



ANIMACIONES

ROADMAP CUATRIMESTRE Y TPE



TRABAJO PRÁCTICO ENTREGABLE N°4



IMPLEMENTAR WEB DE PRESENTACIÓN DE JUEGO

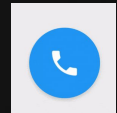
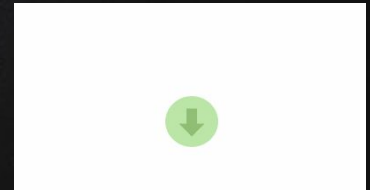
- X ANIMACIONES
- X PARALLAX SCROLLING
- X SPRITESHEET



ANIMACIONES

ANIMACIONES: IMPORTANCIA

- El feedback visual es útil cuando se quiere informar al usuario del resultado de una operación
- En los casos en los cuales la operación no es completada correctamente, una animación funcional indica donde está el error
- Los usuarios quieren conocer el estado del sistema en cualquier momento
- Las apps no deben hacer que el usuario deba adivinar
- Se pueden usar animaciones para indicar que está pasando mediante feedback visual
- Atraer la atención del Usuario o guiar al usuario



ANIMACIONES: PROCEDURALES (JS O CSS)

La posición y orientación de los elementos animados se actualizan por código evaluando un función o rutina...



```
let seconds = 0;  
let clockhand = document.querySelector(...);  
  
setInterval(function() {  
  seconds = (seconds+1) % 60;  
  clockhand.style.transform = 'rotateZ(${360 * seconds / 60.0}deg)';  
}, 1000);
```

DEMO ...



<https://codepen.io/interf-usuario-interac/pen/MWGZxNJ>

<https://codepen.io/interf-usuario-interac/pen/WNJLWeQ>

<https://codepen.io/interf-usuario-interac/pen/GRdPLKL>



PARALLAX SCROLLING

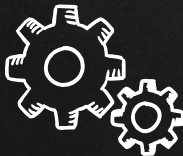
KEYFRAME: ANIMACION

Parallax

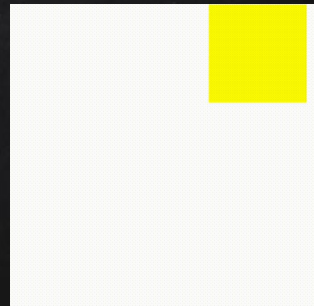


KEYFRAME: ANIMACION

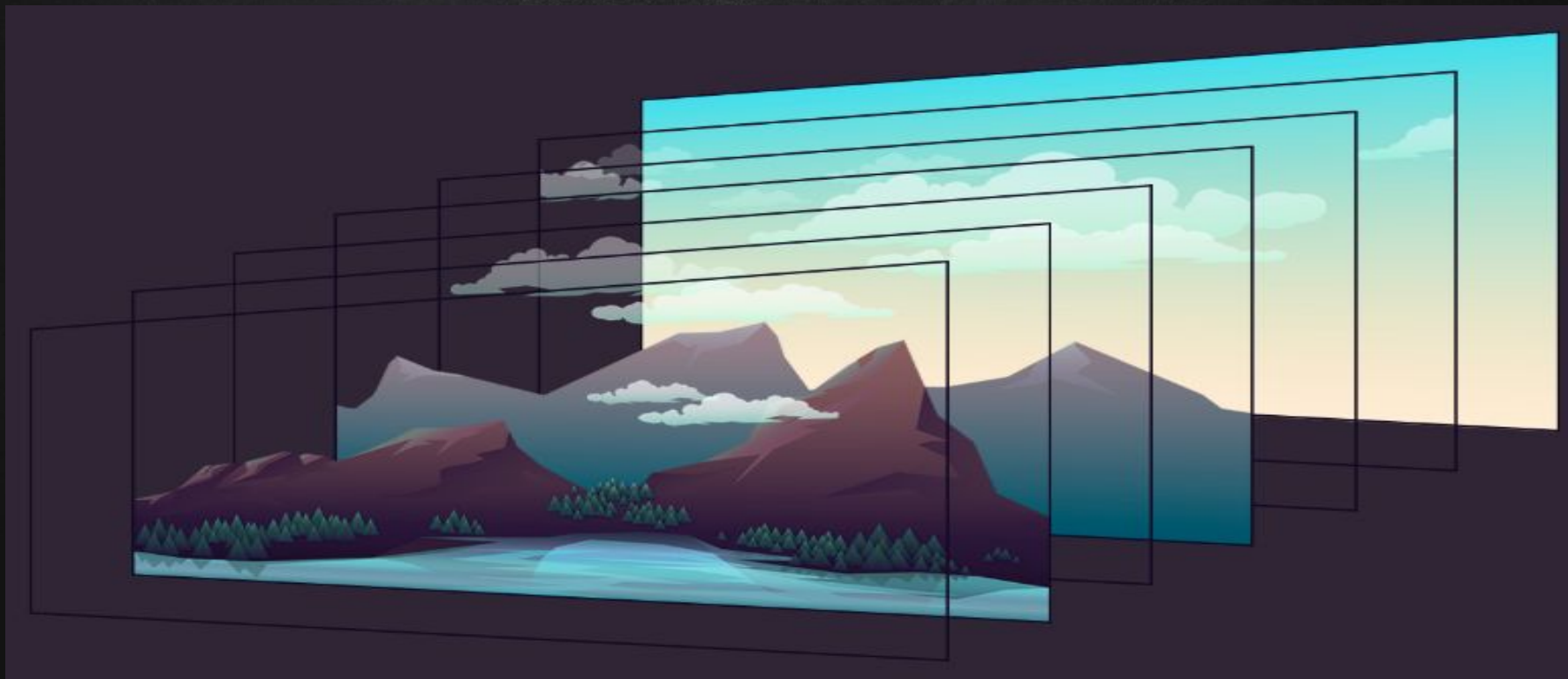
- ✗ Una lista de keyframes (%)
- ✗ Una duración
- ✗ Repetición?
- ✗ Parámetros adicionales
 - Dirección
 - Continua o discreta
 - Mapeo del tiempo



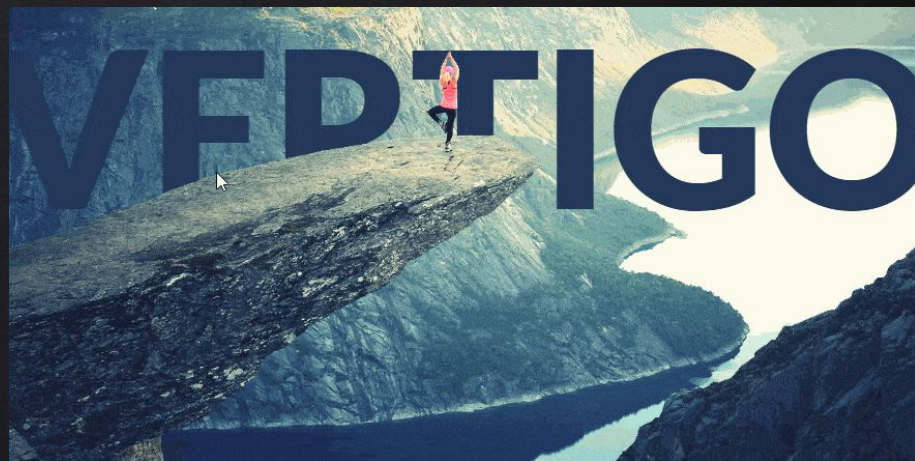
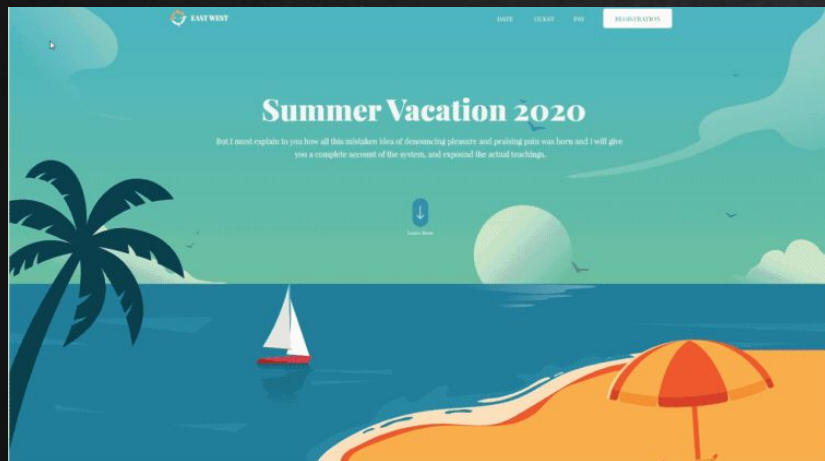
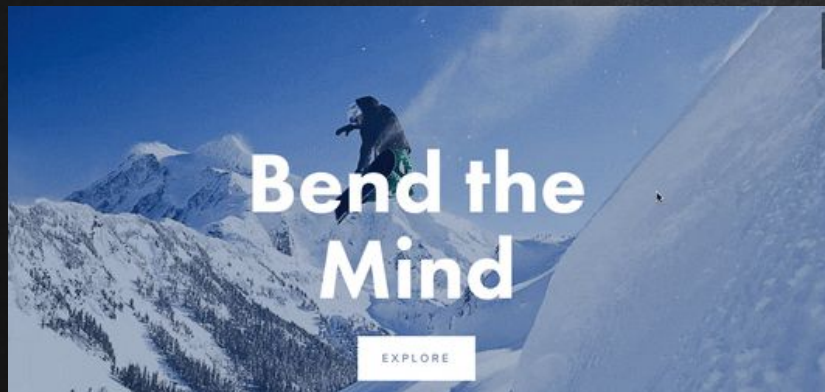
```
@keyframes example {  
  0% {background-color: red; left: 0px; top: 0px;}  
  25% {background-color: yellow; left: 200px; top: 0px;}  
  50% {background-color: blue; left: 200px; top: 200px;}  
  75% {background-color: green; left: 0px; top: 200px;}  
  100% {background-color: red; left: 0px; top: 0px;}  
}  
  
div {  
  width: 100px;  
  height: 100px;  
  position: relative;  
  background-color: red;  
  animation-name: example;  
  animation-duration: 4s;  
}
```



BACKGROUND







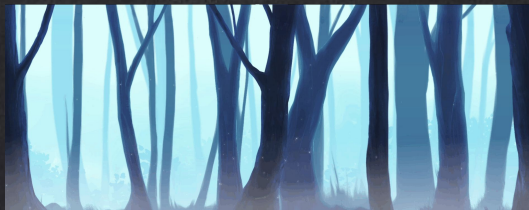
<https://codepen.io/StephenKoller/pen/LWBLNW>

PARALLAX: BACKGROUND HORIZONTAL

- CADA LAYER DEBE SER "TILEABLE" PARA PODER GENERAR UN BACKGROUND INFINITO
- GENERA SENSACIÓN DE PROFUNDIDAD MOVIENDO LAS CAPAS A DIFERENTES VELOCIDADES
- LAS CAPAS MÁS LEJANAS SE MUEVEN MÁS LENTO QUE LAS CAPAS CERCANAS



DEMO ...



```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
  <link rel="stylesheet" href="style.css">
</head>
<body>
  <div class="layer layer-10"></div>
  <div class="layer layer-9"></div>
  <div class="layer layer-8"></div>
  <div class="layer layer-7"></div>
  <div class="layer layer-6"></div>
  <div class="layer layer-5"></div>
  <div class="layer layer-4"></div>
  <div class="layer layer-3"></div>
  <div class="layer layer-2"></div>
  <div class="layer layer-1"></div>
</body>
</html>
```

<https://codepen.io/interf-usuario-interac/pen/dyewLWa>

```
@keyframes scroll {
  0% {background-position-x: 0px;}
  100% {background-position-x: -1920px;}
}

.layer {
  min-height: 500px;
  width: 100%;
  height: 100%;
  position: absolute;
  top: 0;
  left: 0;
  background-repeat: repeat-x;
  background-size: cover;
}

.layer-10 {
  background-image: url("10.png");
  animation: scroll 20s infinite linear;
}

.layer-9 {
  background-image: url("09.png");
  animation: scroll 15s infinite linear;
}

.layer-8 {
  background-image: url("08.png");
  animation: scroll 10s infinite linear;
}

.layer-7 {
  background-image: url("07.png");
  animation: scroll 8s infinite linear;
}

.layer-6 {
  background-image: url("06.png");
  animation: scroll 6s infinite linear;
}
```

PARALLAX: SCROLLING

- CADA ELEMENTO SE MUEVE A DIFERENTE VELOCIDAD A MEDIDA QUE SE HACE SCROLL
- GENERA SENSACIÓN DE DINAMISMO
- EL CAMBIO LO HACEMOS USANDO JS (HAY EJEMPLOS 100% CSS)
- CAMBIANDO A DIFERENTES UNIDADES LA POSICIÓN DE LOS ELEMENTOS



DEMO ...



```
const wrapper = document.getElementById("wrapper");
const parallax0 = document.getElementById("parallax0");
const parallax1 = document.getElementById("parallax1");
const parallax2 = document.getElementById("parallax2");

wrapper.onscroll = function () {
  let y = wrapper.scrollTop;
  parallax0.style.top = 1800 - y * 2.65 + "px";
  parallax1.style.top = 1200 - y * 0.75 + "px";
  parallax2.style.top = 800 + y * 0.5 + "px";
};
```



```
#parallax0,#parallax1,#parallax2 {
  position: absolute;
  height: 450px;
  width: 450px;
  border-radius: 50%;
  text-align: center;
  line-height: 390px;
  box-shadow: 10px 5px 25px #333;
  color: #fff;
}
```

```
#parallax0 {
  background-color: #01579b;
  top: 1800px;
  left: 45%;
  z-index: 22;
}
```

```
#parallax1 {
  background-color: #4f83cc;
  top: 1200px;
  left: 55%;
  z-index: 21;
}
```

```
#parallax2 {
  background-color: #002f6c;
  top: 900px;
  left: 25%;
  z-index: 20;
}
```

PARALLAX: MOUSE MOVING

- CADA ELEMENTO SE MUEVE PARA DIFERENTES LADOS SEGÚN POSICIÓN DEL MOUSE
- GENERA SENSACIÓN DE SEGUIMIENTO
- EL CAMBIO LO HACEMOS USANDO JS
- CAMBIANDO X E Y EN % AL MOVIMIENTO



DEMO ...



```
(function() {  
  // Add event listener  
  document.addEventListener("mousemove", parallax);  
  const elem = document.querySelector("#parallax");  
  // Magic happens here  
  function parallax(e) {  
    let _w = window.innerWidth/2;  
    let _h = window.innerHeight/2;  
    let _mouseX = e.clientX;  
    let _mouseY = e.clientY;  
    let _depth1 = `${50 - (_mouseX - _w) * 0.01}% ${50 - (_mouseY - _h) * 0.01}%`;   
    let _depth2 = `${50 - (_mouseX - _w) * 0.02}% ${50 - (_mouseY - _h) * 0.02}%`;   
    let _depth3 = `${50 - (_mouseX - _w) * 0.06}% ${50 - (_mouseY - _h) * 0.06}%`;   
    let x = `${_depth3}, ${_depth2}, ${_depth1}`;  
    console.log(x);  
    elem.style.backgroundPosition = x;  
  }  
})();
```

```
#parallax {  
  position: relative;  
  width: 100%;  
  height: 100vh;  
  background-image:  
url(https://raw.githubusercontent.com/oscicen/oscicen.github.io/master/img/depth-3.png),  
url(https://raw.githubusercontent.com/oscicen/oscicen.github.io/master/img/depth-2.png),  
url(https://raw.githubusercontent.com/oscicen/oscicen.github.io/master/img/depth-1.png);  
  background-repeat: no-repeat;  
  background-position: center;  
  background-position: 50% 50%;  
}
```

<https://codepen.io/interf-usuario-interac/pen/LYqzRNd>

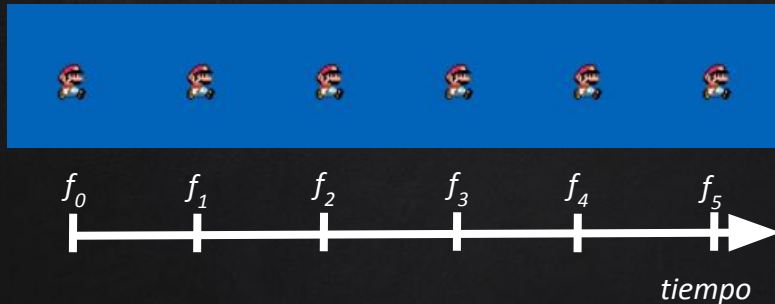




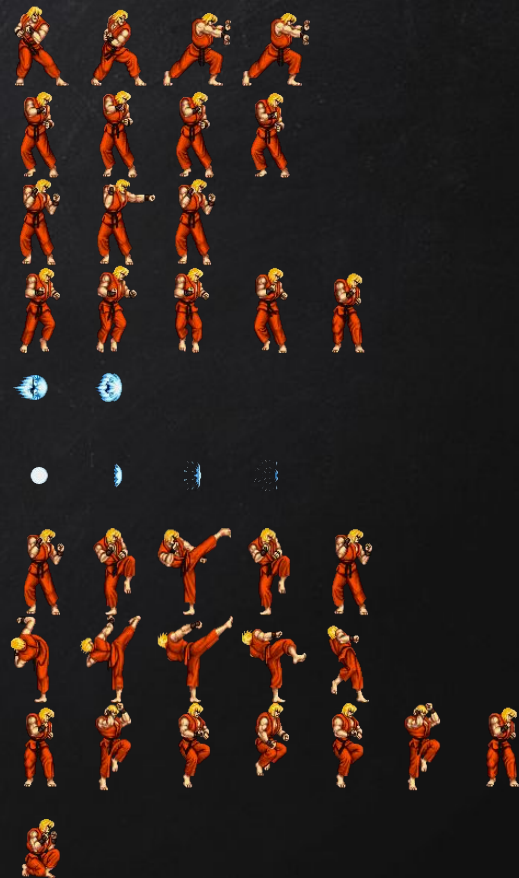
SPRITESHEET

KEYFRAME: CLÁSICO (SPRITESHEET)

- SECUENCIA DE CUADROS (FRAMES) MOSTRADOS EN SECUENCIA PARA DAR SENSACIÓN DE CONTINUIDAD
- MÉTODO CLÁSICO DE ANIMACIÓN (DIBUJOS ANIMADOS Y VIDEOJUEGOS)
- PRIMERA ANIMACIÓN EN 1902

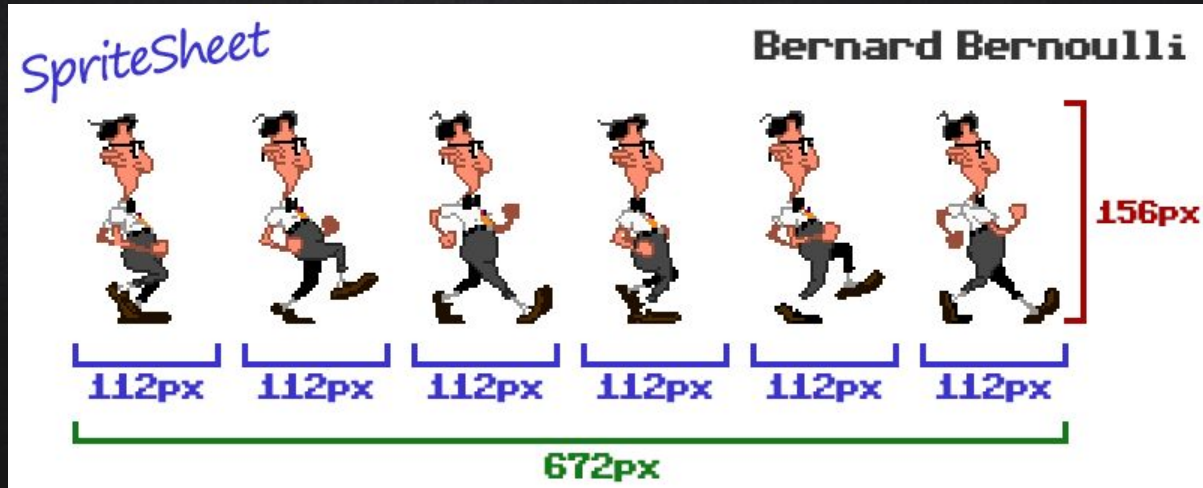


KEYFRAME: EJEMPLOS

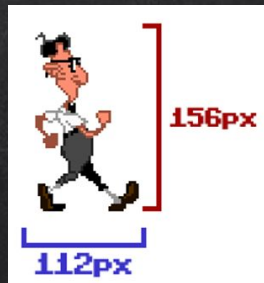


KEYFRAME: QUE VAMOS A EXPLICAR?

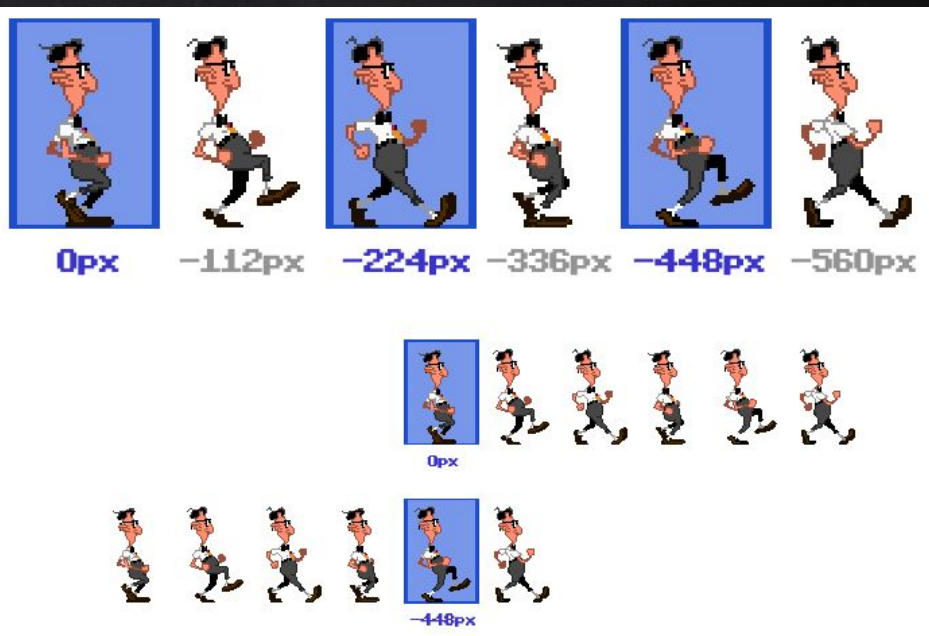
- SIMULAR QUE CAMINA
- SIMULAR QUE AVANZA
- ANIMAR EL SPRITESHEET



```
#bern {  
  width: 112px;  
  height: 156px;  
  background: url(bern.png) left center;  
  animation: walk 8s linear infinite;  
}
```



```
@keyframes walk {  
  0% { background-position: 0px; }  
  100% { background-position: -672px; }  
}
```





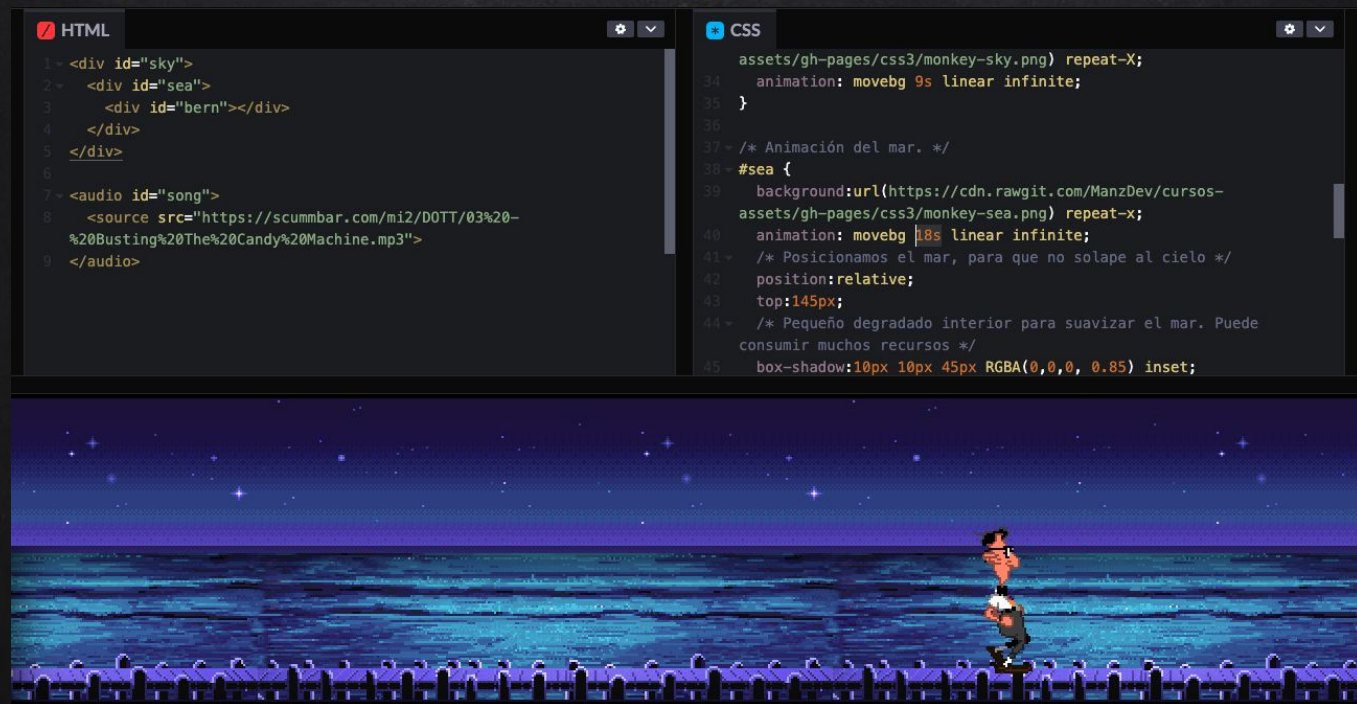
```
#bern {  
  width: 112px;  
  height: 156px;  
  background: url(bern.png) left center;  
  animation: walk 8s linear infinite;  
}
```



```
#bern {  
  width: 112px;  
  height: 156px;  
  background: url(bern.png) left center;  
  animation: walk 8s steps(6) infinite;  
}
```



DEMO ...





THANKS!

