**Lab 8**

**1. Cho biết ý nghĩa của @media (max-width: 768px) trong bài thực hành này.**

Trong bài thực hành này, @media (max-width: 768px) là một quy tắc trong CSS Media Queries, được sử dụng để áp dụng các kiểu dáng (styles) khác nhau cho trang web khi kích thước màn hình tối đa là 768px (thường tương ứng với kích thước của máy tính bảng hoặc màn hình nhỏ). Cụ thể, trong phần CSS của bài thực hành, khi màn hình có độ rộng nhỏ hơn hoặc bằng 768px, các thuộc tính như .card-container, .card h1, và .card p sẽ được điều chỉnh để giảm kích thước (width, height, font-size, margin) nhằm tối ưu hóa giao diện cho các thiết bị có màn hình nhỏ hơn. Điều này giúp đảm bảo trải nghiệm người dùng tốt hơn trên các thiết bị khác nhau.

**2. Cho biết ý nghĩa của transform: rotateY(180deg); trong bài thực hành này.**

Trong bài thực hành này, transform: rotateY(180deg); là một thuộc tính CSS được sử dụng để xoay một phần tử theo trục Y (trục dọc) 180 độ, tạo hiệu ứng lật 3D. Cụ thể, trong phần CSS của thẻ mời, thuộc tính này được áp dụng cho lớp .card.is-flipped và .card-back. Khi thẻ được lật (class is-flipped được thêm vào), mặt trước (card-front) sẽ xoay 180 độ để ẩn đi, trong khi mặt sau (card-back) cũng xoay 180 độ để hiển thị nội dung phía sau. Kết hợp với perspective và backface-visibility: hidden, nó tạo ra hiệu ứng lật thẻ 3D như yêu cầu trong câu 1 của bài thực hành.

**3. Cho biết ý nghĩa của justify-content: space-between; trong bài thực hành này.**

Trong bài thực hành này, justify-content: space-between; là một thuộc tính CSS được sử dụng trong mô hình flexbox để phân bổ không gian giữa các phần tử con bên trong một container flex.

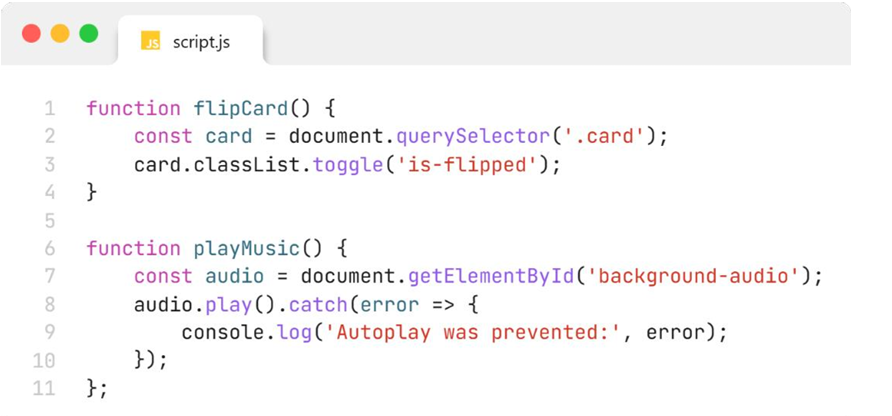
**4. Cho biết ý nghĩa của animation: gradientAnimation 6s ease infinite; trong bài thực hành này, nó có liên quan gì đến @keyframes gradientAnimation không?**

Trong bài thực hành này, animation: gradientAnimation 6s ease infinite; là một thuộc tính CSS định nghĩa một hiệu ứng hoạt hình. Cụ thể:

* gradientAnimation: Tên của hiệu ứng hoạt hình, tham chiếu đến một khối @keyframes có cùng tên.
* 6s: Thời gian thực hiện một chu kỳ hoạt hình là 6 giây.
* ease: Chế độ chuyển động mượt mà, bắt đầu chậm, tăng tốc ở giữa, và chậm lại ở cuối.
* infinite: Lặp lại hiệu ứng vô hạn.

Trong đoạn code, @keyframes gradientAnimation được định nghĩa với các giai đoạn (0%, 50%, 100%) để tạo hiệu ứng chuyển động của nền gradient (linear-gradient). Cụ thể, nó thay đổi vị trí background-position từ 0% sang 100% và quay lại 0% trong 6 giây, tạo cảm giác nền gradient di chuyển liên tục. Vì vậy, animation: gradientAnimation 6s ease infinite; có liên quan trực tiếp đến @keyframes gradientAnimation, vì nó sử dụng định nghĩa từ @keyframes để thực thi hiệu ứng trên các phần tử có áp dụng thuộc tính này (nếu có trong code hoàn chỉnh).

**5. Giải thích hình:**

****

**Hàm flipCard()**:

* **Mục đích**: Hàm này được thiết kế để lật thẻ (card) khi người dùng tương tác (ví dụ: nhấp chuột).
* **const card = document.querySelector('.card');**:
  + Phương thức document.querySelector() chọn phần tử HTML đầu tiên có class là .card (thường là thẻ chứa giao diện thiệp mời). Kết quả được lưu vào biến card.
* **card.classList.toggle('is-flipped');**:
  + Phương thức classList.toggle() thêm class is-flipped vào phần tử card nếu nó chưa có, hoặc xóa đi nếu đã có. Trong CSS (xem PAGE3 của tài liệu), class .is-flipped có thuộc tính transform: rotateY(180deg);, tạo hiệu ứng xoay 180 độ theo trục Y, giúp lật thẻ từ mặt trước sang mặt sau (hoặc ngược lại).
* **Trong bài này**: Hàm này được gọi khi người dùng nhấp vào thẻ (xem onclick="flipCard(); playMusic();" trong index.html từ ảnh trước đó), đáp ứng yêu cầu "Khi nhấn vào tấm thiệp thì thiệp sẽ lật lại" trong bài thực hành.

**Hàm playMusic()**:

* **Mục đích**: Hàm này phát âm thanh nền khi thẻ được lật.
* **const audio = document.getElementById('background-audio');**:
  + Phương thức document.getElementById() chọn phần tử <audio> có id là background-audio (được định nghĩa trong index.html với file âm thanh như ocean-of-dreams-227216.mp3).
* **audio.play().catch(error => { console.log('Autoplay was prevented:', error); });**:
  + audio.play() cố gắng phát file âm thanh.
  + .catch() xử lý lỗi, đặc biệt khi trình duyệt chặn tự động phát âm thanh (do chính sách bảo mật). Nếu lỗi xảy ra, thông báo "Autoplay was prevented:" cùng với chi tiết lỗi sẽ được ghi vào console.
* **Trong bài này**: Hàm này được gọi cùng với flipCard() khi nhấp vào thẻ, thỏa mãn yêu cầu "phát ra âm thanh" trong bài thực hành.

A computer screen shot of a computer program

AI-generated content may be incorrect.

1. **const now = new Date();**:
   * Tạo một đối tượng Date mới đại diện cho thời gian hiện tại của hệ thống. Đối tượng này chứa thông tin về ngày, giờ, phút, giây, v.v.
2. **const seconds = now.getSeconds();**:
   * Lấy giá trị số giây hiện tại (từ 0 đến 59) từ đối tượng now và gán vào biến seconds.
3. **const minutes = now.getMinutes();**:
   * Lấy giá trị số phút hiện tại (từ 0 đến 59) từ đối tượng now và gán vào biến minutes.
4. **const hours = now.getHours();**:
   * Lấy giá trị số giờ hiện tại (từ 0 đến 23) từ đối tượng now và gán vào biến hours.

**Trong bài thực hành:**

* Đoạn mã này là một phần của hàm updateClock() trong file script.js (PAGE9), được sử dụng để lấy thông tin thời gian hiện tại nhằm cập nhật vị trí của các kim đồng hồ (giờ, phút, giây) trong bài thực hành câu 2 ("Thiết kế đồng hồ kim"). Các giá trị này sau đó được sử dụng để tính toán góc quay (secondDeg, minuteDeg, hourDeg) và cập nhật giao diện đồng hồ bằng cách thay đổi thuộc tính transform của các phần tử .hour-hand, .minute-hand, và .second-hand.

**6.Cho biết ý nghĩa của setInterval(updateClock,1000); trong bài thực hành này.**

**updateClock**: Là tên của hàm sẽ được gọi lặp lại. Trong bài thực hành này, hàm updateClock() chịu trách nhiệm cập nhật vị trí của các kim đồng hồ (giờ, phút, giây) dựa trên thời gian hiện tại.

**1000**: Là khoảng thời gian giữa các lần gọi hàm, tính bằng mili giây, tương đương với 1 giây. Điều này có nghĩa là hàm updateClock() sẽ được thực thi lại mỗi giây một lần.

**setInterval** giúp đồng hồ kim hoạt động mượt mà và đồng bộ với thời gian hệ thống.