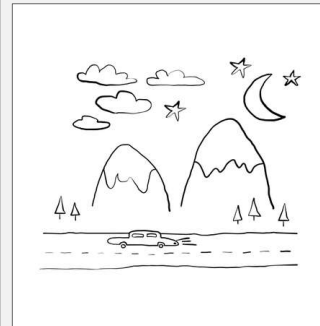


Getting Started With Processing

Ingredients



- what *shape* to draw
- *where* to draw the shape
- *how big* to make the shape
- what *color* to make the shape
 - fill
 - stroke

Ingredients

- what *shape* to draw
 - point, line, ellipse, rectangle



Basic Processing

```

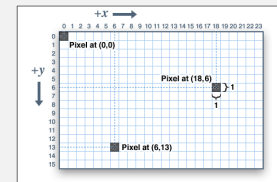
• shape modes
rectMode(CORNER); // interpret values as left,top,width,height
ellipseMode(CORNER); // interpret values as left,top,width,height
rectMode(CENTER); // interpret values as centerx,centery,width,height
ellipseMode(CENTER); // interpret values as centerx,centery,width,height

• draw shapes
line(x1,y1,x2,y2); // draw a line from (x1,y1) to (x2,y2)
point(x,y); // draw a point at (x,y)
rect(x,y,w,h); // draw rectangle (*)
ellipse(x,y,w,h); // draw ellipse (*)
    
```

(*) The meaning of (x,y) depends on the mode (CENTER or CORNER).

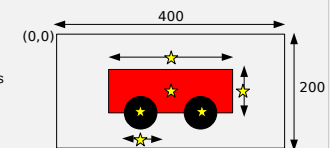
Ingredients

- *where* to draw the shape
 - (x,y) coordinates
 - (0,0) is the upper left corner of the drawing window
- *how big* to make the shape
 - units are in pixels



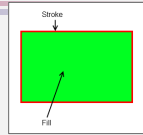
Figuring out position and size –

- identify the quantities you need to find
 - for CENTER mode, need centers of rect and ellipses
 - also need width and height for rect and ellipses
- label what you know
 - window width, height, upper left corner
- compute what you need from what you know
 - eyeball rect center and dimensions from window dimensions
 - eyeball ellipse dimensions from rect dimensions
 - compute ellipse centers from rect center, rect and ellipse dimensions



Ingredients

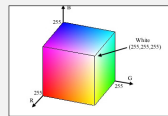
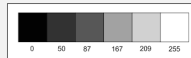
- what *color* to make the shape
 - fill
 - stroke



These set modes that affect subsequent drawing commands.

```

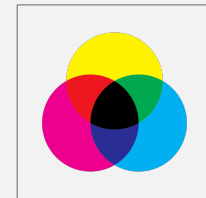
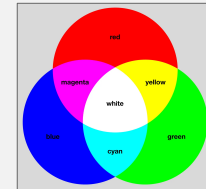
• set line-drawing color
stroke(gray); // grayscale; value is 0-255
stroke(r,g,b); // RGB color; values are 0-255
stroke(r,g,b,a); // RGB color with transparency (a);
                // values are 0-255
• set fill color
fill(gray); // grayscale; value is 0-255
fill(r,g,b); // RGB color; values are 0-255
fill(r,g,b,a); // RGB color with transparency (a);
                // values are 0-255
noFill(); // do not fill area
    
```



- grayscale value 0 (black) to 255 (white)
- RGB color – red, green, blue components with values 0 (off) to 255 (fully on)
- alpha – values 0 (fully transparent) to 255 (opaque)

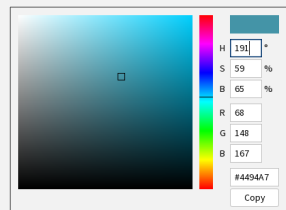
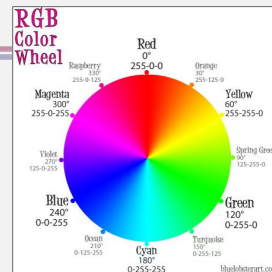
RGB Color

- RGB is an *additive* color model
- additive color mixing applies to light
 - for light, the color we see is the combined energy of the wavelengths present
 - all wavelengths combined makes white
- CMY(K) is a *subtractive* color model
- subtractive color mixing applies to pigments (paint, printer ink)
 - for pigments, some light energy is absorbed and some reflected – the color we see are the reflected wavelengths
 - all pigments combined makes black



RGB Color

- learn the 8 basic colors – red, green, blue, cyan, magenta, yellow, white, black
- to figure out other colors, experiment or use Processing's color selector (Tools → Color Selector...)



Learning Programming

- language *syntax* – what is legal to write down
- language *semantics* – what a statement means
 - (what the computer does when it is executed)
- *patterns* of usage – how to combine language elements to achieve a particular effect

Processing Terminology and Syntax

- programs are called *sketches*
- the *statement* is the basic unit (like a sentence)
 - each statement must end with a semicolon
- *case-sensitive*
 - `rectMode` and `rectmode` are not the same thing
 - avoid defining your own names which differ only in case
- *whitespace* conventions
 - use spaces, tabs, line breaks, blank lines to increase readability
- *comments* are ignored by the computer
 - `//` rest of the line is a comment
 - `/* multi-line comment (between delimiters) */`
 - used to provide explanation to the human reader
 - include your name and a brief description of the program in comments at the beginning of each sketch

1

Basic Processing

```
size(width,height); // open window with specified dimensions

• "other aspects" - set modes that affect subsequent drawing commands
- set line-drawing color
stroke(gray); // grayscale; value is 0-255
stroke(r,g,b); // RGB color; values are 0-255
stroke(r,g,b,a); // RGB color with transparency (a); // values are 0-255
- set fill color
fill(gray); // grayscale; value is 0-255
fill(r,g,b); // RGB color; values are 0-255
fill(r,g,b,a); // RGB color with transparency (a); // values are 0-255
noFill(); // do not fill area

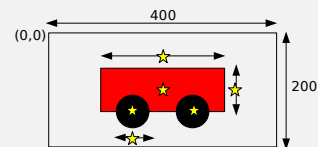
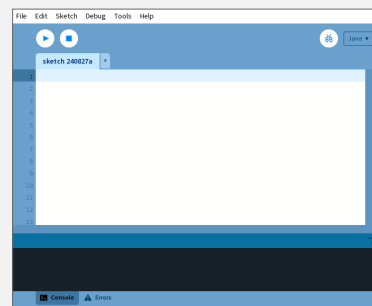
- shape modes
rectMode(CORNER); // interpret values as left,top,width,height
ellipseMode(CORNER); // interpret values as left,top,width,height
rectMode(CENTER); // interpret values as centerx,centery,width,height
ellipseMode(CENTER); // interpret values as centerx,centery,width,height

• drawing commands - draw something on the screen
- clear the background to the specified color
background(gray); // grayscale; value is 0-255
background(r,g,b); // RGB color; values are 0-255
background(r,g,b,a); // RGB color with transparency (a); // values are 0-255
- draw shapes
line(x1,y1,x2,y2); // draw a line from (x1,y1) to (x2,y2)
point(x,y); // draw a point at (x,y)
rect(x,y,w,h); // draw rectangle (*)
ellipse(x,y,w,h); // draw ellipse (*)

(*) The meaning of (x,y) depends on the mode (CENTER or CORNER).
```

CPSC 120: Principles of Computer Science • Fall 2024

12



CPSC 120: Principles of Computer Science • Fall 2024

13