

Transformaciones, Transiciones y Animaciones en CSS

1. Transformaciones CSS

Las transformaciones permiten modificar la posición, tamaño, rotación y deformación de un elemento en el espacio.

Propiedades principales:

translate(x, y): Mueve el elemento en los ejes X e Y.

translateX(x): Mueve el elemento en el eje X.

translateY(y): Mueve el elemento en el eje Y.

scale(x, y): Escala el tamaño del elemento en ambos ejes.

scaleX(x): Escala en el eje X.

scaleY(y): Escala en el eje Y.

rotate(ángulo): Rota el elemento (en grados, radianes, gradianes o giros).

skew(ánguloX, ánguloY): Sesga el elemento en los ejes X e Y.

skewX(ánguloX): Sesga solo en el eje X.

skewY(ánguloY): Sesga solo en el eje Y.

Combinación de transformaciones:

```
selector {  
  
    transform: rotate(45deg) scale(1.5) translate(80px, 80px);  
  
}
```

Origen de la transformación:

```
selector {  
  
    transform-origin: x y; /* Valores: top, bottom, center,  
    left, right, %, px */  
  
}
```

2. Transiciones CSS

Las transiciones permiten cambiar valores de propiedades de forma gradual.

Propiedades principales:

transition-property: Propiedades a animar (ej: `all`, `color`, `transform`).

transition-duration: Duración de la transición (ej: `2s`, `500ms`).

transition-timing-function: Curva de aceleración (ej: `ease`, `linear`, `cubic-bezier`).

transition-delay: Retraso antes de iniciar la transición (ej: `1s`).

Shorthand:

```
selector {  
  
    transition: property duration timing-function delay;  
  
}
```

Ejemplo:

```
selector {  
  
    transition: transform 2s ease-in-out, color 1s linear;  
  
}
```

Optimización:

Usar propiedades que solo afecten a `composite` (ej: `transform`, `opacity`), ya que son más eficientes.

Evitar transiciones simultáneas en propiedades que consuman muchos recursos.

3. Animaciones CSS

Las animaciones permiten definir múltiples estados intermedios mediante `@keyframes`.

Definición con `@keyframes`:

```
@keyframes nombre-animación {  
  
  0% { propiedad: valor; }  
  
  50% { propiedad: valor; }  
  
  100% { propiedad: valor; }  
  
}
```

Propiedades principales:

animation-name: Nombre de la animación definida en `@keyframes`.

animation-duration: Duración (ej: `2s`).

animation-timing-function: Curva de aceleración (ej: `ease`, `linear`).

animation-delay: Retraso antes de iniciar (ej: `1s`).

animation-iteration-count: Número de repeticiones (`infinite` para bucle infinito).

animation-direction: Dirección de la animación (`normal`, `reverse`, `alternate`).

animation-fill-mode: Estilos aplicados antes/después de la animación (`forwards`, `backwards`).

animation-play-state: Estado de la animación (`running`, `paused`).

Shorthand:

```
selector {  
  
  animation: name duration timing-function delay  
            iteration-count direction fill-mode play-state;  
  
}
```

Ejemplo:

```
@keyframes cambia-color {  
  
  0% { background-color: green; }  
  
  50% { background-color: blue; }  
  
  100% { background-color: red; }  
  
}  
  
selector {  
  
  animation: cambia-color 3s ease-in-out infinite;  
  
}
```

4. Clip-path en CSS

La propiedad `clip-path` permite recortar un elemento con una forma específica.

Valores principales:

`circle()`: Recorte circular.

`inset()`: Recorte rectangular.

`ellipse()`: Recorte elíptico.

`polygon()`: Recorte poligonal.

`path()`: Recorte basado en una ruta SVG.

Ejemplo:

```
selector {  
  
    clip-path: circle(50% at 50% 50%);  
  
}
```