

# Animaciones CSS



# Animaciones CSS

## Transform

- Modifican la visualización del elemento, lo "transforman" (rotarlo, trasladarlo, escalarlo, torcerlo).
- NO veo el proceso de transformación.
- 2D y 3D

## Transition

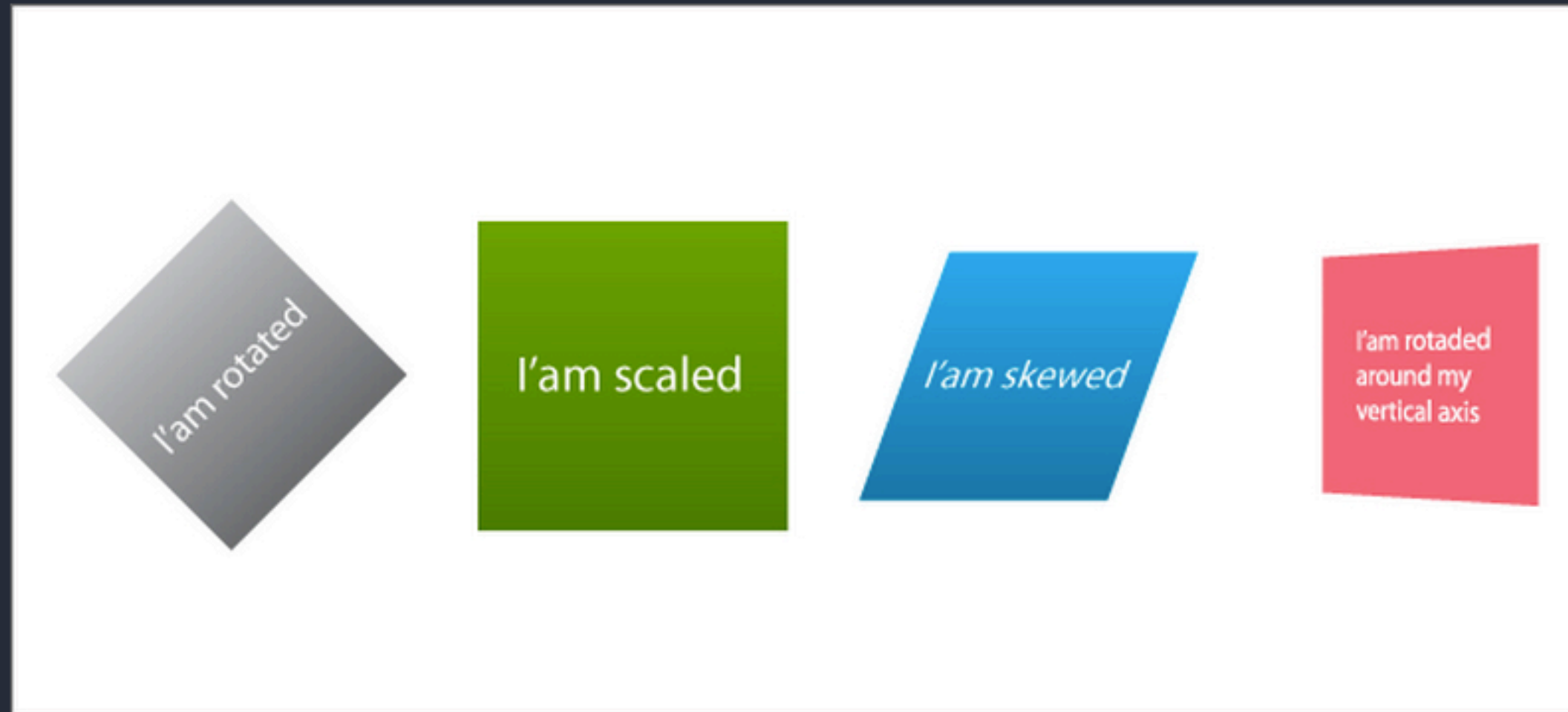
- Controla el proceso de transición de un estado a otro de un elemento, ya tenemos una "animación" sencilla.

## Animation

- Animaciones complejas

# Animaciones CSS

## TRANSFORM (2D)



`transform: rotate(45deg) // transform: scale(1.5) // trasform: skew(-20deg)`

`transform: translate( 20px, 30px)`

`transform: translateX(20px) // translateY(50px)`

# Animaciones CSS

## TRANSFORM (2D) -> ROTATE

```
.ejemplo {  
    -webkit-transform: rotate (45deg);  
    -moz-transform: rotate (45deg);  
    -o-transform: rotate (45deg);  
    -ms-transform: rotate (45deg);  
    transform: rotate (45deg);  
}
```

### Recursos:

<http://prefixr.com/>

### Origen de la rotación

```
transform-origin: left top;  
  
transform: rotate (-20deg);
```



# Animaciones CSS

## TRANSITION

```
div {  
  transition-property: width; (Propiedad que animo, una, varias o todas)  
  transition-duration: 2s; (Duración de la transición)  
  transition-timing-function: linear; (Curva cubic-bezier)  
  transition-delay: 1s; (Tiempo de espera para comenzar la transición)  
}
```

### Shorthand:

```
div {  
  transition: width 2s linear 1s;  
}
```

# Animaciones CSS

## TRANSITION

### Puedo animar una propiedad:

```
div {  
  transition: width 2s ease;  
}
```

### Varias:

```
div {  
  transition: width 2s, height 4s;  
}
```

### O todas:

```
div {  
  transition: all 2s ease 4s;  
}
```

# Animaciones CSS

## TRANSITION

**TRANSFORM** también puedo!:

```
div: hover{  
  transition: transform .5s ease;  
}
```

```
div: hover{  
  transform: scale(1.5);  
}
```

# Animaciones CSS

## ANIMATION

```
div {  
  animation-name: changeBg; (Nombre @keyframes donde está la animación)  
  animation-duration: 5s; (Duración)  
  animation-timing-function: linear; (Cubic bezier)  
  animation-delay: 2s; (Tiempo de espera)  
  animation-iteration-count: infinite; (Repeticiones)  
  animation-direction: alternate; (Va y vuelve, vuelve y va? Va y vuelve, va y vuelve?)  
}
```

### Shorthand:

```
div {  
  animation: changeBg 5s linear 2s infinite alternate;  
}
```



# Animaciones css

ANIMATION -> @KEYFRAMES

**From -> To ( de 0% a 100%)**

```
@keyframes example {  
  from {background-color: red;}  
  to {background-color: yellow;}  
}
```

**Porcentajes:**

```
@keyframes changeBg {  
  0%    {background-color: red;}  
  25%   {background-color: green;}  
  50%   {background-color: blue;}  
  100%  {background-color: yellow;}  
}
```

# Animaciones CSS

ANIMATIONS -> @KEYFRAMES

**Se trata de animar varias propiedades a la vez:**

```
@keyframes example {  
  from {  
    background-color: red;  
    transform: scale(1);  
  }  
  to {  
    background-color: yellow;  
    transform: scale(2);  
  }  
}
```

# Animaciones CSS

@diana\_aceves\_

[www.hoyheaprendido.es](http://www.hoyheaprendido.es)