
INTRODUCTION TO JAVASCRIPT

Lecture 13

TODAY'S TOPICS



- Creating DOM Elements
- **Review:** 99 Luftballons
- **Exercise:** Jeopardy

ANNOUNCEMENTS

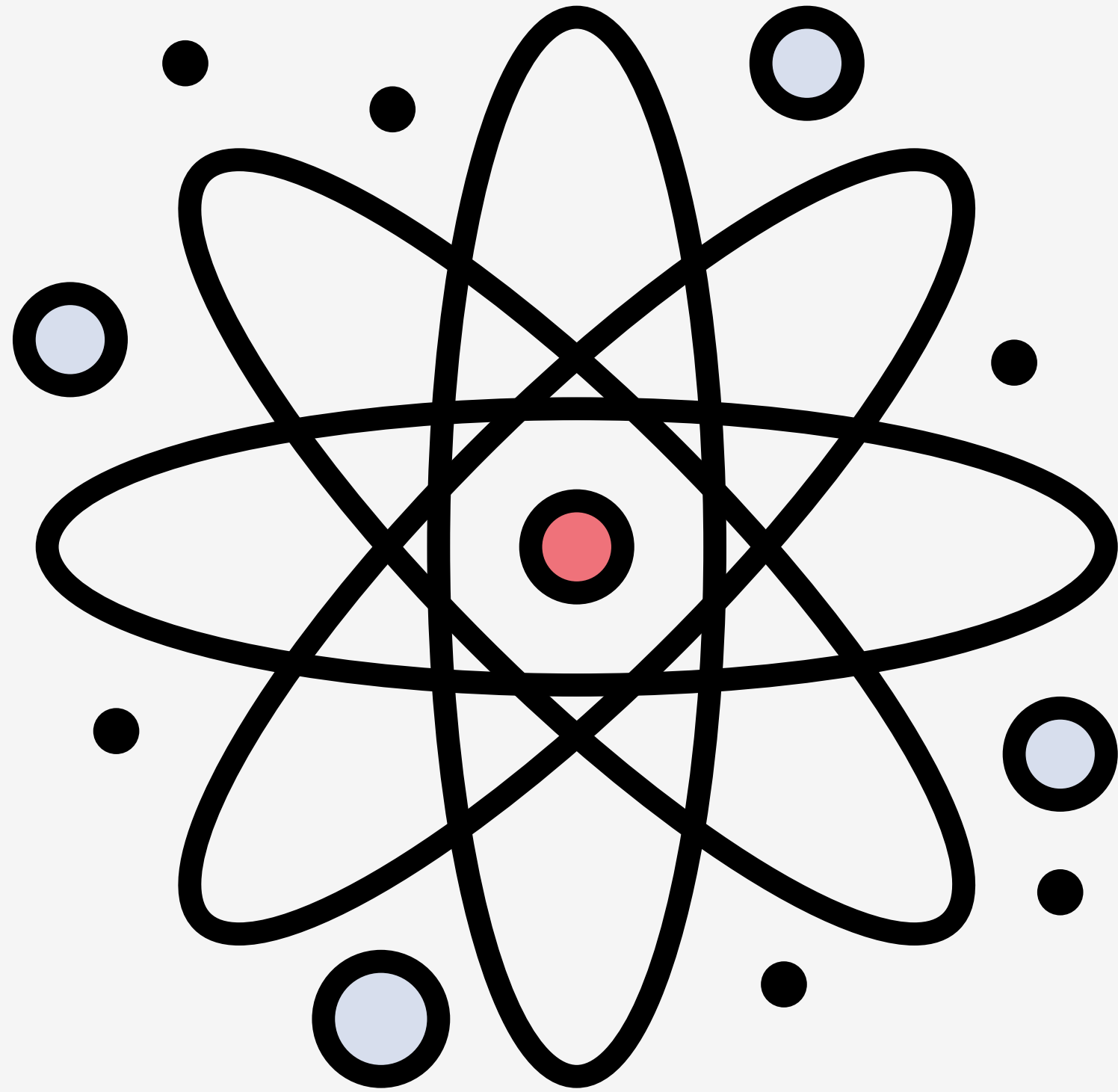


- Sign-in Sheet
- No Classes next week

QUESTIONS?

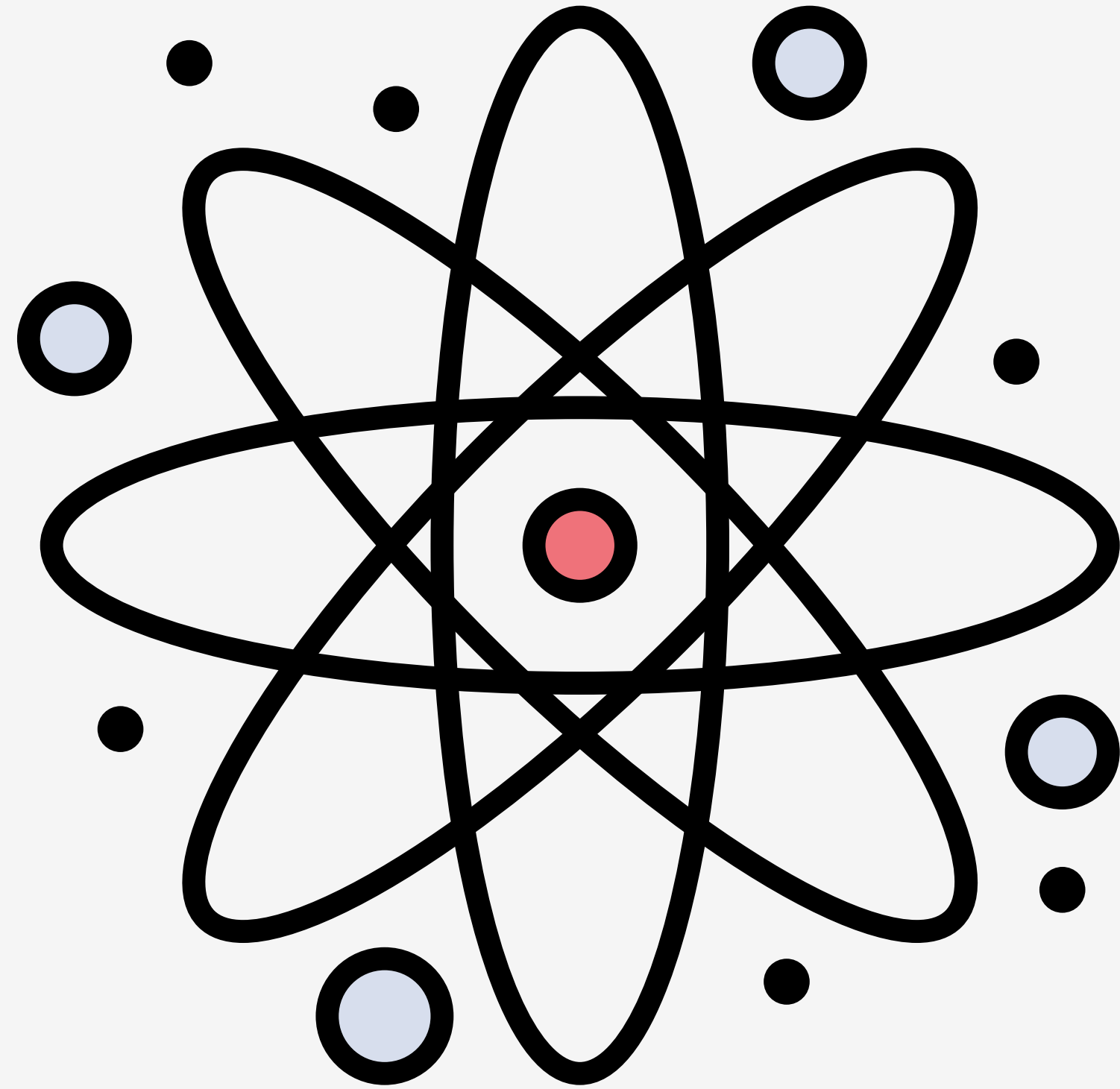
CREATING DOM ELEMENTS

CREATING DOM ELEMENTS



- Two standard ways to create and inserting DOM elements:
 - Using methods
 - Using strings and properties
- Methods are considered to be better of the two options
- Strings and properties is much easier to use

CREATING DOM ELEMENTS



- Utilizes **template literals** and the **innerHTML** property
- **Template literals** will be used to create HTML templates
- The **innerHTML** property can be used to read and update the HTML inside of an element

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

console.log($list.innerHTML) // <li>Milk</li>

$list.innerHTML += '<li>Eggs</li>'
console.log($list.innerHTML)
// <li>Milk</li><li>Eggs</li>
```



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

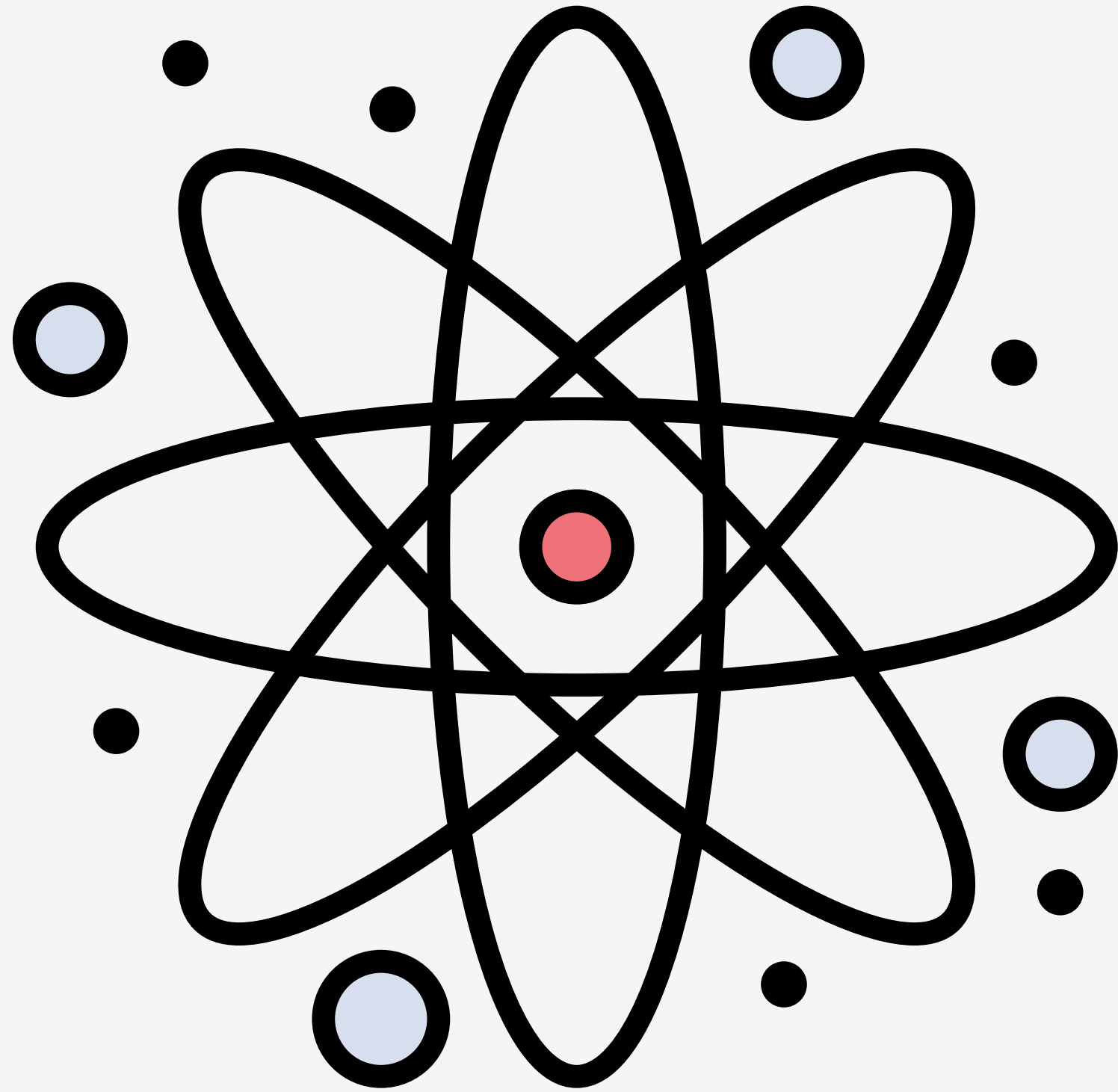
const items = ['milk', 'bread', 'eggs']

for (const item of items) {
  $list.innerHTML += `<li>${item}</li>`
}
```

Bad!

HANDS-ON

CREATING DOM ELEMENTS



- Retrieving and updating the DOM is process intensive
- Do NOT insert DOM elements inside a loop
- Store all the elements in an array or string
- Add all of them to the DOM at once

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

```
const items = ['milk', 'bread', 'eggs']
```

```
const listItems = []
```

```
for (const item of items) {
```

```
    listItems.push(`<li>${item}</li>`) // add to array  
}
```

```
$list.innerHTML += listItems.join(' ') // insert items
```

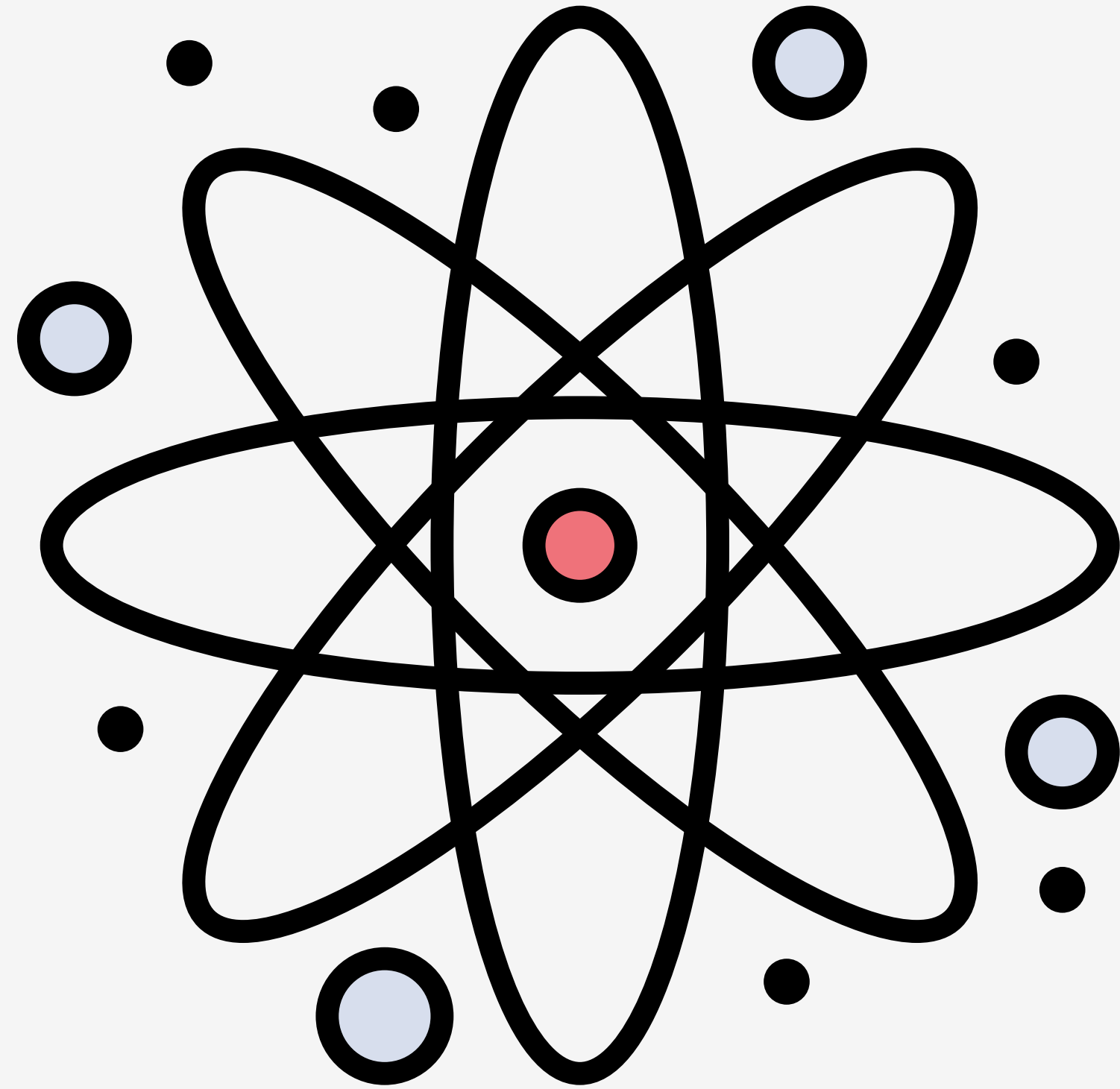
```
const $list = document.getElementById('list')

const items = ['milk', 'bread', 'eggs']

$list.innerHTML += items.map(function (item) {
  return `<li>${item}</li>`
}).join('')
```

HANDS-ON

CREATING DOM ELEMENTS



- The `innerHTML` property is limited to adding at the beginning, end, or replace everything
- The `insertAdjacentHTML()` inserts HTML at a specified position
- There are 4 predefined positions:
 - `beforebegin`
 - `afterbegin`
 - `beforeend`
 - `afterend`

```
const $list = document.getElementById('list')
const newItem = 'apples'

// first item
$list
  .insertAdjacentHTML('afterbegin', `<li>${newItem}</li>`)

// last item
$list
  .insertAdjacentHTML('beforeend', `<li>${newItem}</li>`)

// after first item
$list.firstElementChild
  .insertAdjacentHTML('afterend', `<li>${newItem}</li>`)

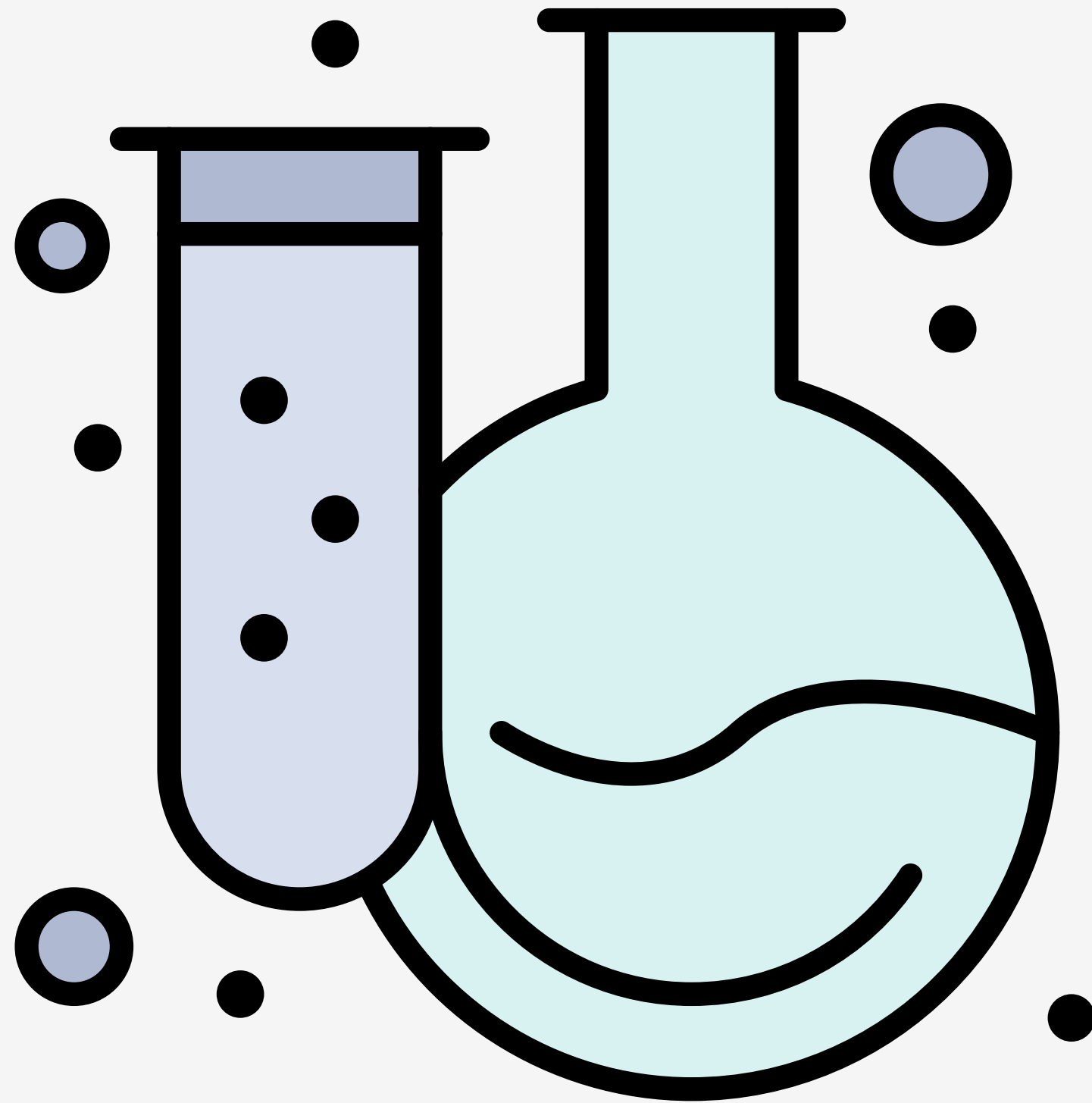
// before last item
$list.lastElementChild
  .insertAdjacentHTML('beforebegin', `<li>${newItem}</li>`)
```


HANDS-ON

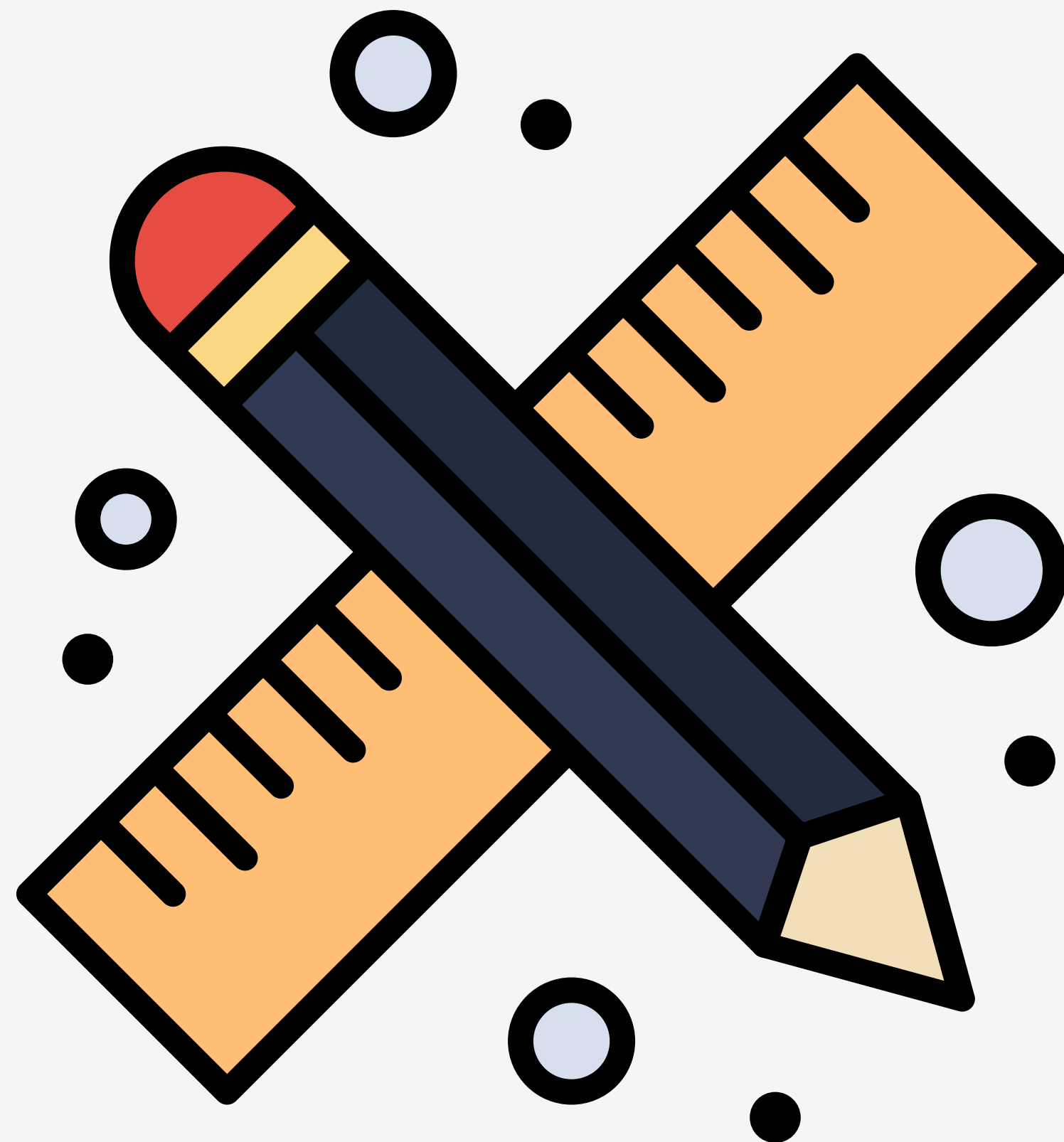
PRACTICE

99 LUFTBALLONS - NOT GRADED

- Practice DOM Creation by reviewing 99 Luftballons

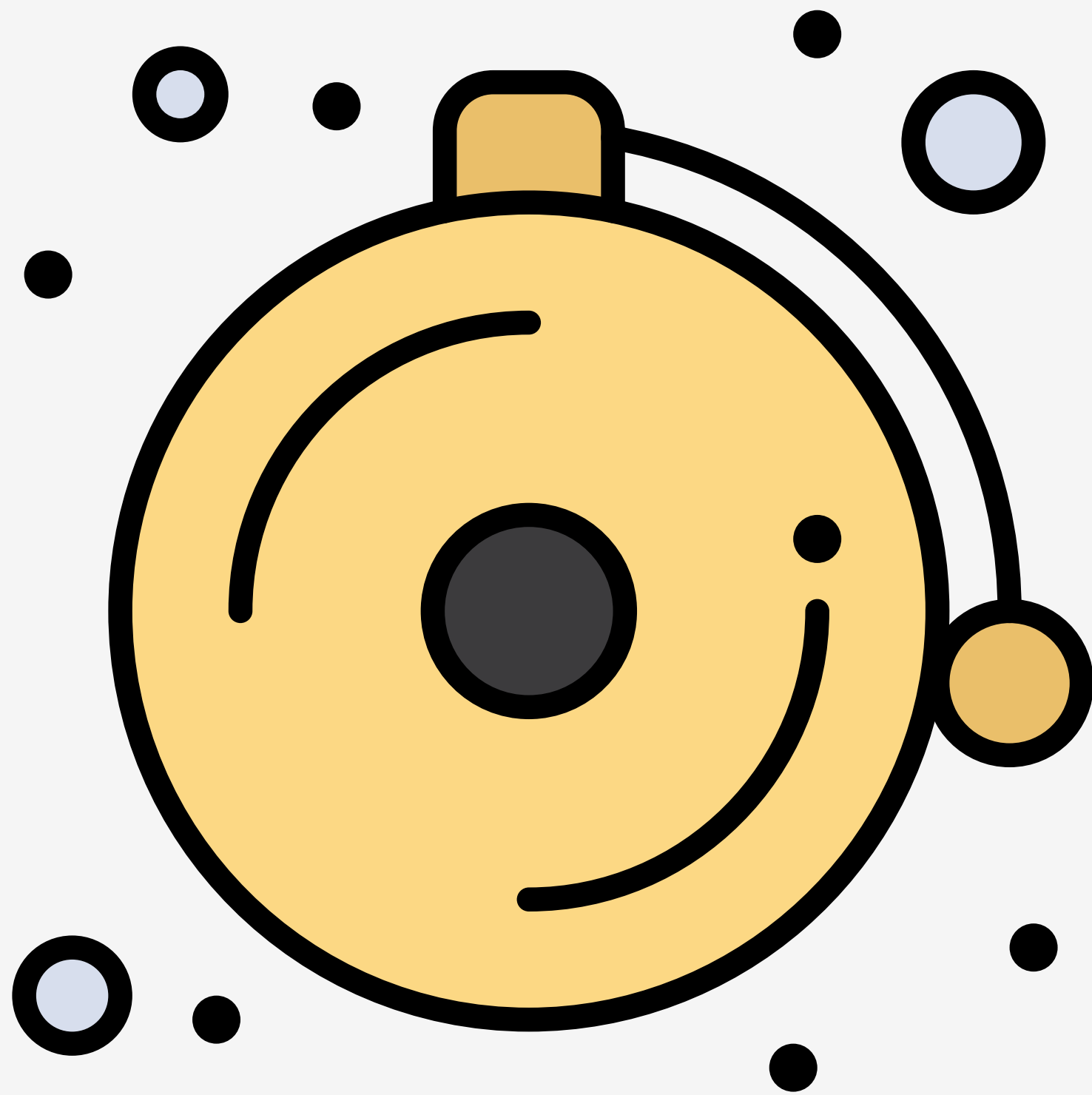


JEOPARDY



- *GITHUB CLASSROOM ASSIGNMENT*
- Create an interactive game of Jeopardy
- Dynamically populate the game board with titles, values, clues, and answers
- Add Event Listeners to populate and show the overlay and to show the answer
- *TEST YOUR CODE*
- Submit the URL to your repository
- *DUE:* Thu. Nov. 21 @ 11:59 PM

NEXT TIME...



- No Classes next week
- DOM Events
- Participation: Color Picker