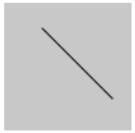
line()

Examples



edit reset copy

```
line(30, 20, 85, 75);
```

edit reset copy

```
line(30, 20, 85, 20);
stroke(126);
line(85, 20, 85, 75);
stroke(255);
line(85, 75, 30, 75);
```

Description

Draws a line (a direct path between two points) to the screen. If called with only 4 parameters, it will draw a line in 2D with a default width of 1 pixel. This width can be modified by using the `strokeWeight()` function. A line cannot be filled, therefore the `fill()` function will not affect the color of a line. So to color a line, use the `stroke()` function.

Syntax

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

```
line(x1, y1, z1, x2, y2, z2)
```

Parameters

x1	Number: the x-coordinate of the first point
y1	Number: the y-coordinate of the first point
x2	Number: the x-coordinate of the second point
y2	Number: the y-coordinate of the second point
z1	Number: the z-coordinate of the first point
z2	Number: the z-coordinate of the second point

point()

Examples

```
point(30, 20);
point(85, 20);
point(85, 75);
point(30, 75);
```

```
point(30, 20);
point(85, 20);
stroke('purple'); // Change th
strokeWeight(10); // Make the
in size
point(85, 75);
point(30, 75);
```

```
let a = createVector(10, 10);
point(a);
let b = createVector(10, 20);
point(b);
point(createVector(20, 10));
point(createVector(20, 20));
```

Description

Draws a point, a coordinate in space at the dimensi
first parameter is the horizontal value for the point,
vertical value for the point. The color of the point i
`stroke()` function. The size of the point can be ch
`strokeWeight()` function.

Syntax

```
point(x, y, [z])
```

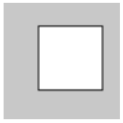
```
point(coordinate_vector)
```

Parameters

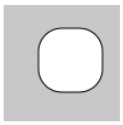
x	Number: the x-coordinate
y	Number: the y-coordinate

rect()

Examples



```
// Draw a rectangle at location (
width and height of 55.
rect(30, 20, 55, 55);
```



```
// Draw a rectangle with rounded
having a radius of 20.
rect(30, 20, 55, 55, 20);
```

```
// Draw a rectangle with rounded
the following radii:
// top-left = 20, top-right = 15,
= 10, bottom-left = 5.
rect(30, 20, 55, 55, 20, 15, 10,
```

Description

Draws a rectangle on the canvas. A rectangle is a four-s
with every angle at ninety degrees. By default, the first
the location of the upper-left corner, the third sets the w
sets the height. The way these parameters are interprete
with the `rectMode()` function.

The fifth, sixth, seventh and eighth parameters, if speci
corner radius for the top-left, top-right, lower-right and
respectively. An omitted corner radius parameter is set
previously specified radius value in the parameter list.

Syntax

```
rect(x, y, w, [h], [tl], [tr], [br], [bl])
```

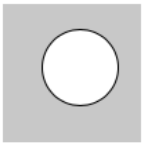
```
rect(x, y, w, h, [detailX], [detailY])
```

Parameters

x	Number: x-coordinate of the rectan
y	Number: y-coordinate of the rectan
w	Number: width of the rectangle.
h	Number: height of the rectangle. (C

ellipse()

Examples



```
ellipse(56, 46, 55, 55);
```

Description

Draws an ellipse (oval) to the screen. By def
set the location of the center of the ellipse, a
parameters set the shape's width and height.
value of width is used for both the width and
width is specified, the absolute value is taker

An ellipse with equal width and height is a c
changed with the `ellipseMode()` function.

Syntax

```
ellipse(x, y, w, [h])
```

```
ellipse(x, y, w, h, [detail])
```

Parameters

x	Number: x-coordinate o
y	Number: y-coordinate o
w	Number: width of the el
h	Number: height of the el
detail	Integer: optional parame This is to specify the nu the perimeter of the ellip draw a stroke for a detai