CSS ANIMACIONES

Estefany AguilarLínea horizontal

# 12 principios de la animación:

1. Squash and Stretch
2. Timing and motion
3. Anticipation
4. Staging
5. Follos through and overlapping action
6. Straight ahead and pose to pose action
7. Slow in and out
8. Arcs
9. Exaggeration
10. Secondary action
11. Solid Drawing.
12. Appeal.

# Contador con Css

* Counter-reset: Crea o reinicia el contador.
* Counter-increment: Incrementa un valor del contador.
* Counter: Función que agrega un valor de un contador.

**HTML:**

* <input type="checkbox">

Agrego un vector de valor, El atributo value es usado para definir el valor enviado por el checkbox. El atributo **checked** se usa para indicar que el elemento está seleccionado. El atributo indeterminate se usa para indicar que el checkbox está en un estado indeterminado

input:checked {

    counter-increment: game;

}

<H3 class="Total-counter">Score:</H3>

.Total-counter::after {

    content: counter(game);

}

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

# Contexto de Apilamiento:

* Identificar los elementos o capas que requerimos. Identificar en que orden irían partiendo de que salen del eje z.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

.phone {

    position: relative;

    border: 8px solid rgb(27, 25, 25); Indicamos el grosor del borde y color que le aplicaremos

    border-radius: 35px; para poner los bordes redondos.

    height: 600px;

    width: 300px;

    box-shadow: 0 19px 38px rgb(0, 0, 0, 0.30), sombra

    0 15px 12px rgb(0, 0, 0, 0.22);

    animation: anim 5s infinite alternate linear; valores de animación, segundos que dura en alternar los colores.

}

@keyframes anim {

    0% {

        background: #5dc2fc;

    }

    25% {

        background: #33ebf185;

    }

    50% {

        background: #1a5b97;

    }

    75% {

        background: #2ae4f1;

    }

     100% {

        background: #e5f1f8;

    }

}

**Posición de los elementos:**

* Utilizamos la propiedad **z-index** en cada clase. Cuando varios elementos se superponen, los elementos con mayor valor z-index cubren aquellos con menor valor.
* **@keyframes:** controlar los pasos intermedios en una secuencia de animación css. Se establecen tiempos de trayectoria (keyframe) compuesto por un % a lo largo de la animación. Debe iniciar de 0% a 100%, si no la declaración keyframe se invalida.
* **Border Radius:** Hay dos formas de aplicarlo:

Con shorthands:

Border-radius: 15px 0px 0px 0px;

O indicando el radio de cada esquina, que sería igual a esto:

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Los valores dados al atributo se empiezan a aplicar desde la esquina superior izquierda en sentido de las manecillas del reloj.

# Animation name y keyframe

* Pueden tener sintaxis con un solo valor o mas de un valor.
* **Keyframe:** se manipula el tiempo desde el principio hasta el final. 0% es tiempo inicial y 100% final.
* También se puede indicar el inicio con from y el final con to.
* **Shorthand:** forma de especificación reducida de escribir las propiedades de estilos.

# Animation Duration:

* Tiempo en el que me voy a demorar en hacer la acción de la animación.

animation-name: blink;

animation-duration: 2s;

animation-iteration-count: infinite;

Indico el nombre de la animación, su duración, y la frecuencia con la que se realizara. En el keyframe debo citar la propiedad de mi elemento que quiero modificar y los valores a modificar.

# Triggers: Layout , Paint y composite

* **Composite:** Ordena las partes de la página. Propiedades como opacity y transrform.
* **Paint:** Rellena pixeles, implica colores, imágenes textos y sombras.
* **Layout:** Diseño de página, ancho, margin, padding, border.

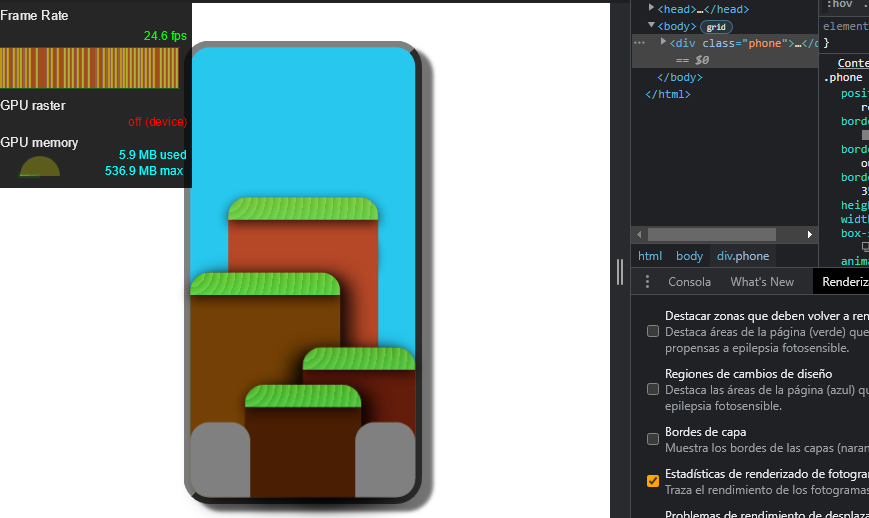
Hay propiedades que utilizan varios procesos , y por ende se demora en renderizar.

# Debbuging de animaciones con Devtools

* Inspeccionar los estados de los códigos.
* Inspect

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente



* En Frame rendering stats, se ve los frams x segundo que no deben pasar de 60.
* Frames son las capas.

# Buenas practicas para optimizar animaciones web

* Usar propiedades que solo pasen por el proceso de composite.
* Homologar propiedades.
* Evitar animar muchas propiedades al mismo tiempo.
* Esconder elementos con : opacity y visibility
* Evita hacer animaciones que ocurran al hacer scroll, ya que el evento que escucha el scroll se ejecuta una gran cantidad de veces. Mejor espera a llegar a cierto punto de la pantalla y ahí ejecutas la animación.

# Titulo

# Titulo

# Titulo

# Titulo

# Titulo

# Titulo

# Titulo

# Titulo

# Titulo

# Titulo