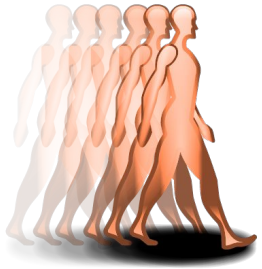

ANIMAÇÃO

animation

As transições dependem de uma interação do usuário para acontecer. Para iniciarmos essas ações automaticamente podemos utilizar a propriedade **animation**.

animation

Animações CSS tornam possível animar transições de um estilo CSS para outro. Animações se consistem de dois componentes: um estilo descrevendo a animação e um set de keyframes que indicam o estado final e inicial do estilo CSS da animação, bem como possíveis waypoints intermediários ao longo do caminho.

animation - configurando

Para criar uma sequência de animação CSS, você estiliza o elemento que deseja animar com a propriedade **animation** ou suas sub-propriedades. Isso permite que você configure o comportamento da animação, bem como outros detalhes, como a sequência de animação deveria progredir. Isso não configura a sequência atual da animação, que é feita usando a regra @keyframes.

animation - configurando

```
.bolinha {
    ...
    position: relative;
    animation: quica 1s infinite;
}

@-webkit-keyframes quica {
    from {
        top: 0;
    }
    to {
        top: 1 em;
    }
}
```

```
@keyframes quica {
    from {
        top: 0;
    }
    to {
        top: 1 em;
    }
}
```

animation - configurando

As sub-propriedades da propriedade **animation** são:

- **animation-delay :** delay entre o tempo em que o elemento é carregado e o início da sequência de animação.

- **animation-direction :** alternar ou não a direção em cada execução durante a sequência ou voltar ao ponto inicial e se repetir.

```
/* Single animation */
animation-direction: normal;
animation-direction: reverse;
animation-direction: alternate;
animation-direction: alternate-reverse;
```

animation - configurando

- **animation-duration :** o tempo que uma animação deveria levar para completar um ciclo.

- **animation-iteration-count** : o número de vezes que uma animação deveria se repetir.

```
/* Keyword value */
animation-iteration-count: infinite;

/* <number> values */
animation-iteration-count: 3;
```

animation - configurando

- **animation-name** : nome da regra descrevendo os keyframes da animação.

- **animation-play-state** : pausar ou resumir a sequência da animação.

```
animation-play-state: running;
animation-play-state: paused;
```

- **animation-timing-function** : como se comporta o ritmo da transição durante o efeito.

```
animation-timing-function: ease;
animation-timing-function: ease-in;
animation-timing-function: ease-out;
animation-timing-function: ease-in-out;
animation-timing-function: linear;
```

animation - Definindo a sequência de animação

Configurada o comportamento da animação, você precisa definir a aparência da animação. Isso é feito por criando dois ou mais keyframes. Cada keyframe descreve como o elemento animado deve se comportar durante a sequência de animação.

Como a comportamento da animação é definida por um estilo CSS que configura a animação, keyframes usam uma porcentagem para indicar o tempo. 0% (**from**) indica o primeiro momento da sequência de animação, enquanto 100%(**to**) indica o estado final da animação. Esses dois tempos devem ser especificados para que o navegador saiba onde a animação deve começar e parar.

animation - Definindo a sequência de animação

```
@keyframes animacao {
    from {
        width: 100px;
        background: black;
    }
    to {
      background: yellow;
        width: 200px;
    }
}
```

```
@keyframes animacao {
    0%{
        width: 100px;
        background: black;
    }
    100%{
      background: yellow;
        width: 200px;
    }
}
```

animation - Definindo a sequência de animação

Você pode opcionalmente incluir keyframes adicionais que descrevem passos intermediários ao longo do caminho do ponto inicial ao ponto final da animação.

```
@keyframes animacaoBolada {
   0%   {
      background: black;
      width: 100px;
   }
   25%  { background: green; }
   50%  { background: blue; }
   75%  { background: red; }
   100% {
      background: yellow;
      width: 200px;
   }
}
```

animation - Fazendo o texto deslizar através da janela do navegador

Esse exemplo simples estiliza o elemento <p> onde o elemento desliza para dentro vindo de fora da lateral direita da janela do navegador.

Perceba que animações como essa podem fazer com que a página se torne mais larga que a janela do navegador. Para evitar esse problema coloque o elemento a ser animado dentro de um container, e atribua overflow:hidden ao container.

animation - Fazendo o texto deslizar através da janela do navegador

Esse exemplo simples estiliza o elemento <p> onde o elemento desliza para dentro vindo de fora da lateral direita da janela do navegador.

Perceba que animações como essa podem fazer com que a página se torne mais larga que a janela do navegador. Para evitar esse problema coloque o elemento a ser animado dentro de um container, e atribua overflow:hidden ao container.

animation - Fazendo o texto deslizar através da janela do navegador

```
p {
  animation-duration: 3s;
  animation-name: slidein;
}

@keyframes slidein {
  from {
    margin-left: 100%;
    width: 300%
  }

  to {
    margin-left: 0%;
    width: 100%;
  }
}
```

animation - faça repetir-se

Para fazer a animação repetir, simplesmente use a propriedade **animation-iteration-count** para indicar a quantidade de vezes que a animação deve se repetir. Neste caso, vamos usar infinite para que a animação se repita indefinidamente:

```
p {
  animation-duration: 3s;
  animation-name: slidein;
  animation-iteration-count: infinite;
}
```

animation - Fazendo a animação se mover para trás e para frente

Com o exemplo anterior, fizemos a animação se repetir, mas é muito estranho tê-la pulando lá do início toda vez que a animação inicia. O que nós realmente queremos é que a animação se mova para trás e para frente por toda tela. Isso é facilmente realizado se adicionarmos o valor alternate à propriedade **animation-direction**:

```
p {
  animation-duration: 3s;
  animation-name: slidein;
  animation-iteration-count: infinite;
  animation-direction: alternate;
}
```