Trong CSS animation, giá trị thứ ba trong thuộc tính **animation** xác định hành vi khi hoạt hình kết thúc. Có ba loại giá trị mà bạn có thể sử dụng:

1. **forwards**: Khi hoạt hình kết thúc, phần tử sẽ giữ lại trạng thái cuối cùng của nó. Nghĩa là các thuộc tính được xác định trong keyframe cuối cùng sẽ được áp dụng cho phần tử sau khi hoạt hình kết thúc.
2. **backwards**: Phần tử sẽ áp dụng trạng thái đầu tiên trong keyframes ngay khi hoạt hình được kích hoạt. Điều này có nghĩa là các thuộc tính được xác định trong keyframe đầu tiên sẽ được áp dụng cho phần tử trước khi hoạt hình bắt đầu chạy.
3. **both**: Phần tử sẽ thực hiện cả hành vi "forwards" và "backwards". Nghĩa là phần tử sẽ giữ lại trạng thái cuối cùng của hoạt hình sau khi kết thúc và áp dụng trạng thái đầu tiên của hoạt hình khi bắt đầu chạy.
4. infinite : Lặp vô hạn

Ví dụ: **animation: moveRight 0.3s forwards;** trong đó **forwards** là giá trị thứ ba của thuộc tính **animation**. Trong ví dụ này, khi hoạt hình **moveRight** kết thúc, phần tử sẽ giữ lại vị trí dịch chuyển cuối cùng của nó.

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Phần mềm đa phương tiện

Mô tả được tạo tự động

Tất cả các giá trị sau đây đều là các kiểu hàm thời gian (timing function) trong CSS animation:

1. **linear**: Đây là kiểu mặc định. Hoạt hình sẽ diễn ra với tốc độ đồng đều và không có hiệu ứng chậm hoặc tăng tốc.
2. **ease**: Đây là một kiểu thời gian phổ biến khác. Hoạt hình sẽ diễn ra với hiệu ứng chậm ban đầu, tăng tốc giữa chừng và chậm lại trước khi kết thúc. Kiểu này tạo ra một hiệu ứng mượt mà và tự nhiên.
3. **ease-in**: Hoạt hình sẽ diễn ra với hiệu ứng chậm ban đầu và tăng tốc sau đó.
4. **ease-out**: Hoạt hình sẽ diễn ra với hiệu ứng tăng tốc ban đầu và chậm lại trước khi kết thúc.
5. **ease-in-out**: Hoạt hình sẽ diễn ra với hiệu ứng chậm ban đầu, tăng tốc giữa chừng và chậm lại trước khi kết thúc.
6. **step-start**: Hoạt hình sẽ diễn ra theo từng bước, bắt đầu ngay từ đầu mỗi bước mà không có hiệu ứng chuyển đổi mượt mà.
7. **step-end**: Hoạt hình sẽ diễn ra theo từng bước, kết thúc ngay khi đạt đến cuối mỗi bước mà không có hiệu ứng chuyển đổi mượt mà.
8. **steps(n)**: Hoạt hình sẽ diễn ra theo từng bước với tổng cộng n bước. Giá trị n được xác định bởi bạn.

Ví dụ: **animation-timing-function: linear;** trong đó **linear** là kiểu hàm thời gian được sử dụng. Khi áp dụng kiểu thời gian này, hoạt hình sẽ diễn ra với tốc độ đồng đều mà không có hiệu ứng chậm hoặc tăng tốc.

Bạn có thể chọn bất kỳ kiểu hàm thời gian nào phù hợp với hiệu ứng hoạt hình mà bạn muốn áp dụng bằng cách sử dụng thuộc tính **animation-timing-function** trong CSS.