**✅ 1. padding – Khoảng cách bên trong phần tử**

**🧠 Cú pháp:**

padding: <top> <right> <bottom> <left>;

**📌 Ví dụ:**

css

Sao chépChỉnh sửa

padding: 10px 20px 15px 5px;

➡️ Padding trên: 10px, phải: 20px, dưới: 15px, trái: 5px.

**🪄 Viết ngắn gọn:**

| **Cú pháp** | **Diễn giải** |
| --- | --- |
| padding: 10px; | Tất cả 4 phía đều 10px |
| padding: 10px 20px; | Trên-dưới 10px, trái-phải 20px |
| padding: 10px 20px 30px; | Trên 10px, trái-phải 20px, dưới 30px |

**✅ 2. margin – Khoảng cách bên ngoài phần tử**

**🧠 Cú pháp giống padding**

margin: 10px 15px 5px 20px;

**📌 Ví dụ:**

css

Sao chépChỉnh sửa

margin: 0 auto;

➡️ Trên-dưới 0, trái-phải tự căn giữa (hay dùng để căn giữa khối div có width xác định).

**✅ 3. position – Định vị phần tử**

**🧠 Cú pháp:**

position: static | relative | absolute | fixed | sticky;

**📌 Ý nghĩa:**

| **Giá trị** | **Giải thích** | **Ví dụ thường gặp** |
| --- | --- | --- |
| static | Mặc định, không định vị | Các thẻ <p> |
| relative | Định vị tương đối vị trí gốc | Làm cha cho absolute |
| absolute | Định vị tuyệt đối trong phần tử cha gần nhất có relative | Tooltip, icon trong input |
| fixed | Dính vào cửa sổ trình duyệt | Navbar cố định |
| sticky | Dính khi cuộn đến một vị trí | Tiêu đề bảng, thanh công cụ |

**✅ 4. z-index – Lớp chồng hiển thị**

**🧠 Cú pháp:**

z-index: <số nguyên>;

* Số lớn hơn thì hiển thị trên.
* Chỉ hoạt động với các phần tử có position khác static.

**📌 Ví dụ:**

css

Sao chépChỉnh sửa

.box1 {

position: absolute;

z-index: 2;

}

.box2 {

position: absolute;

z-index: 1;

}

➡️ box1 sẽ nằm trên box2.

**✅ 5. display – Kiểu hiển thị phần tử**

**🧠 Cú pháp:**

display: block | inline | inline-block | flex | grid | none;

| **Giá trị** | **Giải thích** |
| --- | --- |
| block | Chiếm toàn bộ chiều ngang (ví dụ: div) |
| inline | Hiển thị nội tuyến (ví dụ: span) |
| inline-block | Nội tuyến nhưng có thể set width, height |
| flex | Kích hoạt Flexbox |
| grid | Bố cục dạng lưới |
| none | Ẩn phần tử (không chiếm chỗ) |

**✅ 6. justify-content – Căn hàng ngang trong Flexbox**

**🧠 Cú pháp:**

justify-content: flex-start | flex-end | center | space-between | space-around | space-evenly;

**📌 Ví dụ:**

css

Sao chépChỉnh sửa

.flex-container {

display: flex;

justify-content: center;

}

➡️ Các phần tử con được căn giữa theo chiều ngang.

| **Giá trị** | **Hiệu ứng** |
| --- | --- |
| flex-start | Dồn trái |
| flex-end | Dồn phải |
| center | Căn giữa |
| space-between | Cách đều, sát hai bên |
| space-around | Cách đều 2 bên mỗi item |
| space-evenly | Cách đều hoàn toàn |

**✅ 7. align-items – Căn chiều dọc trong Flexbox**

**🧠 Cú pháp:**

align-items: stretch | flex-start | flex-end | center | baseline;

**📌 Ví dụ:**

css

Sao chépChỉnh sửa

.flex-container {

display: flex;

align-items: center;

}

| **Giá trị** | **Hiệu ứng** |
| --- | --- |
| flex-start | Dồn lên trên |
| flex-end | Dồn xuống dưới |
| center | Căn giữa dọc |
| stretch | Kéo giãn theo chiều dọc |
| baseline | Căn theo dòng chữ |

**✅ 8. transition – Hiệu ứng chuyển động**

**transition – Hiệu ứng chuyển động**

**transition là một thuộc tính trong CSS cho phép bạn thêm hiệu ứng chuyển động mượt mà khi thay đổi một thuộc tính nào đó của phần tử. Nó giúp tạo ra các hiệu ứng động như khi thay đổi màu sắc, kích thước, vị trí, v.v., mà không cần phải sử dụng JavaScript.**

**Cú pháp:**

**css**

**Sao chépChỉnh sửa**

**transition: [property] [duration] [timing-function] [delay];**

**Các thành phần:**

1. **property:**
   * **Thuộc tính CSS mà bạn muốn thêm hiệu ứng. Ví dụ: width, height, color, background-color, transform, v.v.**
   * **Nếu bạn muốn hiệu ứng cho tất cả các thuộc tính thay đổi, bạn có thể dùng từ khóa all.**
   * **Ví dụ: background-color, transform, opacity.**
2. **duration:**
   * **Thời gian (duration) mà hiệu ứng sẽ diễn ra, có thể sử dụng đơn vị thời gian như s (giây) hoặc ms (miligiây).**
   * **Ví dụ: 0.3s (0.3 giây), 500ms (500 miligiây).**
3. **timing-function:**
   * **Kiểu chuyển động của hiệu ứng, có thể là:**
     + **ease (mặc định): Chuyển động bắt đầu chậm, tăng tốc, rồi chậm lại.**
     + **linear: Chuyển động đều đặn, không thay đổi tốc độ.**
     + **ease-in: Bắt đầu chậm rồi tăng tốc.**
     + **ease-out: Bắt đầu nhanh rồi chậm lại.**
     + **ease-in-out: Bắt đầu chậm, tăng tốc giữa chừng, rồi chậm lại ở cuối.**
     + **cubic-bezier: Cho phép bạn tự định nghĩa tốc độ chuyển động với các tham số cụ thể.**
   * **Ví dụ: ease, linear, ease-in-out.**
4. **delay:**
   * **Độ trễ trước khi hiệu ứng bắt đầu diễn ra, có thể sử dụng đơn vị thời gian như s (giây) hoặc ms (miligiây).**
   * **Ví dụ: 1s (1 giây), 200ms (200 miligiây).**

**✅ 9. transform – Biến đổi hình dạng**

**transform – Biến đổi hình dạng**

**Thuộc tính transform trong CSS cho phép bạn thay đổi các thuộc tính của phần tử như vị trí, kích thước, góc quay, hay độ nghiêng mà không làm ảnh hưởng đến bố cục hoặc dòng chảy của phần tử trong trang. Bạn có thể sử dụng một hoặc nhiều hàm biến đổi kết hợp với nhau để tạo ra các hiệu ứng động cho phần tử.**

**Cú pháp:**

**transform: [function];**

**Các hàm (functions) mà bạn có thể sử dụng trong transform gồm có:**

* **translate(): Di chuyển phần tử.**
* **scale(): Phóng to hoặc thu nhỏ phần tử.**
* **rotate(): Xoay phần tử.**
* **skew(): Nghiêng phần tử.**
* **matrix(): Áp dụng phép biến đổi ma trận.**

**Các hàm trong transform:**

**1. translate(x, y) – Di chuyển phần tử**

* **Tác dụng: Di chuyển phần tử theo các trục X và Y (hệ tọa độ 2D).**
* **Cú pháp: translate(x, y)**
  + **x: Di chuyển theo trục hoành (trái/phải) với đơn vị là px, %, v.v.**
  + **y: Di chuyển theo trục tung (lên/xuống) với đơn vị là px, %, v.v.**
  + **Nếu chỉ cung cấp một giá trị cho x, y sẽ mặc định là 0.**

**Ví dụ:**

**.box {**

**transform: translate(50px, 100px);**

**}**

* **Giải thích: Phần tử .box sẽ di chuyển 50px sang phải và 100px xuống dưới.**

**2. scale(x, y) – Phóng to / Thu nhỏ phần tử**

* **Tác dụng: Phóng to hoặc thu nhỏ phần tử theo các trục X và Y.**
* **Cú pháp: scale(x, y)**
  + **x: Tỷ lệ phóng to/thu nhỏ theo trục X.**
  + **y: Tỷ lệ phóng to/thu nhỏ theo trục Y.**
  + **Nếu chỉ cung cấp một giá trị cho x, y sẽ có giá trị bằng nhau, tức là phần tử sẽ được phóng to/thu nhỏ đồng đều.**

**Ví dụ:**

**.box:hover {**

**transform: scale(1.2);**

**}**

* **Giải thích: Khi hover vào phần tử .box, phần tử sẽ phóng to lên 20% so với kích thước ban đầu.**

**3. rotate(deg) – Xoay phần tử**

* **Tác dụng: Xoay phần tử một góc nhất định.**
* **Cú pháp: rotate(deg)**
  + **deg: Góc xoay, có đơn vị là deg (độ). Một vòng tròn đầy đủ là 360 độ.**

**Ví dụ:**

**.box {**

**transform: rotate(45deg);**

**}**

* **Giải thích: Phần tử .box sẽ được xoay 45 độ theo chiều kim đồng hồ.**

**4. skew(x, y) – Nghiêng phần tử**

* **Tác dụng: Nghiêng phần tử theo các trục X và Y.**
* **Cú pháp: skew(x, y)**
  + **x: Góc nghiêng theo trục X.**
  + **y: Góc nghiêng theo trục Y.**
  + **Các giá trị được tính bằng độ (deg).**

**Ví dụ:**

**.box {**

**transform: skew(20deg, 10deg);**

**}**

* **Giải thích: Phần tử .box sẽ bị nghiêng 20 độ theo trục X và 10 độ theo trục Y.**

**5. matrix(a, b, c, d, e, f) – Áp dụng phép biến đổi ma trận**

* **Tác dụng: Áp dụng phép biến đổi phức tạp qua ma trận 2D.**
* **Cú pháp: matrix(a, b, c, d, e, f)**
  + **Đây là một phép toán ma trận có 6 tham số, được dùng để áp dụng các phép biến đổi như dịch chuyển, xoay, phóng to, thu nhỏ, và nghiêng trong một phép toán duy nhất.**

**Ví dụ:**

**.box {**

**transform: matrix(1, 0.2, -0.2, 1, 50, 100);**

**}**

* **Giải thích: Phần tử .box sẽ được dịch chuyển và xoay một cách phức tạp, tùy theo các tham số ma trận đã cho.**

**Kết hợp nhiều hàm transform**

**Bạn có thể kết hợp nhiều hàm transform để thực hiện nhiều phép biến đổi cùng lúc.**

**Ví dụ:**

**.box:hover {**

**transform: scale(1.1) rotate(10deg) translate(50px, 0);**

**}**

* **Giải thích: Khi hover vào phần tử .box, phần tử này sẽ phóng to 10%, xoay 10 độ và di chuyển 50px sang phải, tất cả trong một lần thay đổi.**

**✅ 10. text-align – Căn lề văn bản**

**🧠 Cú pháp:**

text-align: left | right | center | justify;

**📌 Ví dụ:**

css

Sao chépChỉnh sửa

h1 {

text-align: center;

}

| **Giá trị** | **Giải thích** |
| --- | --- |
| left | Căn trái |
| right | Căn phải |
| center | Căn giữa |
| justify | Căn đều hai bên (giống báo in) |