

Animaciones y JS

Módulo 05

Introducción

Animation + JS

Implementación

Animation + JS,

Con javascript podemos controlar nuestra **animación y esta parte es la más interesante de todo lo aprendido.**

Para poder comenzar es importante en todo caso, **o en la mayoría de los casos trabajar** con la propiedad **animation-play-state** en **paused** como vemos en la imagen de la derecha,



```
#cubo {  
  width: 100%;  
  height: 100%;  
  position: absolute;  
  transform-style: preserve-3d;  
  transform: translateZ(-150px);  
  animation-name: girar;  
  animation-duration: 4s;  
  animation-delay: 2s;  
  animation-play-state: paused; }
```

Implementación

Animation + JS,

Luego desde nuestro **archivo podemos a través de botones o cualquier otro elemento** con el cual pueda interactuar el usuario activar tal animación cambiando el valor de **animation-play-state**,



```
#cubo {  
  width: 100%;  
  height: 100%;  
  position: absolute;  
  transform-style: preserve-3d;  
  transform: translateZ(-150px);  
  animation-name: girar;  
  animation-duration: 4s;  
  animation-delay: 2s;  
  animation-play-state: paused; }
```

Implementación

Animation + JS,

Desde nuestro archivo de javascript podemos modificar cuantos valores de propiedades de css creamos convenientes siempre utilizando el método style para poder acceder a la propiedad en cuestión.

Luego si la propiedad tuviese un valor compuesto por ejemplo **background-color** trabajamos con **camelCase**,



```
personalizados.html JS animacion.js X
> JS animacion.js > ...
1 document.getElementById('cambiar').onclick = cambiar;
2
3 function cambiar() {
4   document.getElementById('caja').style.backgroundColor = 'red';}
5
```

Implementación

Animation + JS,

Por ejemplo **en el caso del trabajo con animaciones una opción puede ser la siguiente,**



```
<> personalizados.html    JS animacion.js X
js > JS animacion.js > ...
1  document.getElementById('btn_girar').onclick = girar;
2  document.getElementById('btn_parar').onclick = parar;
3
4
5  ✓ function girar() {
6    document.getElementById('cubo').style.animationPlayState = 'running';
7    document.getElementById('cubo').style.animationDelay = '3s';}
8
```

Animation-iteration-count

Implementación

animation-iteration-count

Indica cuantas veces se reproduce la animación,

El valor predeterminado es 1, sin embargo podemos poner un número entero que representa la **cantidad de veces que la animación va a reproducirse**.

Es importante saber que contamos con el valor **infinite** si queremos que la **animación genere un loop**.



iteration count + direction

Implementación

iteration-count + direction

Cuándo trabajamos con iteration count en valores **superiores a 1** es importante saber cómo será la animación ya que si el valor de direction lo dejamos en **reverse o normal** notaremos un **corte que en general no es agradable**.

Por eso generalmente en las **reglas notamos lo siguiente**,

```
#cubo {  
  width: 100%;  
  height: 100%;  
  position: absolute;  
  transform-style: preserve-3d;  
  transform: translateZ(-150px);  
  animation-name: girar;  
  animation-duration: 4s;  
  animation-delay: 2s;  
  animation-play-state: paused;  
  animation-iteration-count: infinite;  
  animation-direction: alternate;  
}
```



Implementación

animation-fill-mode

Un ejemplo completo de animación incluyendo animation-fill-mode sería,

```
<style>
div {
  width: 200px;
  height: 200px;
  border-radius: 20px;
  padding: 20px;
  background: blue;
  animation: animacion 3s linear forwards infinite alternate ; }

@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>
```



Revisión

- Repasar los conceptos vistos de **animation**.
- 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!