reason.

Conditionals

Conditional statements give you the ability to control the flow of your program, letting it make decisions on what code to execute. The if statement allows you to control if a program enters a section of code or not based on whether a given condition is true or false.

→ **MDN Conditionals**

Andreas Schlegel
andreas.schlegel@lasalle.edu.sg

Conditionals

```
if(x == 0) {

  // execute code here if x is 0

} else if (x == 1) {

  // execute code here if x is 1

} else {

  // execute code here if x is
  // neither 0 nor 1

}
```

Andreas Schlegel
andreas.schlegel@lasalle.edu.sg

Conditionals

Conditionals are useful for a program to make decisions.

Andreas Schlegel
andreas.schlegel@lasalle.edu.sg

Loops

A loop is a sequence of instructions that is continuously repeated until a condition is reached. Often a loop counts up from 0 to a higher number and increases by 1 for each iteration. Loops 'can be nested, a loop within a loop.

→ **MDN** Loops and iterations

Loops

```
for(var i=0; i<10; i++) {
  rect(100, 100 + i * 20, 100, 10);
}
```

number of loop iterations

prefix

```
for(var i=0; i<10; i++) {
  rect(100, 100 + i * 20, 100, 10);
}
```

function body

a variable that serves as counter

```
// this loop will run 10 times.
// I will start at 0 and count up to
// 9, after which the loop concludes
// and the program continues with
// the next operation.
// 10 rectangles will be draw from
// along the y axis
```

Andreas Schlegel
andreas.schlegel@lasalle.edu.sg

Loops

Loops are useful to perform a particular operation multiple times instead of writing the same code number of times.

Arrays

In JavaScript an array is a single variable that is used to store a range of different elements. It is often used when we want to store list of elements and access them by a single variable.

→ **MDN Arrays**

Andreas Schlegel
andreas.schlegel@lasalle.edu.sg

Arrays

```
// declare and initialise an array
var arr = []

// assign a value at index 0
arr[0] = 100

// use the push command to add a new
// value to the tail of the array
arr.push(200)

// get the length/size of the array
arr.length

// and there is more, follow the link
```

⟶ **MDN Arrays**

Andreas Schlegel
andreas.schlegel@lasalle.edu.sg

Arrays

Arrays are useful to store a collection of data, it is often useful to think of an array as a collection of variables.

Objects

An object is a collection of related data and/or functionality which usually consists of several variables and functions.

→ **MDN Object Guides and Classes**

Andreas Schlegel
andreas.schlegel@lasalle.edu.sg

Objects

```
// an object in javascript
var person = {
  name: ['Bob', 'Smith'],
  age: 32,
  gender: 'male',
  interests: ['music', 'skiing'],
  greeting: function() {
    alert('Hi! I\'m ' +
    this.name[0] + '.');
  }
};
// call function greeting
person.greeting()
```

⟶ **MDN Object Guides and Classes**

Andreas Schlegel
andreas.schlegel@lasalle.edu.sg

Objects

```
// class example
class ClassObject {
  constructor() {
    this.x = Math.random(0,100)
  }

  update() {
    console.log(this.x)
  }
}
// create multiple copies of class ClassObject
var o1 = new ClassObject()
var o2 = new ClassObject()

o1.update()
o2.update()

// for more information on how objects and class
// classes work, follow the links below
```

⟶ **MDN Object Guides and Classes**

Andreas Schlegel
andreas.schlegel@lasalle.edu.sg

Objects

Objects are useful to structure your program and control the state of your data more efficiently.

Andreas Schlegel
andreas.schlegel@lasalle.edu.sg

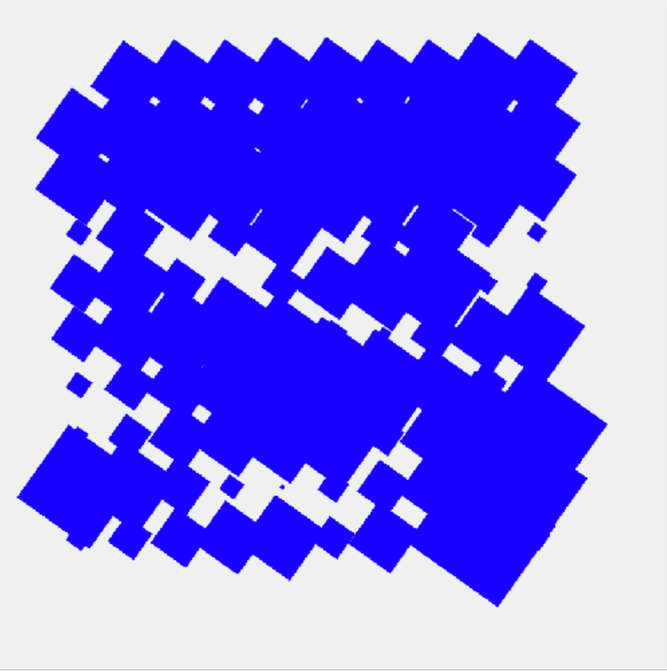
Shapes

Some p5js shapes, and there are many more, do follow the link below.

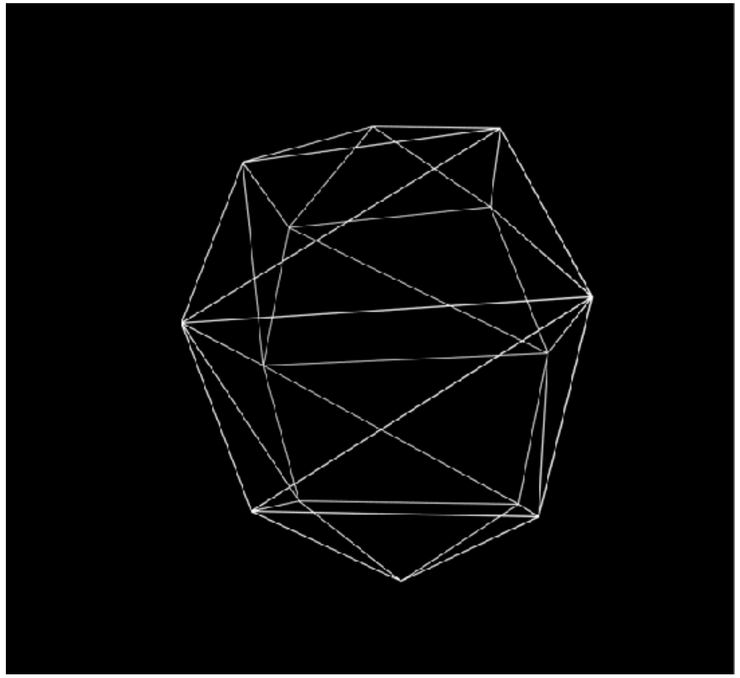
```
rect(x, y, w, h)
ellipse(x, y, w, h)
triangle(x1, y1, x2, y2, x3, y3)
line(x1, y1, x2, y2)
vertex(x, y)

beginShape()
  vertex(x1, y1)
  vertex(x2, y2)
  vertex(x3, y3)
endShape()
```

⟶ p5js reference scroll down to section shape

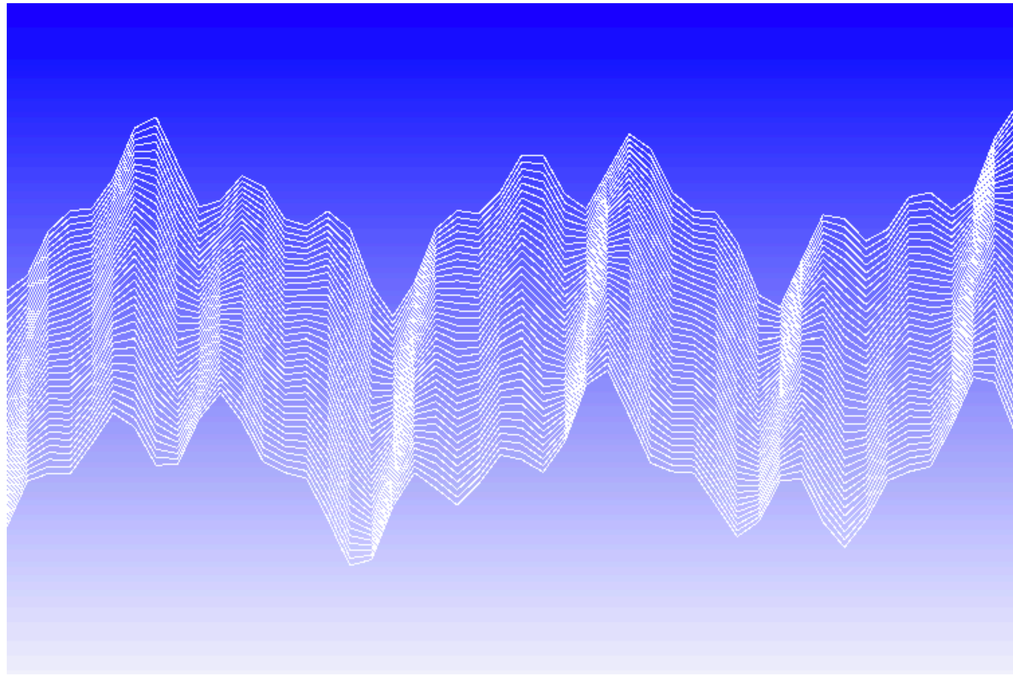
Andreas Schlegel
andreas.schlegel@lasalle.edu.sg



2D **Rectangle** rect() in a 10x10 grid



3D **Sphere** sphere() with low sphereDetail



2D **Line** beginShape(), endShape() and vertex()

Shapes

Andreas Schlegel
andreas.schlegel@lasalle.edu.sg

Shapes

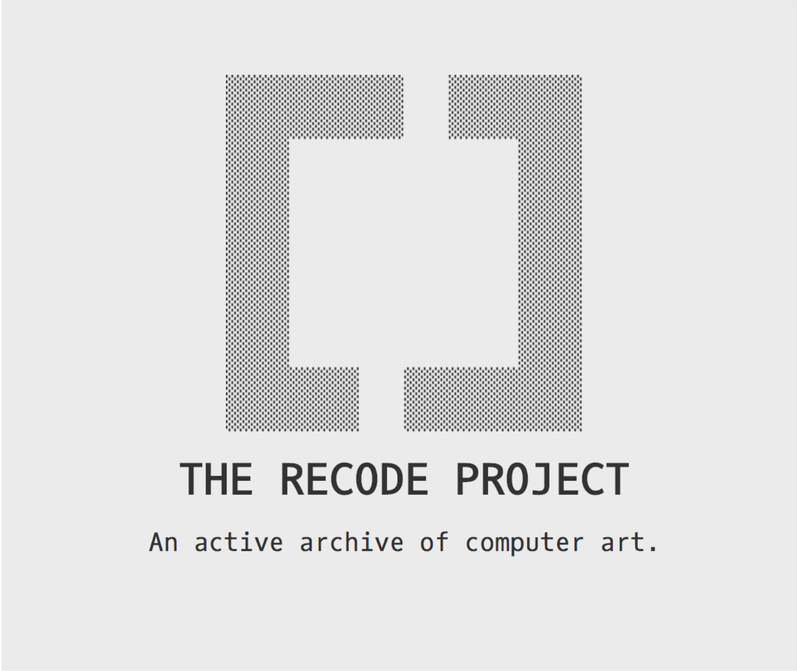
Transformations allow you to manipulate coordinates, rotation or scale of shapes.

```
translate(x, y, z)
rotate(angle)
rotateX(angle)
rotateY(angle)
rotateZ(angle)
scale(x, y, z)
push()
pop()

// for more in depth explanations with
// examples follow the link below
```

⟶ Gene Kogan's overview of p5js-transformations

Andreas Schlegel
andreas.schlegel@lasalle.edu.sg

Resources



The ReCode Project is a community-driven effort to preserve computer art by translating it into a modern programming language.
recodeproject.com

Generative Design: Visualize, Program, and Create with JavaScript in p5.js. www.generative-gestaltung.de/2