# p5.js

a cheat sheet for visual artists

## program structure

```
// runs once before setup
function preload(){
  // img = loadImage(imageFilePath);
  // snd = loadSound(soundFilePath);
}

// runs once when program starts or after preload
function setup(){
  // createCanvas(width, height);
  // initializations
}

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

## system variables

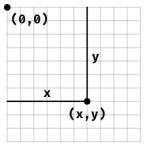
```
windowWidth / windowHeight
// width / height of inner window
width / height
// width / height of canvas
frameCount
// number of frames since program started
mouseX / mouseY
// horizontal / vertical mouse position
key
// most recent character typed
keyCode
// most recent key pressed: BACKSPACE, DELETE,
// ENTER, TAB, ESCAPE, SHIFT, CONTROL, OPTION, ALT,
// UP_ARROW, DOWN_ARROW, LEFT_ARROW, RIGHT_ARROW
```

## system events

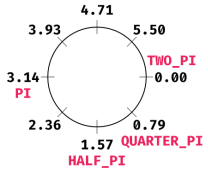
```
// a key is pressed
function keyPressed(){
}
// mouse moves and a button is not pressed
function mouseMoved(){
}
// a mouse button is pressed
function mousePressed(){
}
// mouse button pressed and then released
function mouseClicked(){
}
// window changes size
function windowResized(){
  // resizeCanvas(windowWidth, windowHeight);
}
```

## 2d space

grid system




angles in radians

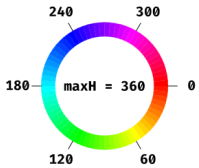


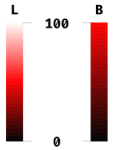

```
push(); // save current styles and transformations
  translate(xDisplace, yDisplace);
  scale(xFactor, yFactor);
  rotate(angle);
  // drawing instructions
pop(); // restore styles and transformations

angleMode(DEGREES);  //deafult: RADIANS
```

## color

```
background(color);  // fill the background

fill(color);  // set the fill color
noFill();  // disables fill
stroke(color);  // set the stroke color
noStroke();  // disables stroke

fill('red');  // color string
fill('#222222');  // 6-digit hex fill
fill(gray);
fill(gray, alpha);
fill(red, green, blue);
fill(red, green, blue, alpha);

colorMode(HSL, maxH, maxS, maxL, maxA);
fill(hue, saturation, lightness, alpha);

colorMode(HSB, maxH, maxS, maxB, maxA);
fill(hue, saturation, brightness, alpha);

colorMode(RGB, 255);  // restore default color mode

colorVariable = color(120, 50, 90);
```

hue, lightness, brightness



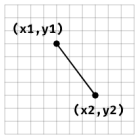

## shapes

```
ellipseMode(CORNER);  // default: CENTER
rectMode(CENTER); // default: CORNER

strokeWeight(weight);
// set the stroke's width in pixels

line(x1, y1, x2, y2);

ellipse(x, y, width, height);

rect(x, y, width, height);

arc(x, y, width, height, start, stop);

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

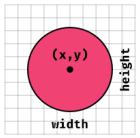
## typography

```
font = loadFont(filePath);
// loads TTF or OTF font

textFont(font, size);
// set the font and size

textSize(size);
// set only size

textAlign(horizontal, vertical);
// horizontal: LEFT (default), CENTER or RIGHT
// vertical: TOP, BOTTOM, CENTER or BASELINE (default)

text("string", x, y);
text("multiline string", x, y, boxwidth, boxheight);
```

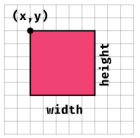
line()



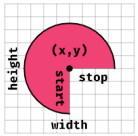

ellipse()



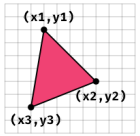

rect()



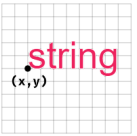

arc()




vertex()




text()




## non-visual feedback

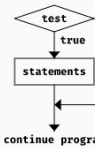
```
print("string: " + variable);
// report data to the output console

// double slash to comment code (program skip
```

## math

```
+ - / *  //basic math operators

random(low, high); // ranged random number
noise(x, y, z); //Perlin noise between 0 and

round(value); //nearest integer

map(value, in1, in2, out1, out2);
//map a value from input range to output rang

dist(x1, y1, x2, y2);
//calculate distance between 2 points
```

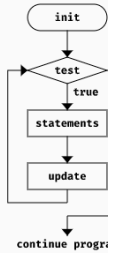
## if/then logic

```
if ( test ) {
  statements
}

==  //equal to
!=  //not equal
>   //greater than
<   //less than
>=  //greater than or equal
<=  //less than or equal

&&  //both operands true
||  //either operand true
!   //reverses operand
```




## for loop logic

```
for ( init; test; update ) {
  statements
}

//example
for (var i = 0; i < 10; i++) {
  print(i);
}
```




Compiled by Antonio Belluscio codesthesia.net CC
Based on Ben Moren's p5