SESSION TEN
OCTOBER 31, 2023

FUNCTIONS + THE DOM

ENDLESS APPRECIATION
TO ERIC LI

DAVIS SCHERER
SCHERERD@NEWSCHOOL.EDU

AGENDA

1. JAVASCRIPT + YOU
2. ANATOMY OF A FUNCTION
3. INTERACTING WITH THE DOM
4. BREAK
5. PRACTICE
6. HOMEWORK

YOUR JAVASCRIPT

Today, we're taking a step back.
Let's talk about how to think about functions,
and what we're actually doing when we
are using JS to affect our HTML.

YOUR JAVASCRIPT

How have you found working with JS?
What kind of scripts did you look at?
Let’s discuss the JS you broke down.

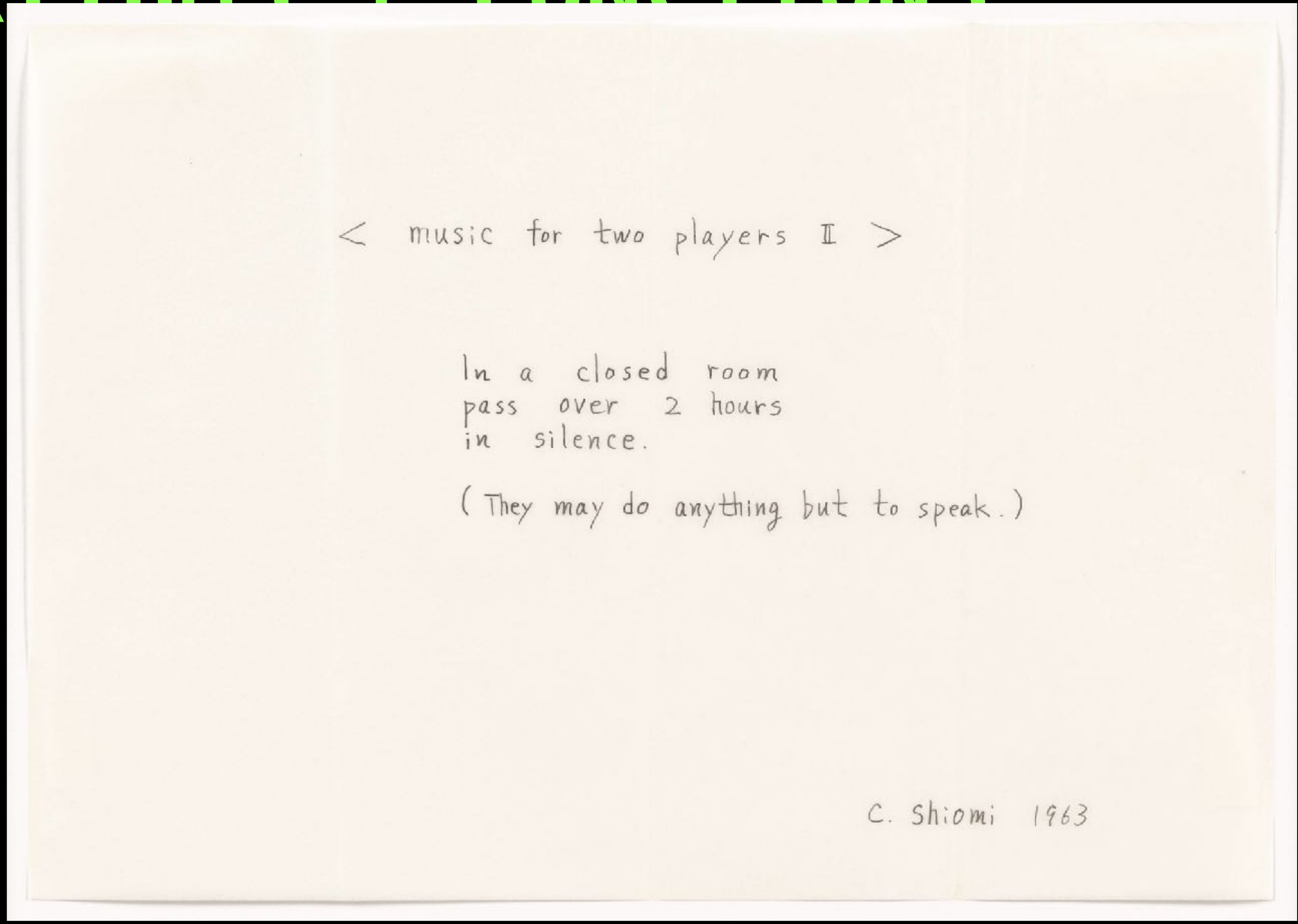
ALGORITHMS + FUNCTIONS

All an algorithm is is a series of instructions. It is a sequence of events targeted towards an end.

ALGORITHMS + FUNCTIONS

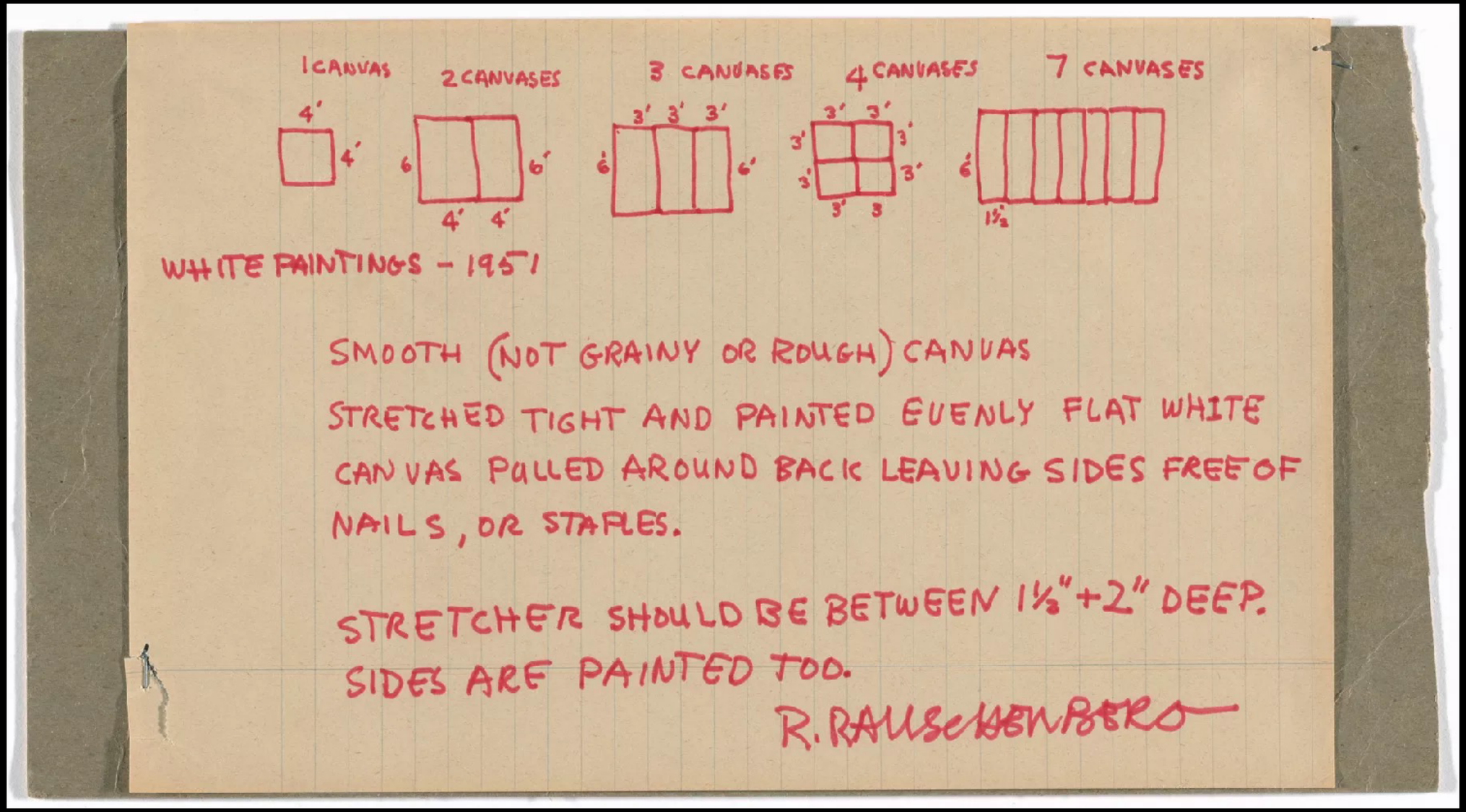
A recipe is an algorithm.
Directions are an algorithm.
An artwork can be an algorithm.

ALGORITHMS + FUNCTIONS



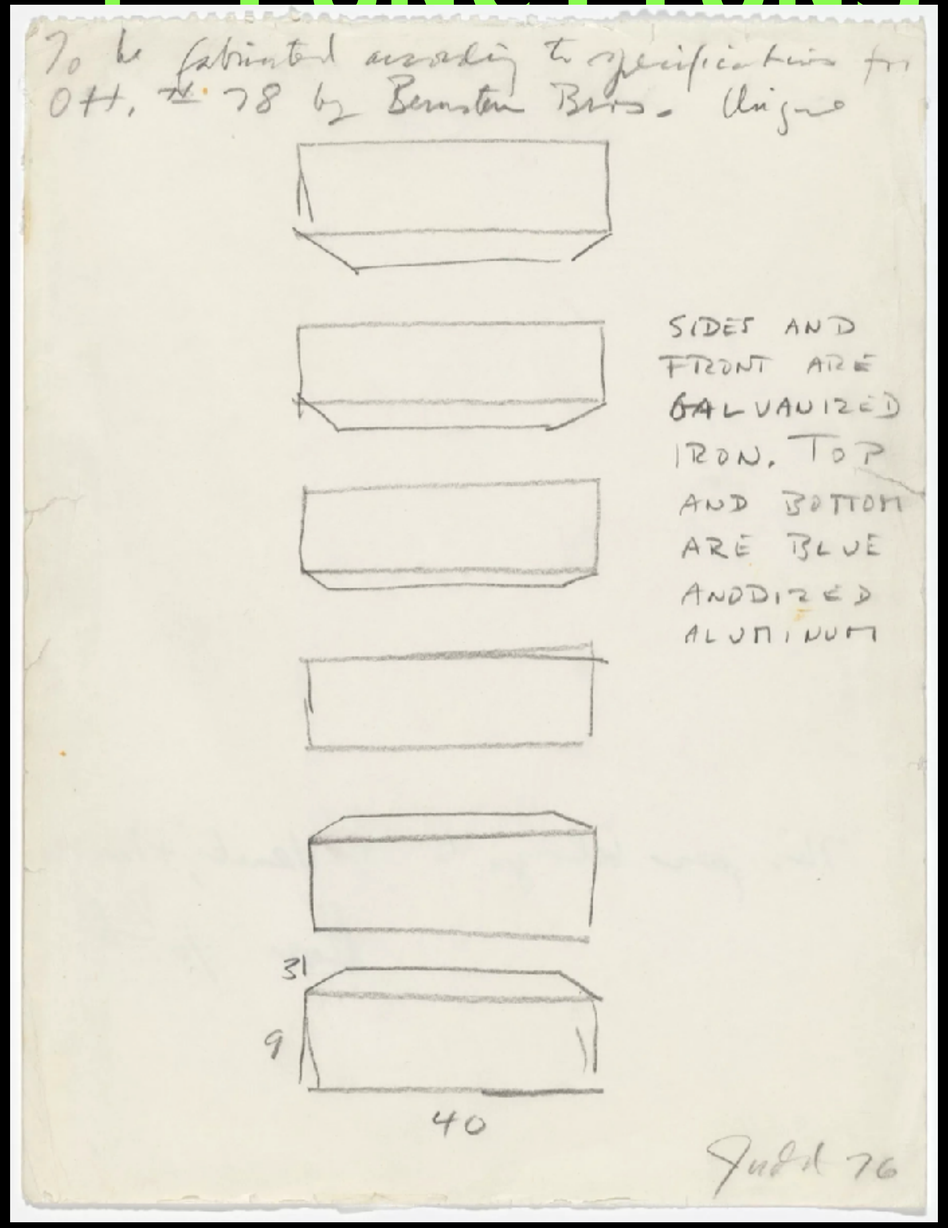
Mieko Shiomi < *music for two players II* > 1963

ALGORITHMS + FUNCTIONS



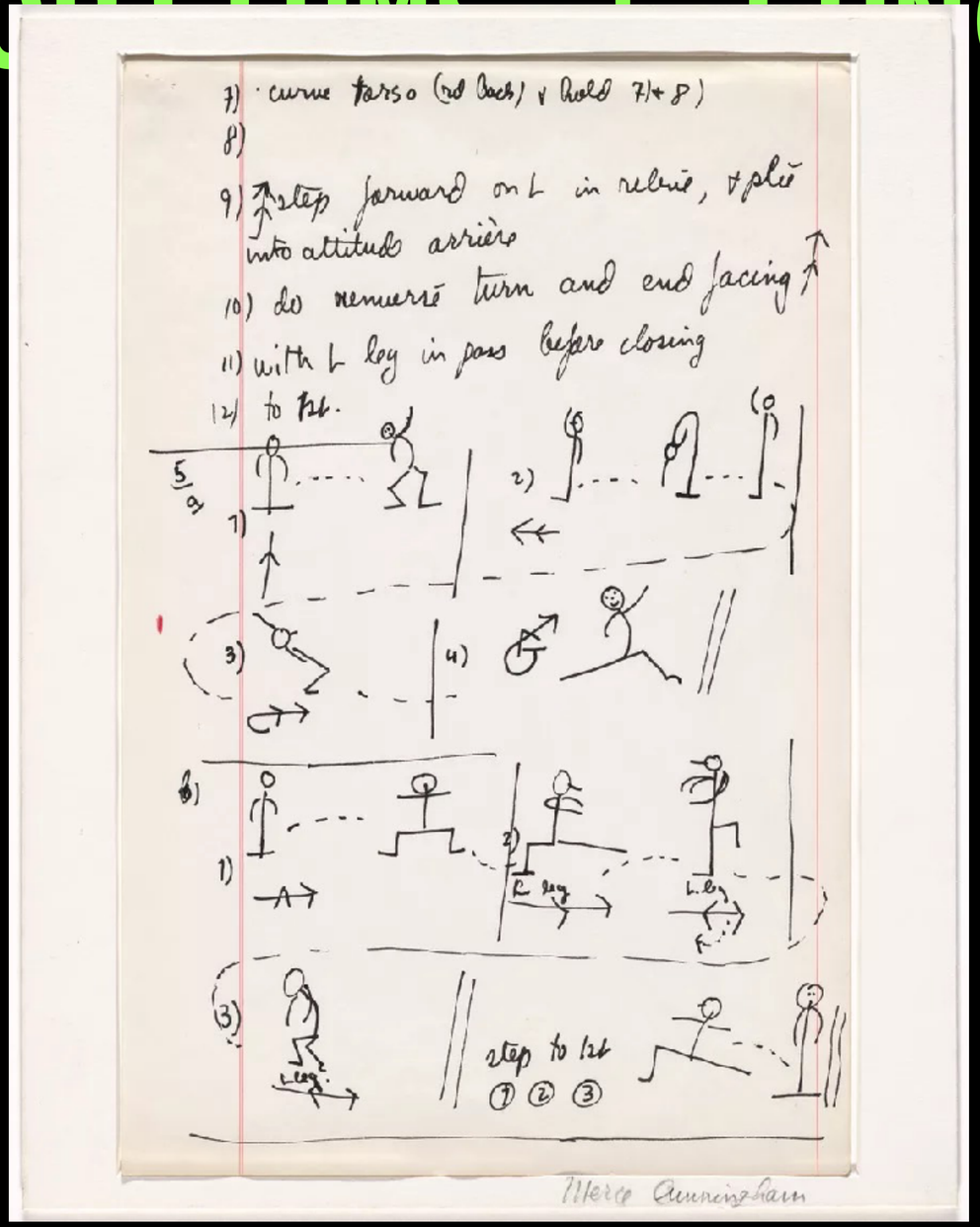
Robert Rauschenberg *White Paintings 1951*, 1965

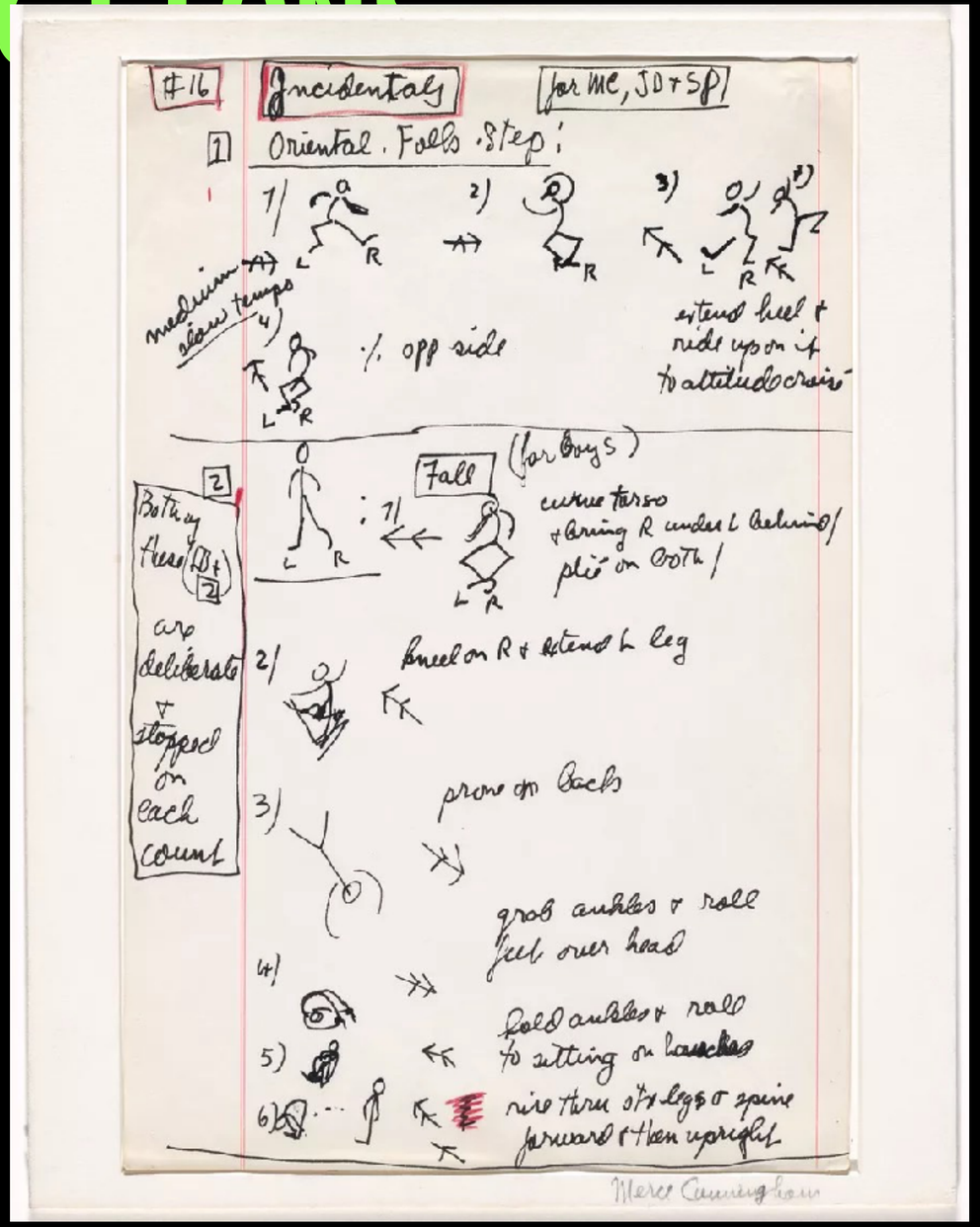
ALGORITHMS + FUNCTIONS



Donald Judd, *Instruction Drawing for Otterlo Show Wall Sculpture*, 1975

ALGORITHMS + FUNCTIONS





ALGORITHMS + FUNCTIONS

Functions in javascript are just machine-readable instructions.

ALGORITHMS + FUNCTIONS

A classic example:
How would you explain making a peanut butter and jelly sandwich to someone who's never done it before?
Write your answer.

ALGORITHMS + FUNCTIONS

That’s how we should approach functions in code—either when reading or writing them. The best algorithms are **specific, ordered, and streamlined.**

FUNCTIONS IN JS

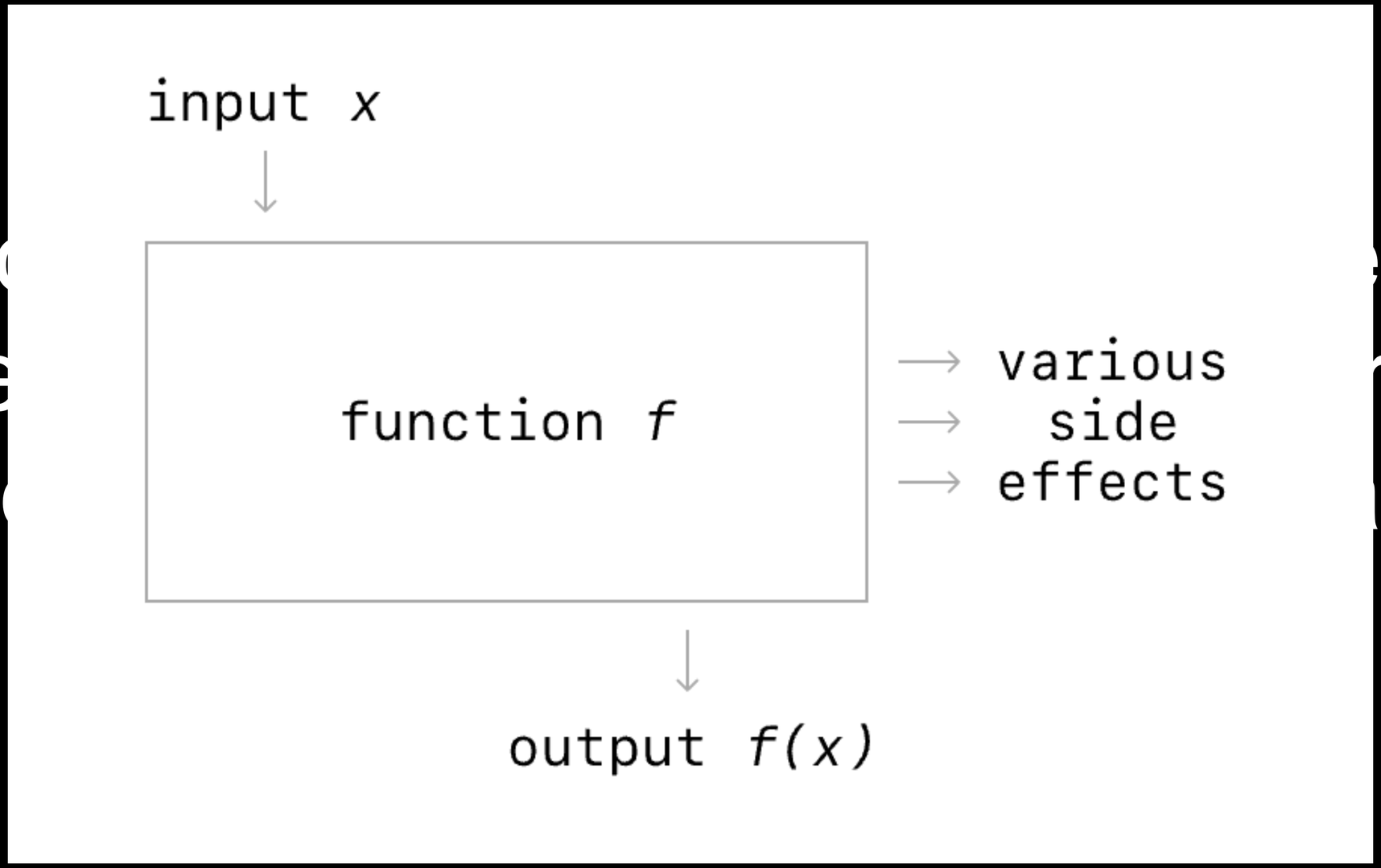
“ A function in JavaScript is similar to a procedure—a set of statements that performs a task or calculates a value, but for a procedure to qualify as a function, it should take some input and return an output where there is some obvious relationship between the input and the output. To use a function, you must define it somewhere in the scope from which you wish to call it.” —MDN

WHY FUNCTIONS?

Functions can be **reused**. In the interest of expediency and not repeating yourself, consider containing anything you may want to do again in a function.

WHY FUNCTIONS?

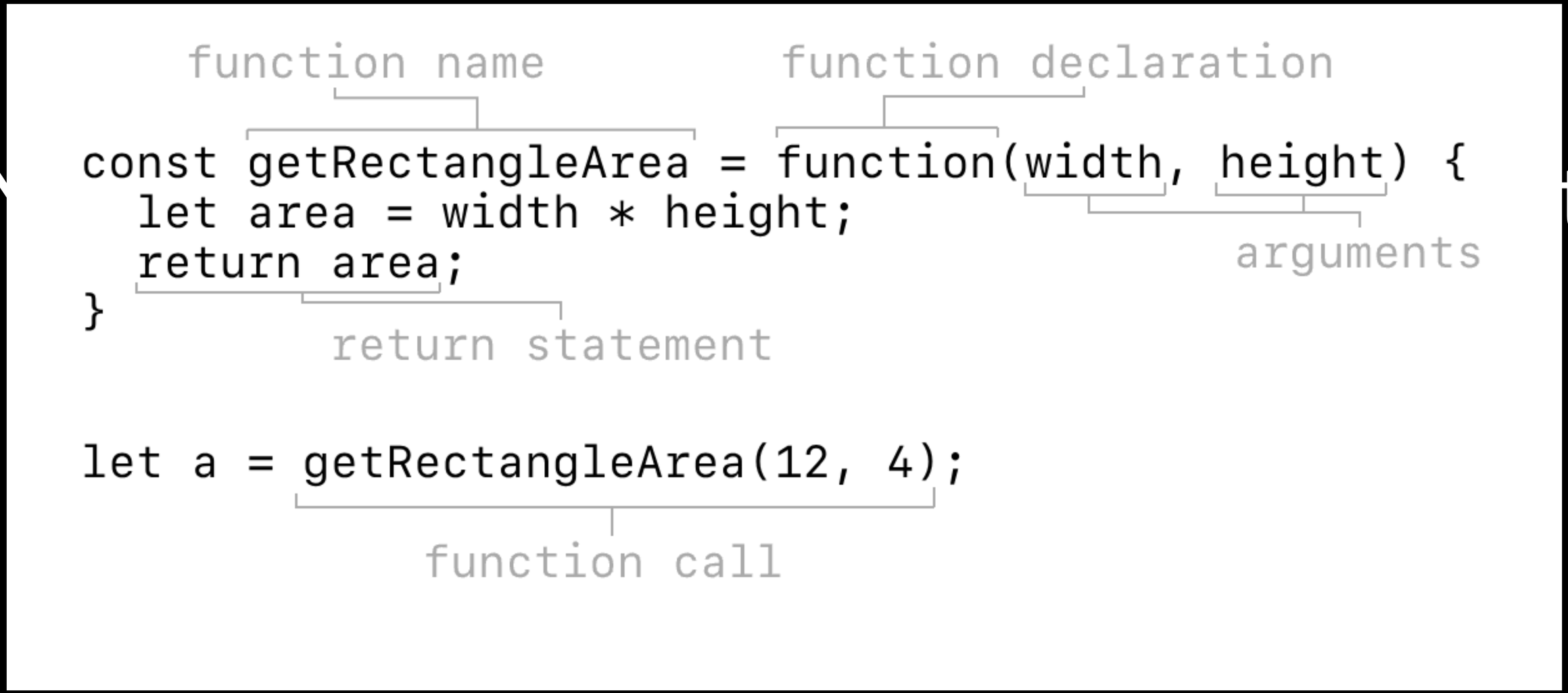
Functi erest of
expe rself,
consid y want



FUNCTIONS IN JS

You can recognize a function in javascript from its telltale parentheses—empty or otherwise.

FUNCTIONS IN JS



FUNCTIONS IN JS

1. Create a name for the function

FUNCTIONS IN JS

2. Declare the function's *arguments*—essentially, what is input to be acted upon

FUNCTIONS IN JS

3. Implement a function’s behavior (in the curly brackets)

FUNCTIONS IN JS

4. Specify any return (output) of the function

FUNCTIONS IN JS

We touched on **libraries** last class—libraries, essentially, are just a set of purpose-built functions.

THE DOM

The DOM (*Document Object Model)*
is an abstraction of your HTML / CSS
so that it can be directly affected by
JavaScript.

THE DOM

JavaScript has an *object* called `document` which represents the HTML of a website as an entity.

THE DOM

```
<body>
  <h1 id="heading">This is a
  heading.</h1>
  <p>This is a paragraph.</p>
  <p>This is another
  paragraph.</p>
  <ul class="list">
    <li class="list_item">List
  item 1</li>
    <li class="list_item">List
  item 2</li>
  </ul>
</body>
```

```
// Matches the first element with an id="heading" attribute
  let h1 = document.getElementById('heading'); // Returns <h1>

  // Selects the first element which matches the given CSS selector
  let p = document.querySelector('p'); // Returns <p>
  let div = document.querySelector('div'); // Returns null

  // Selects the first element which matches the given CSS selector
  let ps = document.querySelectorAll('p'); // Returns [<p>, <p>]
  let divs = document.querySelectorAll('div'); // Returns []

  // How do we access all list items?
  let liEls= document.querySelectorAll('li');
  let liEls = document.querySelectorAll('.list_item');
  let liEls = document.querySelectorAll('.list > .list_item');
```

THE DOM

```
<body>
  <h1 id="heading">This is a
  heading.</h1>
  <p>This is a paragraph.</p>
  <p>This is another
  paragraph.</p>
  <ul class="list">
    <li class="list_item">List
  item 1</li>
    <li class="list_item">List
  item 2</li>
  </ul>
</body>
```

```
// Matches the first element with an id="heading" attribute
let h1 = document.getElementById('heading'); // Returns <h1>

// Selects the first element which matches the given CSS selector
let p = document.querySelector('p'); // Returns <p>
let div = document.querySelector('div'); // Returns null

// Selects the first element which matches the given CSS selector
let ps = document.querySelectorAll('p'); // Returns [<p>, <p>]
let divs = document.querySelectorAll('div'); // Returns []

// How do we access all list items?
let liEls= document.querySelectorAll('li');
let liEls = document.querySelectorAll('.list_item');
let liEls = document.querySelectorAll('.list > .list_item');
```

THE DOM

So what can we do?

THE DOM

```
let h1 = document.getElementById('heading'); // Returns <h1>
  h1.innerText; // Gets the rendered text contained within an element, This is a heading.
  h1.innerHTML; // Gets the inner HTML of an element, <a href="/">This is a heading.</a>

  h1.innerText = 'New heading'; // Sets the value of innerText of our h1 to 'New heading'
  h1.innerHTML = '<a href="/">New heading</a>'; // Same here, but can use HTML elements

  h1.style; // Gets the inline styles of an element.
  window.getComputedStyle(h1); // Get all computed styles of the element.
  Window is used because it represents the visual, not the Document.

  h1.style.color = 'red'; // Sets the color of the element to 'red'

  // Sets the background color. You can either use CSS property names as in the first line,
  or camelCase the property as in the second line.
  h1.style['background-color'] = 'green';
  h1.style.backgroundColor = 'green';

  // Use classList to add, remove, or toggle the red class on an element.
  h1.classList.add('red');
  h1.classList.remove('red');
  h1.classList.toggle('red');
```

THE DOM

We can also create new elements in JavaScript by manipulating the DOM.

THE DOM

```
// Create a new div element.
  const newDiv = document.createElement('div');
  // Give it some copy.
  newDiv.innerText = 'Hi, I’m a newly created element!';

  // Add the element into the DOM.
  document.body.appendChild(newDiv);
```

YEHWAN SONG

yhsong.com

YEHWAN SONG

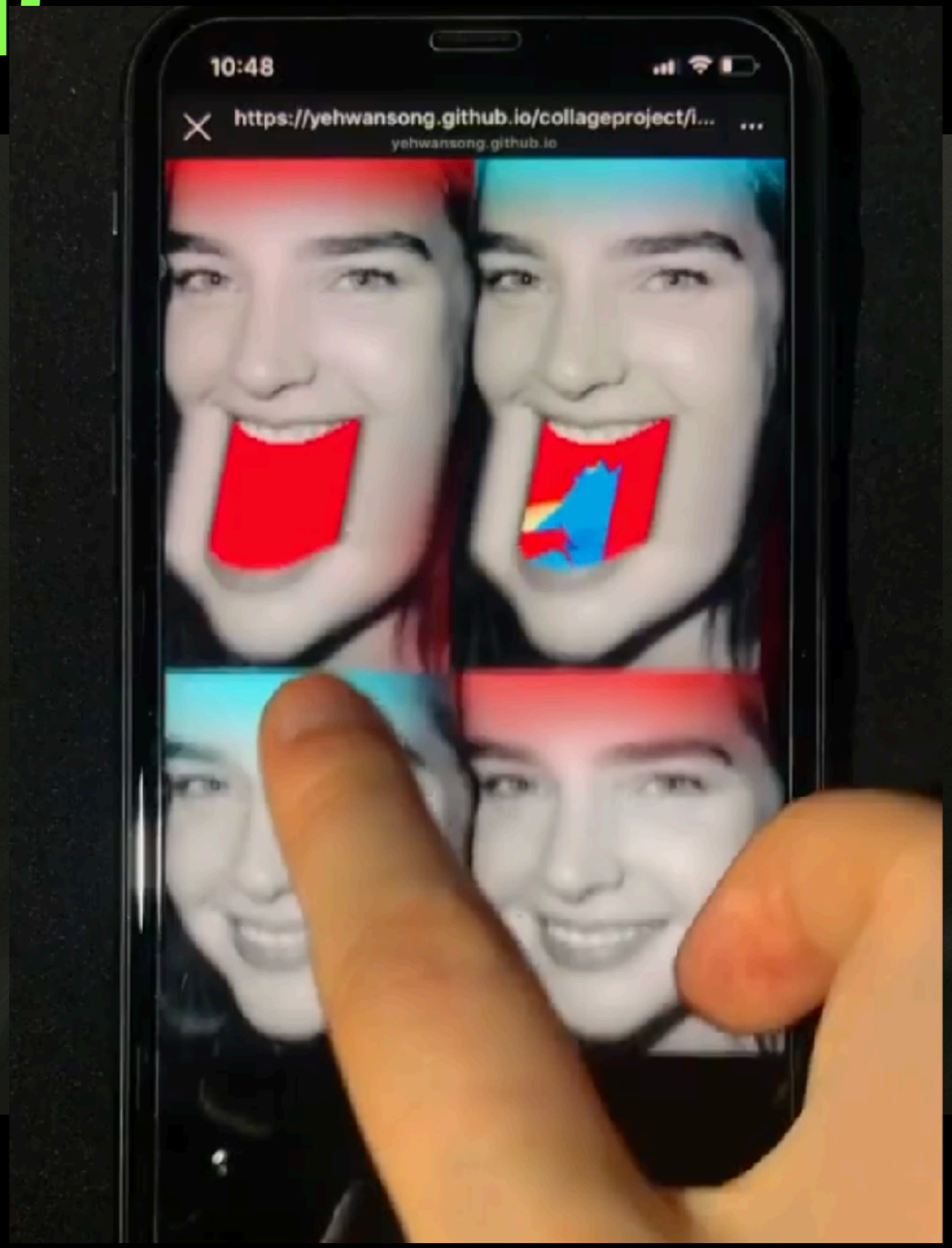

Archive
Yaeji
Shows
Merch
Listen
Video
Subscribe
Contact
Album Out Now

YEHWAN SONG

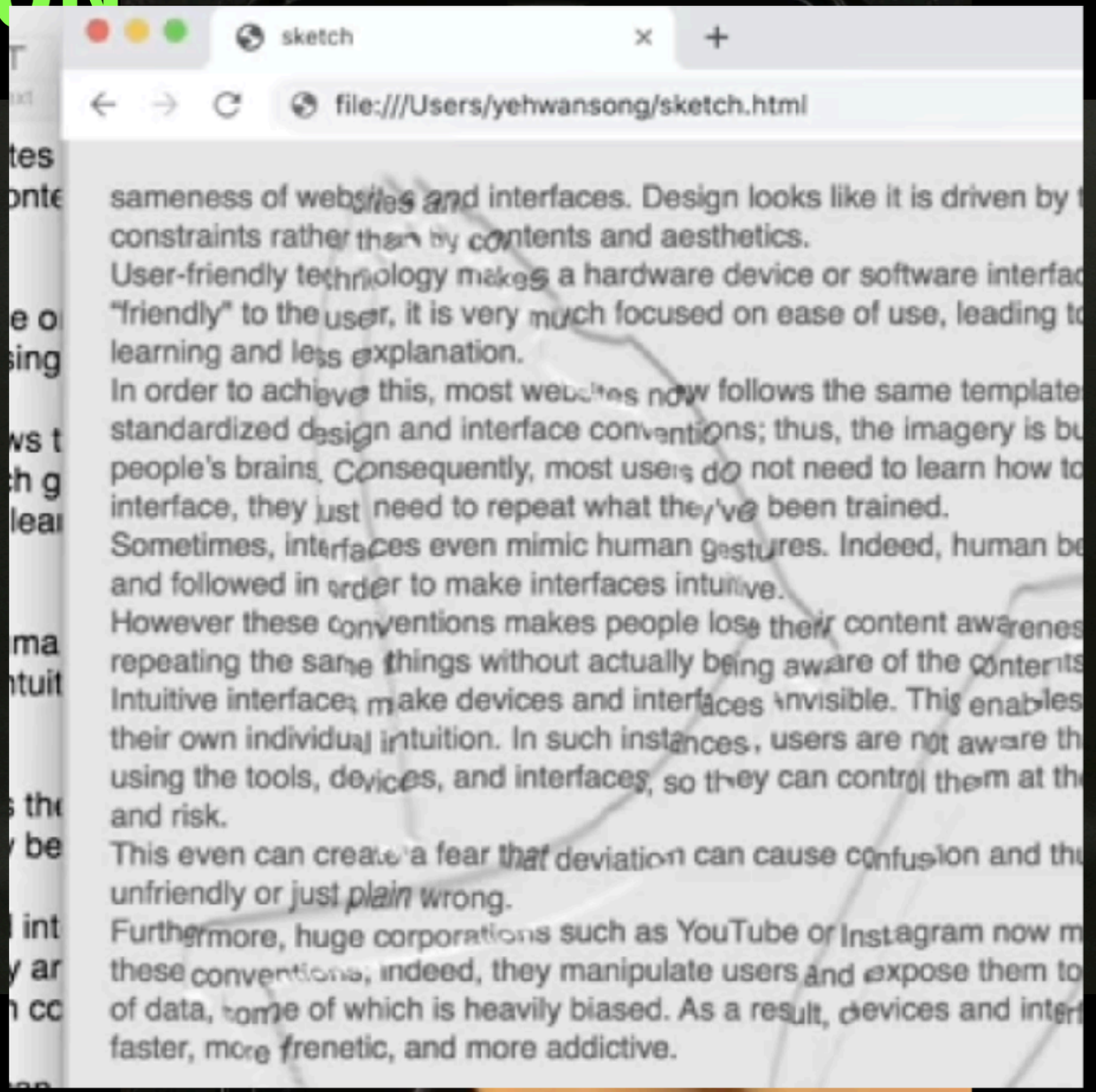

HOME
ARCHIVE
GAME
애프터 메니페스토 텍스트에서 목소리로
아직 생성 가운데 있는 사이버 공간이라는 미로 속에서 정책과 결정과 실천이 제 길을 찾을 수 있도록 하나의 긍정적인 시각을 만들어내고자 한다.
(피에르 레비, 17)
한국에서는 이미 1980년대에서 90년대에 새로운 영토로 등장해 사이버 공유지로 상상되었던 인터넷에서 여성과 성소수자에 대한 혐오가 점증되기 시작했다.
(손희정 2017b, 137)
나선의 춤에 갇혀 있다는 점에서는 마찬가지이지만 나는 여신보다 응앵응이 되겠다.

YEHWAN SONG

YEHWAN SONG


10:48
https://yehwansong.github.io/collageproject/i...
yehwansong.github.io

YEHWAN SONG


sketch
file:///Users/yehwansong/sketch.html
sameness of websites and interfaces. Design looks like it is driven by
constraints rather than by contents and aesthetics.
User-friendly technology makes a hardware device or software interfac
"friendly" to the user, it is very much focused on ease of use, leading t
learning and less explanation.
In order to achieve this, most websites now follows the same template
standardized design and interface conventions; thus, the imagery is bu
people's brains. Consequently, most users do not need to learn how to
interface, they just need to repeat what they've been trained.
Sometimes, interfaces even mimic human gestures. Indeed, human be
and followed in order to make interfaces intuitive.
However these conventions makes people lose their content awarenes
repeating the same things without actually being aware of the contents
Intuitive interfaces make devices and interfaces invisible. This enables
their own individual intuition. In such instances, users are not aware th
using the tools, devices, and interfaces, so they can control them at th
and risk.
This even can create a fear that deviation can cause confusion and th
unfriendly or just plain wrong.
Furthermore, huge corporations such as YouTube or Instagram now m
these conventions, indeed, they manipulate users and expose them to
of data, some of which is heavily biased. As a result, devices and inter
faster, more frenetic, and more addictive.

BREAK (10m)

THE DOM

Let’s practice together.

HOMEWORK

1. HC 8

2. READ *DIGITAL POETICS* by Taeyoon Choi.
   Be prepared to discuss. (It’s short!)
   https://taeyoonchoi.com/poetic-computation/digital-poetics/

SESSION TEN
OCTOBER 31, 2023

THANK YOU

DAVIS SCHERER
SCHERERD@NEWSCHOOL.EDU