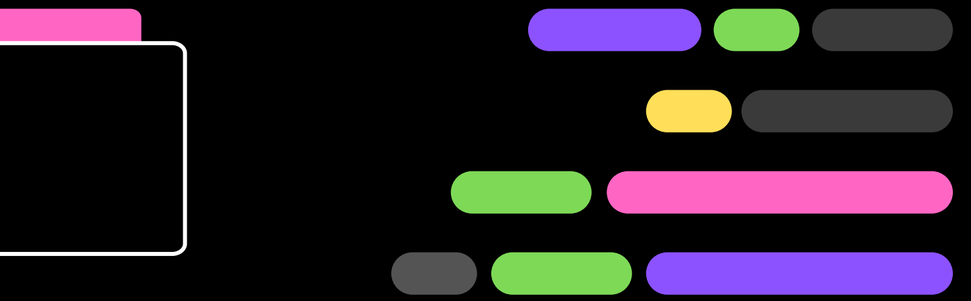
INTRODUCTION TO

{WEB DEVELOPMENT}

DAY THREE



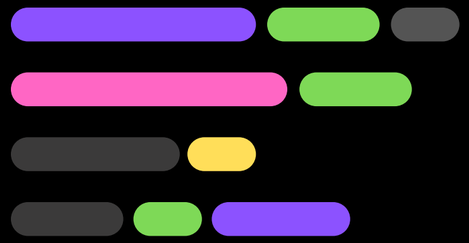
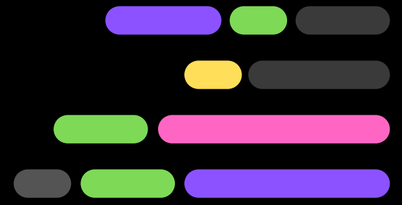
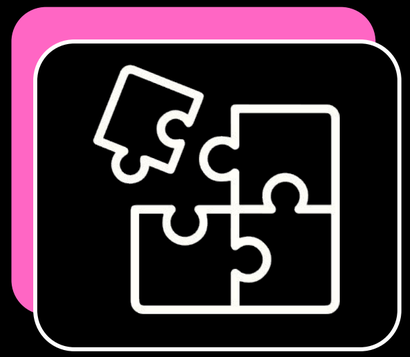


TABLE OF CONTENTS



IPO ALGORITHMS



Input

Receive user input.

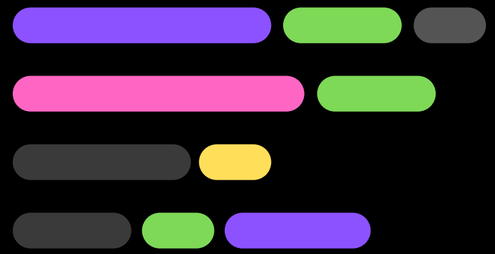
Process

Apply a process such as math, string concatenation or an if statement.

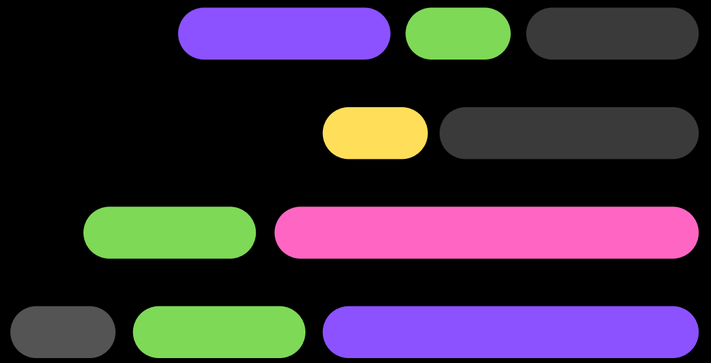
Output

Display the processed data on the page.

01 IF STATEMENTS



```
5   if (statement1 === true) {
6       // Run code
7   } else if (statement2 === true) {
8       // Run code
9   } else {
10      // Run code
11  }
12
13  // Run rest of program
```



COMPARISON & LOGICAL OPERATORS

===	is equal to	size === “small” or number === 3
!==	is not equal to	productStatus !== “sold-out” or number !== 8
>	is greater than	number > 2
<	is lesser than	number < 9
>=	is greater than or equal to	number >= 5
<=	is lesser than or equal to	number <= 23
&&	and	number >= 1 && number <= 10 (1-10)
||	or	number < 0 || number >= 10 (negative or 10+)

CODE ALONG -

ICE CREAM COST CALCULATOR

PRACTICE: NUMBER COMPARISON

Using the start code, create a program that compares the size of 2 numbers and outputs a conclusion. It should include:

- An If Statement
- Well formatted code
  - Comments (IPO Algorithm)
- A conclusion that outputs to the site

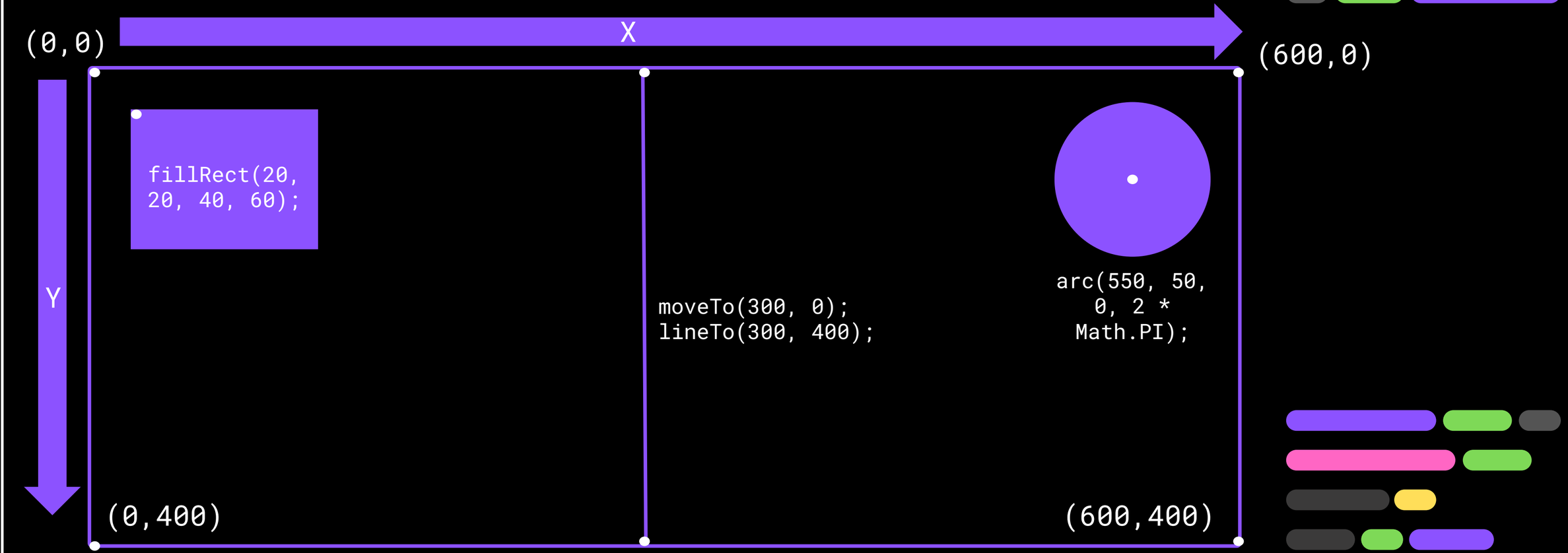
02

DRAWING & ANIMATION

```
// Canvas Set Up
let cnv = document.getElementById("my-cnv");
let ctx = cnv.getContext("2d");
cnv.height = 400;
cnv.width = 600;
```

```
// Canvas Set Up
// 1. Find the canvas element
// 2. Create a drawing object
// 3. Set the canvas height
// 4. Set the canvas width
```

COORDINATES



COLOUR

In JavaScript, colours can be referenced in multiple ways:

- RBG [ex rgb(138,43,226)]
- Hex Codes [ex #8A2BE2]
- Colour Name [ex “blueViolet”]
- HSL [ex hsl(271.15, 75.93%, 52.75%)]

The colour of a shape, line, or text can be defined using:

- fillStyle = colour
- strokeStyle = colour

DRAW FUNCTIONS

fillStyle / strokeStyle	Defines the fill / outline color of any stroke or fill function after it is defined. Is set to black by default
fillRect(x, y, width, height) / strokeRect(x, y, width, height)	Draws a filled in / outlined rectangle at the given coordinates with the given size
beginPath()	Declares a new path (Used for drawing lines, circles, and shapes other than rectangles)
moveTo(x, y)	Moves your “pen” to a given coordinate (does not draw a line)
lineTo(x, y)	Draws a line to a given coordinate (will not appear until stroke() is called)
stroke()	Draws the line / outline of the shape
fill()	Draws the filled in shape

All must be used with ctx.___
Ex:

```
ctx.fillStyle = “red”;
ctx.fillRect(10, 10, 50, 70);
```

Must be assigned a value.
Ex:

```
ctx.fillStyle = “purple”;
ctx.font = “sans Serif”;
```

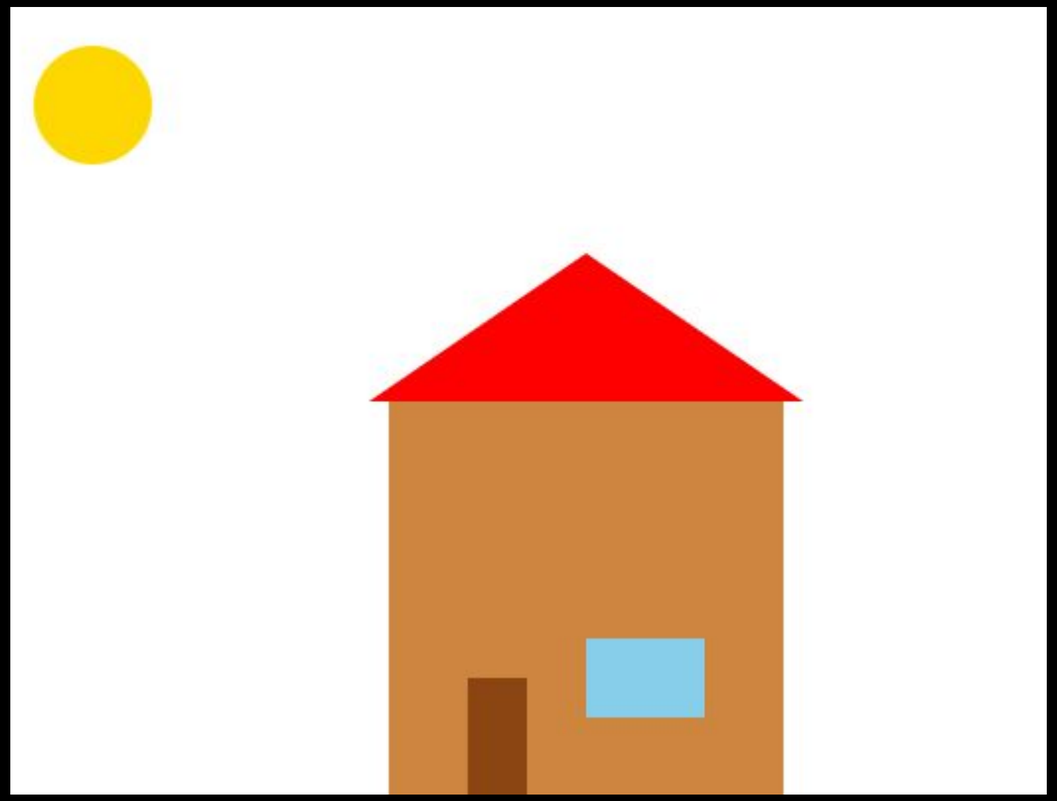
DRAW FUNCTIONS

closePath()	Makes a line to the starting point of the path
arc(x, y, radius, startAngle, endAngle) circle: arc(x, y, 0, 2 * Math.PI)	Draws an arc or circle with the given parameters, angles are in radians. (Does not draw until stroke() or fill() is called)
font	Defines the font for drawn text
fillText(text, x, y) / strokeText(text, x, y)	Draws filled in / outlined text
drawImage(image, x, y, width, height)	Draws an image from a given file path. Width and height are optional

All must be used with ctx.___
Ex:
```
ctx.fillStyle = “red”;
ctx.fillRect(10, 10, 50, 70);
```

Must be assigned a value.
Ex:
```
ctx.fillStyle = “purple”;
ctx.font = “sans Serif”;
```

PRACTICE - RECREATE AN IMAGE



Try your best to recreate the image, or create your own! The image should include:

- A circle
- Rectangles of varying size, dimensions, and colour
- A custom shape using lineTo() and closePath() (such as a triangle, octagon, rhombus, etc)

Feel free to reference the previous slides, google, or previous code.