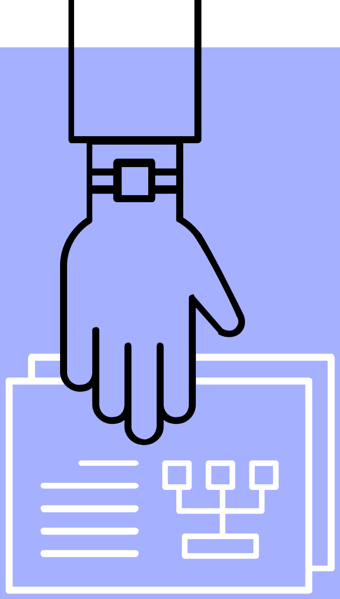
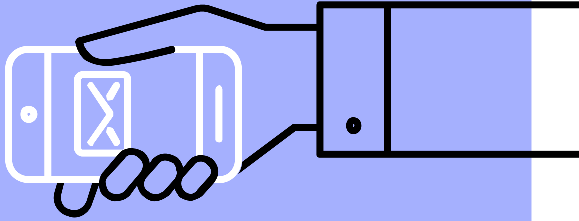
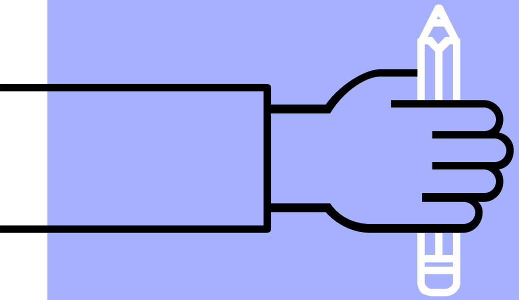
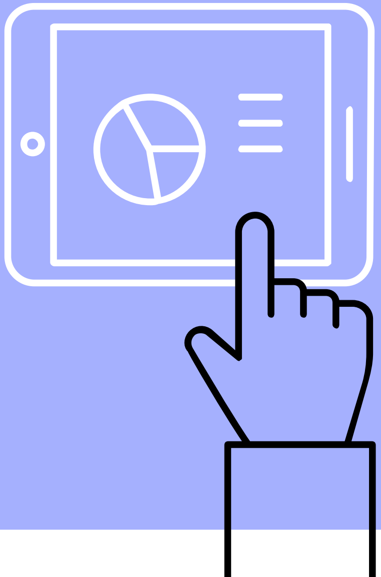




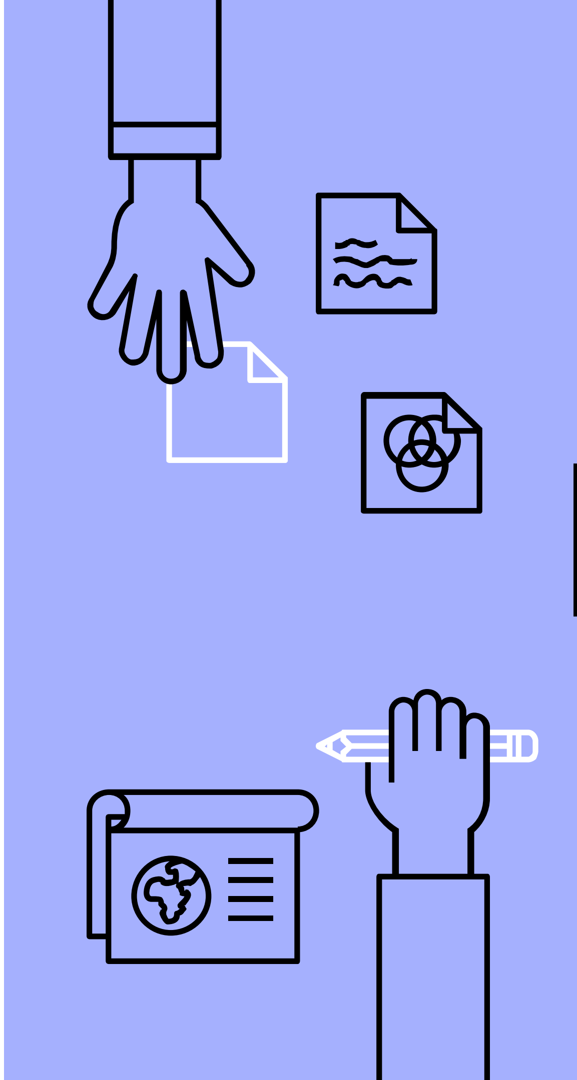
Intro to Javascript

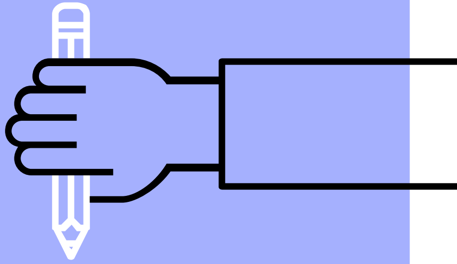




Do Now (5 mins)

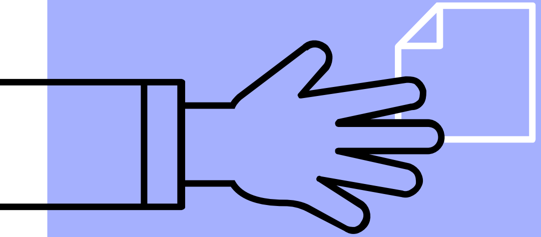
Name as many user inputs as you can think of for an electronic device.





Objectives

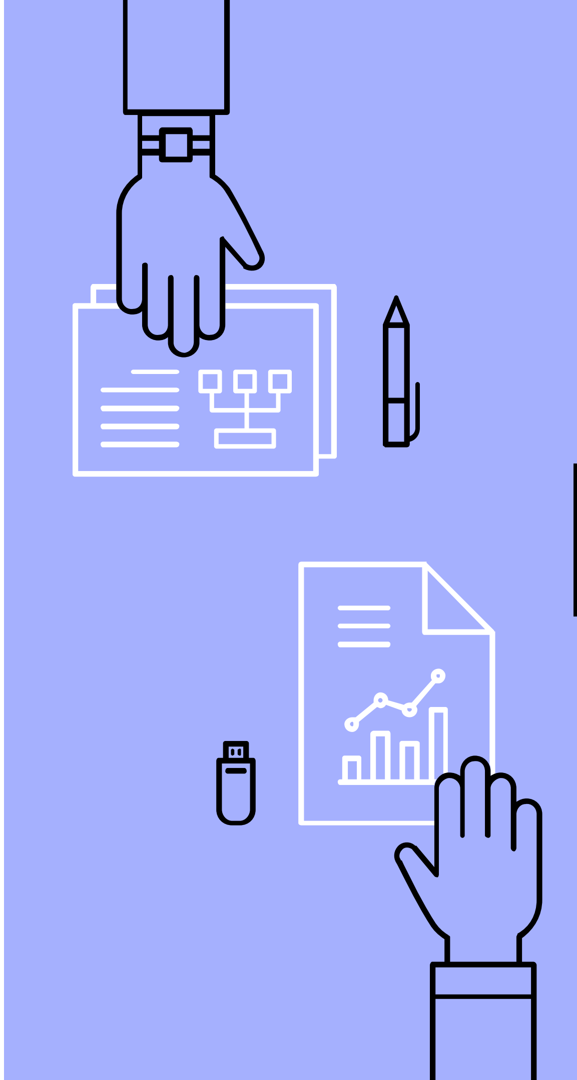
- Learn how to code user input
- Add event listeners to an ID
- Create a dummy login page



User input

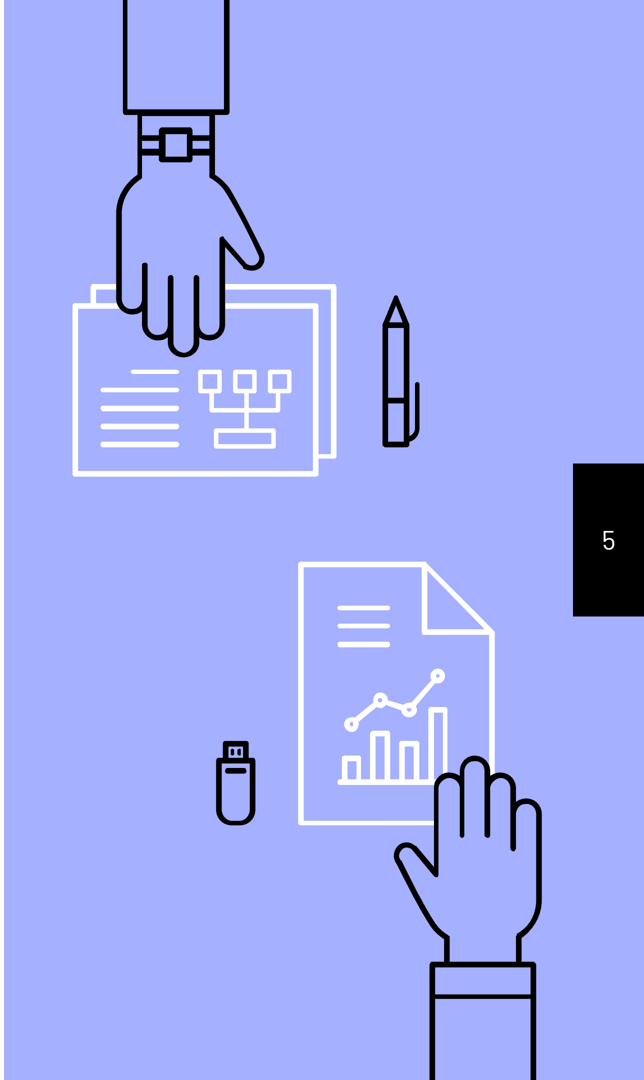
User input is the way humans interact with an electronic device. There are 3 main input types:

- Click- clicking a mouse (left, right, double, triple, etc)
- Keyboard- typing different keys
- Touch- tap, swipe, long press, etc



Inputs and event listeners

We will be looking at two new HTML elements and how we can manipulate them with JS. They work just like any element, but to make it functional, we need to grab the ID using JS, just like we did with span.



Button

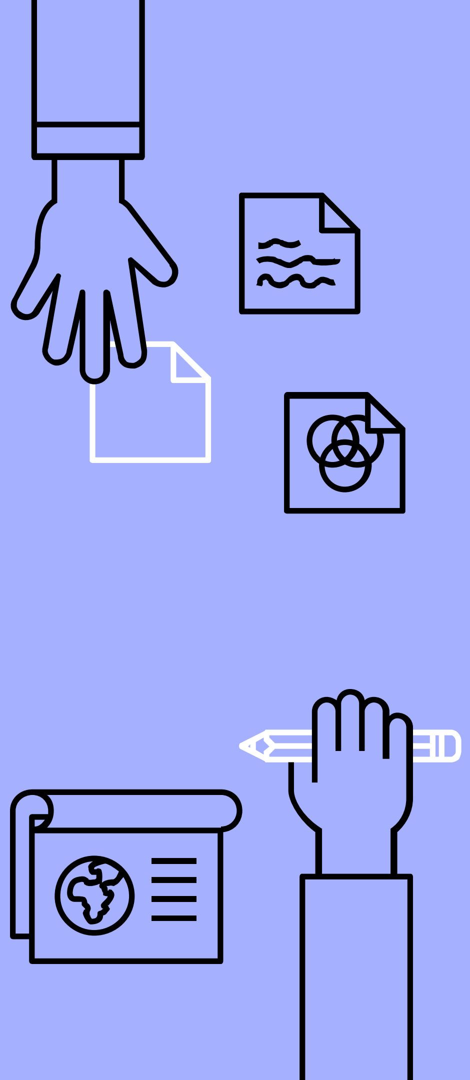
HTML

```
<button id=”clickMe”>
This is a button!
</button>
```

JS

```
let clickButton= document.querySelector(“#clickMe”);
```

- Remember, you can use let or var to create a variable



Input

The first one is a text field to type in words.

You can also create a button by specifying the type of input, but it’s up to you which you use.

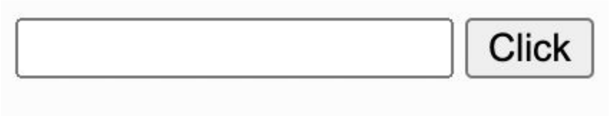
** NOTE: Inputs do not have closing tags! All information exists inside of its own tag.

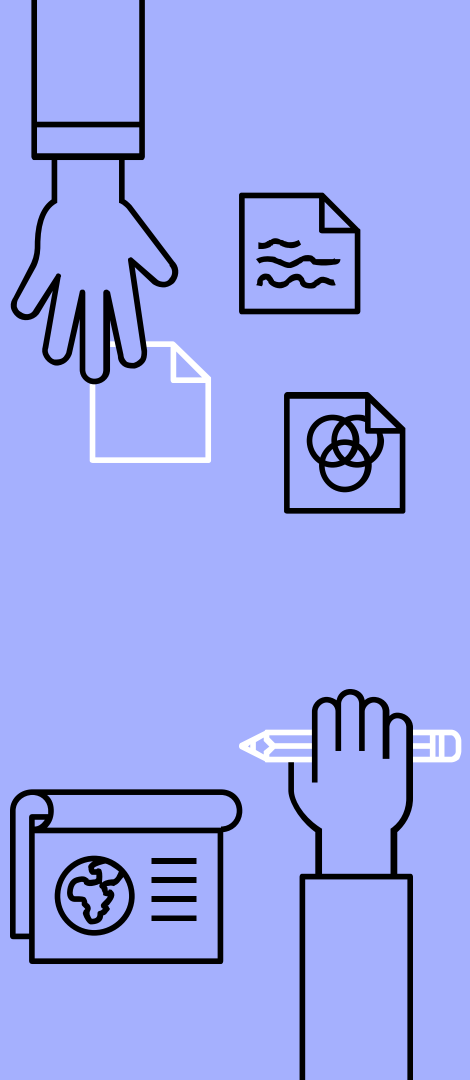
HTML

```
<input type=”text”>


<input type=”button”
value=”Click”>
```

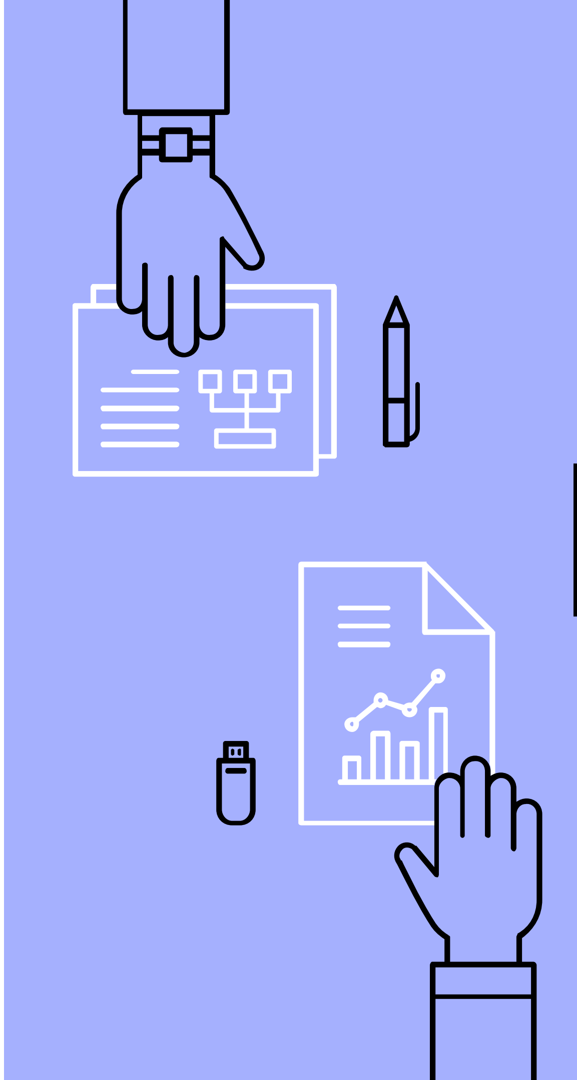
PREVIEW





Using IDs (& JS!)

Using JS, you can grab an ID or class from an input, store it in a variable, and now you can use **event listeners** to make the input do different things like go to a different webpage, make something appear/disappear, log in, finish a search or query, etc. Using event listeners expands your use of JS immensely.



Sample Event Listener

Let’s say you click a button, like on slide 6:

```
clickButton.addEventListener(‘click’, e=>{
CODE HERE;
}
```

- variable taken from the ID button
- addEventListener is appended; it adds the event to the ID and waits for a response
- click is the type of response it is listening for
- e=> is what happens after the event is true

Sample Event Listener

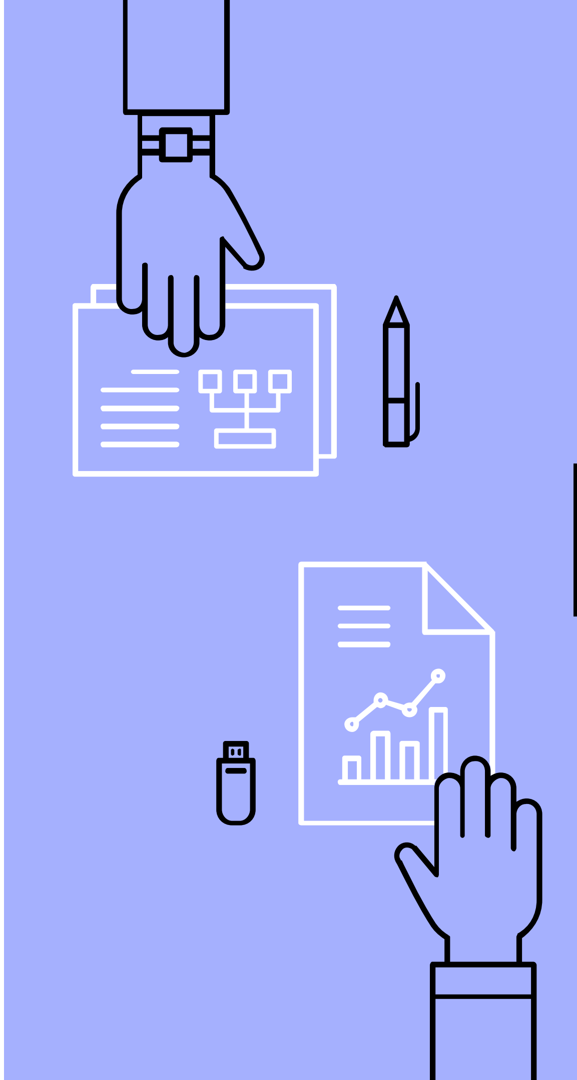
Here’s what it would look like in psuedocode:

```
clickButton.addEventListener(‘click a mouse’,
when someone left clicks the mouse button=>{
if (password is correct){
    log in to instagram;
} otherwise {
    Error message, try again;
};
}
```

Sample Event Listener

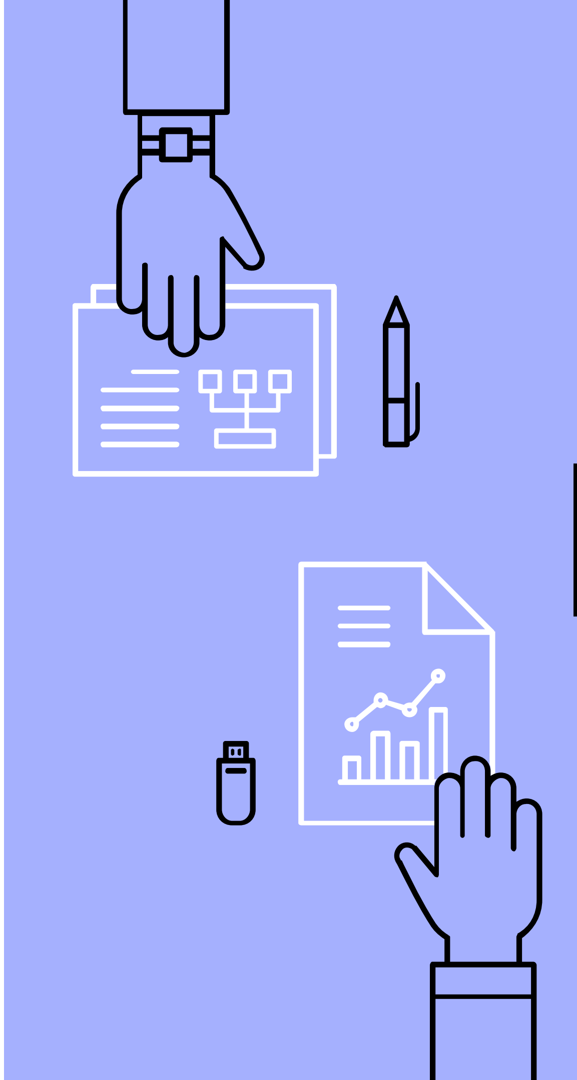
Here’s what it would look like in JS code:

```
clickButton.addEventListener(‘click’, e=>{
if (password==”parkeast123”){
      console.log(“user has logged in”);
} else {
      console.log(“password incorrect”);
};
}
```



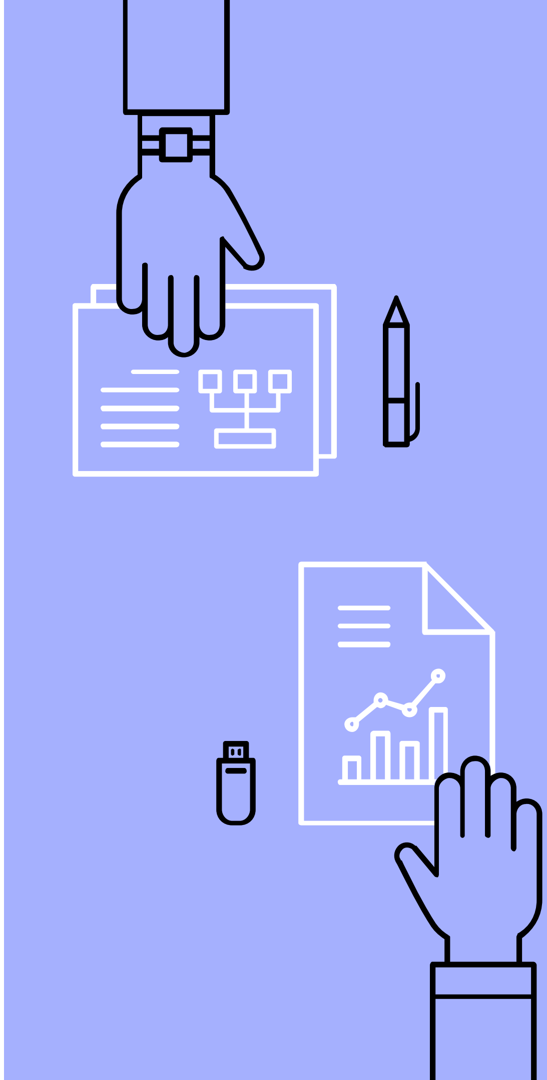
Let’s add one more thing…

Let’s say you want to make sure to lock the account after 3 attempted logins. You can add a variable that counts how many times someone tried to log in with a variable. If the number ever reaches 3, you can lock the account for a certain amount of time or until you reset the password.



With password check (nested conditional)

```
clickButton.addEventListener(‘click’, e=>{
if (password==”parkeast123”){
      console.log(“user has logged in”);
} else {
      console.log(“password incorrect”);
};
      if (wrong>=3){
            Error message, lock account;
}
};
}
```



Code-along!

Let’s work through today’s lab together.