

JAVASCRIPT

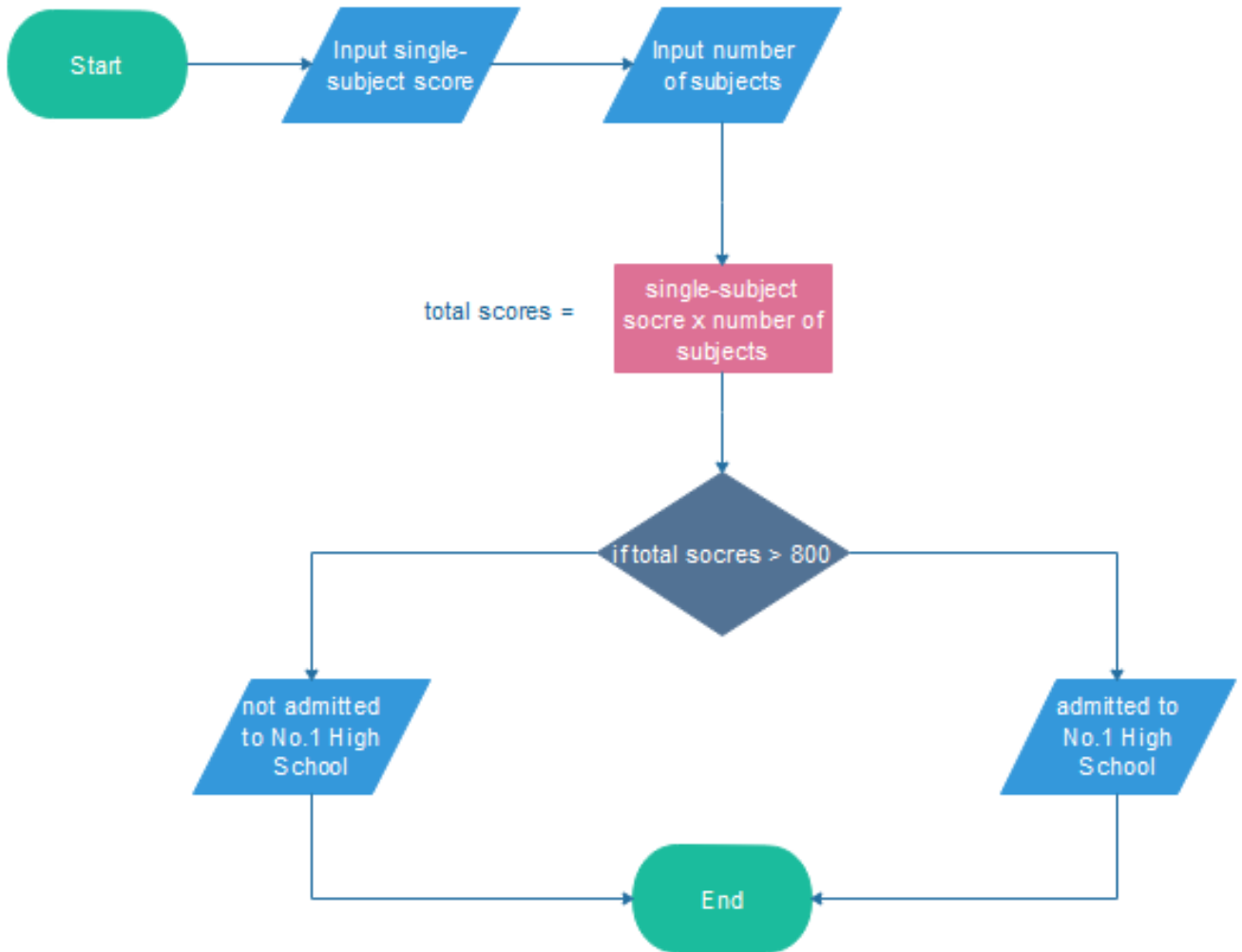
**PROGRAM DESIGN AND USEFUL
FUNCTIONS**

PROJECT: HANGMAN

- Learn a couple useful functions
- Watch game (Hangman.html)
- Flowchart main logic
- Figure out spacing for underlines and letters (as letters are guessed)
- Learn a few methods
- Figure out spacing for hangman
- Put it all together

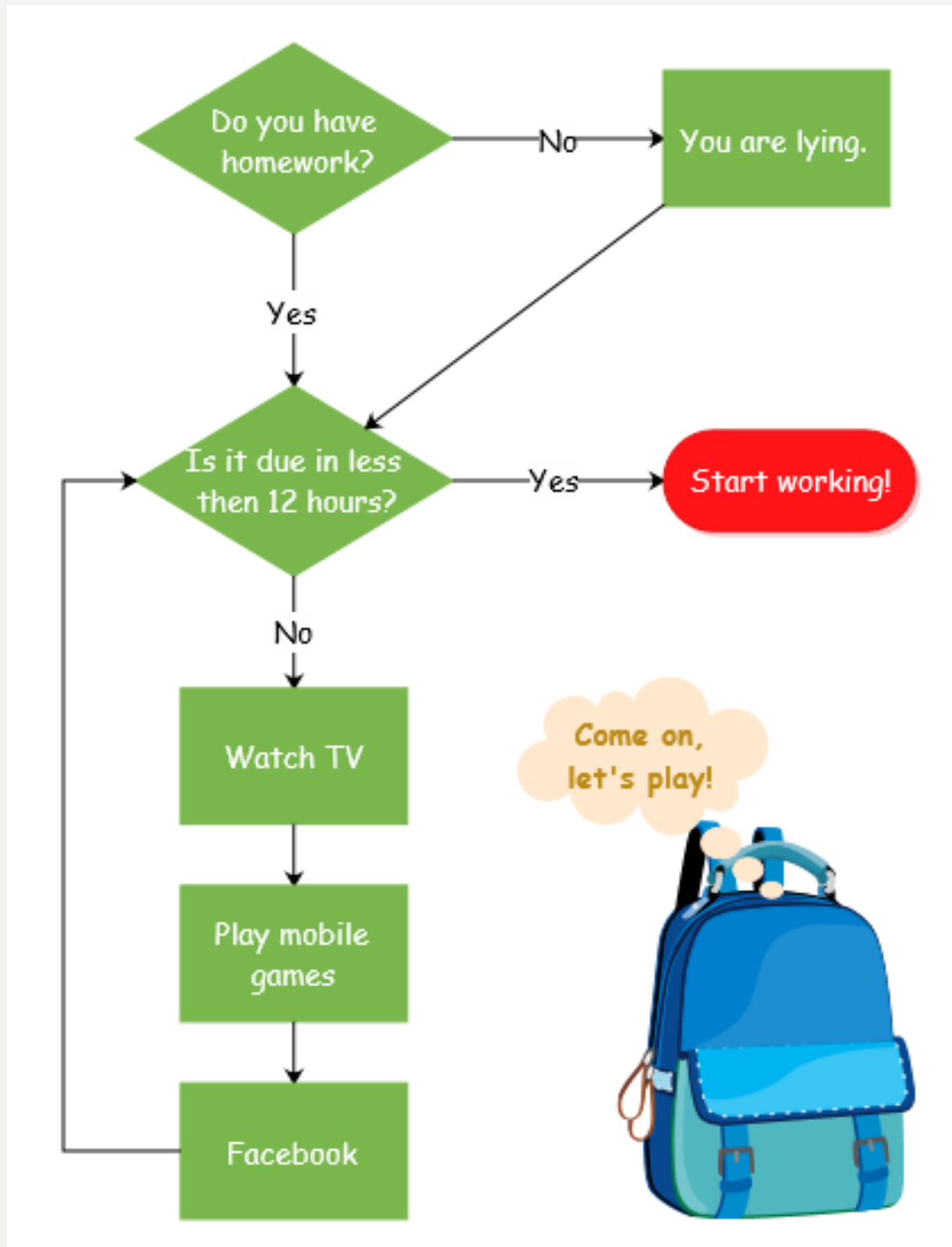
FLOWCHART EXAMPLE #1

Am I Admitted to Preferred High School?



FLOWCHART EXAMPLE #2

Should I do my homework now?



FLOWCHART SYMBOLS



Terminator. starting or ending point of the system.



Process. A box indicates some particular operation



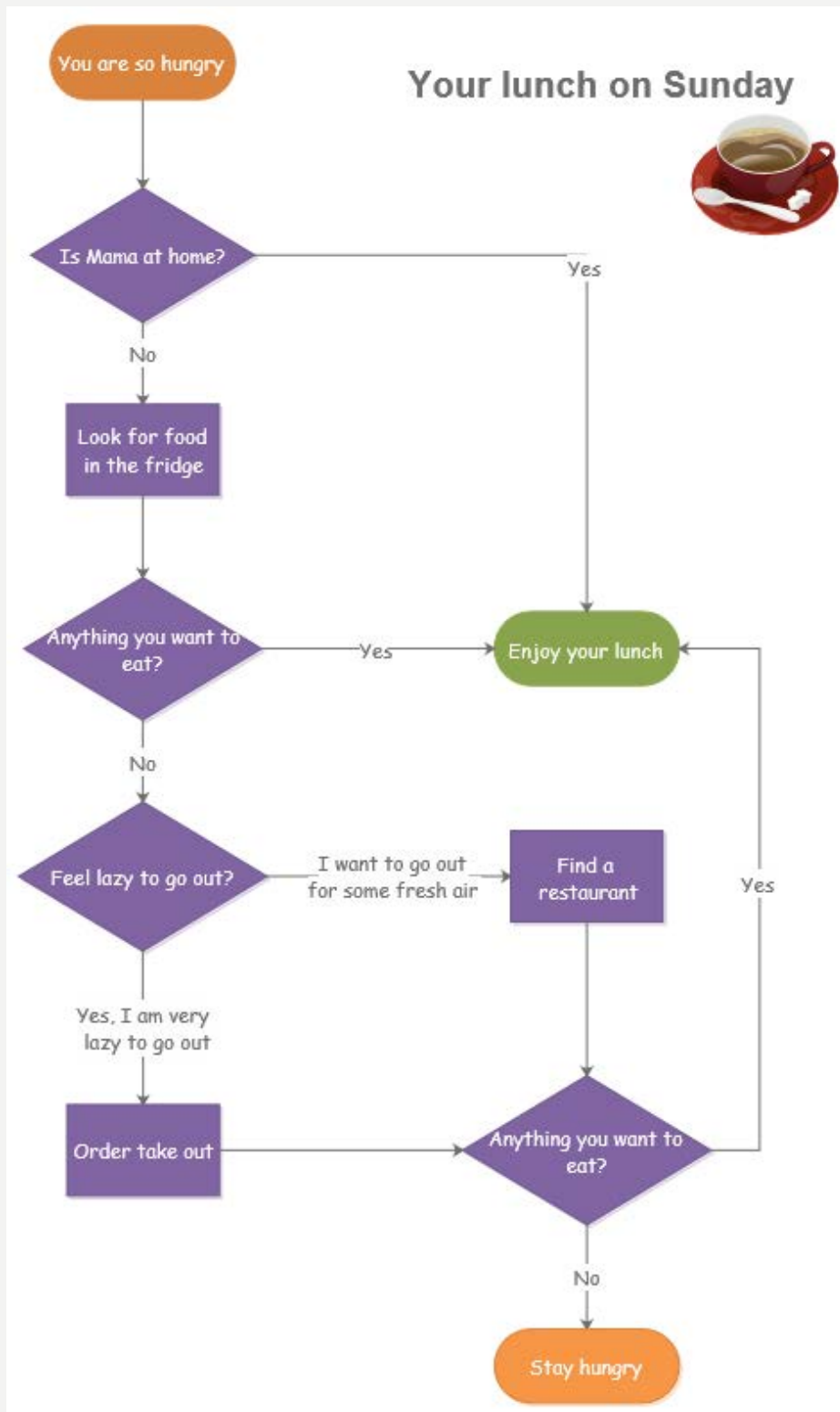
Input-Output. Get entry from user, display message. Could just use process for this, for simplicity.



Decision. A diamond represents a decision or branching point. Lines coming out from the diamond indicates different possible situations, leading to different sub-processes.

FLOWCHART EXAMPLE #3

What should I eat for lunch?



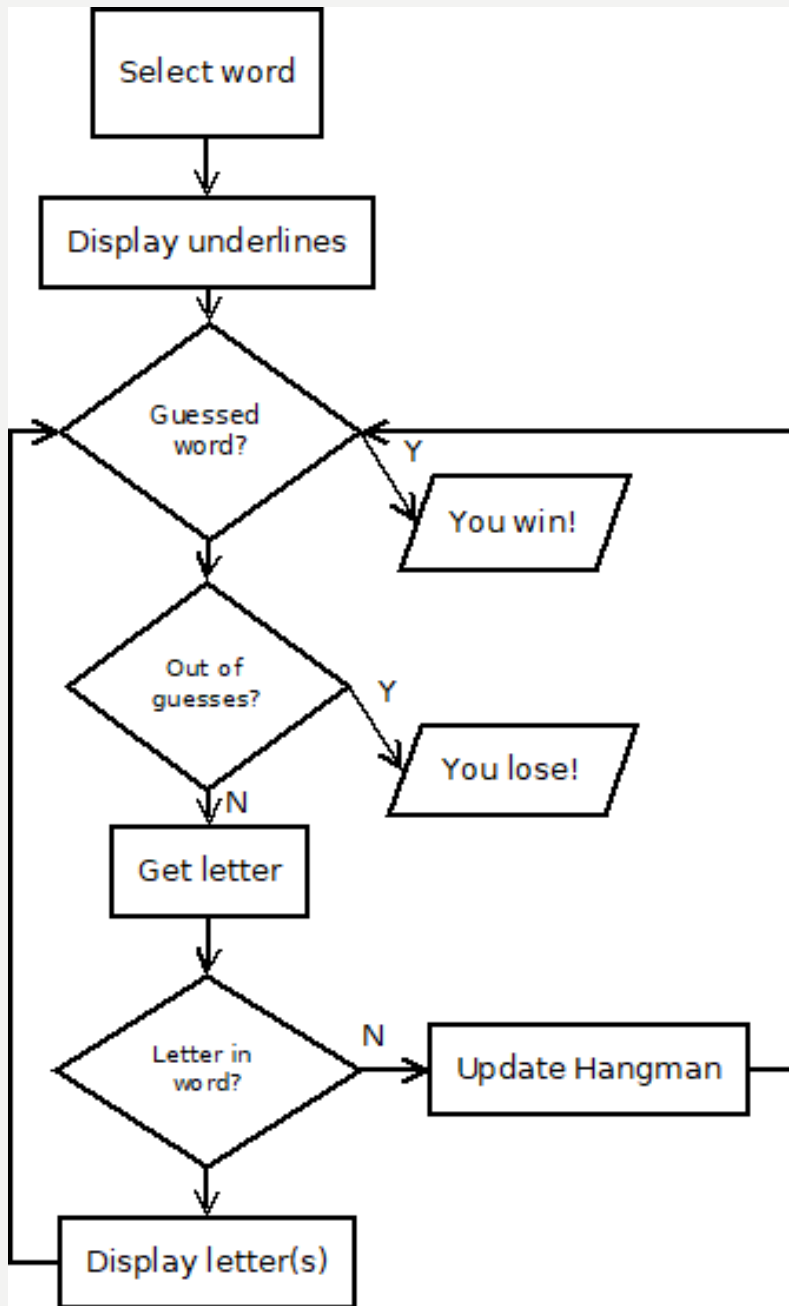
TRY IT – WITH A PARTNER

- Watch the game
- Draw a flowchart for the main program logic
 - Select the word, get a guess, etc.
 - Flowchart does not contain JavaScript syntax. For example, you might have a process box that says Select Word. This will use a math function named random ... but that's unnecessary detail for the flowchart.
 - Just use one process box for drawing the hangman
 - We will do more design on hangman drawing soon

Pedagogy sidebar.

- You can provide a flowchart OR ask students to do it
- Best to use flowcharts for complex processes (just boring for routine tasks)
- There are automated tools (e.g., Dia) that streamline the process.

HANGMAN FLOWCHART



Pedagogy sidebar: Flowcharts can be more detailed, such as specifying how variables should be updated. But this is often more tedious than helpful.

Flowchart can help you identify tasks you don't know how to code. *What have you not yet learned??*

RANDOM

- Games often rely on random behavior (shuffling cards, rolling die, etc.)
- Computer generates pseudo-random numbers (digital code is not like a coin toss, but can appear so)
- `Math.random() ;`
 - Generates a number from `[0, 1)`
 - Includes 0, excludes 1
 - To get number from 0 to 9:
 - `Math.floor(Math.random() * 10) ;`
 - Floor example. Assume 0.28 is generated.
 - $0.28 * 10 = 2.8$. But we want integer values.
 - Floor truncates, becomes 2.
 - What other numbers become 2?

Common: use random to select array location

- `Math.floor(Math.random() * words.length) ;`

Pedagogy sidebar: examples clarify how functions work. English descriptions can be confusing.

PROMPT

```
var person = prompt("Please enter  
your name", "Harry Potter");  
  
if (person == null || person == "")  
{  
    txt = "User cancelled input.";  
} else {  
    txt = "Hello " + person +  
        "! How are you today?";  
}
```

- Causes a pop-up box to appear
- Clunky way to input, but easy to learn
- Data is often input via forms. Common in business, but beyond our scope.

WHILE LOOP

Sometimes we want to continue an action until a certain condition is true

```
// continues previous example
while (ticketsRemaining > 0) {
    var tickets =
        prompt("Buy how many tickets?");
    if (tickets != null && tickets != "") {
        ticketsRemaining -= tickets;
    }
    console.log(ticketsRemaining);
}
// actions AFTER the loop ends
alert("No more tickets!");
```

Pedagogy sidebar: Common mistake is to not update the variables involved in the condition. To debug, put console.log statements inside the loop to help you see how the values are changing.

WHILE WITH BOOLEAN VAR

- Condition for while loop to continue is a Boolean (true/false)
- Can be evaluated statement (tickets < 100)
- Can also be a variable with a Boolean value
- If keepGoing is true, !keepGoing is false (sometimes conditions are easier to express/test this way, such as !gameOver)

```
var keepGoing = true;
var number = Math.floor(Math.random()*10);
var numGuesses = 0;
while (keepGoing) {
    var guess = prompt("Guess the number");
    if (guess != null && guess != "") {
        if (guess == number) {
            alert("That's right!");
            keepGoing = false;
        } else if (numGuesses >= 3) {
            alert("Too bad, you didn't guess");
            keepGoing = false;
        }
    }
    numGuesses++;
}
```

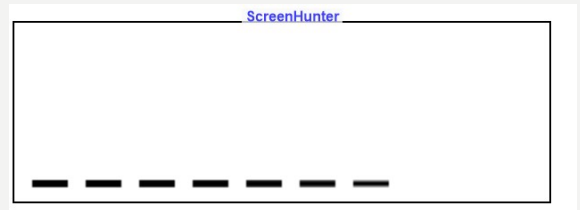
Pedagogy sidebar: how similar should the example be to the assignment? For first lessons, very similar. As students gain experience, increase challenge by providing examples that need more modifications.

UNDERLINE SPACING

- Goal: display the correct # of underlines based on the selected word
- Challenge: the underlines will be drawn (using JS commands we already learned). We need to figure out x and y coordinates so spacing is reasonable.

- Will appear on screen as:

margin|line|margin|line|margin|line



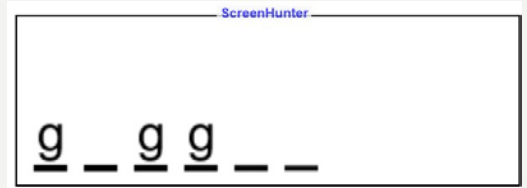
- Size assumptions:
 - margin (space before/between lines) is 10
 - line length is 20 (twice as much as margin)
- Does the y value change?
- How does the x value change each time?

Letter (ix)	start x	end x
0	10	30
1	40	60
2	70	90

Pedagogy sidebar: coding SPEED is based on knowing what you want to do and how to do it! Trial and error is SLOW.

Some students are analytical thinkers, may not need examples. Many/most students will benefit from this technique.

LETTER SPACING



- Goal: display the guessed letters
- JS command: `ctx.fillText(letter, x, y);`
- Each letter should be indented slightly from its underline
- A formula for this will likely be cleaner than lots of additions
- Technique:
 - write out the first few examples (2-3 is often useful to see the pattern)
 - Convert the pattern to a formula
- Size assumptions:
 - margin (space before/between lines) is 10, line length is 20 ($2 * \text{margin}$)
 - indent (from start of line to start of letter) is 2
- y is constant again, but needs to be above line (see descenders in image)

ix	x		formula
0	12	<code>margin+indent</code>	$(ix+1)*\text{margin} + \text{indent}$
1	42	<code>margin+line+margin+indent</code>	$(ix+1)*\text{margin} + ix * \text{line length} + \text{indent}$
2	62	<code>margin+line+margin+line+indent</code>	same as above

Do we need a different formula for first letter?

LETTER IN WORD?

- Everyone knows how to google
- BUT some queries are more effective than others
- Be specific and concise about goal
- Include language (i.e., JavaScript)
- The first answer is not always the best
- Built-in commands are preferred (if JS has a function that does what you need, use it!)
- jQuery is a popular JS library. It's very useful for professionals, but you will probably want to avoid it (has lots of power but also a learning curve)
- Some solutions will use “regular expressions” which are also powerful but even more learning curve
- Example queries:
 - JavaScript draw text canvas
 - JavaScript is letter in word
 - yields a number home-grown solutions
 - JavaScript string functions
 - https://www.w3schools.com/jsref/jsref_indexof.asp

As with other aspects of programming, your skill with this will increase over time. The “best” programmers are often just the most persistent. Tell your girls this! (and your boys, too)

MORE DESIGN

- Need to design the hangman
- Note that this could be done before or after game play
- `HangmanDrawing.html`

MODELING

- It's good to show students that sometimes you have to think/struggle (i.e., don't present everything in finished form, show the false starts)
- If you like a student's idea better than yours, adopt it (and give credit)

Hangman Design

Plan your variables
// available to you
canvas.width: 400
canvas.height: 400
// Face location and size
faceX: $c.w/2$
faceY:
faceRadius: $c.h/4$
// Body
bodyLength: $faceR \times 3$
// Arm & Leg. Could be the same or different
armLegLengths: $bodyLength/3$

Observation
center of face, body both centered. One end of each line also at center X
centerX = $c.w/2$

*original - ratio of body parts
scalable - more relation
to canvas
concrete vs
abstract*

CONTINUE PROGRAMMING

- If this is brand new and too fast/confusing, I can revisit with individuals/small groups tomorrow.
- If concepts are good but it will take more time, this might be your post work.
- If you're done or almost done – KUDOS!



END OF DAY 2

**STICK AROUND FOR BROADENING
PARTICIPATION IN COMPUTING**