p5.js

program structure

```
//runs once when program starts
function setup(){
  createCanvas(800,600);
}

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

system variables

windowWidth / windowHeight
width / height of window

width / height
width / height of canvas

mouseX / mouseY
current horizontal / vertical
mouse position

non-visual feedback
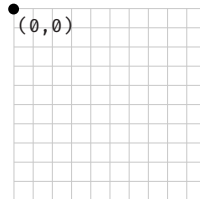
print()
report data to the output console

color

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
color(0, 0, 255**)** p5.Color object

2d primitives

line(x1, y1, x2, y2**)**

ellipse(x1, y1, width, height**)**

rect(x1, y1, width, height**)**

arc(x1, y1, width, height, start, stop**)**

beginShape();
  vertex(x1, y1**);**
  vertex(x2, y2**);**
  vertex(x3, y3**);**
  //add more vertex
endShape(CLOSE**);**

text("string", x, y, boxwidth, boxheight**)**

grid system
(0,0)

line()
(x1,y1)
(x2,y2)

ellipse()
(x1,y1)
height
width

rect()
(x1,y1)
height
width

arc()
height
(x1,y1)
start
stop
width
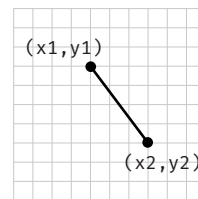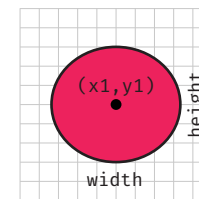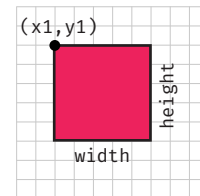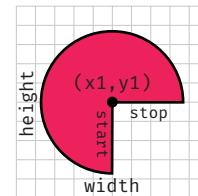
vertex()
(x1,y1)
(x2,y2)
(x3,y3)
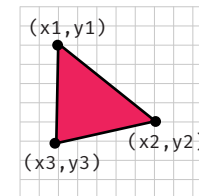
math

+ - / *

random(low,high**)**

map(value, in1, in2, out1, out2**)**
map a value from input to output range

attributes

background(color**)**
set the background color

fill(color**)**
set the fill color

noFill()
disables fill

stroke(color**)**
set the stroke color

strokeWeight(weight**)**
set the stroke's width

noStroke()
disables stroke

ellipseMode(MODE**)**
rectMode(MODE**)**
CENTER,CORNER

textSize(pixels**)**

if/then logic

```
if(test){
  statements
}
```

test
false
true
statements
continue
program
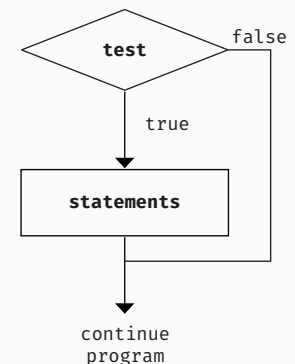
== equal to
!= not equal
> greater than
< less than
>= greater than or equal
<= less than or equal