

# Workshop HTML

## Requisitos:

- Visual Studio Code
- Extensão Live Server(Visual Studio Code)\*

\*opcional

## Localização:

### Link da Sala:

<https://us05web.zoom.us/j/4052134344?pwd=L0orSmVkZVI4WnYxOGwvditwb3R2dz09>

### ID da reunião:

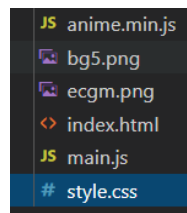
405 213 4344

### Senha de acesso:

JCGM20/21

## 1ºEtapa

Estrutura dos ficheiros iniciais:



Código html inicial:

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width,
initial-scale=1.0">
  <title>Workshop HTML5</title>
  <script src="main.js"></script>
  <script src="anime.min.js"></script>
  <link rel="stylesheet" href="style.css">
</head>

<body onload="onLoad()">
  <div class="logo">
    
  </div>
  <div class="stagger-visualizer"></div>
</body>

</html>
```

## 2ºEtapa

Código css inicial:

```
body,
html {
  position: absolute;
  display: flex;
  flex-wrap: wrap;
  justify-content: center;
  align-items: center;
  width: 100%;
  height: 100vh;
  background-image: url("bg5.png");
  background-size: cover;
  background-repeat: no-repeat;
}

.stagger-visualizer {
  display: flex;
  flex-wrap: wrap;
  justify-content: center;
  align-items: center;
  width: 17rem;
  height: 17rem;
}

.stagger-visualizer div {
  width: 1rem;
  height: 1rem;
  background-color: #000;
}

.logo {
  position: absolute;
  top: 5%;
  transform: translate(-5%, 0);
}
```

### 3º Etapa

Código javascript inicial:

```
function onLoad() {
  const staggerVisualizerEl =
document.querySelector(".stagger-visualizer");
  const fragment = document.createDocumentFragment();
  const grid = [17, 17];
  const col = grid[0];
  const row = grid[1];
  const numberOfElements = col * row;

  for (let i = 0; i < numberOfElements; i++) {
    fragment.appendChild(document.createElement("div"));
  }

  staggerVisualizerEl.appendChild(fragment);

  const staggersAnimation = anime
    .timeline({
      targets: ".stagger-visualizer div",
      easing: "easeInOutSine",
      delay: anime.stagger(50),
      loop: true,
      autoplay: false,
    })
}
```

## 4º Etapa

### Primeira parte da animação

```
...
.add({
  translateX: [
    {
      value: anime.stagger("-.1rem", {
        grid: grid,
        from: "center",
        axis: "x",
      }),
    },
    {
      value: anime.stagger(".1rem", {
        grid: grid,
        from: "center",
        axis: "x",
      }),
    },
  ],
  translateY: [
    {
      value: anime.stagger("-.1rem", {
        grid: grid,
        from: "center",
        axis: "y",
      }),
    },
    {
      value: anime.stagger(".1rem", {
        grid: grid,
        from: "center",
        axis: "y",
      }),
    },
  ],
  duration: 1000,
  scale: 0.5,
  delay: anime.stagger(100, { grid: grid, from: "center" }),
})
```

## 5ºEtapa

Segunda parte da animação

```
...
.add({
  translateX: () => anime.random(-10, 10),
  translateY: () => anime.random(-10, 10),
  delay: anime.stagger(8, { from: "last" }),
})
```

## 6ºEtapa

Terceira parte da animação

```
...
.add({
  translateX: anime.stagger(".25rem", {
    grid: grid,
    from: "center",
    axis: "x",
  }),
  translateY: anime.stagger(".25rem", {
    grid: grid,
    from: "center",
    axis: "y",
  }),
  rotate: 0,
  scaleX: 2.5,
  scaleY: 0.25,
  delay: anime.stagger(4, { from: "center" }),
})
```

## 6ºEtapa

Quarta parte da animação

```
.add({
  rotate: anime.stagger([90, 0], { grid: grid, from: "center" }),
  delay: anime.stagger(50, { grid: grid, from: "center" }),
})
```

## 7ºEtapa

Quinta parte da animação

```
...
.add({
  translateX: 0,
  translateY: 0,
  scale: 0.5,
  scaleX: 1,
  rotate: 180,
  duration: 1000,
  delay: anime.stagger(100, { grid: grid, from: "center" }),
})
```

## 8ºEtapa

Sexta parte da animação

```
...
.add({
  scaleY: 1,
  scale: 1,
  delay: anime.stagger(20, { grid: grid, from: "center" }),
});
```

## 9ºEtapa

Colocar a animação a dar

```
...
staggerAnimation.play();
```