

MATERIAL EXTRA

Animações CSS



growdev

ANIMAÇÕES EM CSS

O CSS surgiu como forma de personalizar códigos feitos em programação, e isso nos leva a muitas possibilidades. Após entender o básico, de que se utiliza em conjunto com arquivos HTML, e auxilia a visualização dos códigos de personalização, nós vamos para um avanço dessa compreensão: as animações.

Com as animações, tornamos o CSS muito poderoso e mais assertivo na hora de criar interfaces com uma boa experiência. Com as interfaces fazendo cada vez mais parte do dia-a-dia, as necessidades de deixá-las mais simples, funcionais e confortáveis aumentaram, e isso também inclui uma interação maior da interface para com o usuário.

Além de se adaptar a tecnologias diferentes como celulares, tablets e relógios, a personalização começou a transformar aplicativos e sites em algo agradável. Botões que se mexem, fotos que desaparecem aos poucos para acostumar o olhar, formas que se alteram conforme a necessidade, tudo está presente na parte de animações. Vamos ver um pouco disso então.



Figura 1 - Exemplo de animação - uxplanet.com

Código

Para criar uma animação, usamos a propriedade “animation” no elemento que queremos aplicar a animação, como por exemplo em uma “div”. E para funcionar, usamos a propriedade “@keyframes” no CSS, onde nós decidimos o nome desta animação e estipulamos o início e o fim dela, como se fosse uma linha do tempo.



Figura 2 - Exemplo de animação com os códigos abaixo

```
<head>
  <style>
    div {
      width: 100px;
      height: 100px;
      background: blue;
      position: relative;
      animation: DivAnimada 5s infinite;
    }

    @keyframes DivAnimada {
      from {
        left: 0px;
      }
      to {
        left: 200px;
      }
    }
  </style>
</head>
```

Nome da animação

Tempo da animação

Repetição

Início

Atributo inicial

Fim

Atributo final

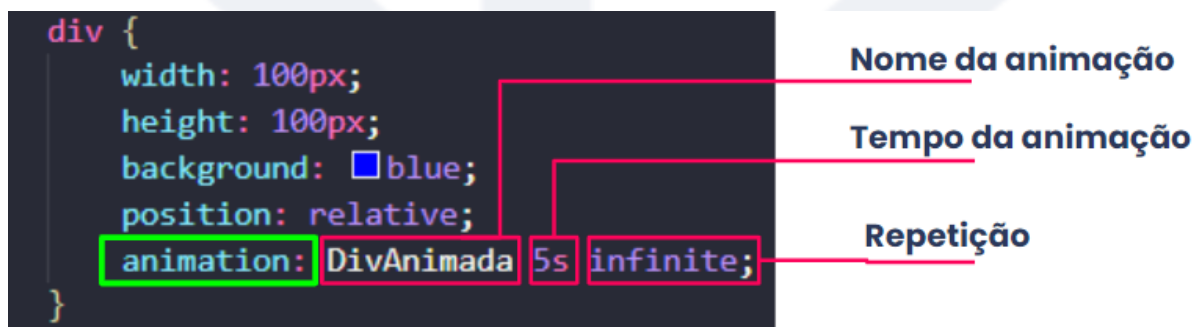
Animation

A propriedade “animation” é uma abreviação de outros comandos como “animation-name” , “animation-duration”, entre outros. Utilizamos a palavra “animation” e já adicionamos os atributos como forma de diminuir linhas.

Atenção!

Sempre adicione a duração da animação, pois o padrão é 0, o que significa que não rodará a animação.

Usando com o base o exemplo mostrado anteriormente, podemos resumir o CSS responsável pela animação usando a abreviação “animation” da seguinte maneira:



Explicando o comando acima:

- **Animation:** Adicionado “animation” dentro da div;
- **Nome da animação:** “DivAnimada”, é o mesmo nome usado em “@keyframes”.
- **Tempo da animação:** É a duração. Está usando a medida em segundos (por isso o valor é 5s).
- **Repetição:** Não é obrigatório; o “infinite” é um exemplo de repetição infinita, mas pode-se usar outros tipos.

A sintaxe de animation padrão é, nesta ordem:

```
animation: name duration timing-function delay iteration-count direction  
fill-mode play-state;
```

Caso deseje ver mais sobre animation, utilize a w3schools:

https://www.w3schools.com/cssref/css3_pr_animation.asp

Keyframes

O termo “keyframe” é bem conhecido, ele é utilizado em outras aplicações que trabalham com animações, indicando ser uma “linha do tempo”. A “@keyframes” do CSS possui atributos que podem ser adicionados para criar uma boa animação.

Nas animações, definimos um ciclo, que pode ter vários pontos entre o início e o fim após ser definido. As transições do ciclo podem ser alteradas para tornar a animação melhor, através dos “eases”. No exemplo mostrado anteriormente, utilizamos apenas um início e um fim:

```
@keyframes DivAnimada {  
  from {  
    left: 0px;  
  }  
  to {  
    left: 200px;  
  }  
}
```

O diagrama mostra o código CSS para keyframes com anotações em português explicando cada parte:

- Início**: aponta para a palavra-chave `from`.
- Atributo inicial**: aponta para o valor `left: 0px;` dentro do bloco `from`.
- Fim**: aponta para a palavra-chave `to`.
- Atributo final**: aponta para o valor `left: 200px;` dentro do bloco `to`.

Explicando o comando acima:

- **@keyframes:** Adicionado “@keyframes” em CSS, ou em style dentro do html; Aqui criamos o nome da animação, neste caso, “DivAnimada”.
- **Início:** É o estado inicial da animação.
- **Atributo inicial:** Tanto a posição quanto os atributos podem ser definidos aqui.
- **Fim:** É o estado final da animação.
- **Atributo final:** são os valores finais da animação.

A sintaxe de “@keyframes” pode variar bastante, inicialmente podemos usar porcentagem entre 0 e 100% para ter controle de variações:

```
@keyframes example {  
  0%    {background-color: red;}  
  27%   {background-color: blue;}  
  60%   {background-color: yellow;}  
  100%  {background-color: green;}  
}
```

Caso deseje ver mais sobre @keyframes, utilize a w3schools:

https://www.w3schools.com/cssref/css3_pr_animation.asp

Easings

As “easing” são formas de mudarmos o tempo da transição do “animation” buscando tornar a animação melhor. É um refinamento que ajuda a controlar o tempo de execução, acelerando no começo e diminuindo no final, ou lenta apenas no meio, etc. Se utiliza a propriedade “animation-timing-function”, que pode ser alterada de duas formas:

1. Com padrões pré-determinados do CSS; ou
2. Utilizando o “cubic-bezier()” para passar os valores.

Padrões pré-determinados

São 5 opções que seguem estes gráficos:



Figura 4 - Padrões do “animation-timing-function”

Visualizando eles e comparando com o padrão “linear”:

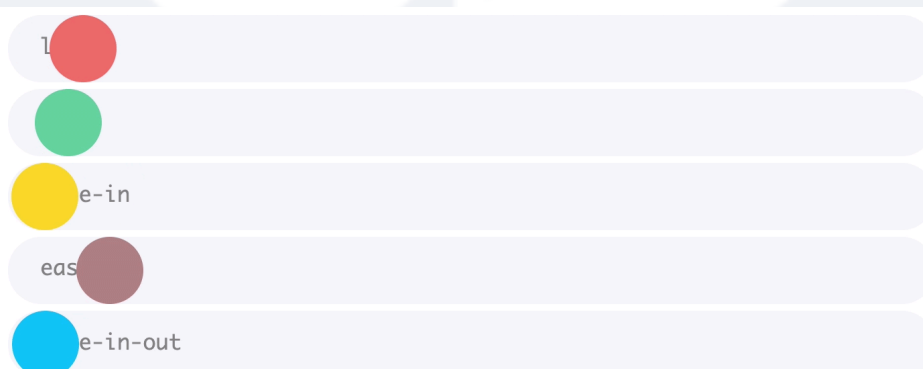


Figura 5 - Padrões do “animation-timing-function” em comparação - desenvolvimentoparaweb.com

Cubic Bezier

Utilizando o “cubic-bezier” podemos alterar manualmente estes gráficos, onde eles são transformados em valores separados por vírgulas. Exemplo:

```
animation: fade-out 1s cubic-bezier(.51,1.93,.66,-1.61);
```

Figura 5 - Usando o "cubic-bezier" no "animation-timing-function"

A sintaxe padrão segue: `cubic-bezier(x1, y1, x2, y2)`. Mas, para facilitar, geralmente se procuram sites para estas alterações que geram o código, como o cubic-bezier.com.

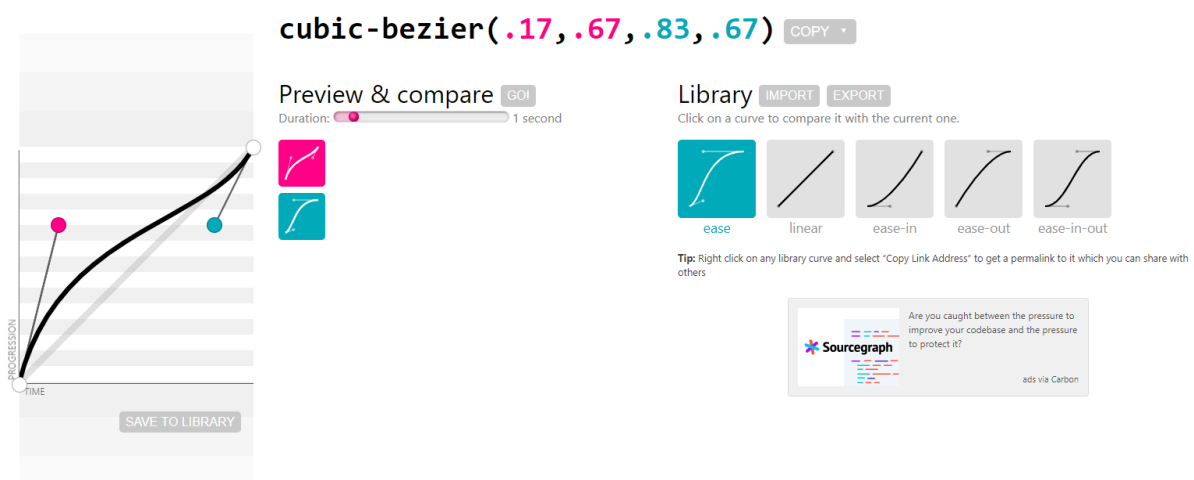


Figura 6 - Site cubic-bezier.com

Caso deseje ver mais sobre eases, utilize a w3schools:

https://www.w3schools.com/cssref/css3_pr_animation-timing-function.asp

Por fim, lembre-se que as animações tem por função ajudar a tornar mais confortável a experiência do usuário, em conjunto com cores, tipografia, formas e todo o design. É um ótimo recurso, aproveite!

Resumo

As animações em CSS são uma grande possibilidade de transformar a interação e a experiência em sites e aplicações melhor. Com o “animation” pode-se criar uma infinidade de animações e transições. Seja mudando a transição em que aparecem os itens da página, personalizando o tempo, ou criando reação para ações, elas acabam sendo muito úteis para a melhor experiência do usuário.

REFERÊNCIAS

1. [Usando animações CSS](#)
2. [CSS Animations](#)
3. [CSS animation-timing-function Property](#)
4. [CSS3 – Animation e regra keyframe - Tableless](#)
5. [animation | CSS-Tricks](#)
6. [CSS3 Animation – Tutorial para iniciantes](#)
7. [CSS: animações com Transition e Animation | Alura](#)

BONS ESTUDOS

