

CSS3

Transformaciones, Transiciones y Animaciones

Ejemplos prácticos

¡BIENVENIDOS!

Soy Juan Diego Pérez Jiménez

Profesor de Informática en FP



<https://www.linkedin.com/in/juandiegoperez>



@pekechis

ÍNDICE

- **Descripción de taller**
- **Requisitos**
- **Transformaciones**
- **Transiciones**
- **Animaciones**
- **Ejemplos**

DESCRIPCIÓN DEL TALLER



TRANSFORMACIONES

TRANSICIONES

ANIMACIONES

REQUISITOS



Conocimiento de **HTML y **CSS** sobre todo a nivel de **selectores** y **propiedades** básicas.**

TRANSFORMACIONES



The **CSS transform property** lets you modify the coordinate space of the CSS visual formatting model. Using it, elements can be **translated, rotated, scaled, and skewed** according to the values set.

<https://developer.mozilla.org/en-US/docs/Web/CSS/transform>

TRANSFORMACIONES

- **Rotación (rotate)**
- **Traslación (translate)**
- **Escalado (scale)**
- **Sesgados (skew)**
- **Perspectiva (perspective)**

TRANSFORMACIONES

ROTACIÓN

```
selector {
```

```
    transform: rotate(angulo);
```

```
}
```

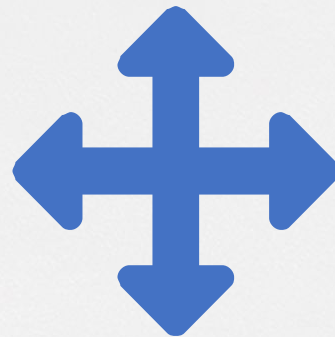
```
// Expresaremos los ángulos como (45deg) o (-45deg)  
// deg por degrees. Se pueden usar otras unidades  
// En el sentido de las aguja del reloj
```



TRANSFORMACIONES

TRANSLACIÓN

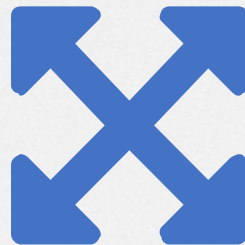
```
selector {  
  //En ambas dimensiones  
  transform: translate(valueX,valueY);  
  //Sólo en X  
  transform: translateX(valueX);  
  //Sólo en Y  
  transform: translateY(valueY);  
}
```



TRANSFORMACIONES

ESCALADO

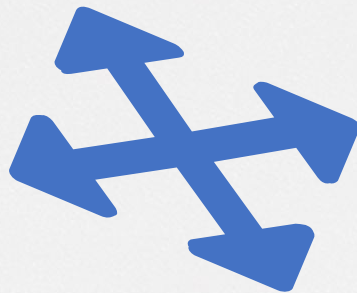
```
selector {  
    //En ambas dimensiones  
    transform: scale(valueX,valueY);  
    //Sólo en X  
    transform: scaleX(valueX);  
    //Sólo en Y  
    transform: scaleY(valueY);  
}
```



TRANSFORMACIONES

SESGADOS

```
selector {  
    //En ambas dimensiones  
    //No funciona en algunos navegadores  
    transform: skew(anguloX,anguloY);  
    //Sólo en X  
    transform: skewX(anguloX);  
    //Sólo en Y  
    transform: skewY(anguloY);  
}
```



TRANSFORMACIONES

VARIAS A LA VEZ

```
selector {
```

```
    //Por ejemplo
```

```
    transform: rotate(angulo) scale(ratio) translate(x,y);
```

```
}
```

TRANSFORMACIONES

ORIGEN DE LA TRANSFORMACIÓN



Es una **propiedad CSS** que establece el punto de referencia para las transformaciones, por defecto es el centro de la caja (50%,50%)

TRANSFORMACIONES

ORIGEN DE LA TRANSFORMACIÓN

```
selector {
```

```
    //Establezco la transformación
```

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

```
    //Modifico el origen
```

```
    //Son válidos top, left, right, bottom, center
```

```
    transform-origin: despX despY;
```

```
}
```

TRANSICIONES



The transition **CSS property** is a **shorthand** property for **transition-property, transition-duration, transition-timing-function, and transition-delay**. Transitions enable you to define the **transition between two states** of an element. Different states may be defined using pseudo-classes like `:hover` or `:active` or dynamically set using JavaScript.

<https://developer.mozilla.org/en-US/docs/Web/CSS/transition>

TRANSICIONES



TRANSICIONES

//En el estado inicial establecemos los valores de iniciales y los valores de // la transición. En el estado final los valores finales.

selector {

 //De una sola vez

 transition: propiedad duración función retraso;

 //Separadamente

 transition-property: propiedad (all, o lista de propiedades CSS separadas por ,)

 transition-duration: duración (s or ms);

 transition-timing-function: función; (ease, linear, ease-in, ease-out, step-start)

 transition-delay: retraso (s or ms);

}

ANIMACIONES



The animation shorthand CSS property applies an animation between styles. It is a shorthand for animation-name, animation-duration, animation-timing-function, animation-delay, animation-iteration-count, animation-direction, animation-fill-mode, and animation-play-state.

<https://developer.mozilla.org/en-US/docs/Web/CSS/animation>

ANIMACIONES

- **animation-duration** (en s o ms)
- **animation-name** (para luego posteriormente las distintas variaciones)
- **animation-delay** (retraso para que comience, s o ms)
- **animation-timing-function** (como transition timing function)

ANIMACIONES

- **animation-iteration-count** (un número de veces o infinite)
- **animation-direction** (normal, reverse, alternate, alternate-reverse)
- **animation-play-state** (running, paused)
-

ANIMACIONES

```
selector {
```

```
    //Especificamos las  
    propiedades de la animación  
    animation-name: ejemplo;  
    animation-.....  
    animation-.....
```

```
}
```

```
//Defino los valores intermedios  
de la animación
```

```
@keyframe ejemplo {
```

```
    0% { propiedades-css-vi }
```

```
    ....
```

```
    ....
```

```
    X% { propiedades-css-vX }
```

```
    ...
```

```
    ...
```

```
    100% {propiedades-css-vf}
```

```
}
```



EJEMPLOS

confinamiento

En el año 2020 la sociedad mundial se vió obligada a confinarse en casa debido a la pandemia global causada por el Covid19.

Numerosas plataformas educativas, muchas sin oficio ni beneficio y con apenas contenidos de calidad, aprovecharon la situación intentando ganar un sector de mercado que no poseían.

Muchas fracasaron. El entorno era





Un mundo de posibilidades

THANKS!

Any questions?

You can find me at @pekechis &
<https://github.com/pekechis>