

---

# INTRODUCTION TO JAVASCRIPT

## Lecture 14

---

# TODAY'S TOPICS



- DOM Events
- Event Listeners
- Event Types
- **Participation:** Color Picker

---

# ANNOUNCEMENTS

- Sign-in Sheet



**QUESTIONS?**

# DOM EVENTS

---

# DOM EVENTS

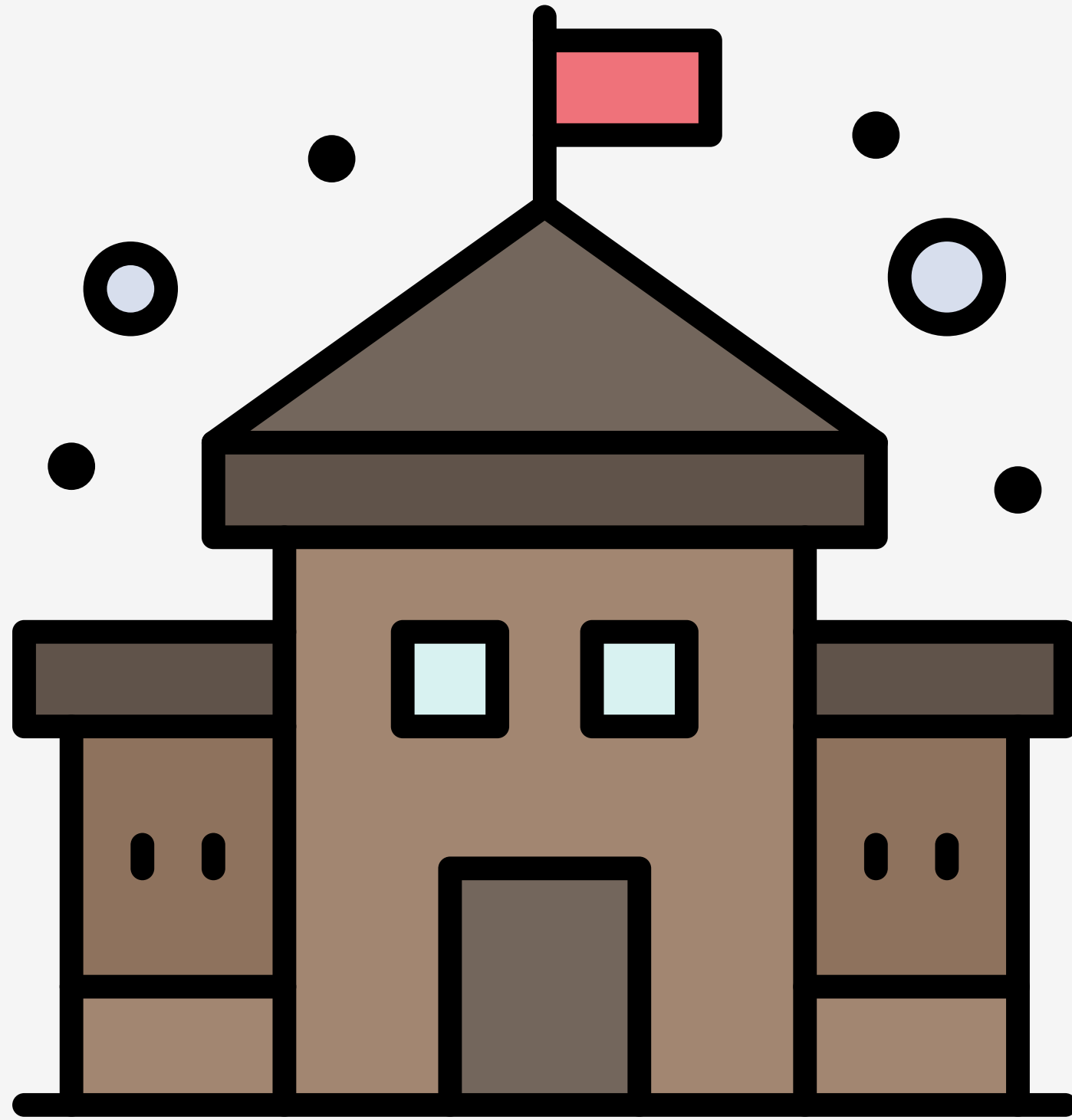


- **DOM Events** are notifications that something interesting has happened
- **DOM Events** can include a user action or change to the status of the web browser
- There are **DOM Events** for the keyboard, mouse, touch, clipboard, media, view, printing, drag & drop, animation, forms, and more
- Each **DOM Event** is represented by an **Event Object**

# EVENT LISTENERS

---

# EVENT LISTENERS



- Event Listeners is the preferred method for responding to DOM Events
- Event Listeners are created by using the `addEventListener()` method
- The method takes two arguments, an event type and a function



```
const $button = document.getElementById('button')
```

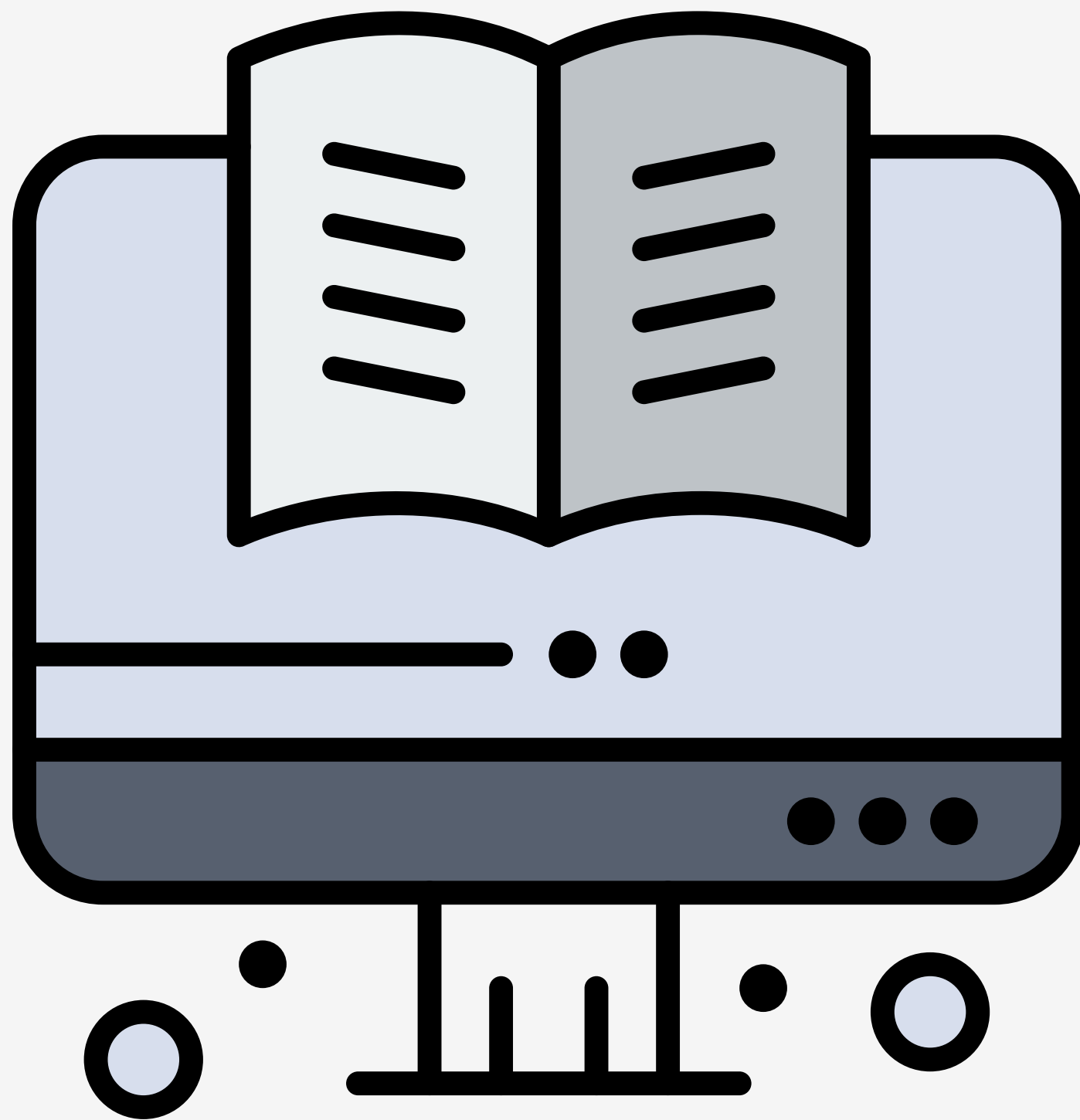
```
// using an anonymous function
```

```
$button.addEventListener('click', function () {  
    console.log(`The button was clicked`)  
})
```

# EVENT TYPES

---

# MOUSE EVENT TYPES



- There are many different ways that a mouse can interact with the page
- The most common mouse event types are: **click**, **mouseover**, and **mouseout**

```
const $button = document.getElementById('button')

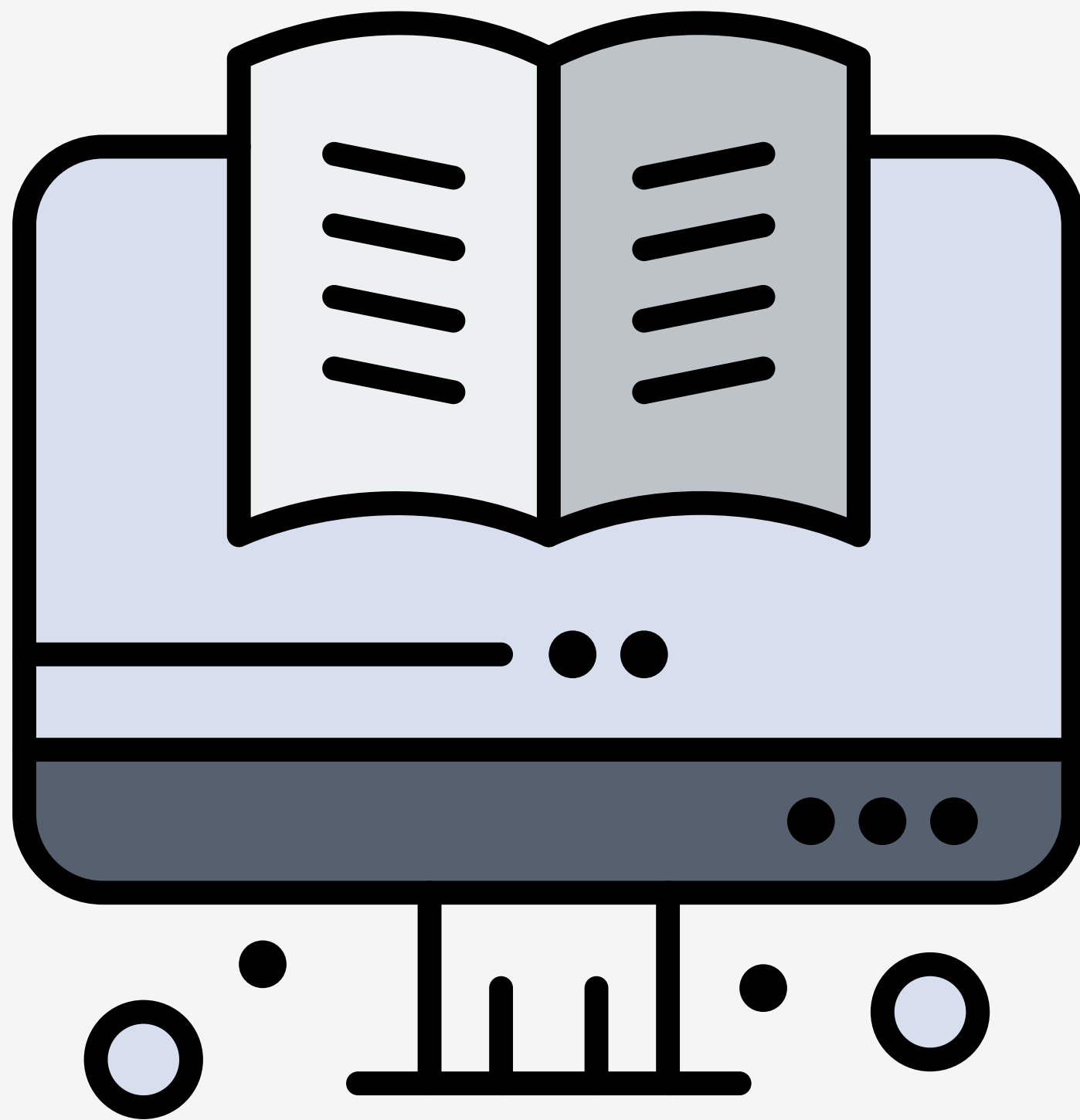
$button.addEventListener('mouseover', function() {
    $button.textContent = `Don't you do it!`
})

$button.addEventListener('click', function() {
    $button.textContent = `You clicked the button!`
})

$button.addEventListener('mouseout', function() {
    $button.textContent = `Don't do it again!`
})
```

---

# KEYBOARD EVENT TYPES

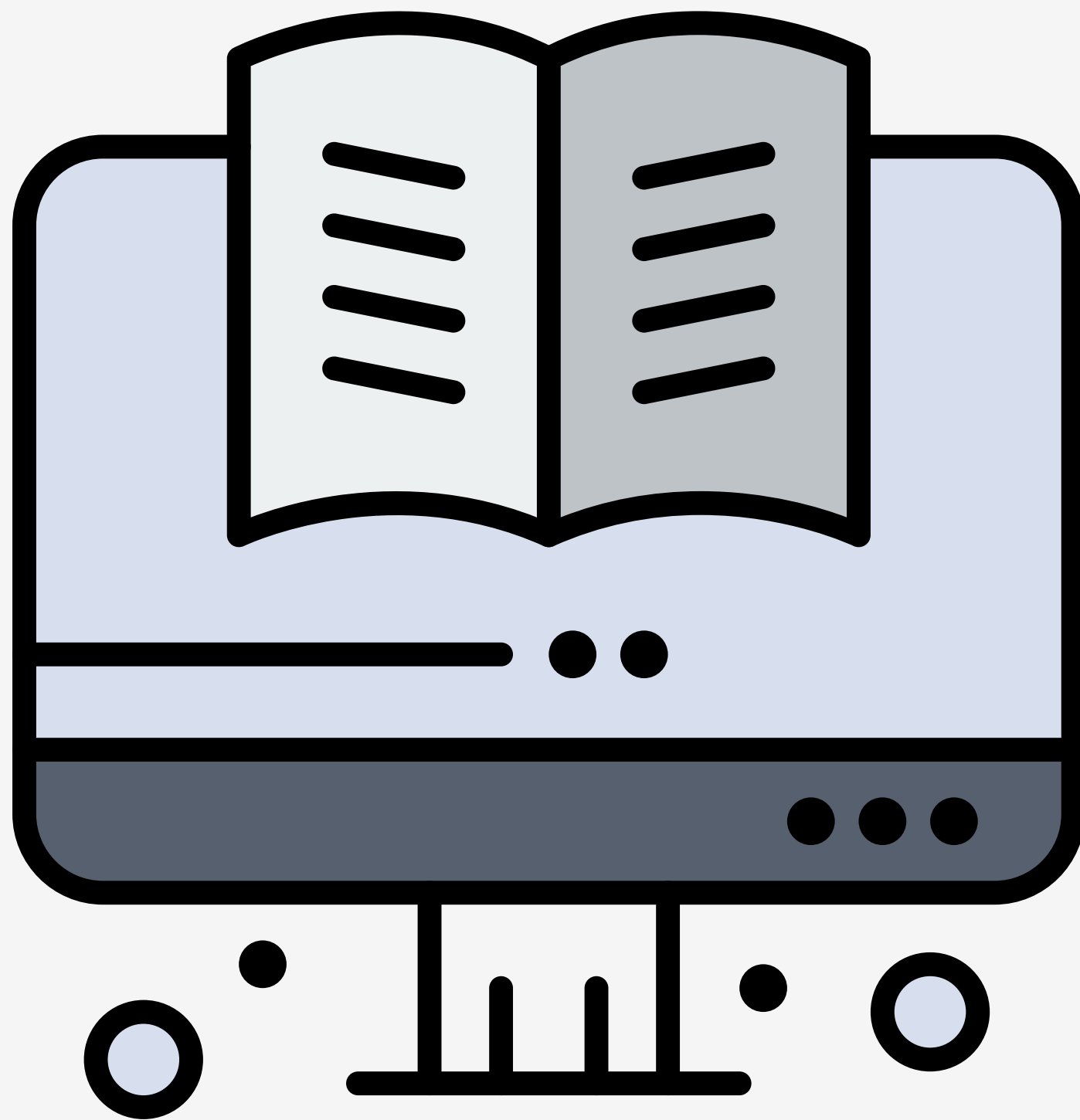


- The keyboard can also be used to interact with the browser
- There are three keyboard event types: **keydown**, **keyup**, and **keypress**
- The Event Object provide a key code that identifies which key was pressed
- The **key** property provides a string that represents the key pressed
- The older **keyCode** property provides a number that represents the key pressed

```
document.addEventListener("keydown", function (e) {  
    if (e.key !== undefined) {  
        console.log(e.key)  
    } else if (e.keyCode !== undefined) {  
        console.log(e.keyCode)  
    }  
})
```

---

# OTHER EVENT TYPES



- There are many other event types that represent browser actions.
- The **change** event type is used for form inputs like `<select>`, `<input>` and `<textarea>`
- The **focus** and **blur** event types are used when element has change of focus
- The **touchstart**, **touchmove**, and **touchend** events types are used with touch screens

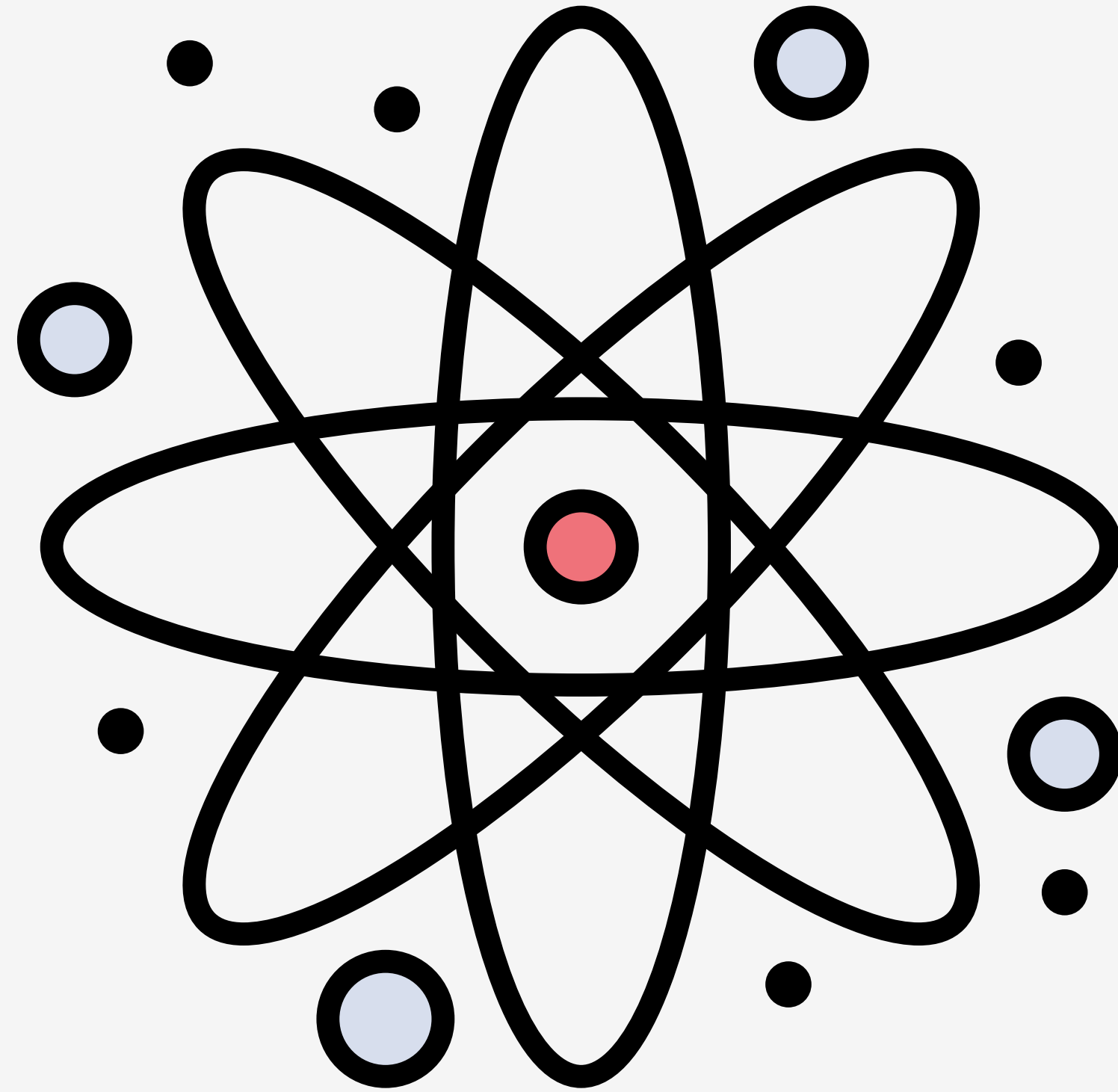
**HANDS-ON**



# **LISTENERS ON MULTIPLE ELEMENTS**

---

# USING A LOOP



- Retrieve all the elements as a **NodeList** or **HTMLCollection**.
- Predefined the event handler function
- Use a loop to add the event listener to each element

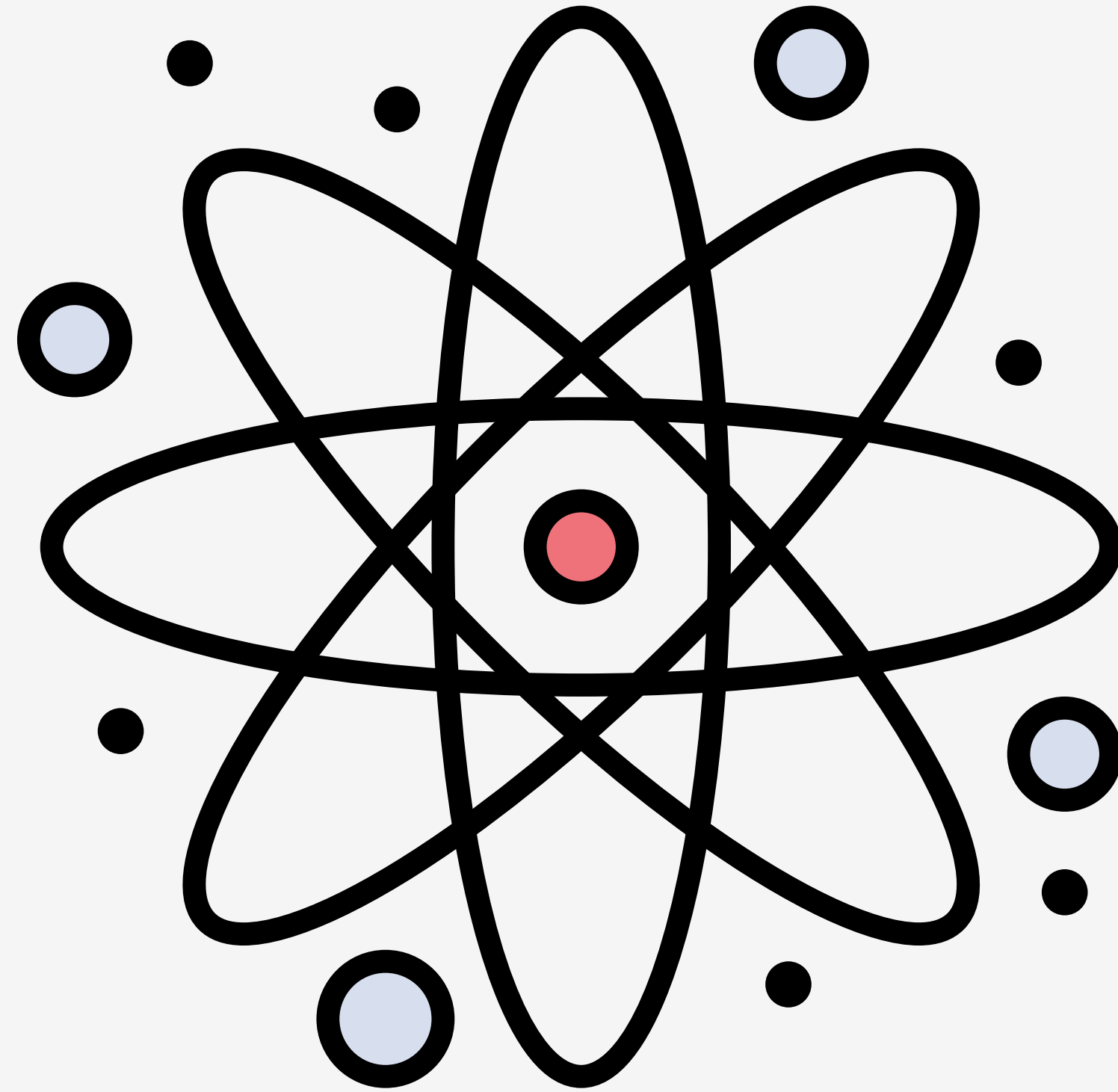
```
// getting all .button elements
const $buttons = document.querySelectorAll('.button')

// predefine button action
function buttonAction (e) {
    console.log(e.target.textContent)
}

// loop over buttons
// add the event listener
for ($button of $buttons) {
    $button.addEventListener('click', buttonAction)
}
```

---

# EVENT DELEGATION



- Utilize the process of event propagation
- Adds event listeners to a parent element of the target element
- Use the `target` property of the Event object to identify which event originated
- Use `classList.contains()` or `closest()` to help identify if `target` is the desired element

```
// parent of the .button elements
const $buttons = document.querySelector('.buttons')

// predefine button action
function buttonAction (e) {
  if (e.target.classList.contains('button')) {
    console.log(e.target.textContent)
  }
}

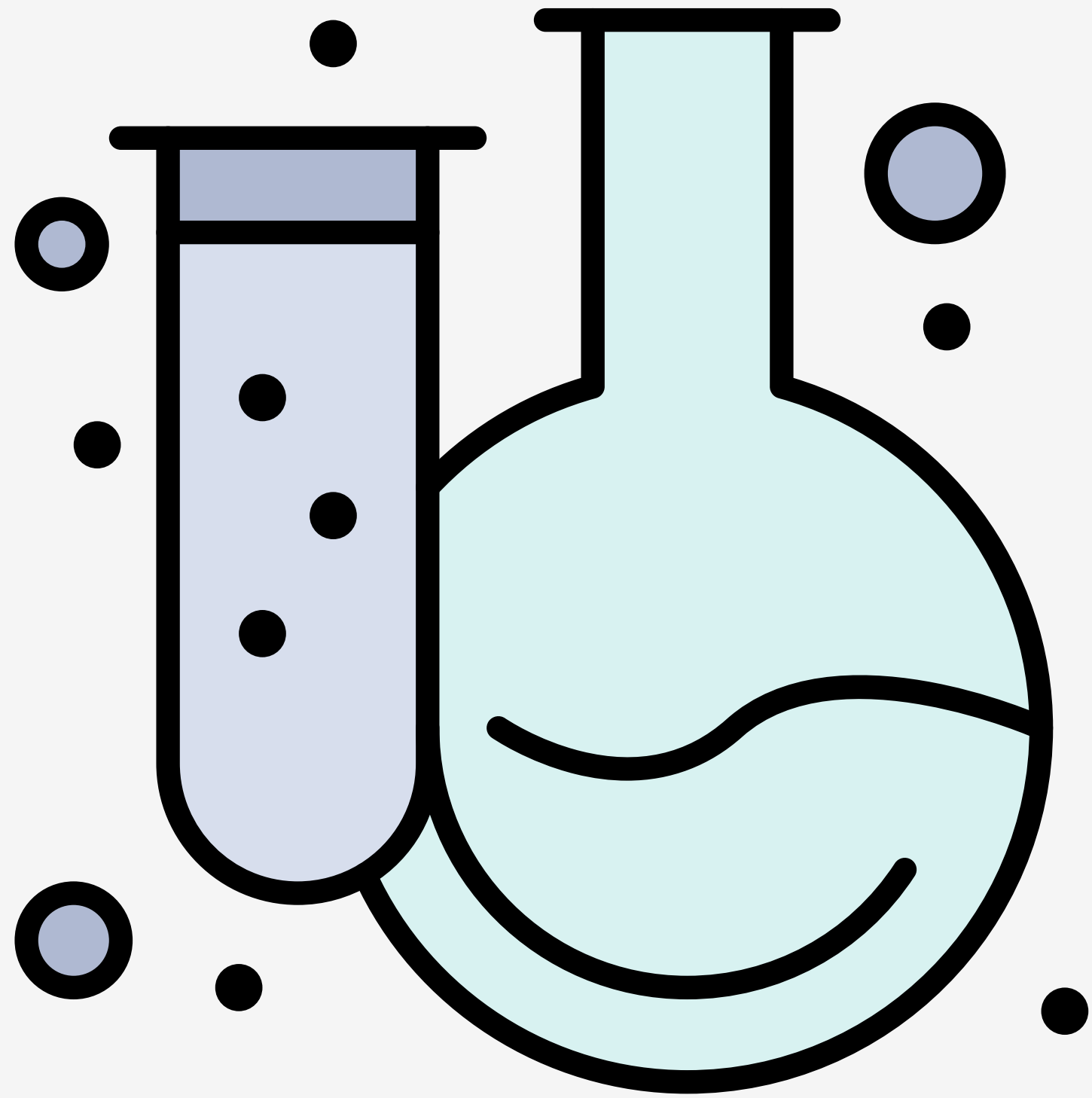
// use event delegation
$buttons.addEventListener('click', buttonAction)
```

**HANDS-ON**

**PRACTICE**

---

# COLOR PICKER

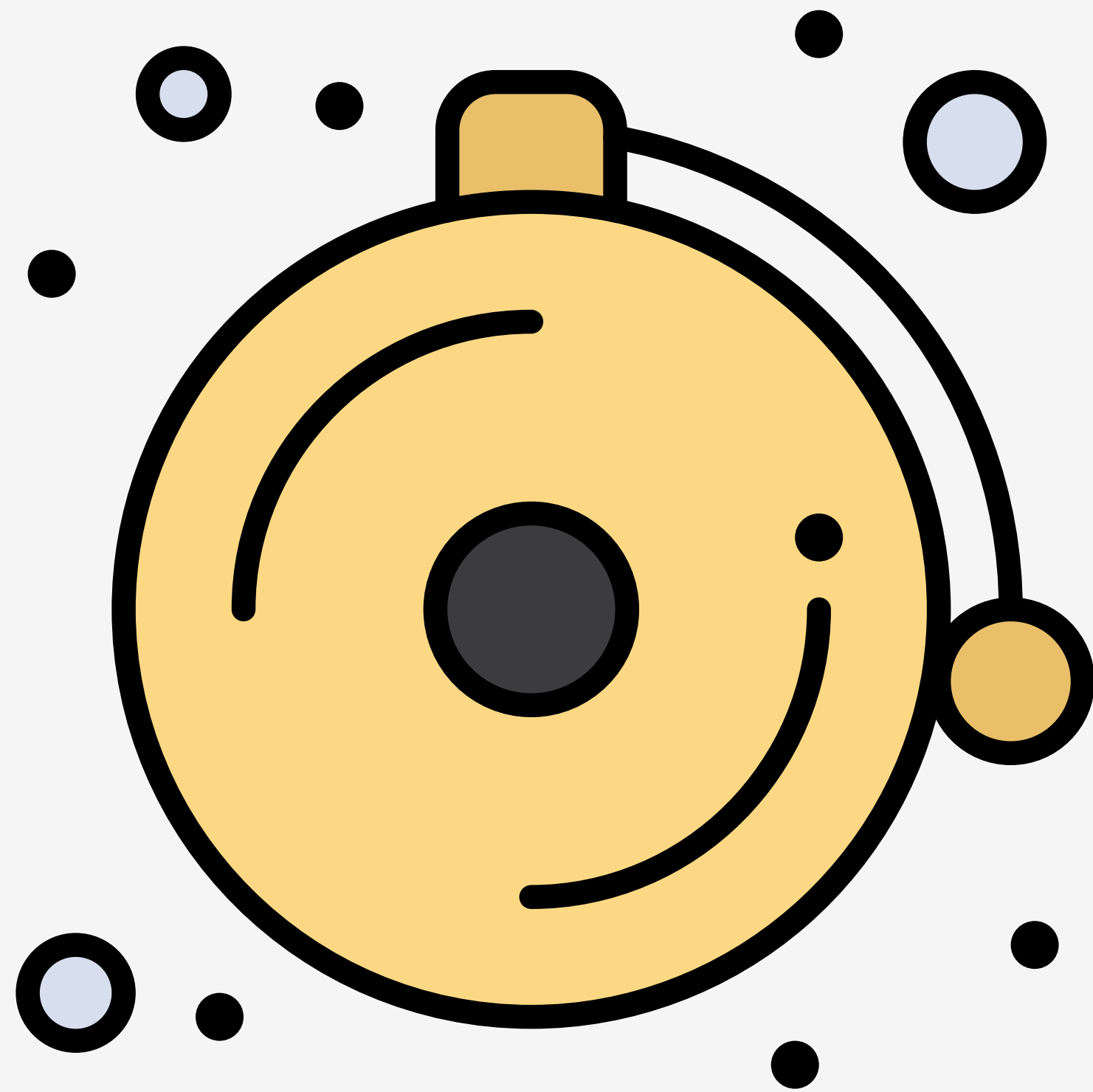


- *FORK THE PEN!*
- Dynamically create color swatches and insert them on to the page
- Add event listeners to the color swatches using event delegation
- Submit the URL to your pen
- *DUE:* Thu. Nov. 7 @ 11:59 PM



---

# NEXT TIME...



- Hands-on: Flashcards
- Project: Memory
- Exercise: Jeopardy