Creative Coding

Professor Danne Woo

creativecode.dannewoo.com

ARTS 249

Spring 2020

Thursday 2:00 PM – 5:50 PM

I-Building 213

Static Visual Design

Week 01: Intro to programming, Processing and creative coding

Week 02: Forms, Shapes and Variables

Week 03: Computational Color and Export

Week 04: Repetition, Decisions and Randomization

Week 05: Functions, Classes and Typography

Week 06: Data Visualization

Week 07: Midterm Presentation

Repetition

Nature

Repetition

Nature

Repetition

Nature

Repetition

Nature

Repetition

Fashion

Repetition

Interior Design

Repetition

Packaging Design

Repetition

Graphic Design

Repetition

Art

Repetition

Art

Repetition

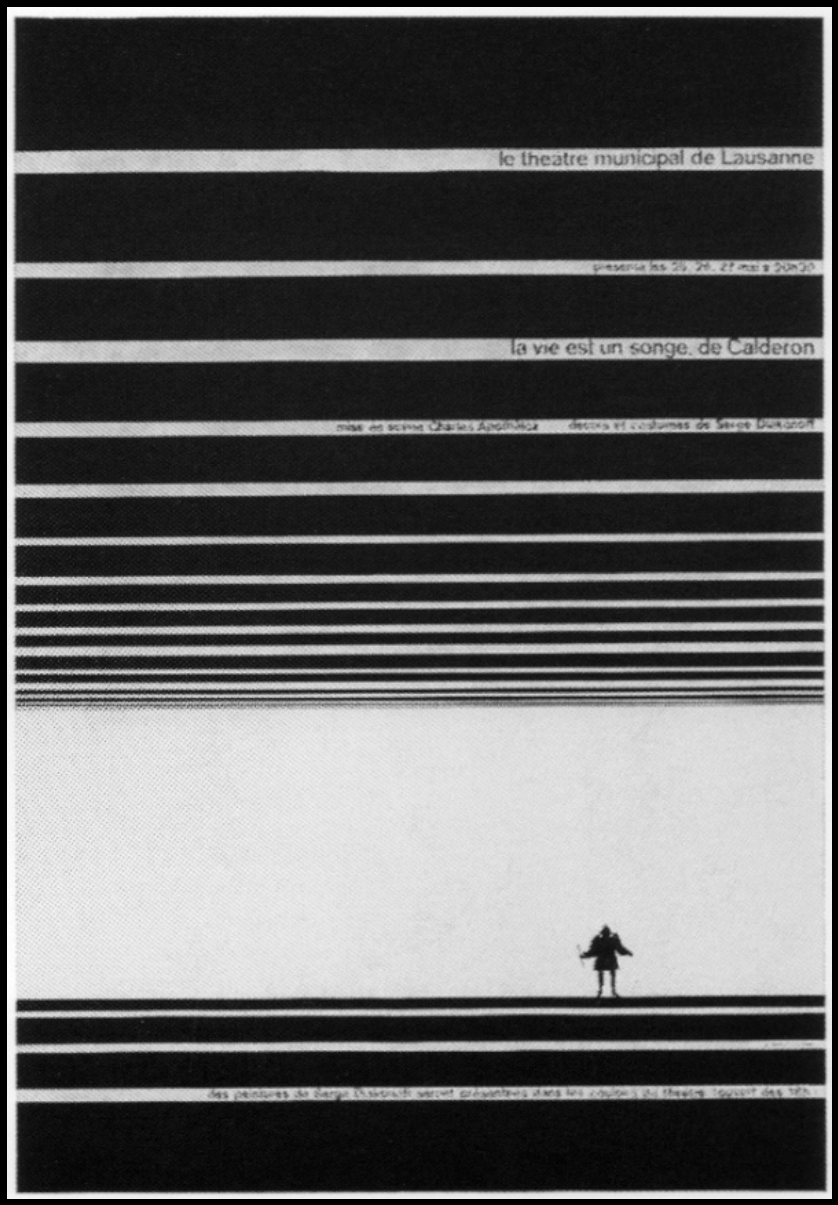
Movement

Repetition

Rhythm

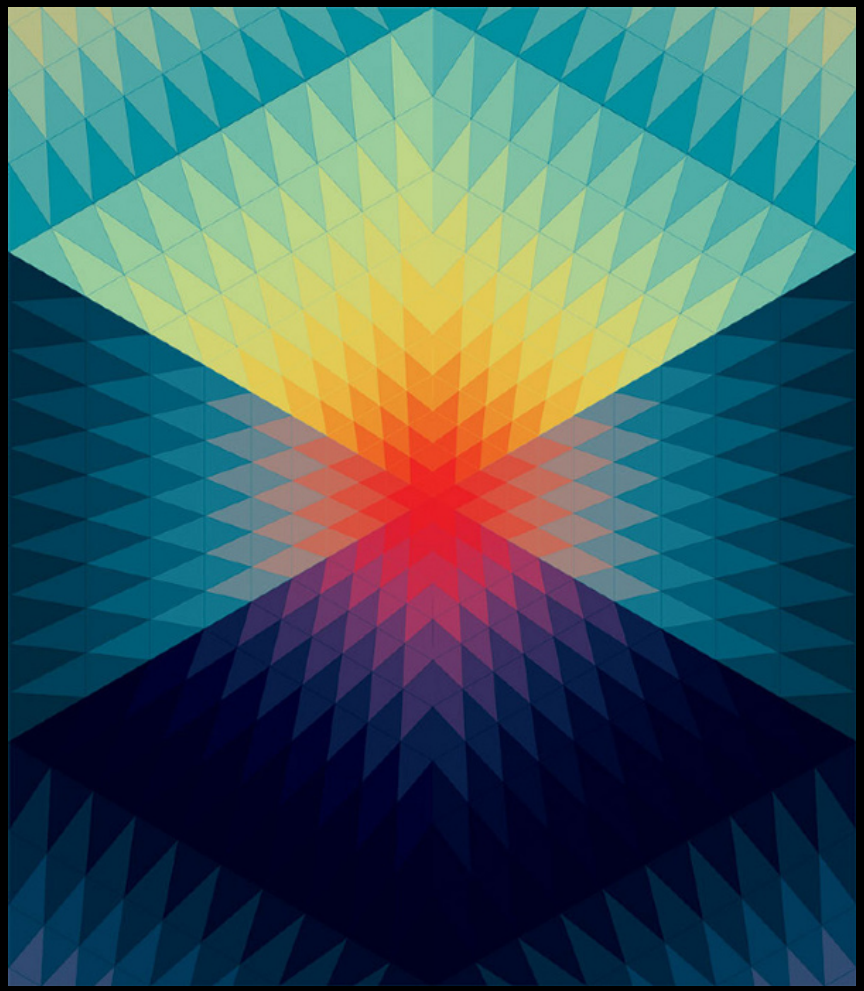
Repetition

Direction



Repetition

Texture

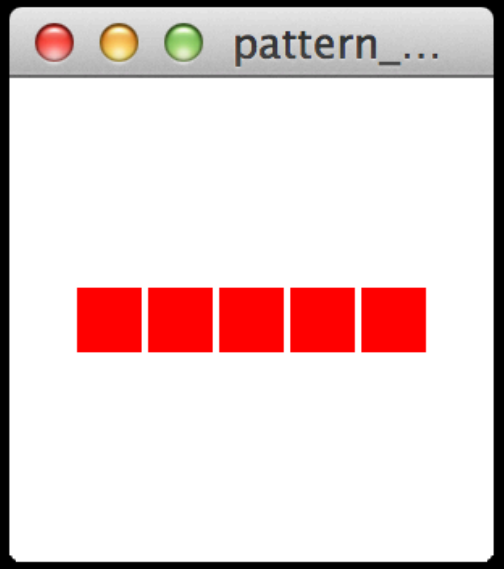


Repetition

for loop

```
for (var i=0; i < 5; i+=1) {
        rect(21*i, 0, 20, 20);
}
```

```
for (variable; test; update) {
        …then do this;
}
```



Repetition

for loop

```
for (var i=0; i < 10; i+=1) {
        fill(36*i, 100, 100);
        rect(width/10*i, 0, width/10, height);
}
```


```
fill(0, 100, 100);
rect(width, 0, width/10, height);
fill(36, 100, 100);
rect(width/10, 0, width/10, height);
fill(72, 100, 100);
rect(width/20, 0, width/10, height);
fill(108, 100, 100);
rect(width/30, 0, width/10, height);
fill(144, 100, 100);
rect(width/40, 0, width/10, height);
fill(180, 100, 100);
rect(width/50, 0, width/10, height);
fill(216, 100, 100);
rect(width/60, 0, width/10, height);
fill(252, 100, 100);
rect(width/70, 0, width/10, height);
fill(288, 100, 100);
rect(width/80, 0, width/10, height);
fill(324, 100, 100);
rect(width/90, 0, width/10, height);
```

Repetition

Arithmetic Syntax

++ increase by 1

-- decrease by 1

+= add assigned value (ie. i += 5, increase i by 5)

-= subtract assigned value (ie. i -= 5, decrease i by 5)

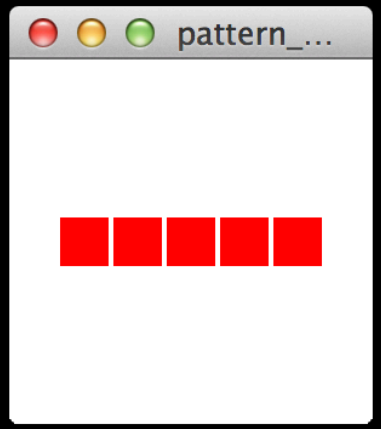
***=** multiply assigned value (ie. i *= 5, multiply i by 5)

/= divide assigned value (ie. i /= 5, divide i by 5)

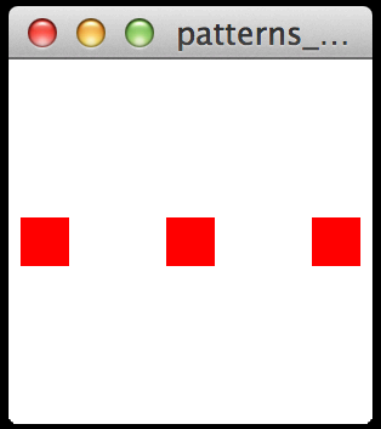
Repetition

for loop

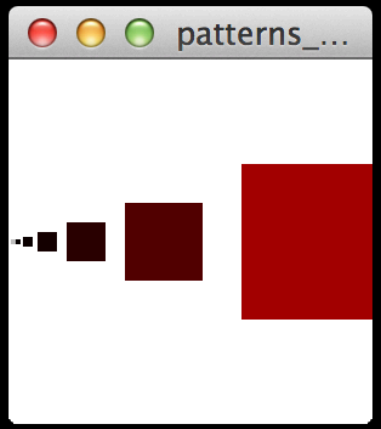
```
for (var i=0; i < 5; i++) {
        rect(21*i, 0, 20, 20);
}
```




```
for (var i=0; i < 9; i+=3) {
        rect(20*i, 0, 20, 20);
}
```



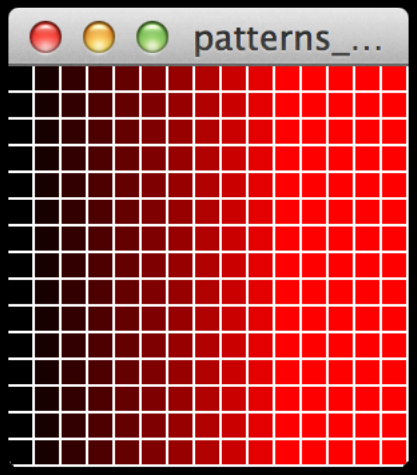

```
for (var i = 1; i < 65; i*=2) {
        fill(0, 100, i);
        rect(2*i, height/2, i, i);
}
```



Repetition

double for loop

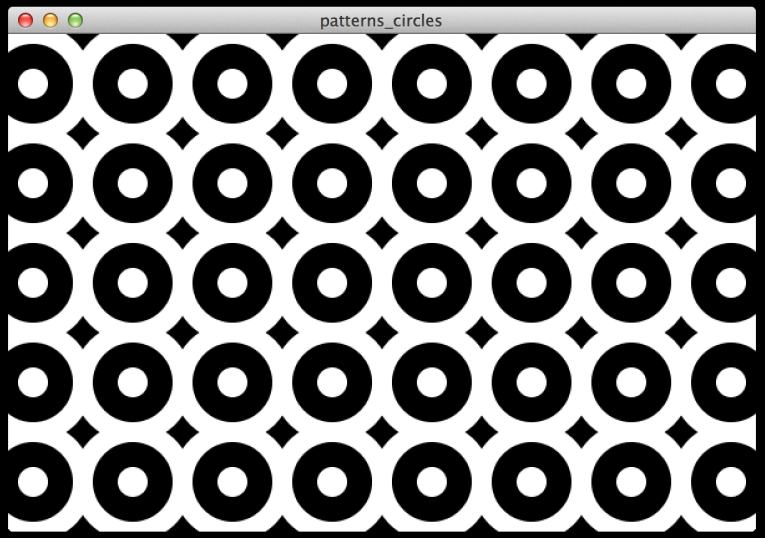
```
for (var i = 0; i < width/10; i++) {
        for (var j = 0; j < height/10; j++) {
                fill(0, 100, 10*i);
                rect(10*i, 10*j, 9, 9);
        }
}
```



Repetition

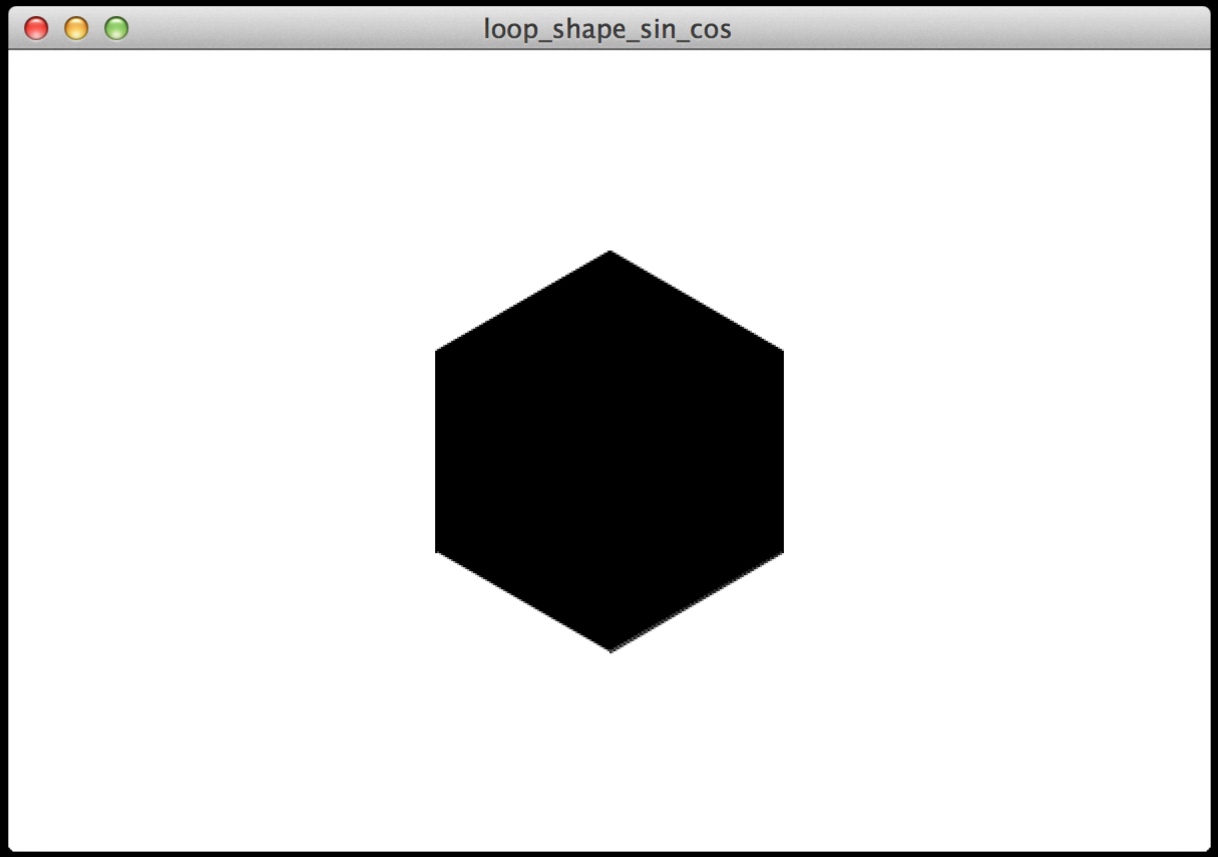
Simple Pattern

```
var circleSize = 80;
translate(circleSize/2, circleSize/2);
for(var x = 0; x < width; x += circleSize) {
    for(var y = 0; y < height; y += circleSize) {
        fill(255);
        ellipse(x, y, circleSize * 1.2, circleSize * 1.2);
        fill(0);
        ellipse(x, y, circleSize * 0.8, circleSize * 0.8);
        fill(255);
        ellipse(x, y, circleSize * 0.3, circleSize * 0.3);
    }
}
```



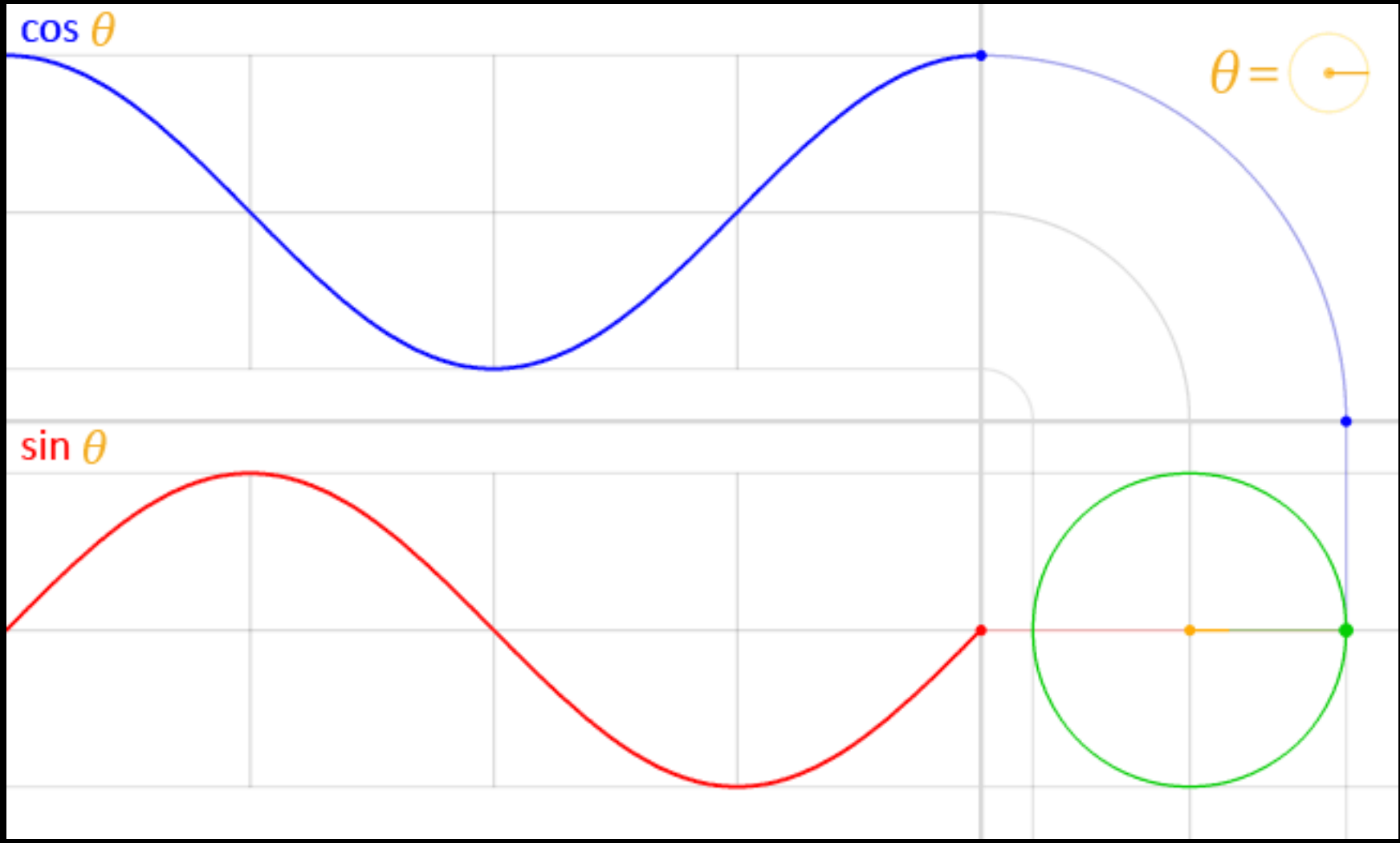
Repetition

Cos and Sin Shapes



Repetition

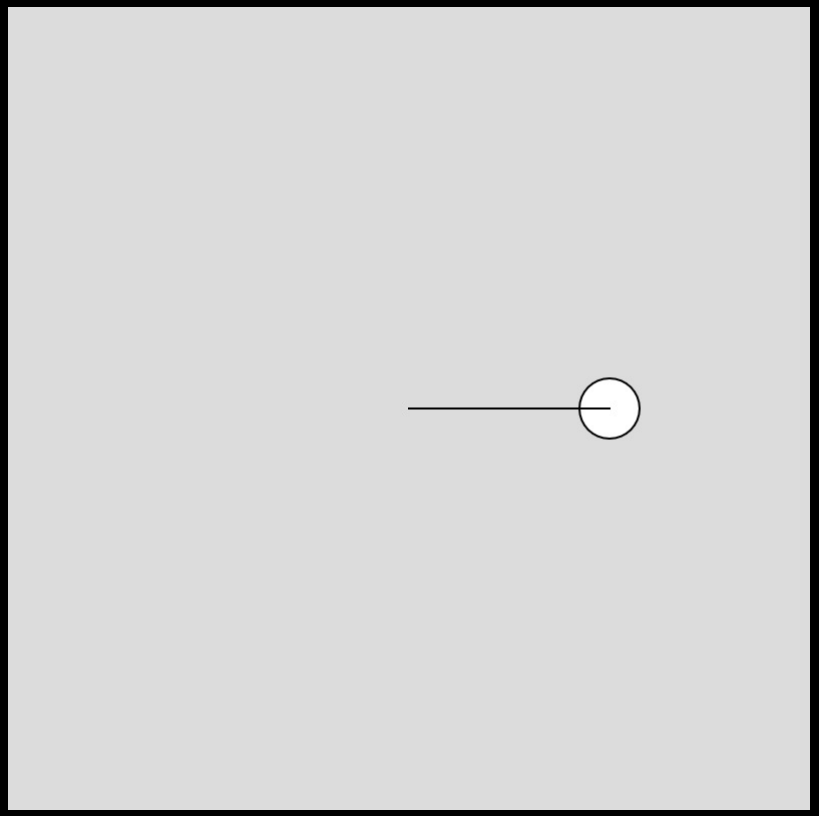
Cos and Sin Shapes



Repetition

Cos and Sin Shapes

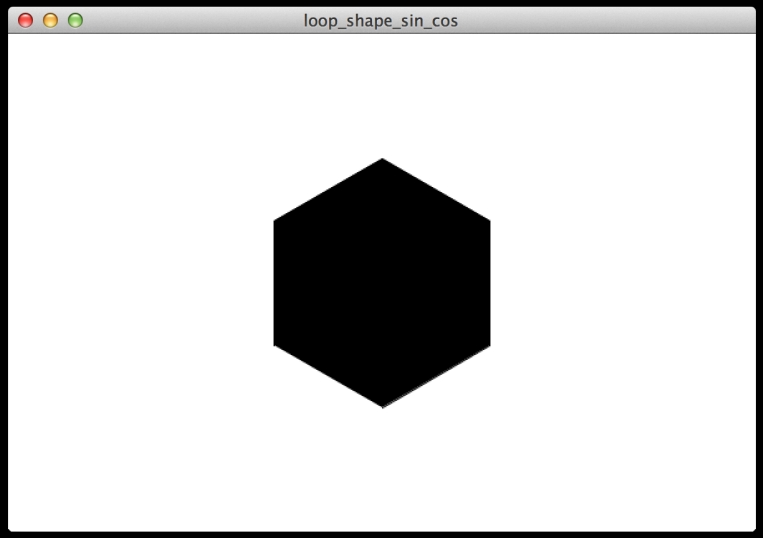
```
var radian = 100;
var degree = 90;

function setup() {
  createCanvas(400, 400);
  background(220);
  translate(width/2, height/2);
  var x = radian * sin(radians(degree));
  var y = radian * cos(radians(degree));
  ellipse(x, y, 30, 30);
  line(0, 0, x, y);
}
```



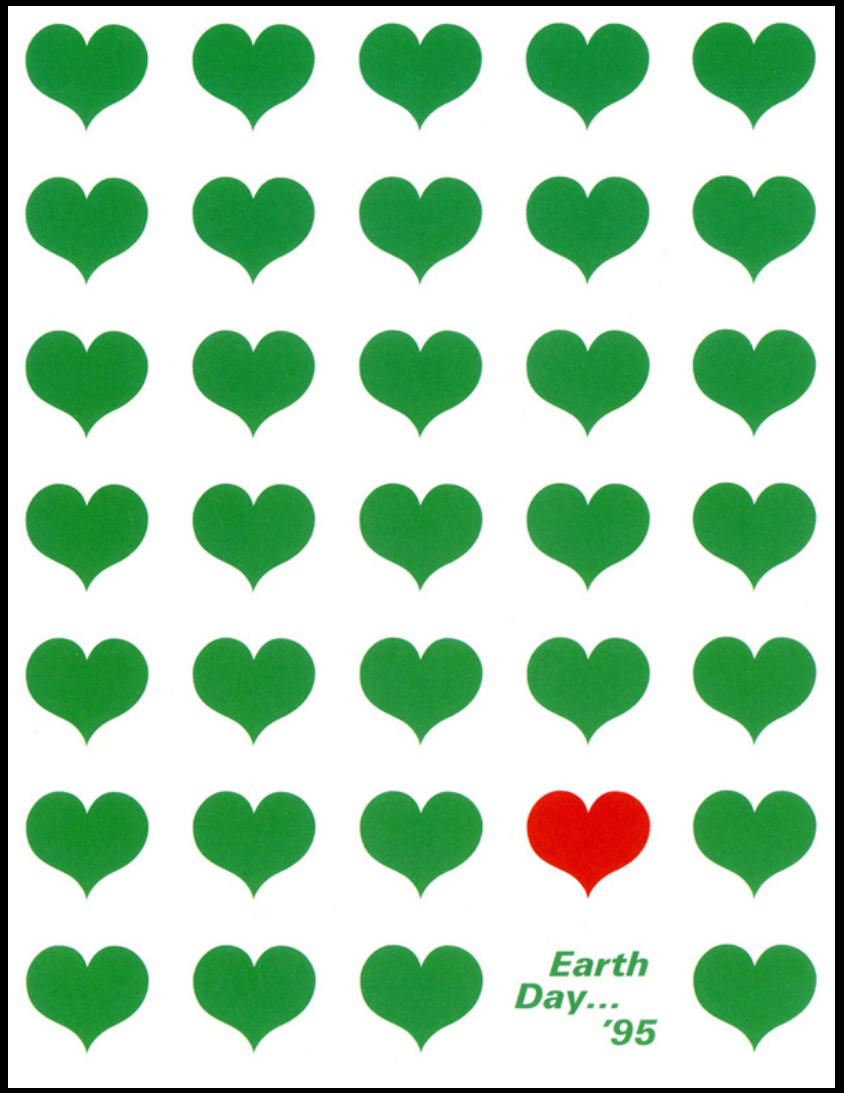
Repetition

Cos and Sin Shapes

```
var points = 6;
var r = 100;

function setup() {
  createCanvas(600, 400);
  background(255);
  translate(width/2, height/2);
  fill(0);

  beginShape();
  for (var i = 0; i < points; i++) {
    var vertexX = r * sin(radians(i * (360/points)));
    var vertexY = r * cos(radians(i * (360/points)));
    vertex(vertexX, vertexY);
  }
  endShape();
}
```



Decisions



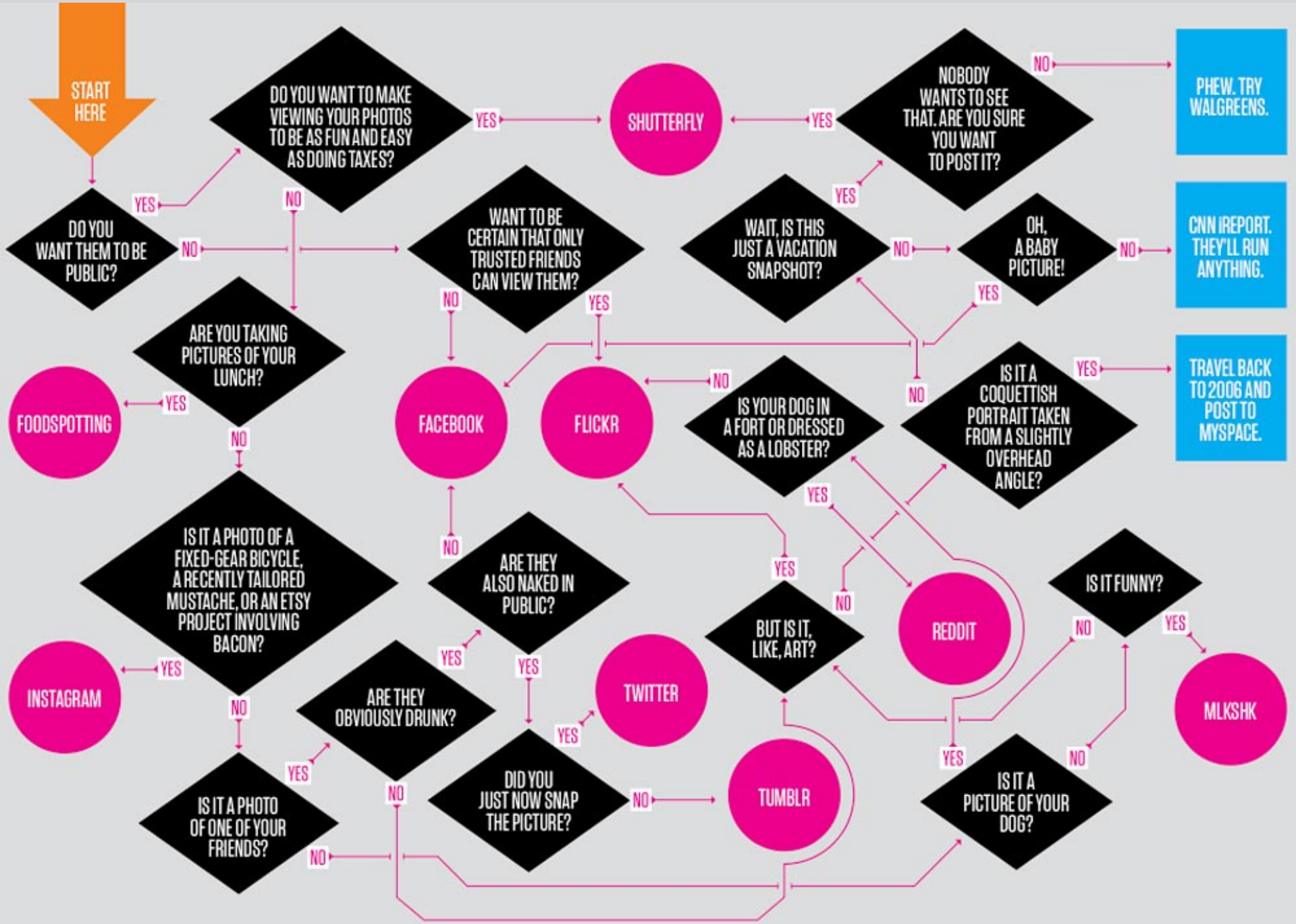
Decisions

if statements

```
if (conditional statement) {
        …then do this;
}
```

```
if (conditional statement) {
        …then do this;
} else {
        …then do this;
}
```

```
if (conditional statement) {
        …then do this;
} else if (conditional statement) {
        …then do this;
} else if (conditional statement) {
        …then do this;
} else if (conditional statement) {
        …then do this;
} else if (conditional statement) {
        …then do this;
} else {
        …then do this;
}
```


START HERE
DO YOU WANT TO MAKE VIEWING YOUR PHOTOS TO BE AS FUN AND EASY AS DOING TAXES?
YES
SHUTTERFLY
YES
NOBODY WANTS TO SEE THAT. ARE YOU SURE YOU WANT TO POST IT?
NO
PHEW. TRY WALGREENS.
YES
DO YOU WANT THEM TO BE PUBLIC?
NO
NO
WANT TO BE CERTAIN THAT ONLY TRUSTED FRIENDS CAN VIEW THEM?
YES
WAIT, IS THIS JUST A VACATION SNAPSHOT?
NO
OH, A BABY PICTURE!
NO
CNN iREPORT. THEY'LL RUN ANYTHING.
YES
YES
NO
NO
ARE YOU TAKING PICTURES OF YOUR LUNCH?
YES
FOODSPOTTING
NO
FACEBOOK
FLICKR
NO
IS YOUR DOG IN A FORT OR DRESSED AS A LOBSTER?
NO
IS IT A COQUETTISH PORTRAIT TAKEN FROM A SLIGHTLY OVERHEAD ANGLE?
YES
TRAVEL BACK TO 2006 AND POST TO MYSPACE.
IS IT A PHOTO OF A FIXED-GEAR BICYCLE, A RECENTLY TAILORED MUSTACHE, OR AN ETSY PROJECT INVOLVING BACON?
NO
ARE THEY ALSO NAKED IN PUBLIC?
YES
YES
NO
BUT IS IT, LIKE, ART?
NO
REDDIT
IS IT FUNNY?
NO
YES
MLKSHK
INSTAGRAM
YES
NO
ARE THEY OBVIOUSLY DRUNK?
YES
NO
TWITTER
YES
YES
NO
IS IT A PICTURE OF YOUR DOG?
YES
NO
IS IT A PHOTO OF ONE OF YOUR FRIENDS?
YES
NO
DID YOU JUST NOW SNAP THE PICTURE?
NO
TUMBLR

Decisions

Syntax

>	greater than
<	less than
>=	greater than or equal to
<=	less than or equal to
==	equal to
!=	not equal to
||	logical OR
&&	logical AND
!	logical NOT

```
if (conditional statement) {
        …then do this;
} else {
        …then do this;
}
```

Decisions

Example if statements

```
if (x == 8) {
   color = red;
} else {
   color = black;
}
```

```
if (x > 3) {
   color = red;
} else {
   color = black;
}
```

```
if (x <= 10) {
   color = red;
} else {
   color = black;
}
```

```
if (x == 3 && y == 4) {
   color = red;
} else {
   color = black;
}
```

```
if (x == 10 || y < 20) {
   color = red;
} else {
   color = black;
}
```

```
if (x != 3 && y != 1) {
   color = red;
} else {
   color = black;
}
```

Decisions

Shorthand if statements

```
if (x == 8) color = red;
else color = black;
```

```
if (x > 3) color = red;
else color = black;
```

```
if (x <= 10) color = red;
else color = black;
```

Decisions

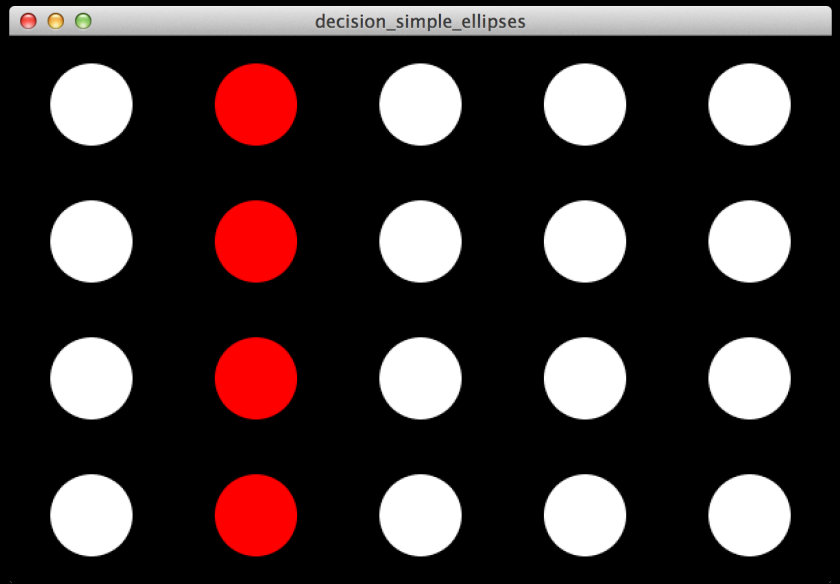
Simple Example

```
var maxCol = 5, maxRow = 4, circleD = 60;

var xSpacing = (width/maxCol);
var ySpacing = (height/maxRow);

for (var x = 0; x < maxCol; x++) {
   for (var y = 0; y < maxRow; y++) {
      if (x == 1) {
         fill(0, 100, 100);
      } else {
         fill(0, 0, 100);
      }
      ellipse(x*xSpacing, y*ySpacing, circleD, circleD);
   }
}
```

Decisions

Simple Example

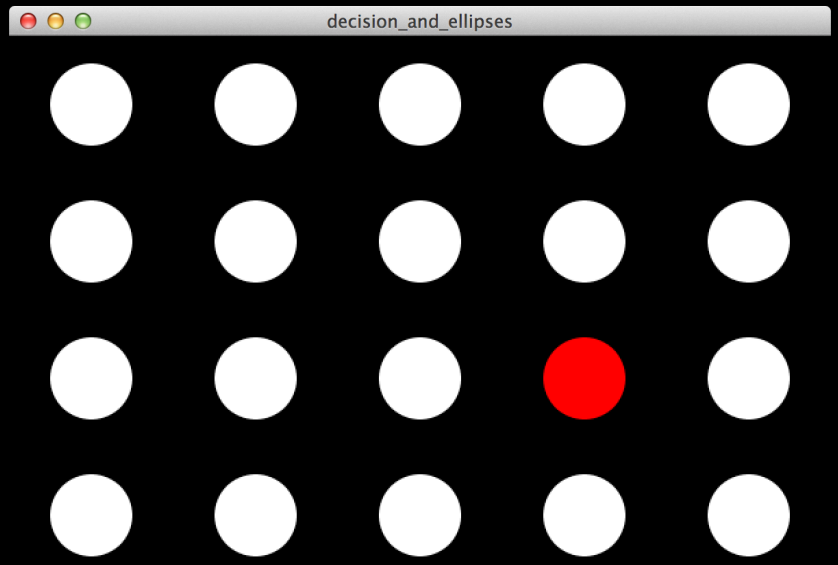
```
var maxCol = 5, maxRow = 4, circleD = 60;

var xSpacing = (width/maxCol);
var ySpacing = (height/maxRow);

for (var x = 0; x < maxCol; x++) {
   for (var y = 0; y < maxRow; y++) {
      if (x == 1) {
         fill(0, 100, 100);
      } else {
         fill(0, 0, 100);
      }
      ellipse(x*xSpacing, y*ySpacing, circleD, circleD);
   }
}
```



Decisions

Simple Example

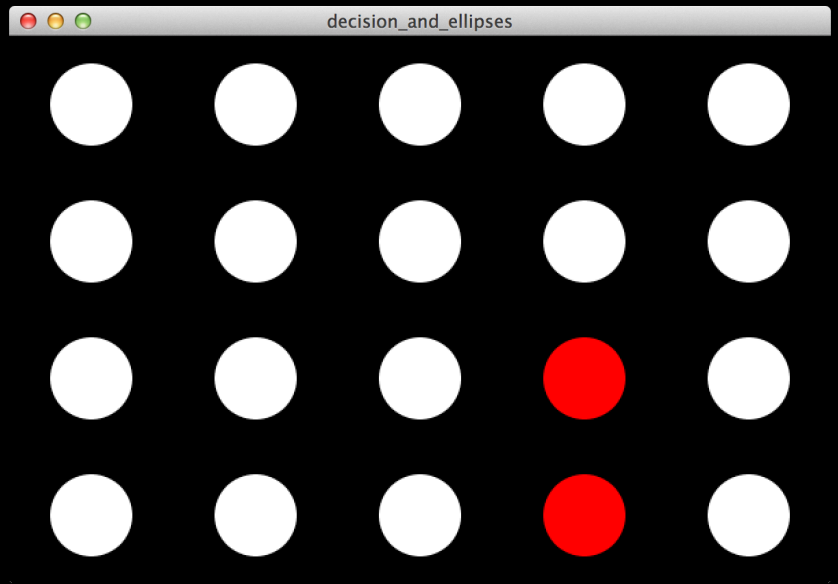
```
var maxCol = 5, maxRow = 4, circleD = 60;

var xSpacing = (width/maxCol);
var ySpacing = (height/maxRow);

for (var x = 0; x < maxCol; x++) {
   for (var y = 0; y < maxRow; y++) {
      if (x == 3 && y == 2) {
         fill(0, 100, 100);
      } else {
         fill(0, 0, 100);
      }
      ellipse(x*xSpacing, y*ySpacing, circleD, circleD);
   }
}
```



Decisions

Simple Example

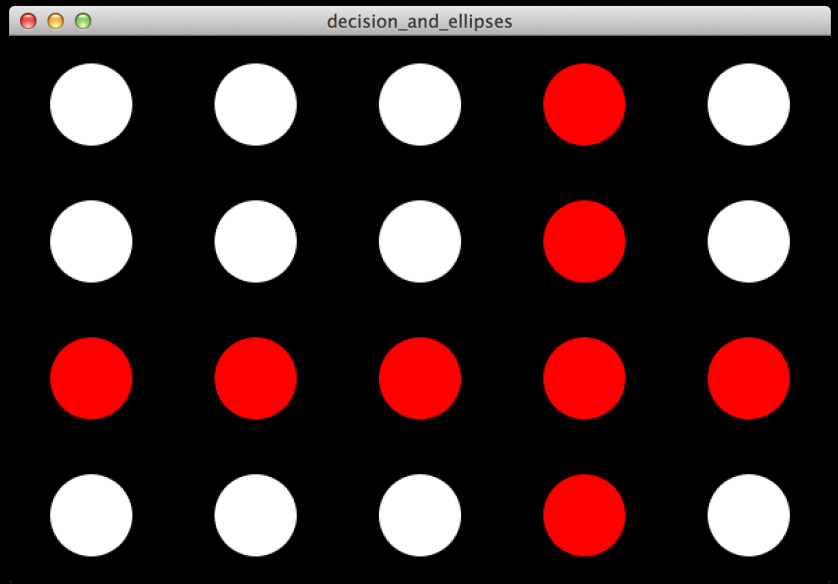
```
var maxCol = 5, maxRow = 4, circleD = 60;

var xSpacing = (width/maxCol);
var ySpacing = (height/maxRow);

for (var x = 0; x < maxCol; x++) {
   for (var y = 0; y < maxRow; y++) {
      if (x == 3 && y >= 2) {
         fill(0, 100, 100);
      } else {
         fill(0, 0, 100);
      }
      ellipse(x*xSpacing, y*ySpacing, circleD, circleD);
   }
}
```



Decisions

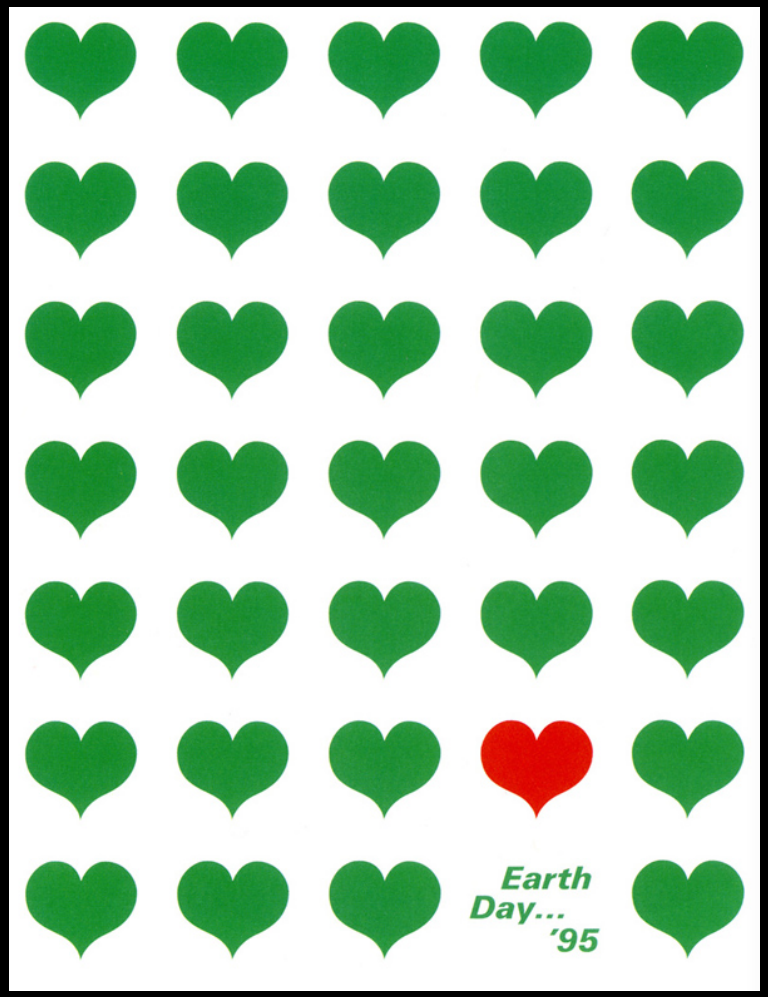
Simple Example

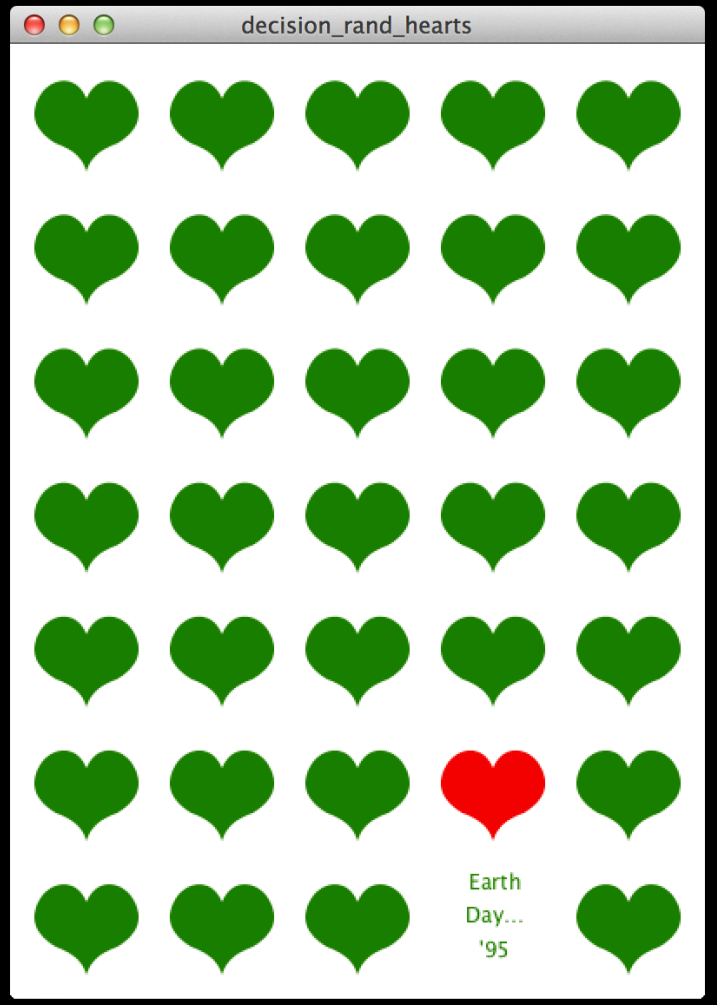
```
var maxCol = 5, maxRow = 4, circleD = 60;

var xSpacing = (width/maxCol);
var ySpacing = (height/maxRow);

for (var x = 0; x < maxCol; x++) {
   for (var y = 0; y < maxRow; y++) {
      if (x == 3 || y == 2) {
         fill(0, 100, 100);
      } else {
         fill(0, 0, 100);
      }
      ellipse(x*xSpacing, y*ySpacing, circleD, circleD);
   }
}
```



Decisions

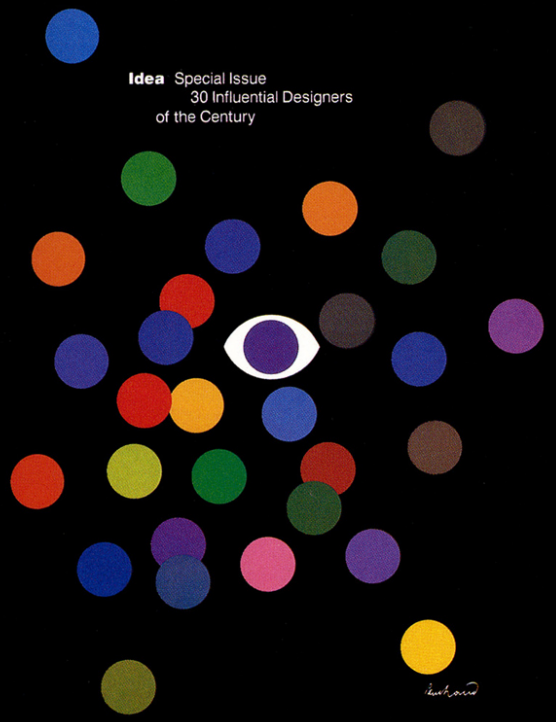
Rand Example

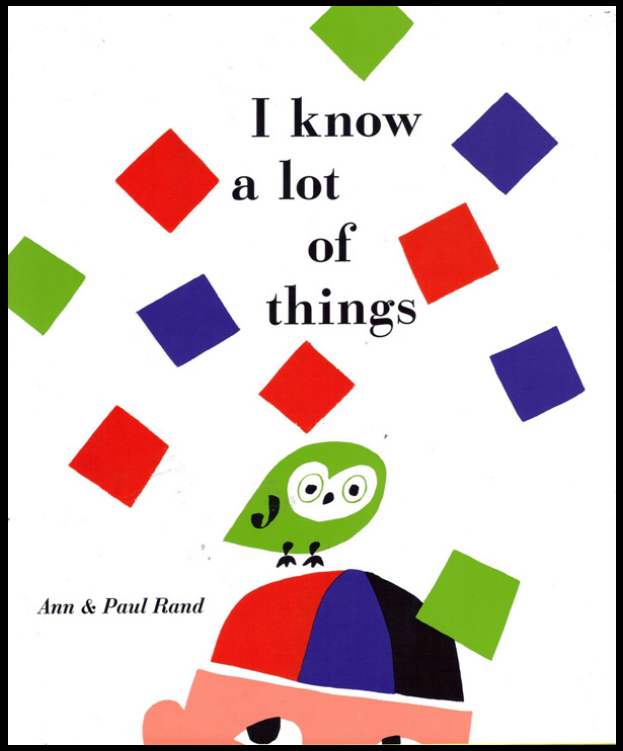




Randomization

Position
Rotation
Color
Dimensions
Opacity

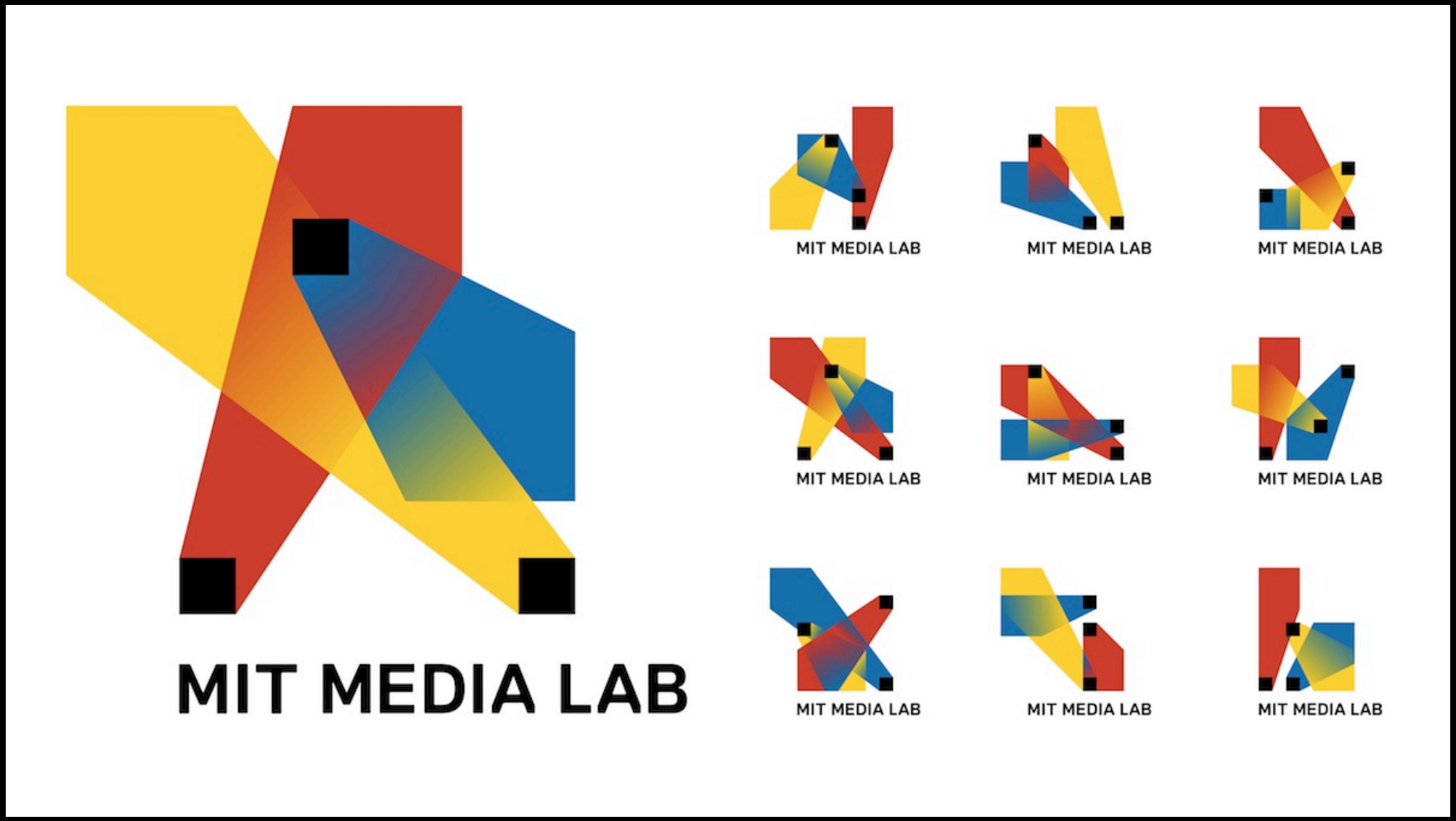


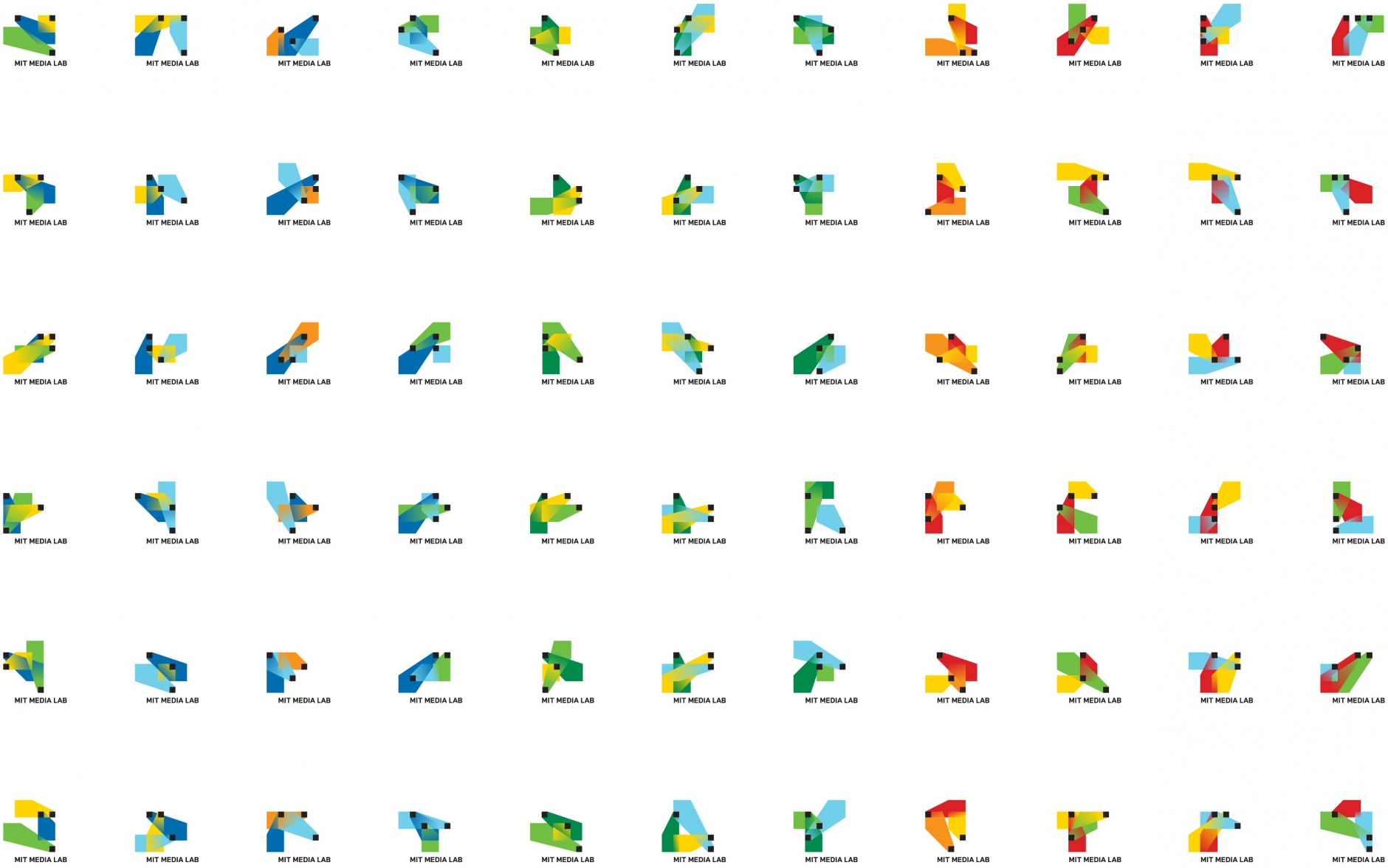


Anything that has a numerical value
can be assigned a random value.

Randomization

MIT Media Lab Logo




MIT MEDIA LAB

Randomization

City of Melbourne Logo

Randomization

Tiny Artists

Randomization

function

random(100);

// Gives a float between 0 and 100

random(50, 100);

// Gives a float between 50 and 100

round(random(100));

// If you want a int rather than a float (which returns a decimal point) you can use round(), ceil() or floor() to round it to the closest, next or previous whole number.

Randomization

Arithmetic Functions

ceil() rounds a decimal point up to a full number

floor() rounds a decimal point down to the full number

round() rounds a decimal point to the closest full number

min() finds the smallest number in a series

max() finds the largest number in a series

Randomization

Arithmetic Functions

ceil()

```
var w = ceil(2.0); // Assign 2 to w
var x = ceil(2.1); // Assign 3 to x
var y = ceil(2.5); // Assign 3 to y
var z = ceil(2.9); // Assign 3 to z
```

floor()

```
var w = floor(2.0); // Assign 2 to w
var x = floor(2.1); // Assign 2 to x
var y = floor (2.5); // Assign 2 to y
var z = floor (2.9); // Assign 2 to z
```

round()

```
var w = round(2.0); // Assign 2 to w
var x = round (2.1); // Assign 2 to x
var y = round (2.5); // Assign 3 to y
var z = round (2.9); // Assign 3 to z
```

min()

```
var u = min(5, 9); // Assign 5 to u
var v = min(-4, -12, -9); // Assign -12 to v
var w = min(12.3, 230.24); // Assign 12.3 to w
```

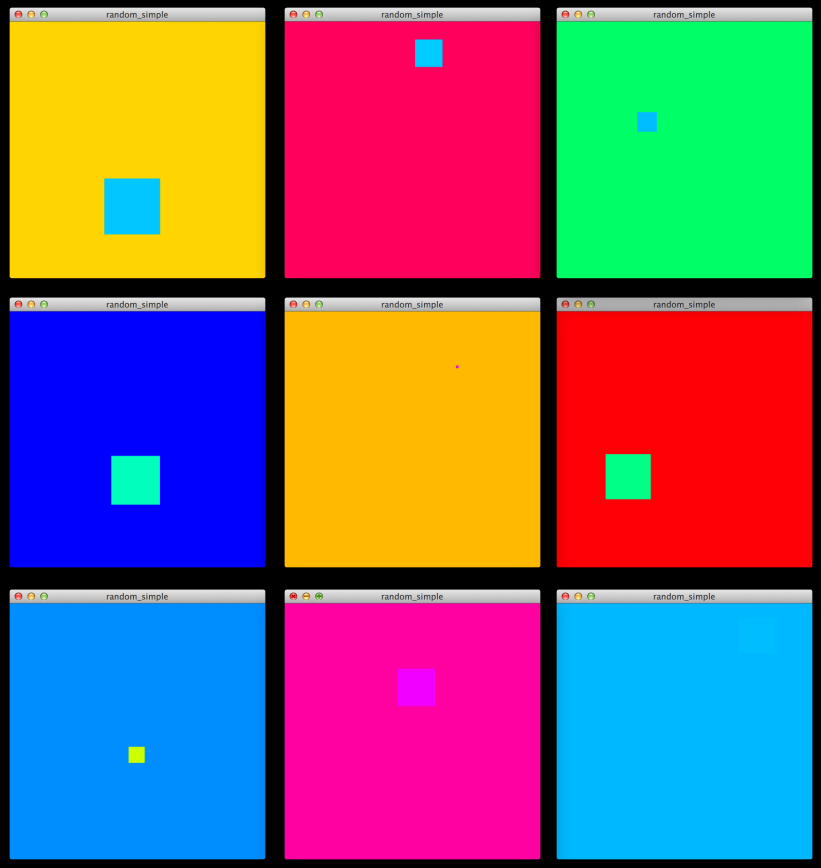
max()

```
var u = max(5, 9); // Assign 9 to u
var v = max(-4, -12, -9); // Assign -4 to v
var w = max(12.3, 230.24); // Assign 230.24 to w
```

Randomization

random(max);

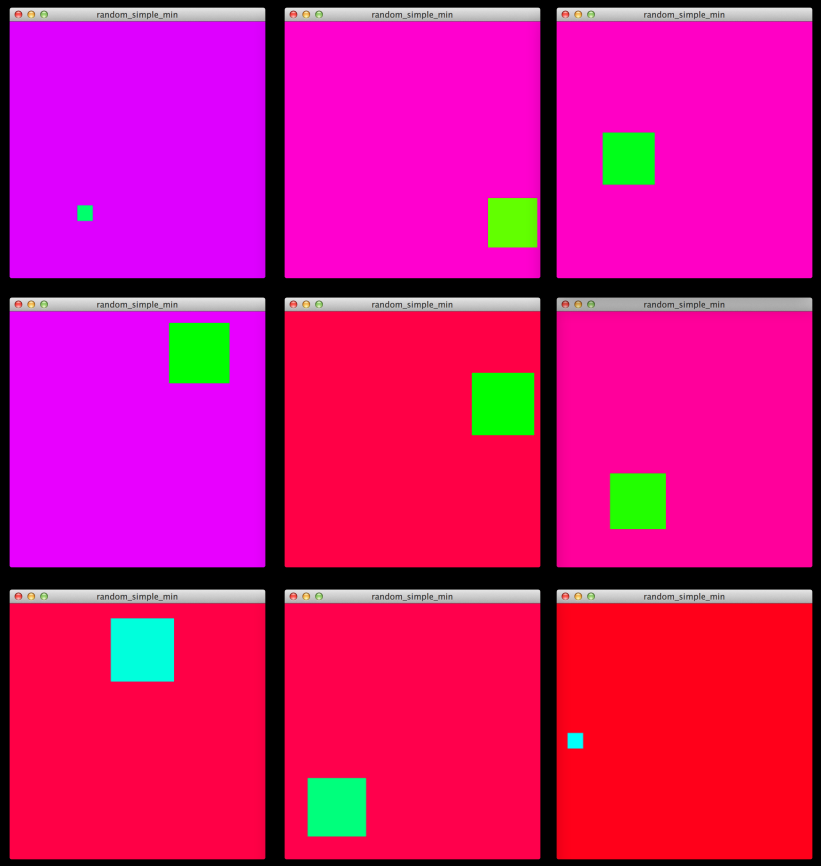
```
var rectSize = random(100);

function setup() {
  createCanvas(400, 400);
  colorMode(HSB, 360, 100, 100);
  var x = random(width - rectSize);
  var y = random(height - rectSize);
  background(random(360), 100, 100);

  noStroke();
  fill(random(360), 100, 100);
  rect(x, y, rectSize, rectSize);
}
```



Randomization

random(min, max);

```
var rectSize = random(20, 100);

function setup() {
  createCanvas(400, 400);
  colorMode(HSB, 360, 100, 100);
  var x = random(width - rectSize);
  var y = random(height - rectSize);
  background(random(270, 360), 100, 100);

  noStroke();
  fill(random(90, 180), 100, 100);
  rect(x, y, rectSize, rectSize);
}
```



Randomization

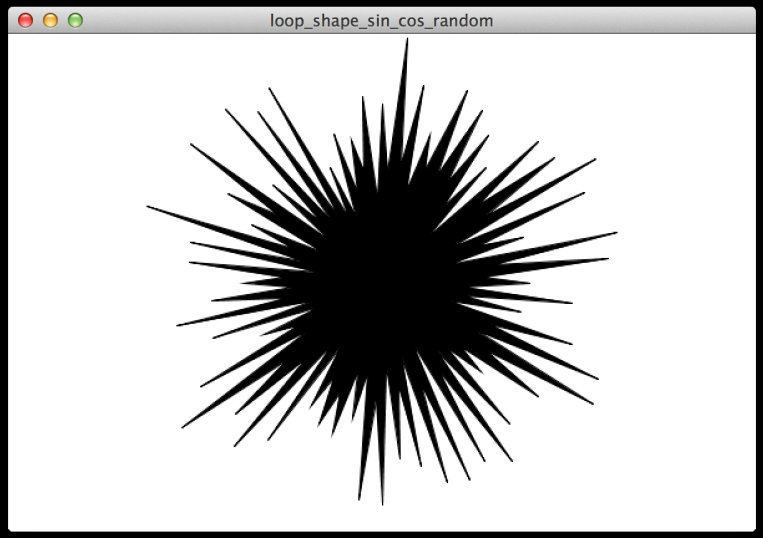
Color Theory

```
var rectSize = 130;

function setup(){
  var origColor = random(180, 360);
  createCanvas(600, 400);
  rectMode(CENTER);
  background(255);
  colorMode(HSB, 360, 100, 100);
  noStroke();
  fill(origColor, 80, 80);
  rect(width/2 - rectSize/2, height/2, rectSize, rectSize);
  fill((origColor + 180) % 360, 80, 80);
  rect(width/2 + rectSize/2, height/2, rectSize, rectSize);
}
```

Repetition

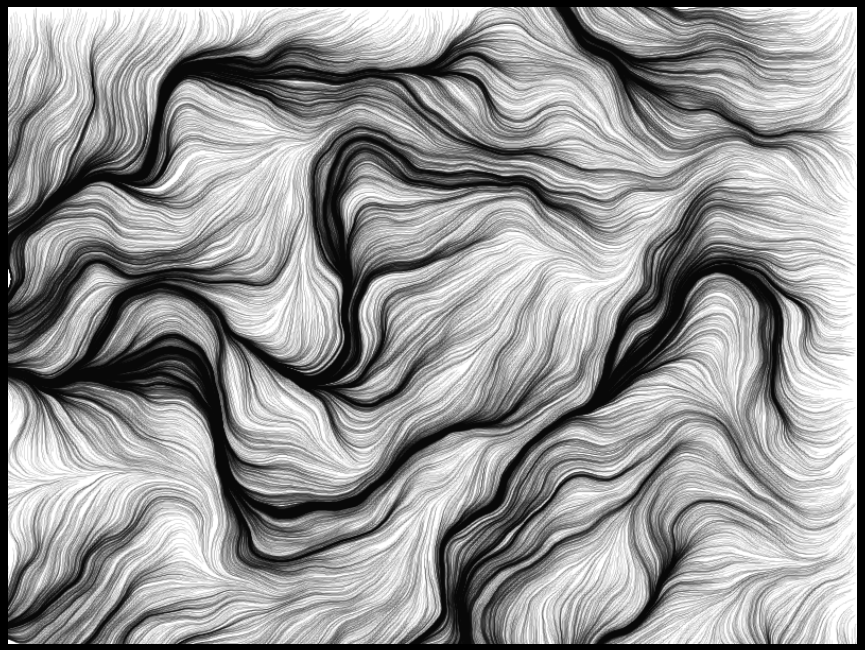
Cos and Sin Shapes Random

```
var points = 120;
var r = 100;

function setup() {
  createCanvas(600, 400);
  background(255);
  translate(width/2, height/2);
  fill(0);

  beginShape();
  for (var i = 0; i < points; i++) {
    var randomValue;
    if (i % 2 == 1) randomValue = -random(0, d/2);
    else randomValue = random(0, d);
    var vertexX = sin(radians(i * (360/points))) * (r + randomValue);
    var vertexY = cos(radians(i * (360/points))) * (r + randomValue);
    vertex(vertexX, vertexY);
  }
  endShape();
}
```



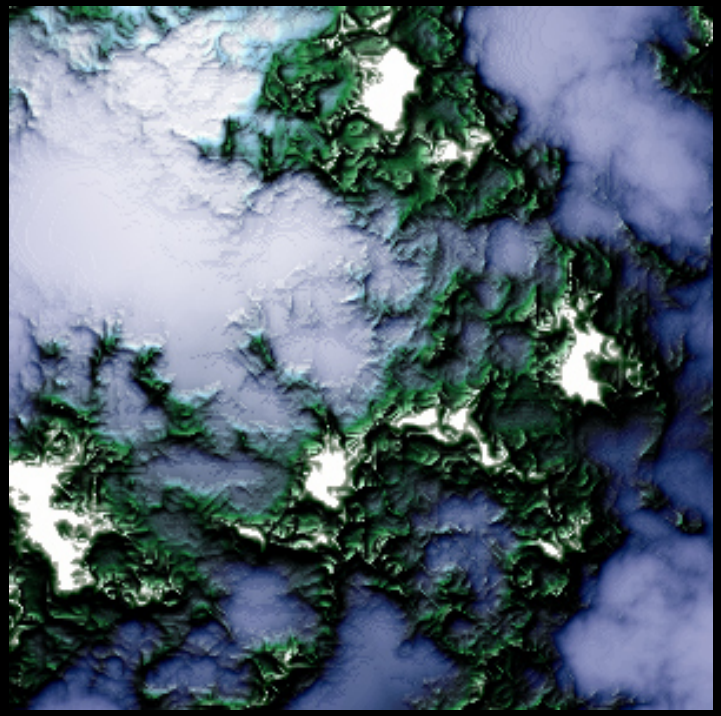
Randomization

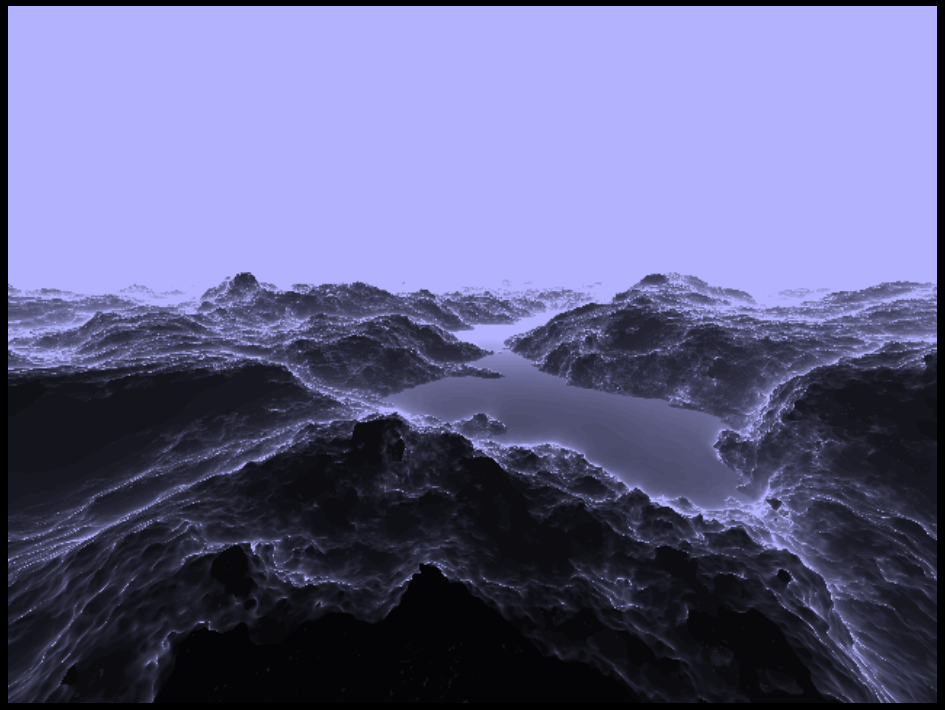
Perlin Noise



Randomization

Perlin Noise



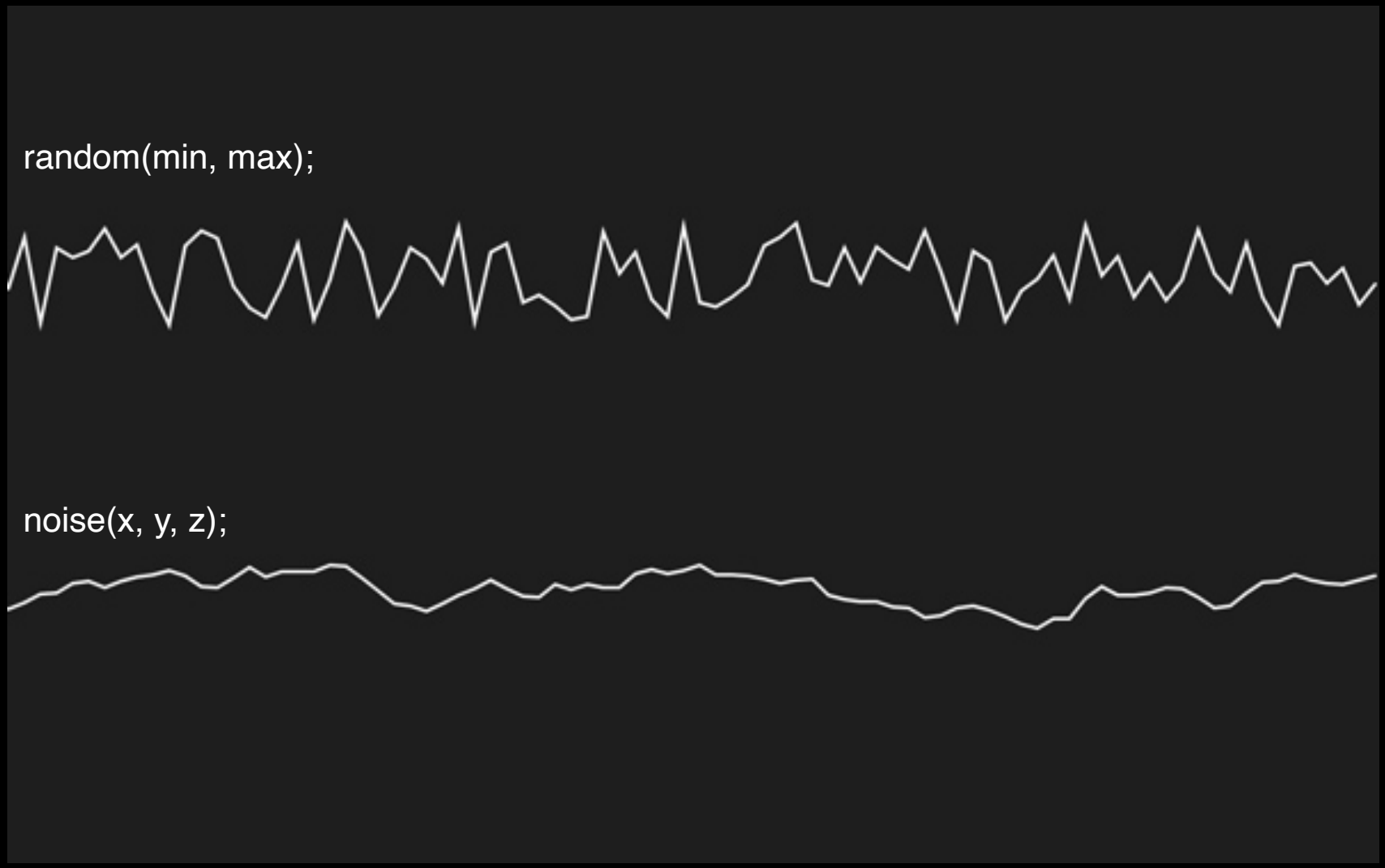


Randomization

Perlin Noise

Randomization

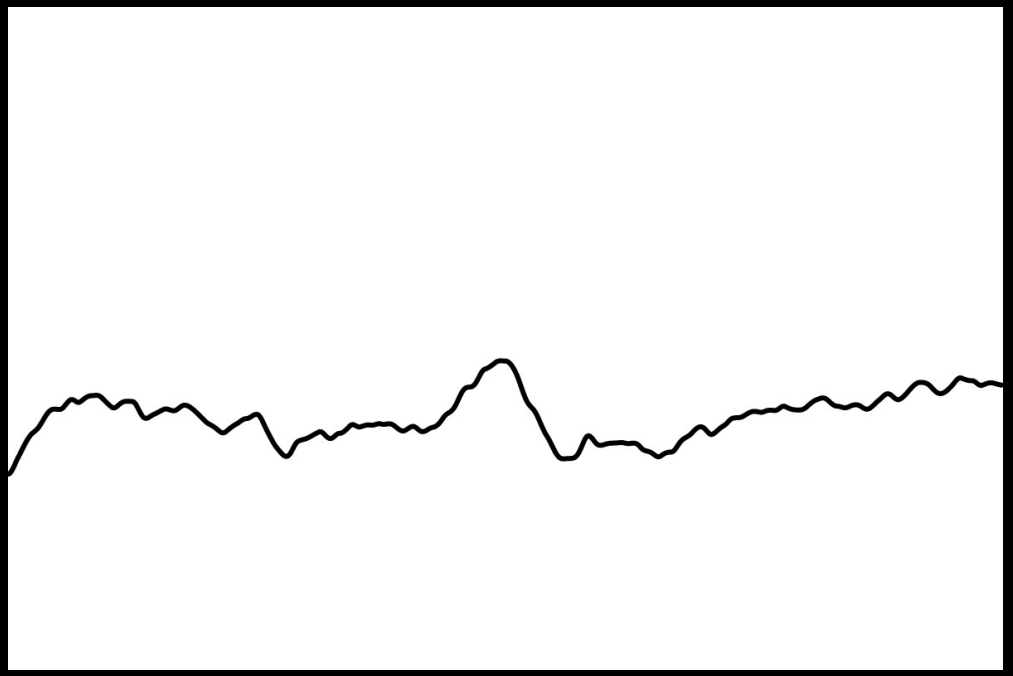
Perlin Noise



Randomization

Perlin Noise

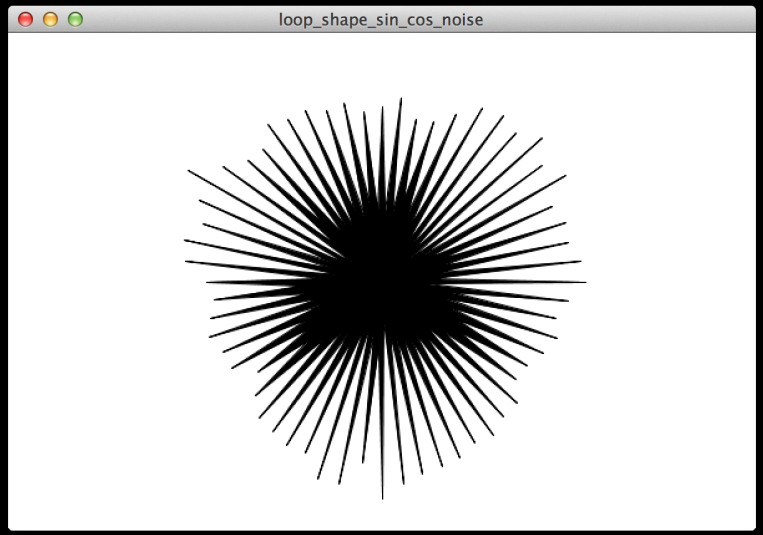
```
function setup() {
  createCanvas(600, 400);
  background(255);
  stroke(0);
  noFill();
  strokeWeight(3);

  translate(0, height/2);
  beginShape();
  var noiseCount = 0;
  for(var i = 0; i < width; i += 1) {
    var ranY = noise(noiseCount);
    vertex(i, ranY * 100);
    noiseCount += 0.02;
  }
  endShape();
}
```



Randomization

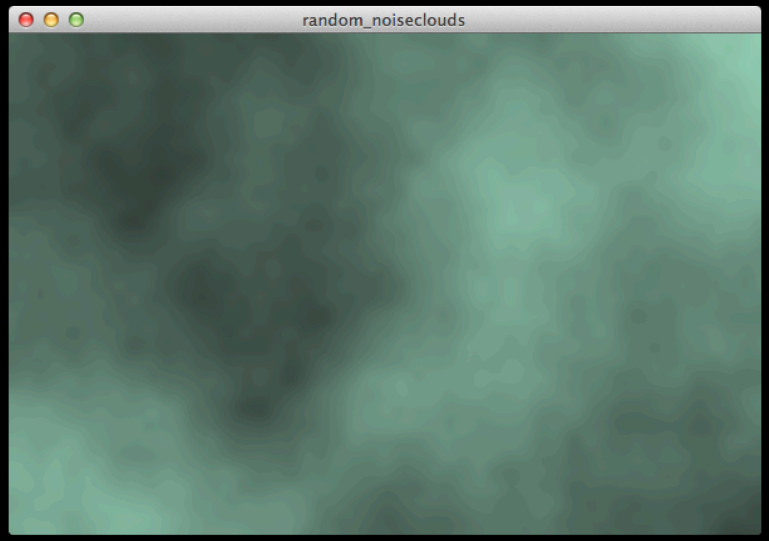
Cos and Sin Shapes Noise

```
var points = 120;
var d = 100;
function setup() {
  createCanvas(600, 400);
  background(255);
  translate(width/2, height/2);
  fill(0);
  beginShape();
  var noiseCount = 0;
  for (var i = 0; i < points; i++) {
    var randomValue;
    if (i % 2 == 1) randomValue = -noise(noiseCount);
    else randomValue = noise(noiseCount);
    var vertexX = sin(radians(i * (360/points))) * (d + (randomValue*100));
    var vertexY = cos(radians(i * (360/points))) * (d + (randomValue*100));
    vertex(vertexX, vertexY);
    noiseCount += 0.2;
  }
  endShape();
}
```



Randomization

Perlin Noise Clouds




```
var noiseVal;
var noiseScale=0.005;
function setup() {
  createCanvas(600, 400);
  colorMode(HSB, 360, 100, 100);
  // noprotect
  for (var y = 0; y < height; y++) {
    for (var x = 0; x < width; x++) {
      noiseDetail(10, 0.5);
      noiseVal = noise((x) * noiseScale, (y) * noiseScale);
      stroke(150, 30, noiseVal*100);
      point(x,y);
    }
  }
}
```

In Class

Six white geometric figures (outlines) superimposed on a black wall.

Use for loops and if statements. Extra credit, add randomization.

Homework

1. Suggested Reading: Pages 43 – 67 and 127 – 130 in Processing, by Casey Reas and Ben Fry
2. Use p5js to design a pattern for wrapping paper. You must use a for loop, if statements and either the random or noise function.
3. Put your p5js sketch in the dropbox folder before our next class and be prepared to talk about it.
4. EXTRA CREDIT: Go to DIA:Beacon to see the art of Sol Lewitt.

Creative Coding

Professor Danne Woo

dwoo@qc.cuny.edu

creativecode.dannewoo.com