

Animaciones y adicionales

Módulo 05

Introducción

Implementación

Cortes en la animación

En las animaciones pueden generarse cortes, es decir inicialmente nuestra animación **puede sólo ser de la siguiente forma,**



```
@keyframes animacion {  
  
  from { background-color: ■ red; transform: scale(2,1.5)}  
  to { background-color : ■ green; transform: scale(0.5,1.5)}  
  
}
```

Implementación

Cortes en la animación

Sin embargo para darle más potencia a la misma, css nos permite agregar **intervalos de animación** por ejemplo [las reglas anteriores son equivalentes a.](#)



```
✓ @keyframes animacion {  
  
  0% { background-color: ■ red; transform: scale(2,1.5)}  
  100% { background-color : ■ green; transform: scale(0.5,1.5)}  
  
}
```

Implementación

Cortes en la animación

De esta manera, **por ejemplo podemos acrecentar la animación de la siguiente forma,**

```
✓ @keyframes animacion {  
  
  0% { background-color: ■ red; transform: scale(2,1.5);}  
  20% { background-color: ■ yellow; transform: scale(2,1.5) rotate(20deg);}  
  50% { background-color: ■ rgb(255, 0, 191); transform: scale(1,1) rotate(-120deg);}  
  100% { background-color : ■ green; transform: scale(0.5,1.5);}  
  
}
```



Implementación

animation-play-state

Podemos generar una alternativa en los estados de la animación,



```
<!DOCTYPE html>
<head>
<style>
div {
  width: 200px;
  height: 200px;
  border-radius: 20px;
  padding: 20px;
  background: blue;
  animation-name: animacion;
  animation-duration: 6s ;
  animation-timing-function: linear;
  animation-delay: 1s;    }

@keyframes animacion {

0% { background: red; width: 300px; transform: rotate(34deg); }
20% { background: blue; height: 30px; }
50% {background: green;}
100% { background: lightblue; }
}

</style>

</head>
<body>

<div> transiciones </div>
</body>
```

Actualización

Nota

¿Que es adicionales?

En esta sección agregaremos propiedades adicionales a tener en cuenta para trabajar con css.

Actualización de 31/01 2021, propiedad object-fit

Esta propiedad es similar al uso que damos a background-size en sus valores posibles.

Sin embargo en este caso , [lo hacemos sobre la propia imagen inserta por ejemplo.](#)



Implementación

object-fit

Si tenemos en nuestro html una imagen inserta,

```

```

Pero esta imagen no queda bien con este tamaño ya que originalmente tiene otro se deformara, pero ahora podemos con esta propiedad adaptarla de formas diversas.

Probemos en nuestros estilos [los siguientes valores posibles.](#)



Implementación

object-fit

Si tenemos en nuestro html una imagen inserta,

```

```

Pero esta imagen no queda bien con este tamaño ya que originalmente tiene otro se deformara, pero ahora podemos con esta propiedad adaptarla de formas diversas.

Probemos en nuestros estilos [los siguientes valores posibles.](#)



object-fit

Ejemplo **desde css,**

```
img {object-fit: fill;}  
img {object-fit: contain;}  
img {object-fit: cover;}  
img {object-fit: scale-down;}
```

No utilizar conjuntamente **ya que siempre quedaría el valor scale-down ya que es el último.**

Ir probando de a **un valor para ver su diferencia.**



Revisión

- Repasar los conceptos vistos de **animation y adicionales**.
- Trabajar con el **bonus track** debajo de todo para practicar los **elementos individualmente**.
- Ver todos los videos y materiales necesarios antes de continuar



¡Muchas gracias!

¡Sigamos trabajando!