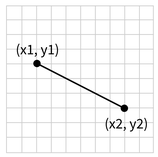
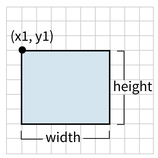
# p5.js

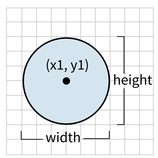
## 2D Primitives



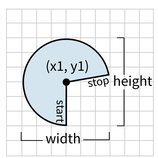

**line(**x1, y1, x2, y2**)**



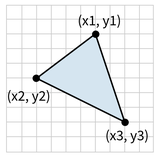

**rect(**x1, y1, width, height**)**




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

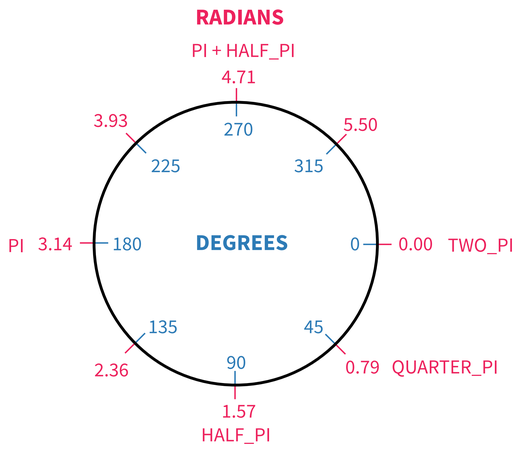



**arc(**x1, y1, width, height, start, stop**)**




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

**text(***string, x, y, x2, y2***)**




## Attributes

**strokeWeight(**weight**)**
set the width of the stroke

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

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

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

**noFill()**
disables fill

**noStroke()**
disables stroke

**ellipseMode(***MODE***)**
CENTER, RADIUS, CORNER, CORNERS

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

**textAlign(***h, v***)**
h: LEFT, CENTER, RIGHT
v: TOP, BOTTOM, CENTER, BASELINE

**textSize(***n***)**

## Colours

fill(120) gray: 0 - 255

fill(255, 0, 0) r, g, b: 0 - 255

fill(255, 0, 0, 50) r, g, b, alpha (0 - 100)

fill('red') color string

fill('#ccc') 3-digit hex

fill('#222222') 6-digit hex

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

## Environment

**print(***string***)**
= console.log()

**cursor(***TYPE***)**
ARROW, CROSS, HAND, MOVE, TEXT

**framerate(**fps**)**
change frame per second

**windowResized()**
called when window resized

## Rendering

**createCanvas(**w, h**)**

**resizeCanvas(**w, h**)**

**createGraphics(**w, h**)**
return new p5.Renderer object
off-screen graphics buffer

## Events

**keyPressed()**

**keyReleased()**

**mouseMoved()**

**mousePressed()**

**mouseClicked()**

## Image

**saveCanvas(***canvas, filename, ext***)**
filename: string
ext: "jpg" or "png"

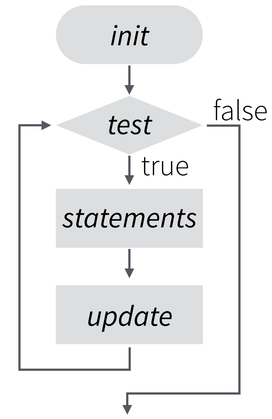
## Structure

function **preload()**{ run once before setup(): asynchronous function
//load data
}
function **setup()**{ run once when the program starts
**createCanvas(**width, height**);** create a canvas, size in pixels
**noLoop()**; stops continuously executing draw()
}
function **draw()**{ run continuously after setup()
//draw
}

**redraw()**
executes the code in draw() once

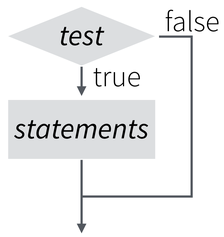
**for loop**

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




**if statement**

```
if (test) {
  statement
}
```



## System Variables

**frameCount**
number of frames since program started

**focused**
confirms if the window is focused

**displayWidth / displayHeight**
width / height of entire screen display

**windowWidth / windowHeight**
width / height of window

**width / height**
width / height of canvas

**key**
most recent key typed

**keyCode**
special keys: BACKSPACE, DELETE, ENTER, RETURN, TAB, ESCAPE, SHIFT, CONTROL, OPTION, ALT, UP_ARROW, DOWN_ARROW, LEFT_ARROW, RIGHT_ARROW

**mouseX / mouseY**
current horizontal / vertical mouse position

**pmouseX / pmouseY**
horizontal / vertical mouse position in the previous frame

**mouseButton**
LEFT, RIGHT, or CENTER

## String

**trim(***string***)**
remove white spaces

**split(***string, delim***)**
split a string into pieces by delimiters

## IO

**loadFont()**
load .otf or ..ttf file in preload()

**loadJSON(**path, [callback], [errorcallback]**)**
loading JSON file

**loadString(**path, [callback], [errorcallback]**)**
loading a text file

**loadTable(**path, [options], [callback**)**
options: “header”, “csv”, “tsv”
returns p5.Table object

## Math

**dist(***x1, y1, x2, y2***)**
calculate distance between 2 points

**map(***v, input1, input2, output1, output2***)**
mapping the value from input to output range

**max()**

**min()**

**round()**

**floor()**

**ceil()**

**degrees(***radians***)**

**radians(***degrees***)**

**random()**

## Conversion

**float()**

**int()**

**str()**

**boolean()**

**hex()**

## p5.dom

**select(***name***)**
select HTML element by ID(“#”) or class(“.”)

**createDiv(***html***)**

**createP(***html***)**

**createSpan(***html***)**

**createImg(***src, [alt]***)**

**createA(***href, html, [targe]***)**

**createSlider(***min, max, [value], [step]***)**

**createButton(***label, [value]***)**

**createCheckbox(***[label], [value]***)**

**createSelect()**
option(*string*) to add item

**createRadio(***[divID]***)**

**createInput(***[value], [type]***)**
type: “text”, “password”

**createFileInput(***[callback], [multiple]***)**

**parent(***parent***)**

**id(***[id]***)**

**class(***[class]***)**

**changed(***function***)**

**input(***function***)**

**mousePressed(***function***)**

## note

## useful links

thecodingtrain.com by daniel shiffman
p5js.org
github.com/processing/p5.js/wiki/Instantiation-Cases