Activity: “Resize the Box” — CSS Units Exploration

🕒 Estimated Duration

30–40 minutes

🎯 Objective

Students will learn how different **CSS measurement units** — **px**, **em**, **rem**, **%**, **vw**, and **vh** — affect the size and responsiveness of web elements.

🧠 Learning Outcome

By the end of this activity, students will:

- Understand the difference between **absolute** (px) and **relative** (%, em, rem, vw, vh) units.
- Recognize how **CSS units** affect **responsiveness** and **scalability**.
- Be able to choose appropriate units for various design scenarios.

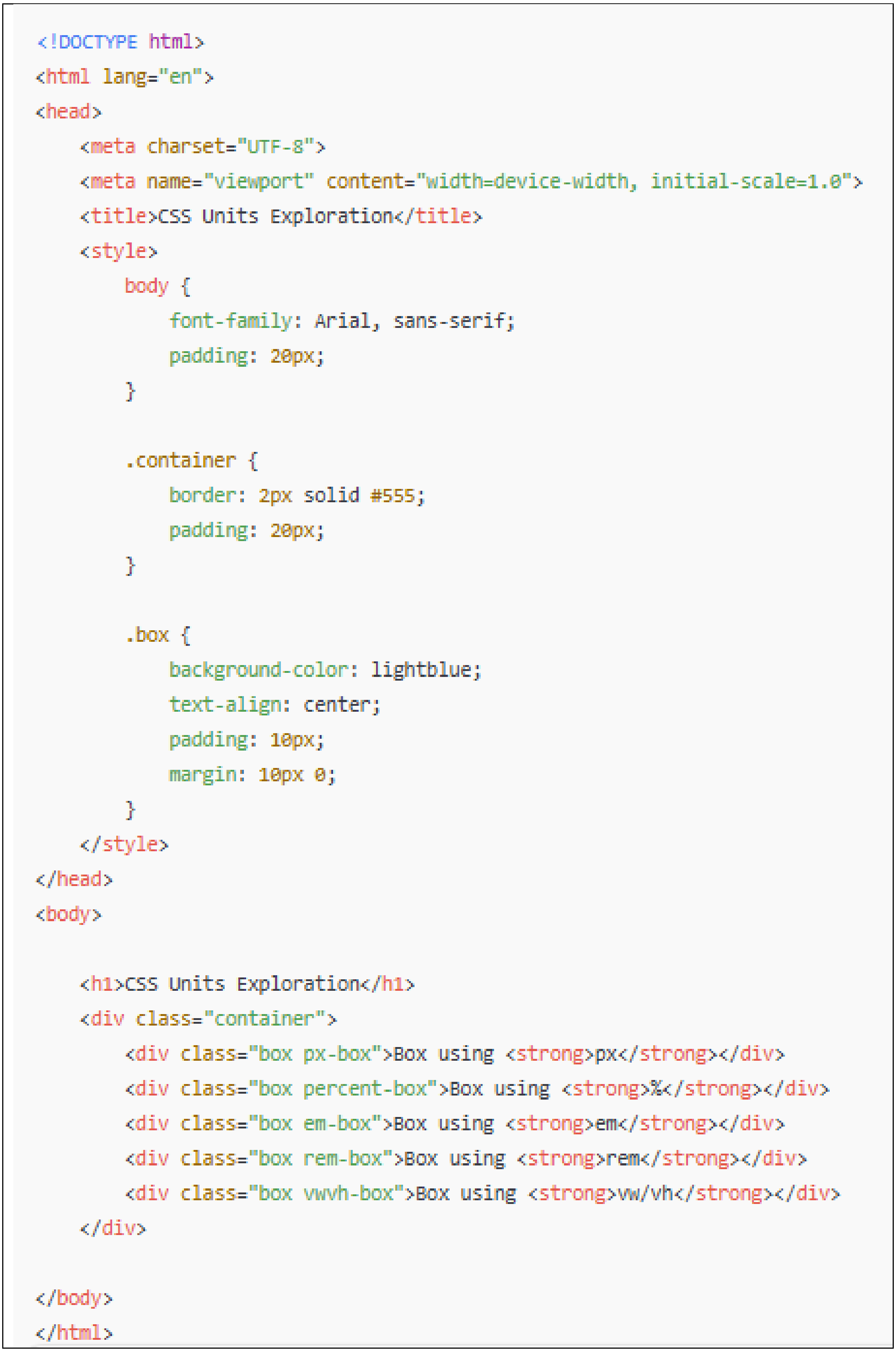
🧰 Materials Needed

- A text editor (e.g., VS Code, Sublime Text)
- A web browser (e.g., Chrome, Edge)
- A simple HTML boilerplate file

📄 Instructions

Step 1: Create the HTML file

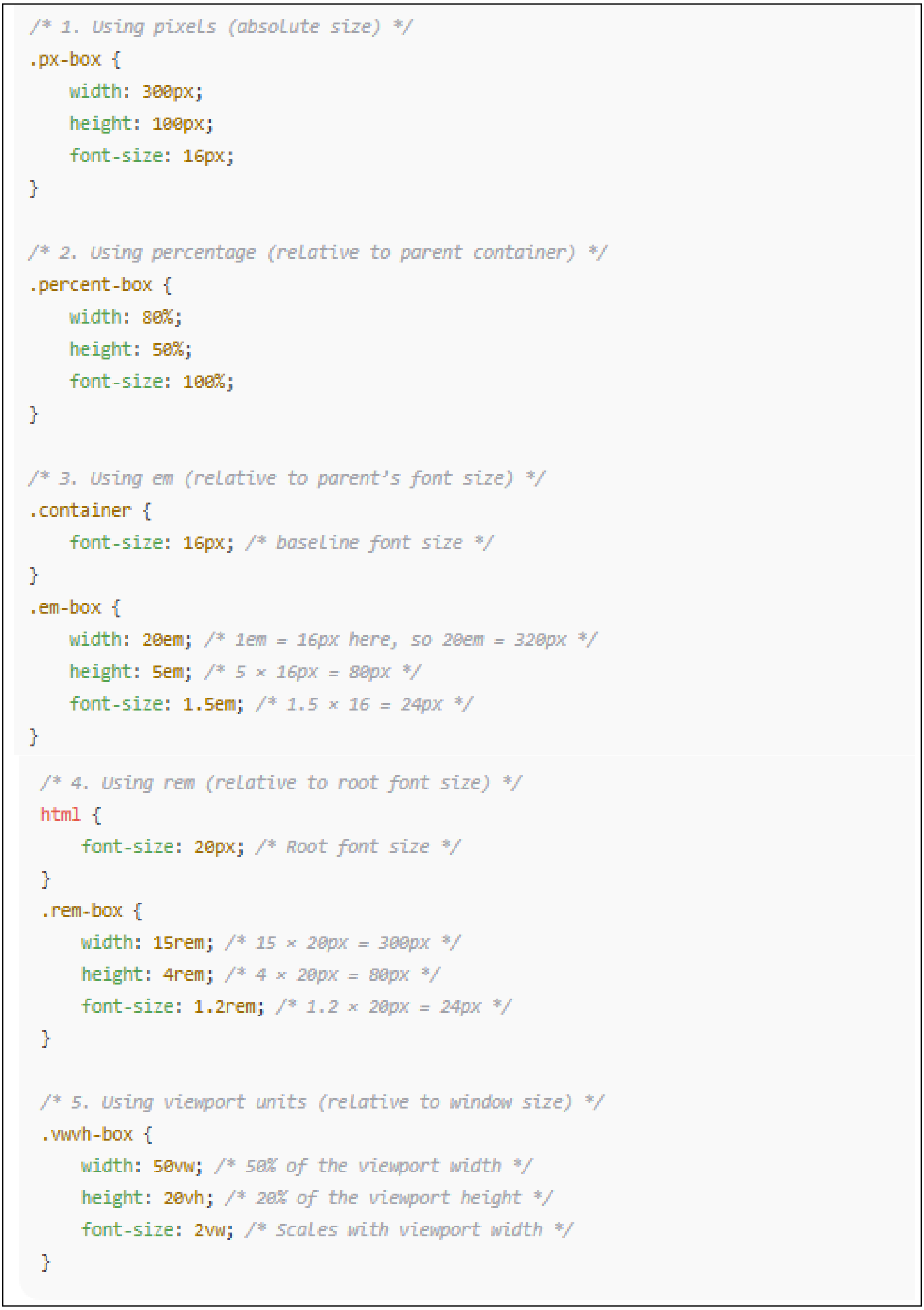
Create a new file named **units.html** and add the following starter code:



```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>CSS Units Exploration</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            padding: 20px;
        }

        .container {
            border: 2px solid #555;
            padding: 20px;
        }

        .box {
            background-color: lightblue;
            text-align: center;
            padding: 10px;
            margin: 10px 0;
        }
    </style>
</head>
<body>

    <h1>CSS Units Exploration</h1>
    <div class="container">
        <div class="box px-box">Box using <strong>px</strong></div>
        <div class="box percent-box">Box using <strong>%</strong></div>
        <div class="box em-box">Box using <strong>em</strong></div>
        <div class="box rem-box">Box using <strong>rem</strong></div>
        <div class="box vwvh-box">Box using <strong>vw/vh</strong></div>
    </div>

</body>
</html>
```

Step 2: Add styles using different units

Inside the **<style>** tag, below the existing rules, add:



```
/* 1. Using pixels (absolute size) */
.px-box {
    width: 300px;
    height: 100px;
    font-size: 16px;
}

/* 2. Using percentage (relative to parent container) */
.percent-box {
    width: 80%;
    height: 50%;
    font-size: 100%;
}

/* 3. Using em (relative to parent's font size) */
.container {
    font-size: 16px; /* baseline font size */
}
.em-box {
    width: 20em; /* 1em = 16px here, so 20em = 320px */
    height: 5em; /* 5 × 16px = 80px */
    font-size: 1.5em; /* 1.5 × 16 = 24px */
}
/* 4. Using rem (relative to root font size) */
html {
    font-size: 20px; /* Root font size */
}
.rem-box {
    width: 15rem; /* 15 × 20px = 300px */
    height: 4rem; /* 4 × 20px = 80px */
    font-size: 1.2rem; /* 1.2 × 20px = 24px */
}

/* 5. Using viewport units (relative to window size) */
.vwvh-box {
    width: 50vw; /* 50% of the viewport width */
    height: 20vh; /* 20% of the viewport height */
    font-size: 2vw; /* Scales with viewport width */
}
```

Step 3: Save and open in a browser

- Open the units.html file in a browser.
- Resize the browser window and **observe how the boxes behave**:
  - The **px** box remains **fixed** in size.
  - The **%** and **vw/vh** boxes **resize dynamically** with the window.
  - The **em** and **rem** boxes scale **relative to font size** changes.

Step 4: Experimentation Tasks

Ask students to:

1. Change the .container font size from 16px to 20px — observe how **em** boxes grow but **rem** boxes stay the same.
2. Resize the browser window — note how the **vw/vh** and **%** boxes adjust.
3. Change the root (html) font-size — observe how it affects **rem** units.

Summary

Unit Type	Example	Calculation	Resizes with Window?	Example Behaviour
px	width: 300px	Fixed	❌	Fixed size
%	width: 80%	Depends on parent width	✅	Grows/shrinks with container
em	width: 20em	20 × parent font size	⚠️	Changes if parent font size changes
rem	width: 15rem	15 × root font size	⚠️	Only changes if root size changes
vw	width: 50vw	50% of viewport width	✅	Adapts to browser width
vh	height: 20vh	Viewport height	✅	Adapts to browser height

Reflection Questions

1. Which CSS unit would you use for a responsive layout and why?

2. How do em and rem units differ when changing parent or root font size?

3. What happens to elements using vw and vh when you resize the browser window?

4. Why might fixed pixel units be less ideal for mobile design?

5. In your own words, explain when you would use each of the following units: px, %, em, rem, vw, vh.