

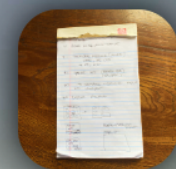
Part Three:

# Create Your Own Web Operating System (OS)

ThomasOS

Always Learning ☁

7/23/2023, 2:46:43 PM



Notes



Angel Hacks (Hackathon Project)

**Carolina Hacks**



Carolina Hacks is South Carolina's first student-led Hackathon for high schoolers.

At our hackathon, high schoolers will

[Watch Documentary](#)

0:00 / 4:07

ALRIGHT

**Giving Giraffee (Hackathon Project)**




**Giving Giraffe**

Clara. Thomas. Dieter

0:00 / 4:17

Welcome



**ThomasOS**

ThomasOS is a feature-limited OS to answer the unanswerable question of "who is Thomas?"


Photo



Notes - Welcome

Welcome  
06/28/2023

Sample Text  
06/28/2023




Welcome to **Hacker Notes**

This is a place where I store my thoughts as they come to mind. What exactly will you find when browsing through these notes? As I ~~once said~~ always say.

I  
a  
r  
s

Epoch (Hackathon I Attended)



Monday morning, [it's] Boxing Day here in Singapore.

0:02 / 4:58

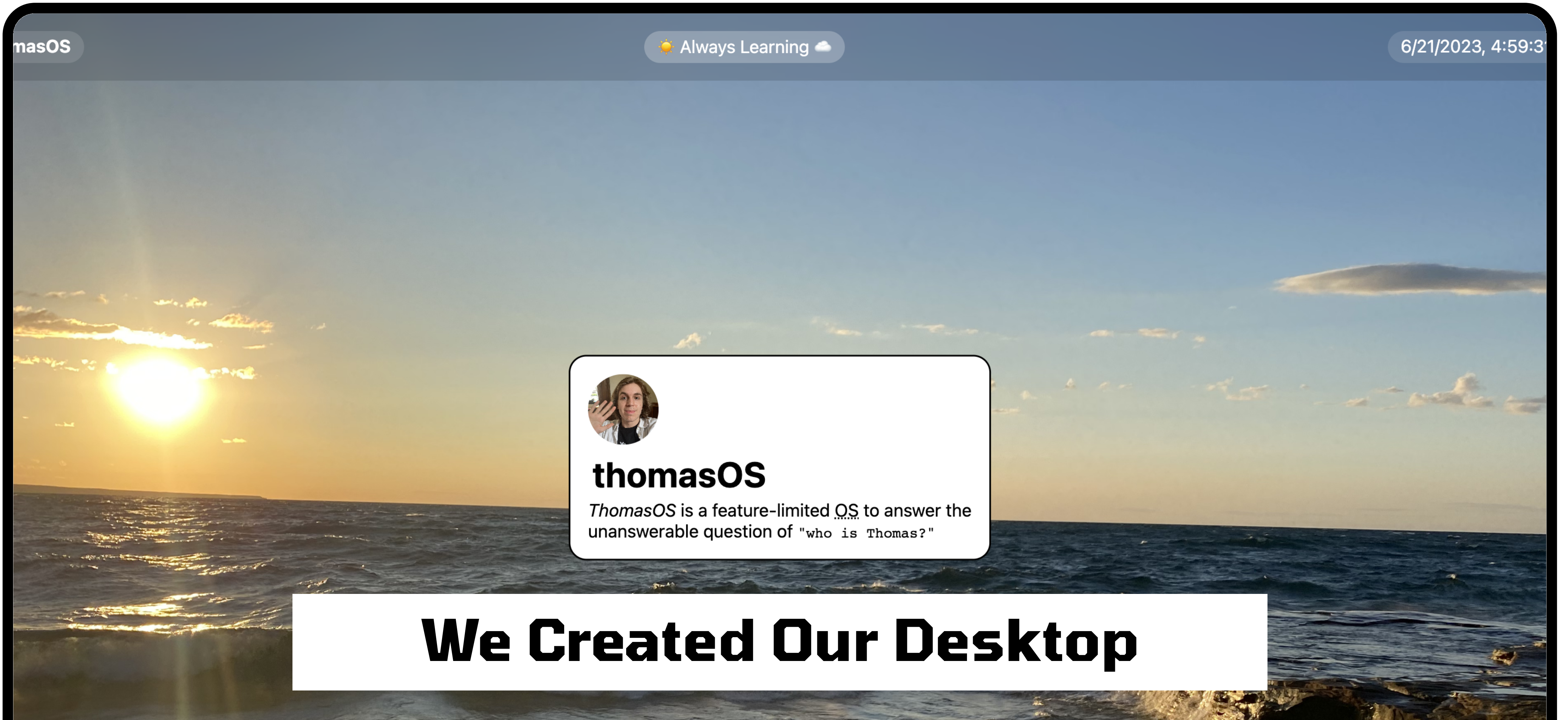
**Build Your Own personalOS  
(that's all about you!)**

**Today We're Going To Be...**

# Making Windows Draggable, Closable, and Openable!



# Where we left off:



thomasOS

Always Learning

6/21/2023, 4:59:3



**thomasOS**

ThomasOS is a feature-limited OS to answer the unanswerable question of "who is Thomas?"

**We Created Our Desktop**

# **Organizing Our Logic**

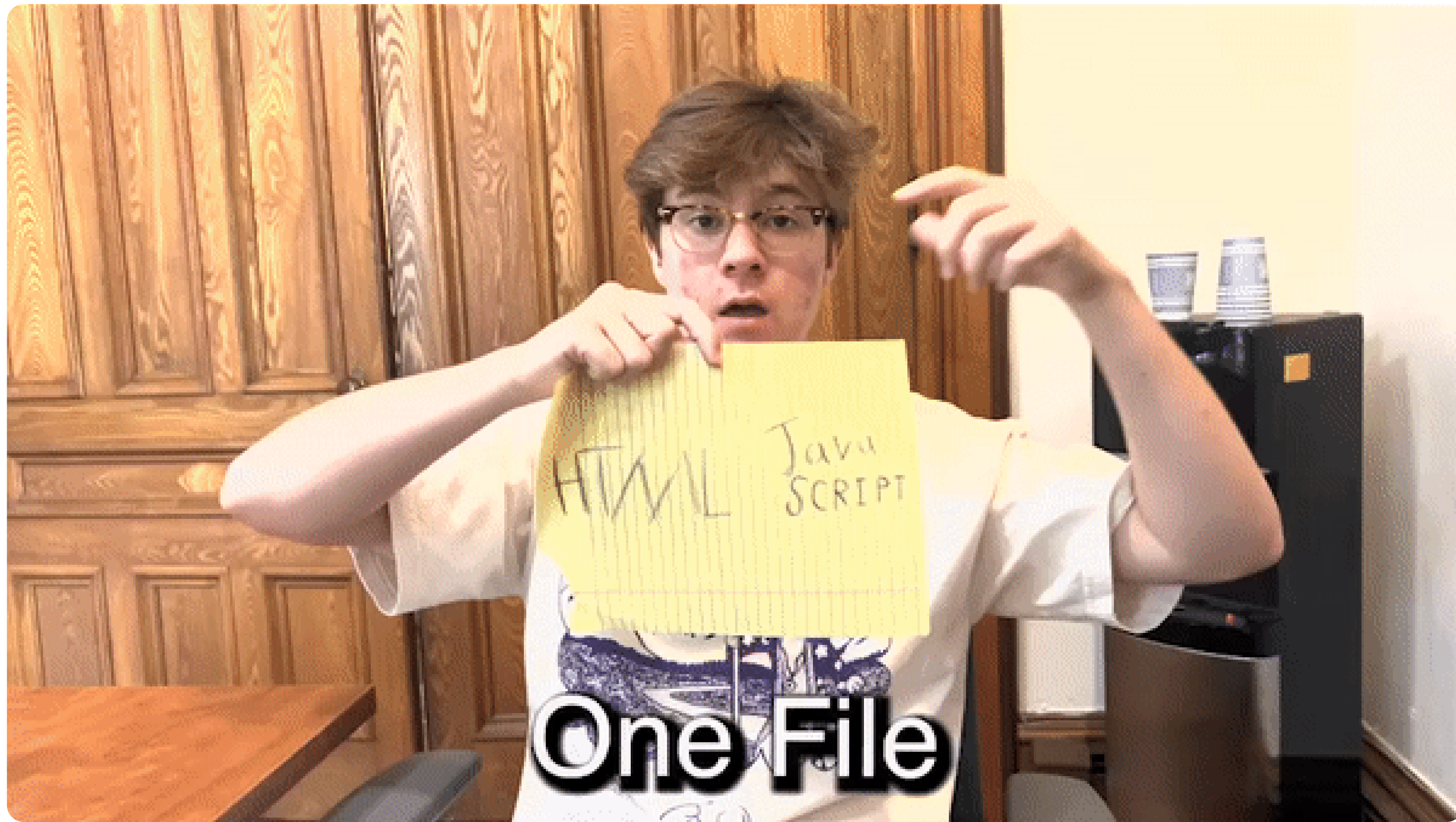
**Our Code is a bit...**

**MESSY**



**We Can Fix This By...**

# Splitting Our Code into Two Files



**One File for Content (HTML)  
& One File for Logic (JavaScript)**

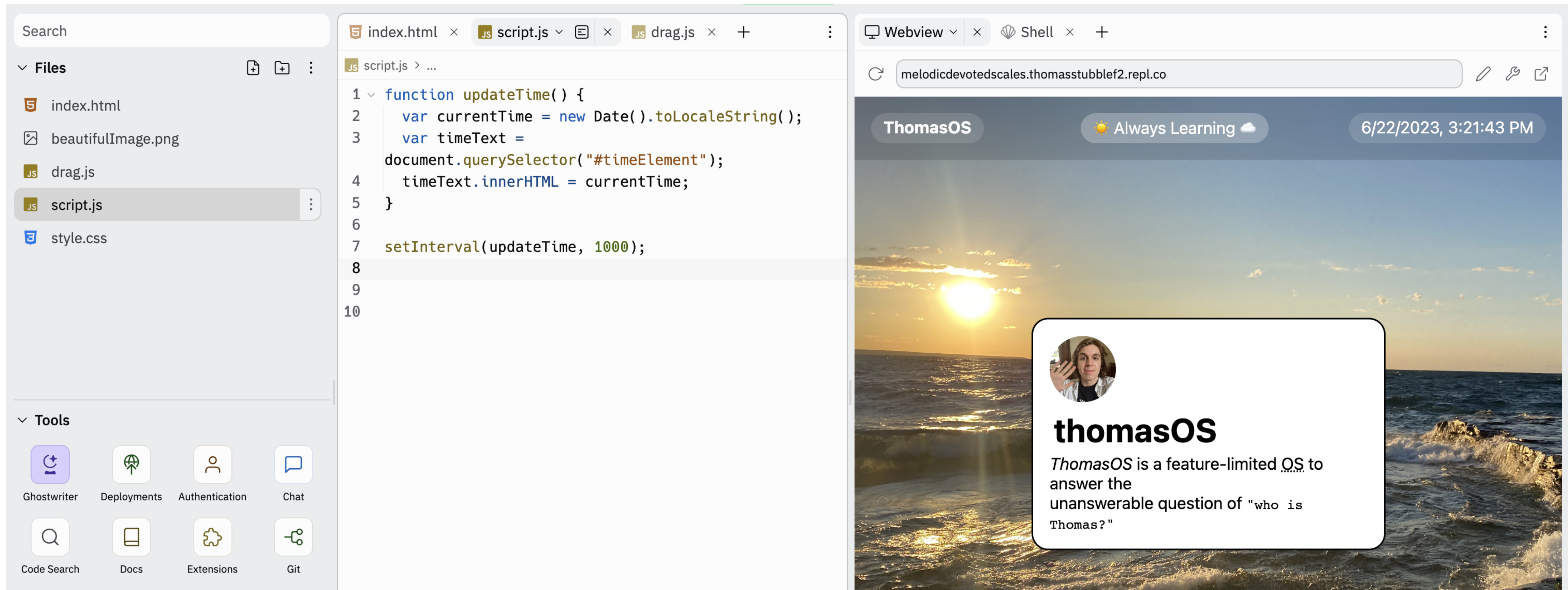
**We need these two files  
to communicate**

# We can do that by...

```
<body>  
    (content)  
    <script src="script.js"></script>  
</body>
```

## Linking the Script to our HTML

# & Copy Our JavaScript Code Into Script.js



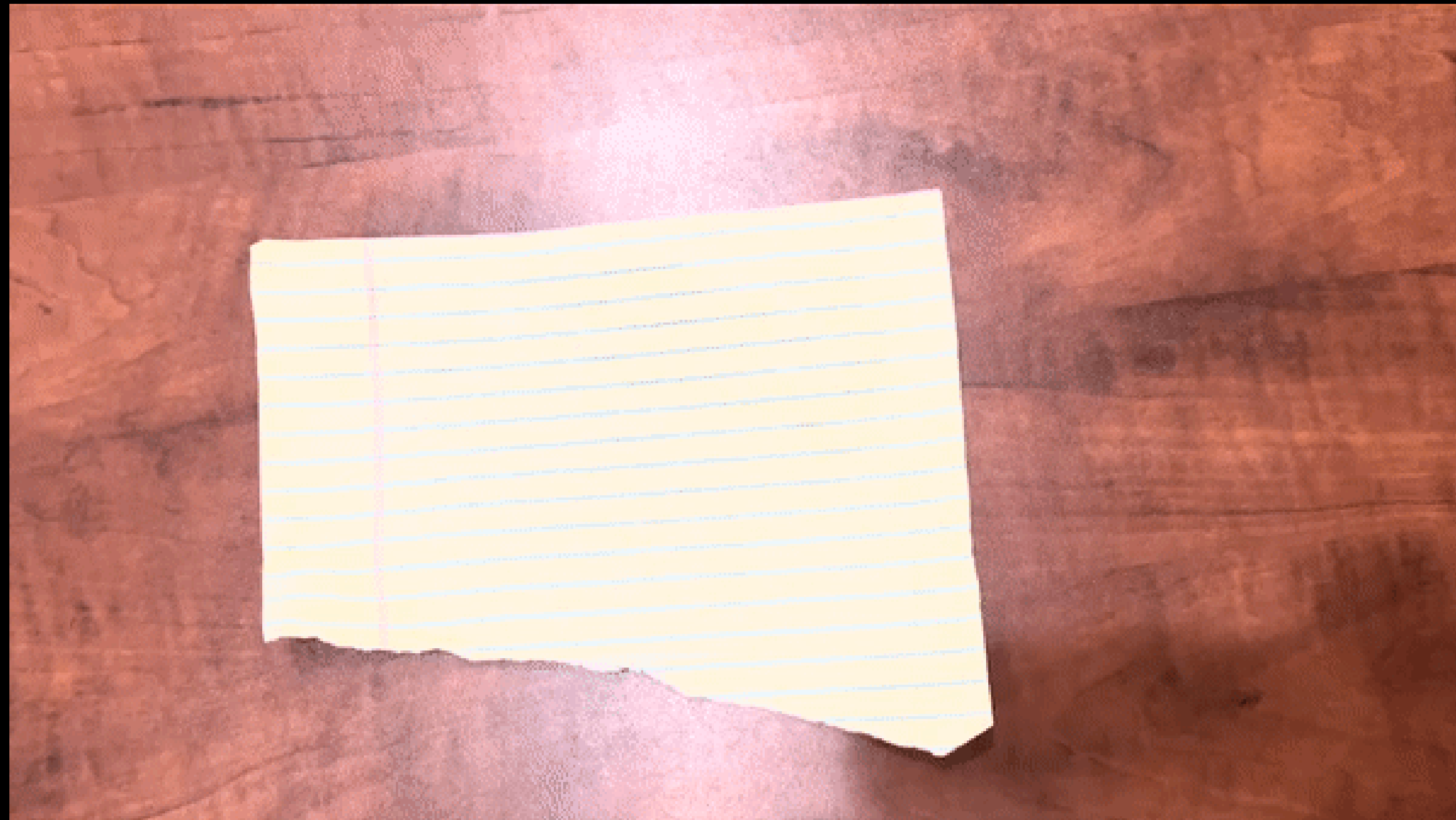
The image shows a development environment with a code editor on the left and a web browser on the right. The code editor displays the following JavaScript code in `script.js`:

```
1 function updateTime() {  
2   var currentTime = new Date().toLocaleString();  
3   var timeText =  
4     document.querySelector("#timeElement");  
5     timeText.innerHTML = currentTime;  
6 }  
7 setInterval(updateTime, 1000);  
8  
9  
10
```

The web browser shows a page titled "ThomasOS" with a status bar displaying "Always Learning" and the date/time "6/22/2023, 3:21:43 PM". The page features a sunset background and a profile card for "thomasOS" with a profile picture and a bio: "ThomasOS is a feature-limited OS to answer the unanswerable question of 'who is Thomas?'".

**Awesome It Works!**

# Let's Make the Window Movable

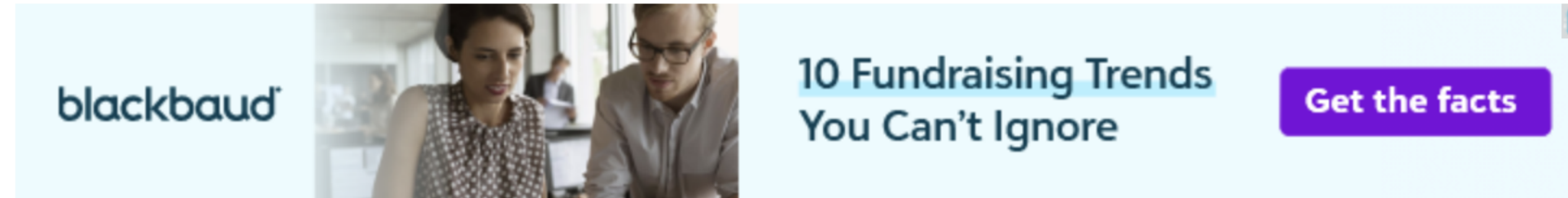




**Try Googling:**

**"How to Make an HTML  
Element Draggable?"**

- Device Look
- Contenteditable Border
- Placeholder Color
- Disable Resizing of Textarea
- Disable Text Selection
- Text Selection Color
- Bullet Color
- Vertical Line
- Dividers
- Text Divider
- Animate Icons
- Countdown Timer
- Typewriter
- Coming Soon Page
- Chat Messages
- Popup Chat Window
- Split Screen
- Testimonials
- Section Counter
- Quotes Slideshow
- Closable List Items
- Typical Device Breakpoints
- Draggable HTML Element**
- JS Media Queries
- Syntax Highlighter
- JS Animations
- JS String Le Google Chrome
- JS Exponentiation



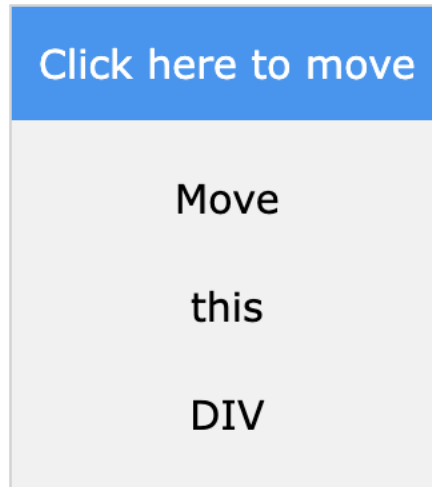
# How TO - Create a Draggable HTML Element

[← Previous](#)

[Next →](#)

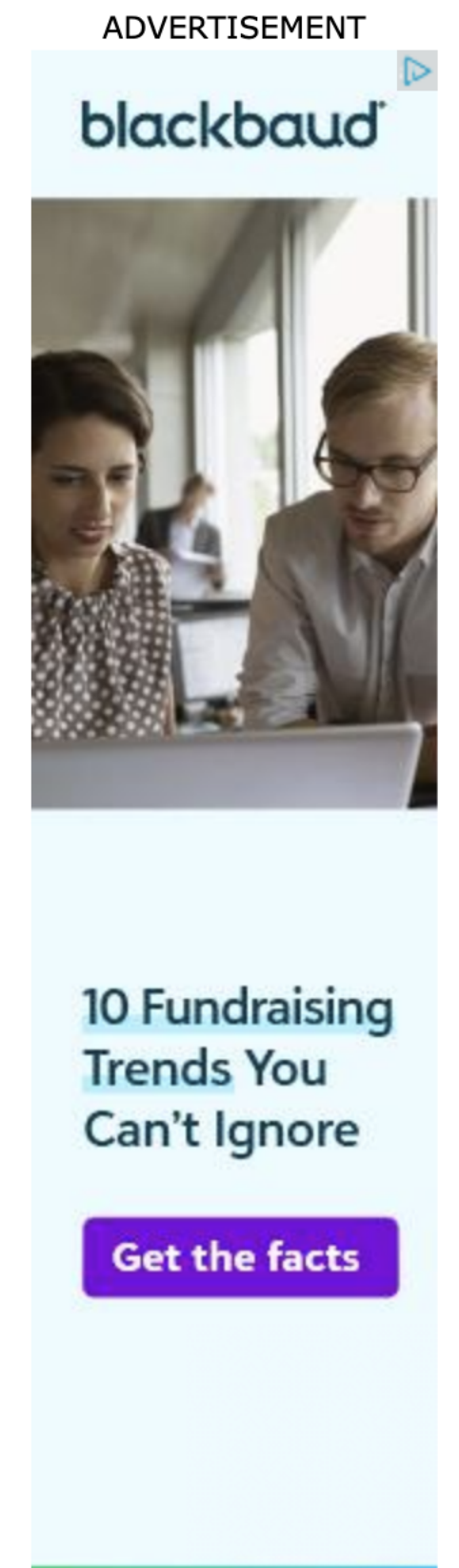
Learn how to create a draggable HTML element with JavaScript and CSS.

## Draggable DIV Element




## Create a Draggable DIV Element

Step 1) Add HTML:



Build your

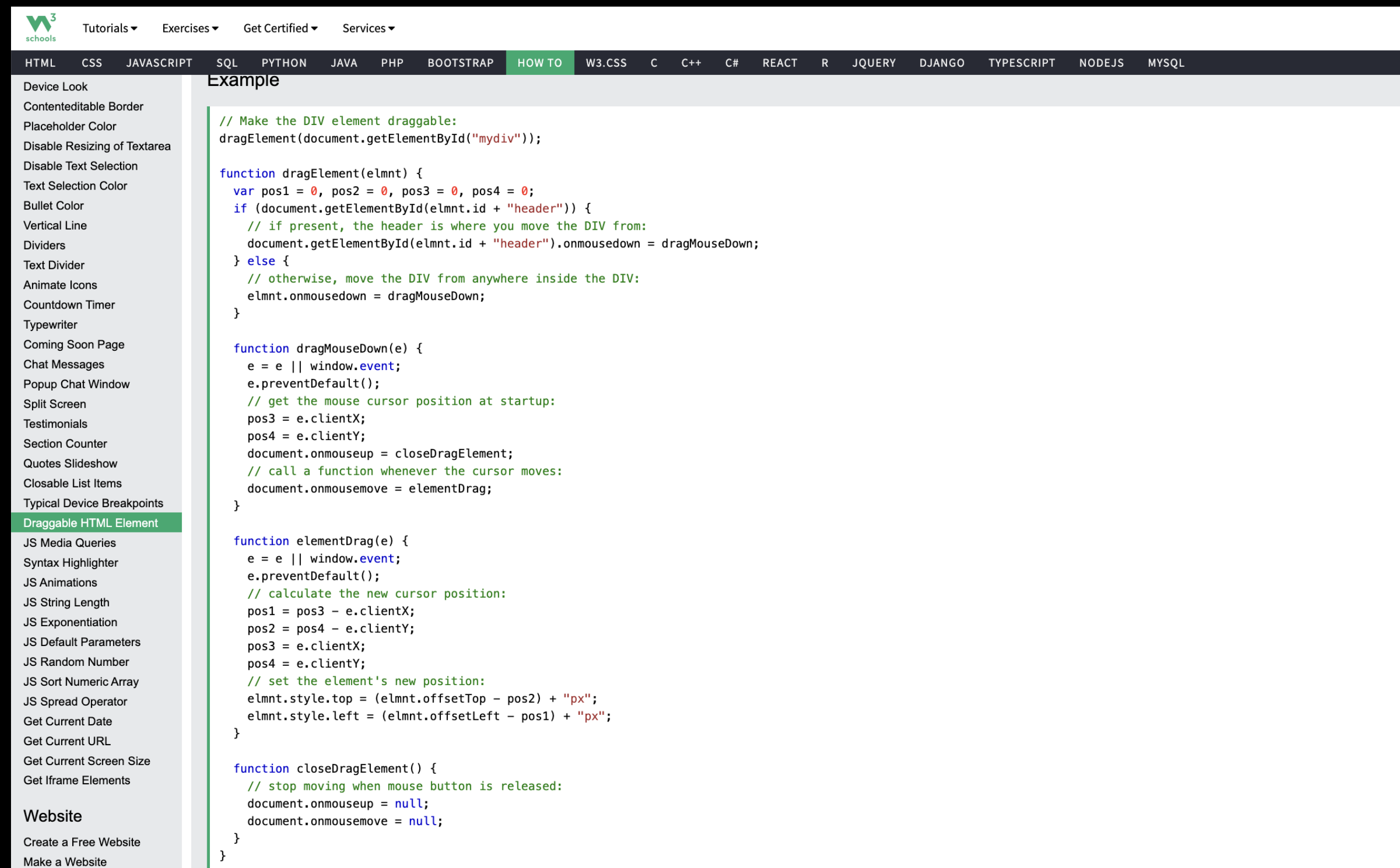
## **According to this site, we need to**

- 1. Apply an id to our welcome window div**
- 2. Make it display absolute (already done )**
- 3. Create a function for dragging the element (that gets the position of the cursor and sets the position of the element to the position of the mouse)**
- 4. Run said function on the welcome window div.**

# Applying an ID to Element

```
<div id="welcome" styles="(styles)">  
    (welcome screen content)  
</div>
```

# Import Dragging Logic from W3 Site [hack.af/make-draggable](http://hack.af/make-draggable)



The screenshot shows a web browser interface with a navigation menu at the top. The menu includes 'Tutorials', 'Exercises', 'Get Certified', and 'Services'. Below the menu is a horizontal navigation bar with various programming and web development topics: HTML, CSS, JAVASCRIPT, SQL, PYTHON, JAVA, PHP, BOOTSTRAP, HOW TO, W3.CSS, C, C++, C#, REACT, R, JQUERY, DJANGO, TYPESCRIPT, NODEJS, and MYSQL. The 'HOW TO' tab is selected. On the left side, there is a sidebar with a list of categories, including 'Device Look', 'Contenteditable Border', 'Placeholder Color', 'Disable Resizing of Textarea', 'Disable Text Selection', 'Text Selection Color', 'Bullet Color', 'Vertical Line', 'Dividers', 'Text Divider', 'Animate Icons', 'Countdown Timer', 'Typewriter', 'Coming Soon Page', 'Chat Messages', 'Popup Chat Window', 'Split Screen', 'Testimonials', 'Section Counter', 'Quotes Slideshow', 'Closable List Items', 'Typical Device Breakpoints', 'Draggable HTML Element', 'JS Media Queries', 'Syntax Highlighter', 'JS Animations', 'JS String Length', 'JS Exponentiation', 'JS Default Parameters', 'JS Random Number', 'JS Sort Numeric Array', 'JS Spread Operator', 'Get Current Date', 'Get Current URL', 'Get Current Screen Size', 'Get Iframe Elements', 'Website', 'Create a Free Website', and 'Make a Website'. The 'Draggable HTML Element' category is highlighted. The main content area is titled 'Example' and contains the following JavaScript code:

```
// Make the DIV element draggable:
dragElement(document.getElementById("mydiv"));

function dragElement(elmnt) {
  var pos1 = 0, pos2 = 0, pos3 = 0, pos4 = 0;
  if (document.getElementById(elmnt.id + "header")) {
    // if present, the header is where you move the DIV from:
    document.getElementById(elmnt.id + "header").onmousedown = dragMouseDown;
  } else {
    // otherwise, move the DIV from anywhere inside the DIV:
    elmnt.onmousedown = dragMouseDown;
  }

  function dragMouseDown(e) {
    e = e || window.event;
    e.preventDefault();
    // get the mouse cursor position at startup:
    pos3 = e.clientX;
    pos4 = e.clientY;
    document.onmouseup = closeDragElement;
    // call a function whenever the cursor moves:
    document.onmousemove = elementDrag;
  }

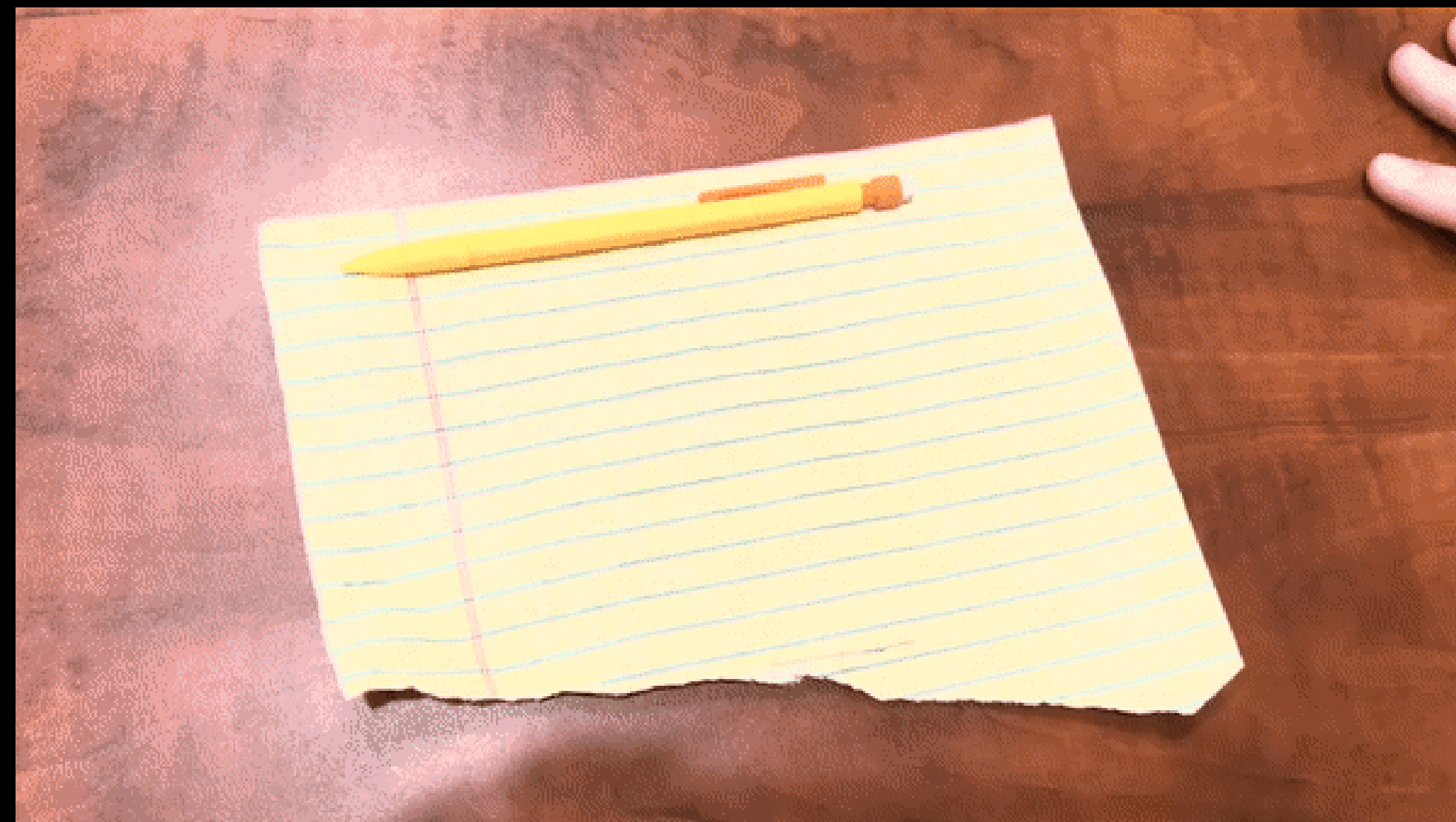
  function elementDrag(e) {
    e = e || window.event;
    e.preventDefault();
    // calculate the new cursor position:
    pos1 = pos3 - e.clientX;
    pos2 = pos4 - e.clientY;
    pos3 = e.clientX;
    pos4 = e.clientY;
    // set the element's new position:
    elmnt.style.top = (elmnt.offsetTop - pos2) + "px";
    elmnt.style.left = (elmnt.offsetLeft - pos1) + "px";
  }

  function closeDragElement() {
    // stop moving when mouse button is released:
    document.onmouseup = null;
    document.onmousemove = null;
  }
}
```

**Super confused? Here's what is happening:**

**The `dragElement` function lets you drag and move the window around.**

**If there's a header element (like a handle), it hooks up the magic of dragging to it (but not the rest of your window).**



**We can apply the logic by simply calling the dragElement function**

```
dragElement(document.getElementBy  
Id("window_name"));
```



# Inside your welcome window, add a handle!

```
<div id="welcome">  
  <p id="welcomeheader">  
    Handle  
  </p>  
</div>
```

**“window\_nameheader”**

Handle



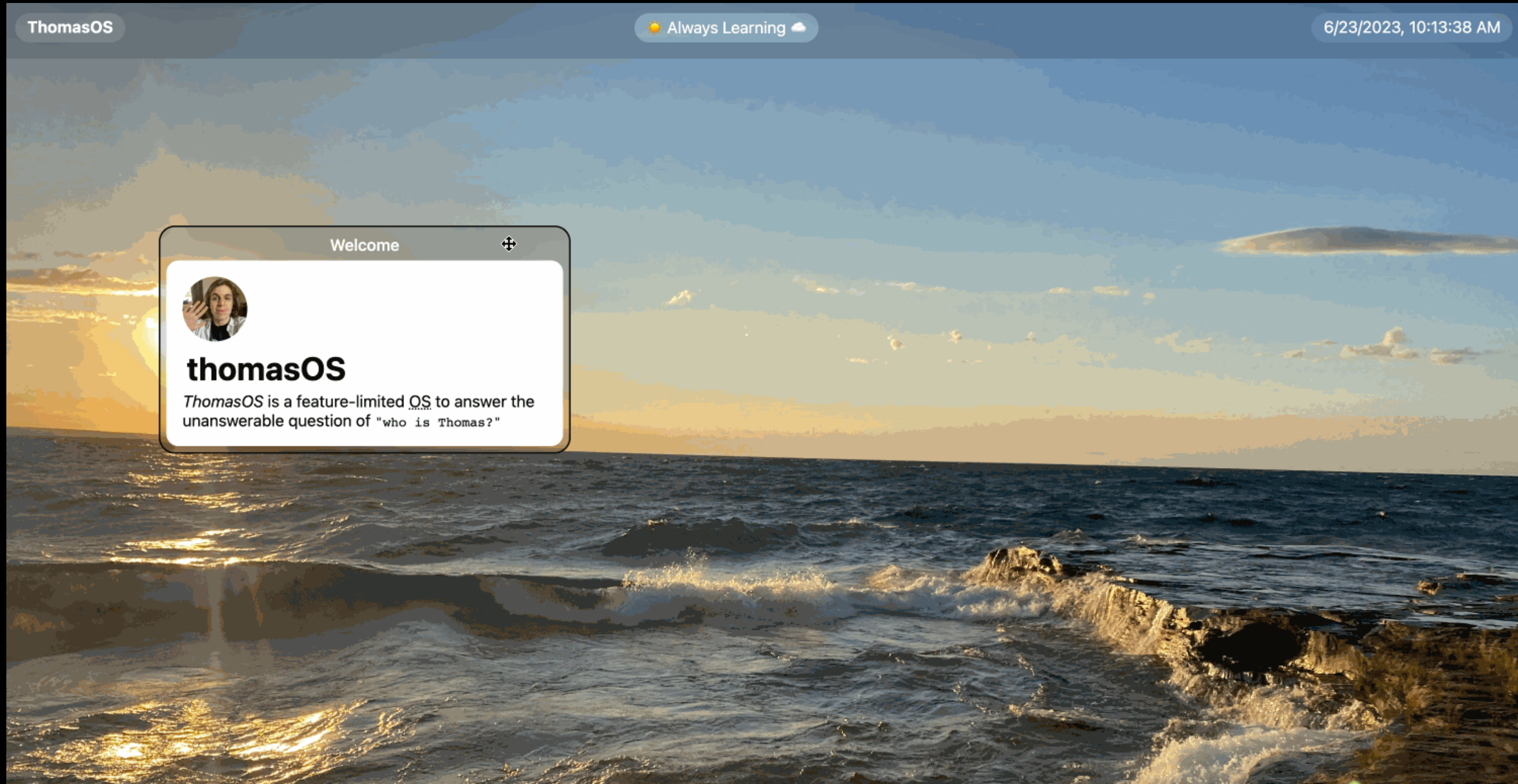
**thomasOS**

*ThomasOS* is a feature-limited OS to answer the unanswerable question of "who is Thomas?"

**Now Customize the Handle to look however you'd like! Have fun with it!**

**Feel free to throw in an icon or just make it a dark section**

# One way:



ThomasOS

Always Learning

6/23/2023, 10:13:38 AM

Welcome



**thomasOS**

ThomasOS is a feature-limited OS to answer the unanswerable question of "who is Thomas?"

# Opening & Closing The Window

Alright, so let's break down this challenge

- 1. We need a way to toggle whether the window is visible**
  - We can use the display property (which can be set to none to hide an element)**
- 2. We need to create a function to open the window**
  - We can pass in an element and enable its visibility**
- 3. We need to create a function to close the window**
  - We can pass in an element and disable its visibility**
- 4. Create buttons for opening & closing the window**
  - using the onClick function on our elements we can run these functions**

# Let's Select the Screen

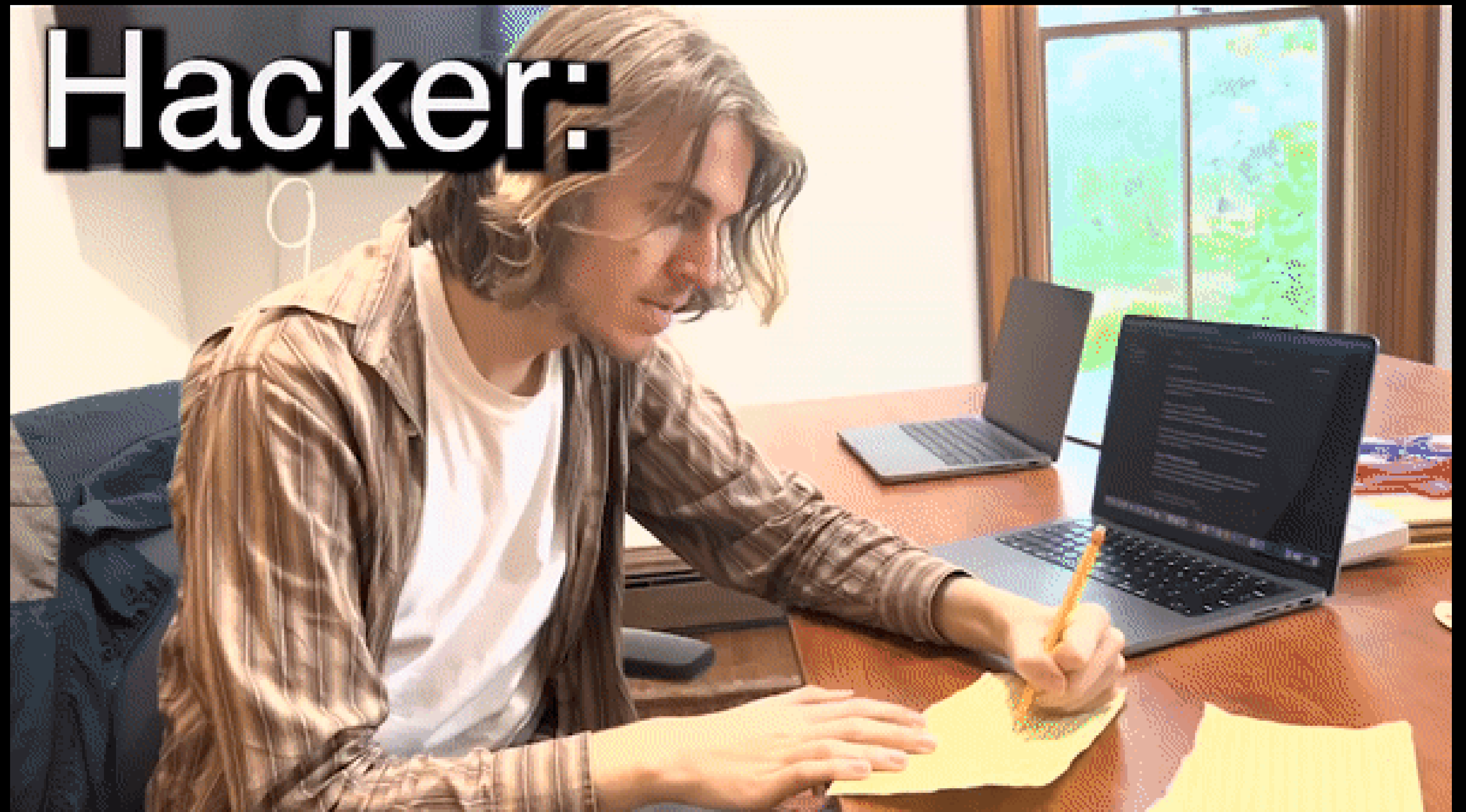
```
var welcomeScreen = document.querySelector("#welcome")
```

# Let's make a function to close our window

```
function closeWindow(element) {  
    element.style.display = "none"  
  
}
```

**Still don't understand functions?**

**Here ya go:**





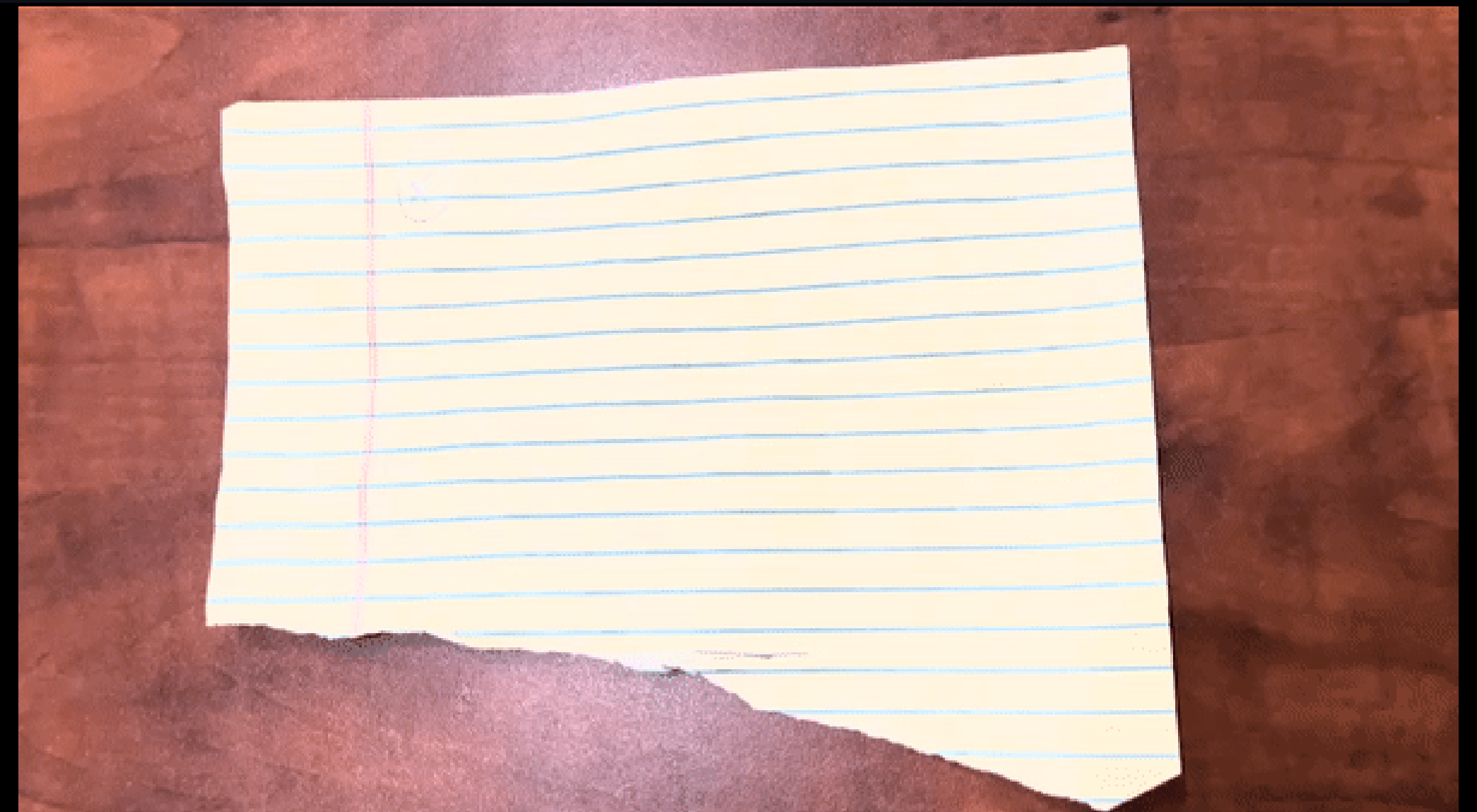
# Let's make a function to open our window

```
function openWindow(element) {  
    element.style.display = "flex"  
}
```

# Create Close Button

```
<p style="cursor: pointer" id="welcomeclose">Close</p>
```

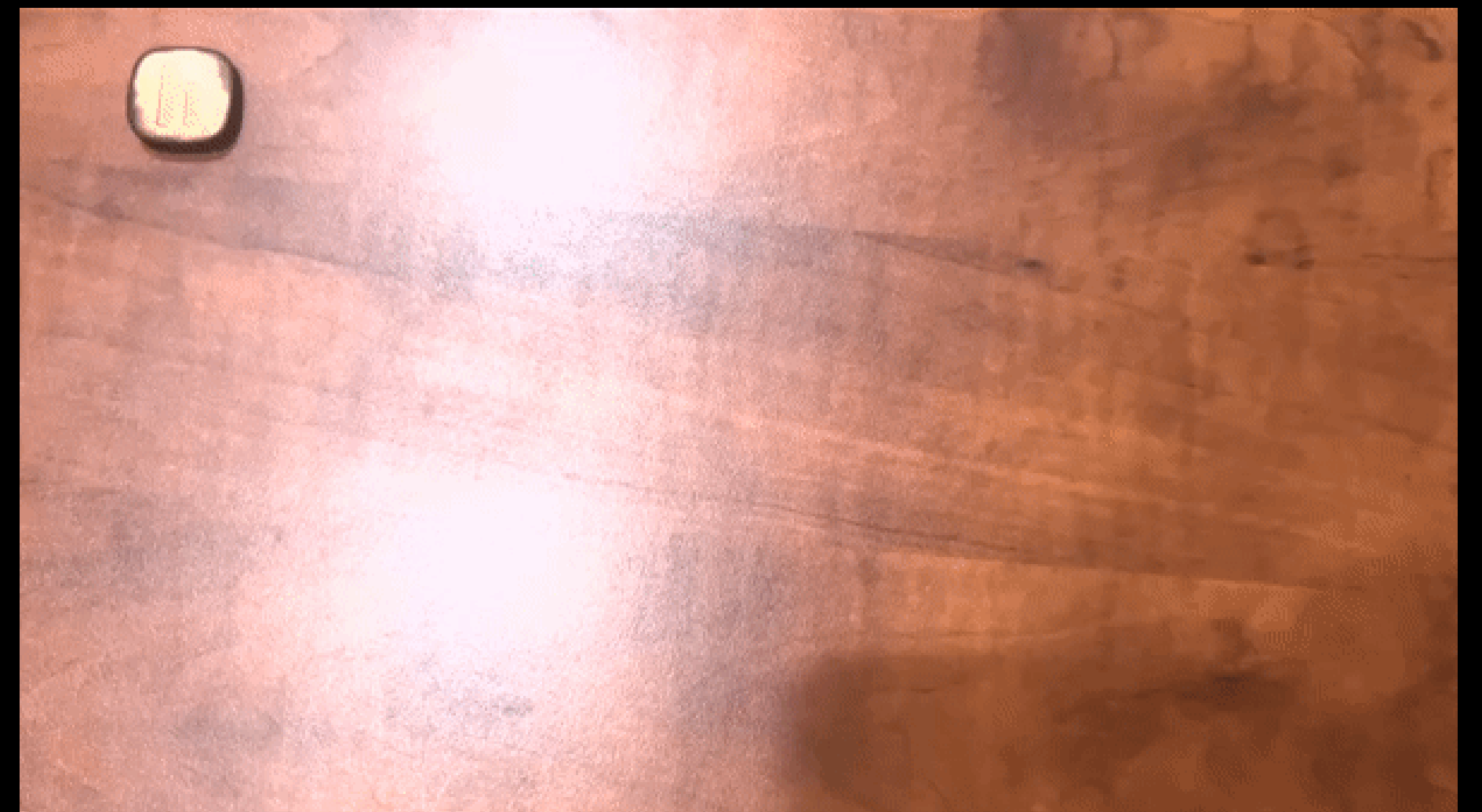
**^ Inside welcome  
screen**



# Create Open Button

```
<p style="cursor: pointer" id="welcomeopen">ThomasOS</p>
```

**^ In tab bar  
or where ever  
you want**



# Select Buttons!

```
var welcomeScreenClose = document.querySelector("#welcomeclose")
```

```
var welcomeScreenOpen = document.querySelector("#welcomeopen")
```

# Add Event Listeners



**Event Listener**

# Add Event Listeners

```
welcomeScreenClose.addEventListener("click", function() {  
    closeWindow(welcomeScreen);  
});
```

```
welcomeScreenOpen.addEventListener("click", function() {  
    openWindow(welcomeScreen);  
});
```

**Awesome it's works!**

**Now Style your Buttons!**

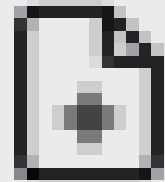
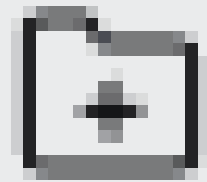

# one way to do it:

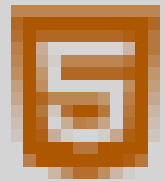




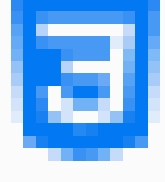





**Our code is still a bit  
messy...**

**To fix this we can separate  
our content from our styles!**

Files   

-  index.html 
-  beautifulImage.png
-  dragElement.js
-  script.js
-  style.css 

 index.t

>  div

<

---

da

re

12

13

# Inside styles.css...

```
.className {  
    property1: 5;  
    property2: "16px";  
    property3: "hidden";  
}
```

# For Example:

```
.closebutton {  
  width: 16px;  
  height: 16px;  
  cursor: pointer;  
  background-color: #EC6B5E;  
  border-radius: 16px;  
  border: solid 1px rgba(0, 0, 0, 0.25);  
  margin-left: 6px;  
}
```

```
<div class="closebutton" id="welcomeclose"></div>
```

**HTML ^**

**<- CSS**

# For Example:

```
.window {  
  border: solid;  
  display: flex;  
  flex-direction: column;  
  border-radius: 16px;  
  position: absolute;  
  backdrop-filter: blur(4px);  
  background-color: rgba(0, 0, 0, 0.125);  
}
```

```
.windowheader {  
  width: 100%;  
  display: flex;  
  align-items: center;  
  cursor: grab;  
  justify-content: space-between;  
  margin-top: 8px;  
}
```

```
.headertext {  
  margin: 0px;  
  color: #fff;  
  font-weight: 500  
}
```

**Remember this is**

**YOUR Operating System**

**Have fun with it!**

**See You Next Time :D**