

ECA  
CHECK-IN



# Welcome to Hack Club!

**Workshop:** Painting App (Drawing on a HTML Canvas)

10/11/2021

BENEFITS

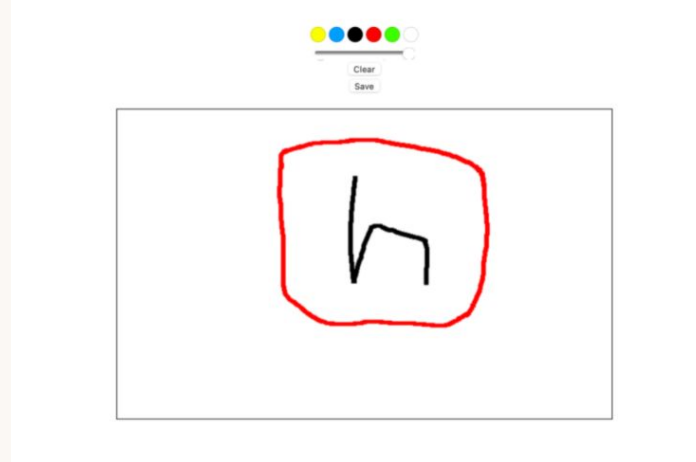


# Painting App



# Objective

- Create a painting application using HTML, CSS & Javascript



# Setting up

Create a new Glitch website



## Website

⚡ STATIC SITE

Build a website from scratch using the basics: HTML, CSS and JavaScript. It could be a splash page, VR web experiment, or personal page.

Remix: **Hello Website** |

**Blank Website**

# Setup basic HTML

- Create a `<div>` tag in the body of the page (div stands for division)
  - Give it the class -> main-container (for CSS)
- Next, we want to create a basic canvas using the `<canvas>` tag, and add the parameters below
  - id -> paint-canvas (which will allow us to grab it in the javascript)
  - width -> 640 (sets width to 640)
  - height -> 400 (sets height to 400)

# Setup our JS file

- Add a `window.onload` method (this will run when the html has finished loading)
- Grab the paint-canvas using `document.getElementById()`
- Define the bindings of the canvas -> `canvas.getBoundingClientRect()`
- Define context using the canvas -> `canvas.getContext("2d")`
- Set `context.strokeStyle` to "black" (color of the strokes)
- Set `context.lineWidth` to 2 (size of the strokes)
- Create the following variables and initialize
  - `mouseX` -> 0
  - `mouseY` -> 0
  - `isDrawing` -> false

# Now let's get to Drawing

- Let's add event listeners for mouse movements (use `canvas.addEventListener`)
- We need to write methods 3 different events
  - `mousedown` (to start drawing)
  - `mousemove` (to actually draw)
  - `mouseup` (to stop drawing)
- Here are some of the methods we will be using to draw

```
context.beginPath();  
context.moveTo(mouseX, mouseY);  
context.lineTo(mouseX, mouseY);  
context.stroke();
```

# Let's create setMouseCoordinates()

- Create a function called setMouseCoordinates that takes in the event value
- In the function we want to,
  - Set the mouseX variable to `event.clientX - bindings.left`
  - Set the mouseY variable to `event.clientY - bindings.top`
- This sets the variables to have x and y values inside of the canvas



# Let's Polish with some CSS

- Let grab the class main-container add add the following properties
  - `display: flex`
  - `flex-direction: column` (specifies for only horizontally)
  - `justify-content: center` (centers div)
  - `align-items: center`
- Let's grab the canvas and add the following props
  - `border: 1px black solid` (adds a border to the canvas)
  - `cursor: crosshair` (set the cursor on the canvas to a crosshair)

# Possible Improvements

## Buttons the change the color

Add multiple buttons and inside of the JS change the `strokestyle` to "black"

## Slider to change stroke width

Create a slider using the input tag and have the js set `context.lineWidth` whenever it changes

## Clear Button + Save Button

Create buttons and inside of the js have it clear the canvas.

# Hacking time

