

# Animação

## Relembrando

Temos uma outra propriedade que já foi vista anteriormente que é utilizada para animar elementos, a propriedade `transition`.

O `transition` é utilizado para animar um elemento em estado diferente, por exemplo, um elemento `button` vai ter um `background red`, mas ao passar o mouse sobre ele o seu `background` ficará `blue`. Nesse caso o `transition` é utilizado para tornar essa transição mais suave.

- `transition-property`
- `transition-duration`
- `transition-delay`
- `transition-timing-function`

## Atalho para a propriedade `transition`

### Transition (shorthand)

```
transition: property duration timing-function delay;
```

## Animation & Keyframes

O `keyframe` é utilizado para criar animações. Com o `animation` definimos a animação para o elemento.

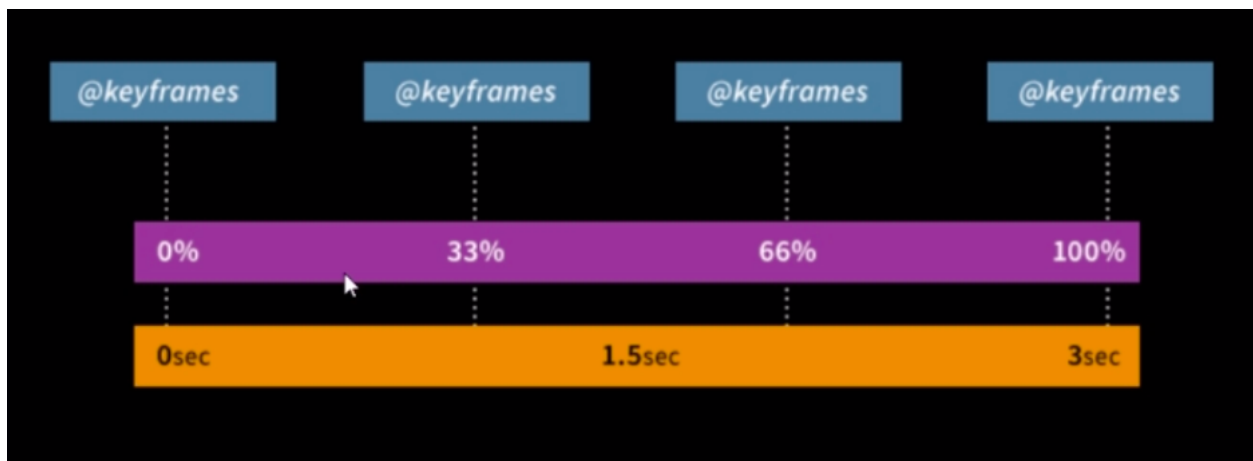
Animation x Transition

# transition vs animation

- apenas para transições simples (sem keyframes)
  - dependem de interação do usuário (:hover, :focus, :active) ou do javascript para adicionar/remover uma classe
- transições complexas (com um ou mais keyframes)
  - não dependem de interação do usuário (mas ainda podemos animar com javascript - adicionar/remover uma classe)
  - maior controle sobre a animação criada

## Timeline

A baixo temos uma imagem que demonstra um exemplo de timeline, que é a linha de tempo que a animação irá ocorrer. No exemplo a baixo temos uma animação de 3 segundos possuindo 4 quadros (keyframes), nesse caso o total de tempo será dividido pelo total de quadros, no caso o primeiro quadro é o estado inicial do elemento.



## Dicas

Somente valores numéricos podem ser animados. Ex: 10px, .5, 58%, 10em, #cf0, red. Valores como auto, arial e hidden, e entre outro não são numéricos.

Propriedades mais performática para animações são: transforme e opacity

Se preciso utilize will-change para tentar melhorar a performance da animação .

O que o will-change faz ?

Ele informa ao browser que em algum momento terá uma animação. O browser por sua vez já reserva um espaço da memória para a animação. Não é recomendado utilizar vários will-change, pois assim você terá o efeito inverso, ao invés de ter mais desempenho terá menos.

O will-change é aplicado no elemento que terá animação seguido da propriedade que será animada, conforme exemplo a baixo:

```
div{
  will-change: color;
}
```

No exemplo acima estou avisando para o browser que em algum momento a div sofrerá uma alteração em sua cor.

## Definindo um Keyframe

```
@keyframes animaCor{
  from{
    color: red;
  }
  to{
    color: blue;
  }
}
```

```
@keyframes animaCor{
  0%{
    color: red;
  }
  100%{
    color: blue;
  }
}
```

Com keyframes intermediários:

```
@keyframes animaCor{
  0%{
    color: red;
  }
  50%{
    color: orange;
  }
}
```

```
}
100%{
  color: blue;
}
}
```

Definindo a animação no elemento

```
div{
  animation: animaCor 2s;
}
```

## Alguns atributos da propriedade animation

**Animation-name:** Recebe o nome da animação na qual foi declarada através de um `@keyframe`.

**Animation-duration:** Recebe o tempo que a animação irá ocorrer.

**Animation-timing-function:** Ele recebe como será a aceleração da animação, podendo receber os seguintes valores: `linear`, `ease-in`, `ease-out`, `ease-in-out`, `cubic-bezier`.

**Animation-iteration-count:** Definir a quantidade de vezes na qual a animação pode acontecer, ela recebe os seguintes valores: `quantidade de vezes` ou `infinite`.

**Animation-direction:** recebe a direção na qual a animação irá ocorrer, podendo receber os seguintes valores: `alternate` (a direção fica alternando), `reverse` (inicia-se do final para o início), `alternate-reverse` (que mistura o `alternate` e o `reverse`).

**Animation-delay:** Defini um tempo de atraso para iniciar a animação.

**Animation-fill-mode:** Caso o elemento tenha propriedade que serão mudadas na animação, o fill-mode permite definir quando será aplicado as propriedade da animação. Ela pode receber os seguintes valores: `backwards` (antes de iniciar a animação os valores já são aplicados na animação), `forwards` (após finalizar a animação os valores se mantém aplicado no elemento) e `both` (uma mistura do `backwards` e do `forwards`).

**Animation-play-state:** ele defini o status da animação, se está sendo executada ou pausada, recebendo os seguintes valores: `running` (executando) e `paused` (pausado).

