



Prof. Válter Brandenburger Vasconcellos
sorvalter@gmail.com

Filtros

- **blur(px)** - nível de ofuscamento
- **brightness(%)** - ajusta o brilho
- **contrast(%)** - ajusta o contraste
- **drop-shadow(h, v, blur, color)** - aplica sombras
- **grayscale(%)** - transforma as cores do elemento para uma escala cinza
- **hue-rotate(deg)** - altera as cores do elemento
- **invert(%)** - inverte as cores do elemento
- **opacity(%)** - controla a opacidade do elemento
- **saturate(%)** - altera a saturação do elemento
- **sepia(%)** - transforma as cores do elemento para um tom sépia



without filter



filter: `blur(4px);`



filter: `hue-rotate(50deg);`



filter: `brightness(150%);`



filter: `invert(100%);`



filter: `grayscale(100%);`



filter: `saturate(150%);`



filter: `sepia(70%);`

```
/* <filter-function> values */
```

```
filter: blur(5px);
```

```
filter: brightness(0.4);
```

```
filter: contrast(200%);
```

```
filter: drop-shadow(16px 16px 20px blue);
```

```
filter: grayscale(50%);
```

```
filter: hue-rotate(90deg);
```

```
filter: invert(75%);
```

```
filter: opacity(25%);
```

```
filter: saturate(30%);
```

```
filter: sepia(60%);
```

```
/* Multiple filters */
```

```
filter: contrast(175%) brightness(3%);
```

```
filter: drop-shadow(3px 3px red) sepia(100%) drop-shadow(-3px -3px blue);
```

```
/* Use no filter */
```

```
filter: none;
```

<https://maujor.com/tutorial/css3-filters.php>

Gradientes

Gradientes

Com já foi visto na UC14, gradiente ou degradê é uma combinação de duas ou mais cores em transições:



Gradientes

Podem ser inseridas no fundo da página ou de um elemento por exemplo.

Antes de existir esta possibilidade com o CSS3, eram feitas com imagens no Photoshop (mais trabalhoso e pesado).

Gradientes Lineares

background-image {

linear-gradient (cor1, cor2)

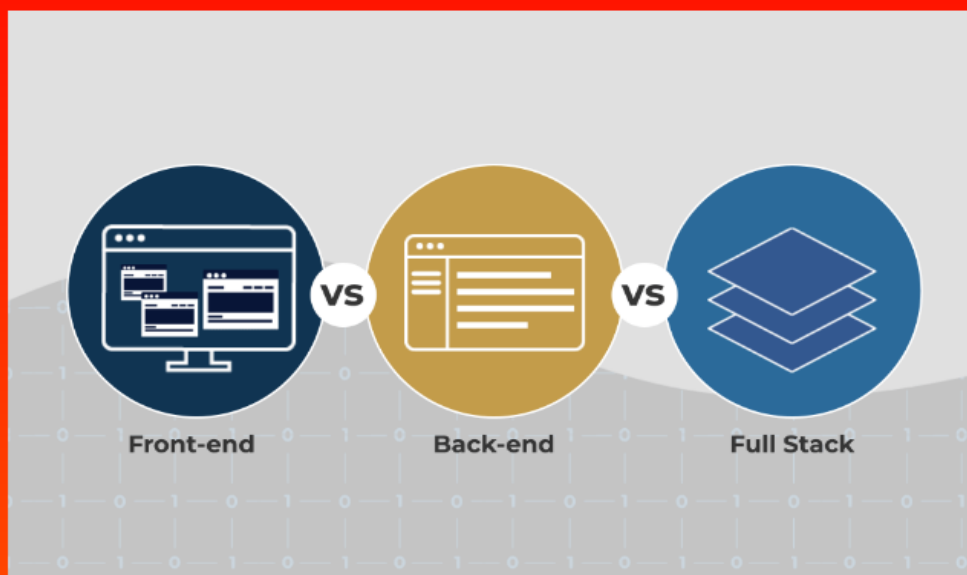
}

Gradientes Lineares

```
body{  
    background-image:linear-gradient(red,yellow) ;  
}
```

Back-end vs Front-end vs Fullstack: Escolha o seu futuro como programador!

Front-end



O front-end é todo o código da aplicação responsável pela apresentação do software (client-side).

Em se tratando de aplicações web, é exatamente o código do sistema que roda no navegador.


Um desenvolvedor front-end, geralmente, trabalha com linguagens como HTML, CSS e JavaScript, além de frameworks e bibliotecas, como por exemplo Angular, React, Vue.js, etc.

Existem diversas oportunidades no mercado para desenvolvedores especialistas em front-end. Nestes casos, esses programadores não conhecem nada ou conhecem muito pouco de back-end.

São pessoas que preferiram se especializar no front da aplicação.

Podemos definir direção do gradiente em relação a primeira cor escolhida:

```
body{  
    background-image:linear-gradient(to right,red,yellow) ;  
}
```



to right: para direita

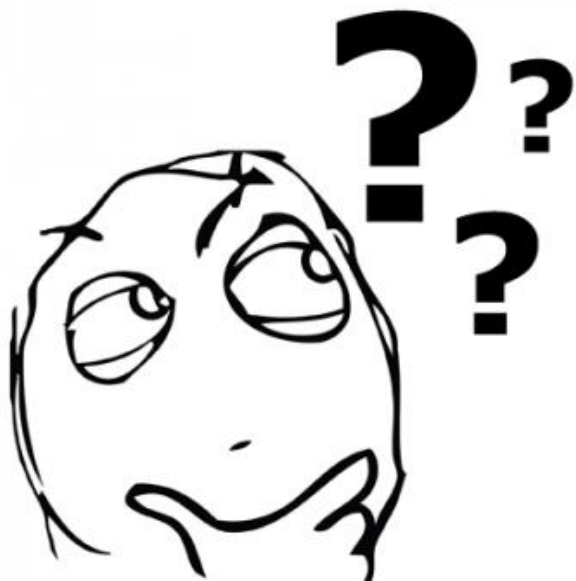
to left: para esquerda

to top: para cima

to bottom: para baixo

Gradientes Lineares

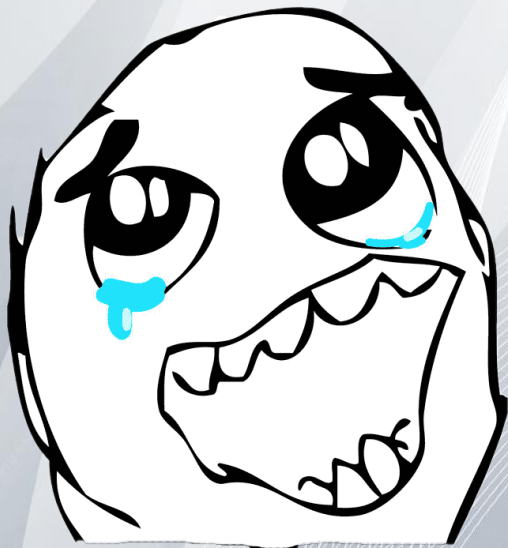
E tem como angular a direção?



Gradientes Lineares

Sim!!!

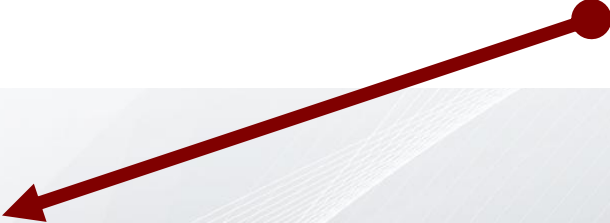
**Usando inclusive quantos graus se
deseja rotacionar!**



Podemos definir direção do gradiente em relação a primeira cor escolhida:

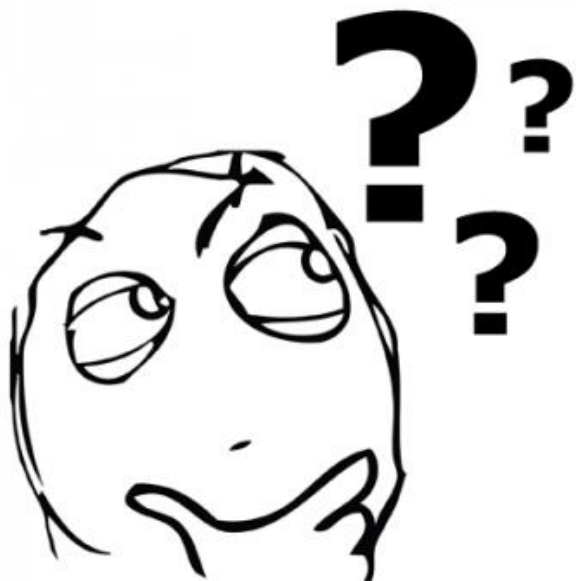
```
body{  
    background-image: linear-gradient(45deg, red, yellow) ;  
}
```

Valor em graus usando
deg (*degrees*)



Gradientes Lineares

**E para gradientes com mais
de duas cores?**



Gradientes Lineares

```
body{  
    background-image:linear-gradient(green,red,yellow) ;  
}
```



Gradientes Lineares

Todas as opções de posicionamento vistas com duas cores valem para três ou mais cores.

Podemos também usar transparências nos gradientes, usando o RGBA:

```
body{  
  background-image: linear-gradient(to top, rgba(255,0,0,0),  
    rgba(255,0,0,1));  
}
```


Gradientes Radiais

Criar gradientes radiais é tão fácil quanto lineares.

Basta trocar a propriedade *linear-gradient* por ***radial-gradient***.

Gradientes Radiais

```
body{  
    background-image: radial-gradient(red, yellow);  
}
```

Gradientes Radiais

```
body{  
    background-image: radial-gradient(green, red, yellow);  
}
```

Gradientes Radiais

```
body{  
    background-image: radial-gradient(green 10%, red 30%,  
    yellow 60%);  
}
```

Colocar % após a cor
determina a
quantidade/distribuição
desta no gradiente radial

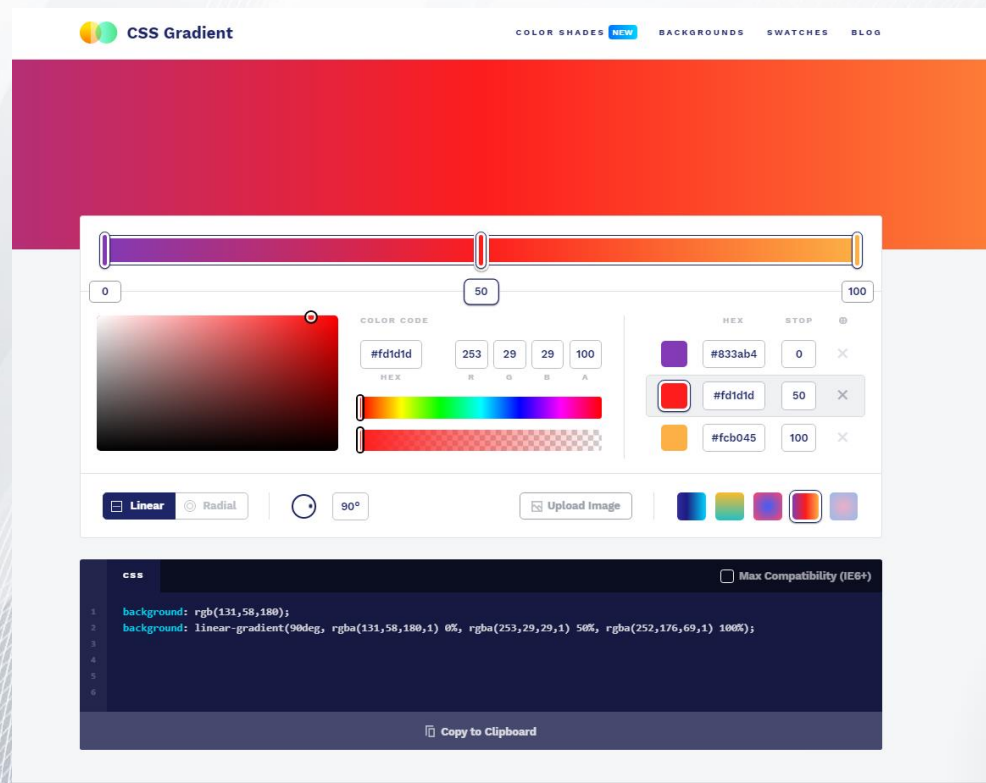


Gradientes Radiais

```
body{  
  background-image: radial-gradient(circle, green 10%,  
    red 30%, yellow 60%);  
}
```


Gradientes

<https://cssgradient.io>



Gradientes

<https://www.colorzilla.com/gradient-editor/>

Ultimate CSS Gradient Generator

A powerful Photoshop-like CSS gradient editor from [ColorZilla](https://www.colorzilla.com/).

FOR FIREFOX FOR CHROME GRADIENT GENERATOR

Presets



Name:



 hue/saturation  reverse  import css  import image

Stops

Opacity:

Location:

Color:

Location:

Sponsor

Preview




Orientation: Size: x ☐ IE

CSS

[switch to scss](#)

```
/* Permalink - use to edit and share this gradient:
https://colorzilla.com/gradient-editor/#1e5799+0,2989d8+50,207cca+51,7db9e8+100;Blue+Gloss
*/
background: #1e5799; /* Old browsers */
background: -moz-linear-gradient(top, #1e5799 0%, #2989d8 50%, #207cca 51%, #7db9e8 100%); /* FF3.6-15 */
background: -webkit-linear-gradient(top, #1e5799 0%, #2989d8 50%, #207cca 51%, #7db9e8 100%); /* Chrome10-25,Safari5.1-6 */
background: linear-gradient(to bottom, #1e5799 0%, #2989d8 50%, #207cca 51%, #7db9e8 100%); /* W3C, IE10+, FF16+, Chrome26+, Opera12+, Safari7+ */
filter: progid:DXImageTransform.Microsoft.gradient(
startColorstr='#1e5799',
endColorstr='#7db9e8',GradientType=0 ); /* IE6-9 */
```

Color format: ☒ Comments

 [Link to, save or share the current gradient using its permalink.](#)

Animações

Animações

Fazer animações para web a alguns anos era utilizar o Adobe Flash (atualmente descontinuado) ou Javascript (o que ainda é uma ótima opção)

Animações

Mais uma opção surgiu com o CSS3, a propriedade **animation**.

Com ela é possível criar animações fáceis e com resultados muito bons, sem a complexidade do Javascript.

Animações

A base de toda animação são os ***keyframes***, ou seja, **quadros-chave**.

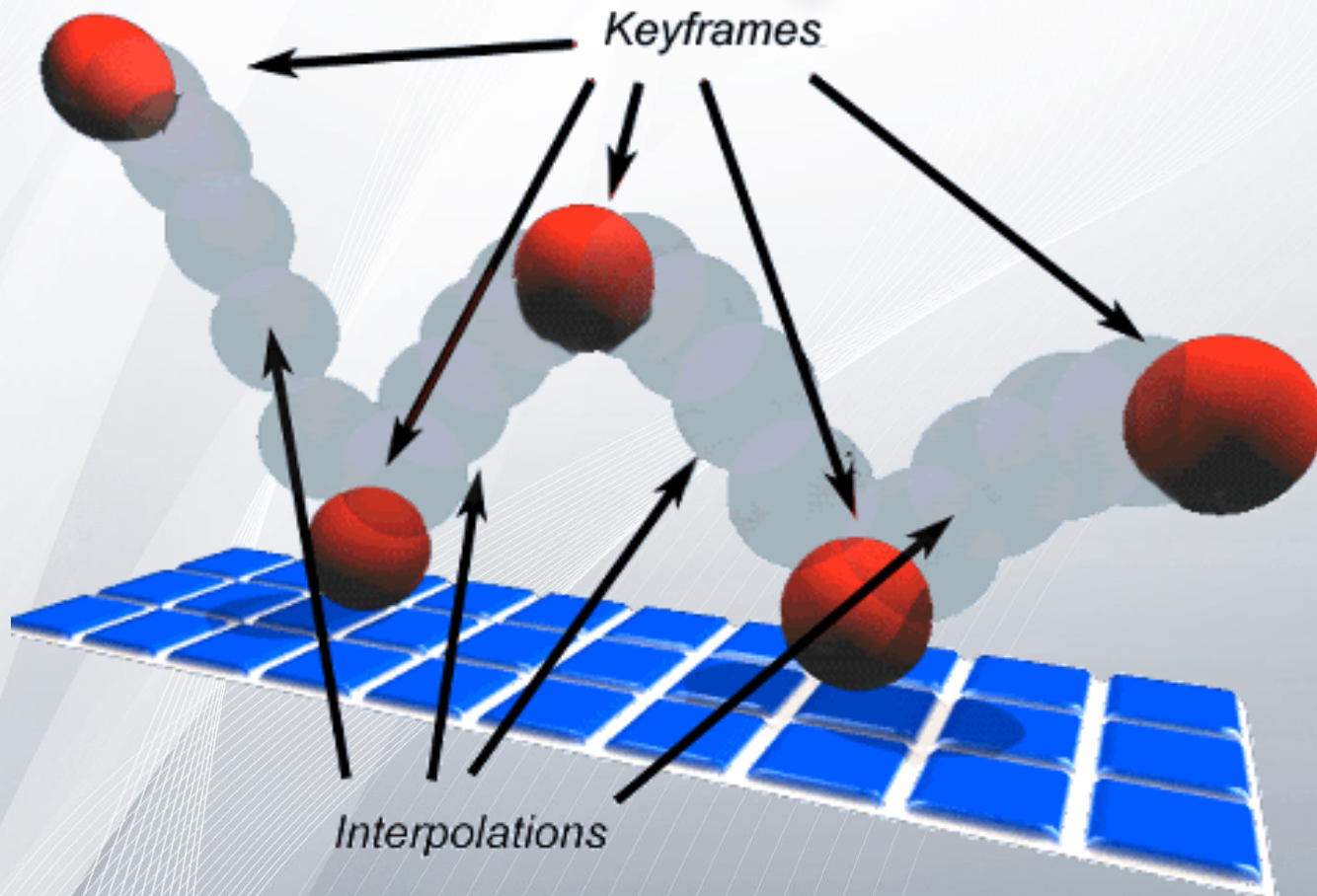
Eles definem os pontos que serão mudados algo, os pontos de alteração. A partir destes pontos é criado as transições de um estado a outro automaticamente.

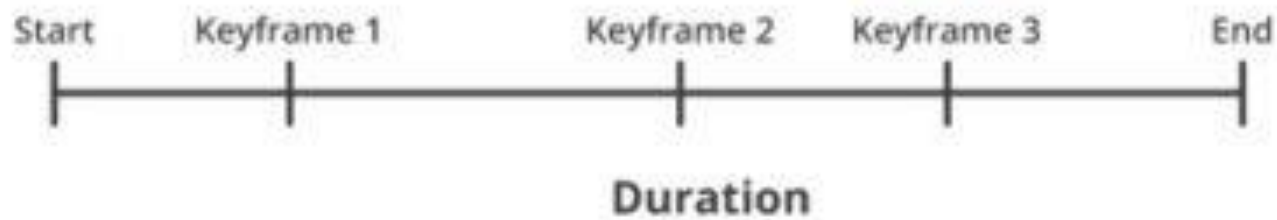
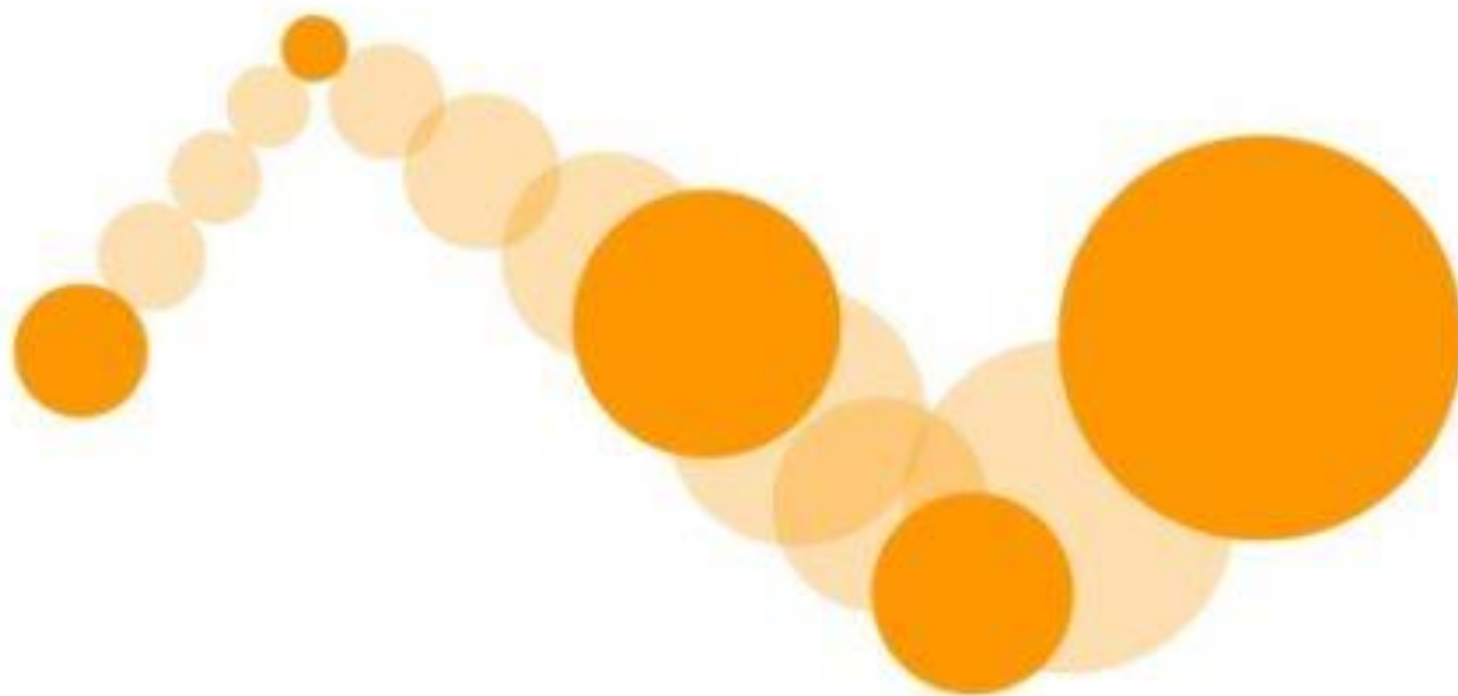


Fecomércio RS

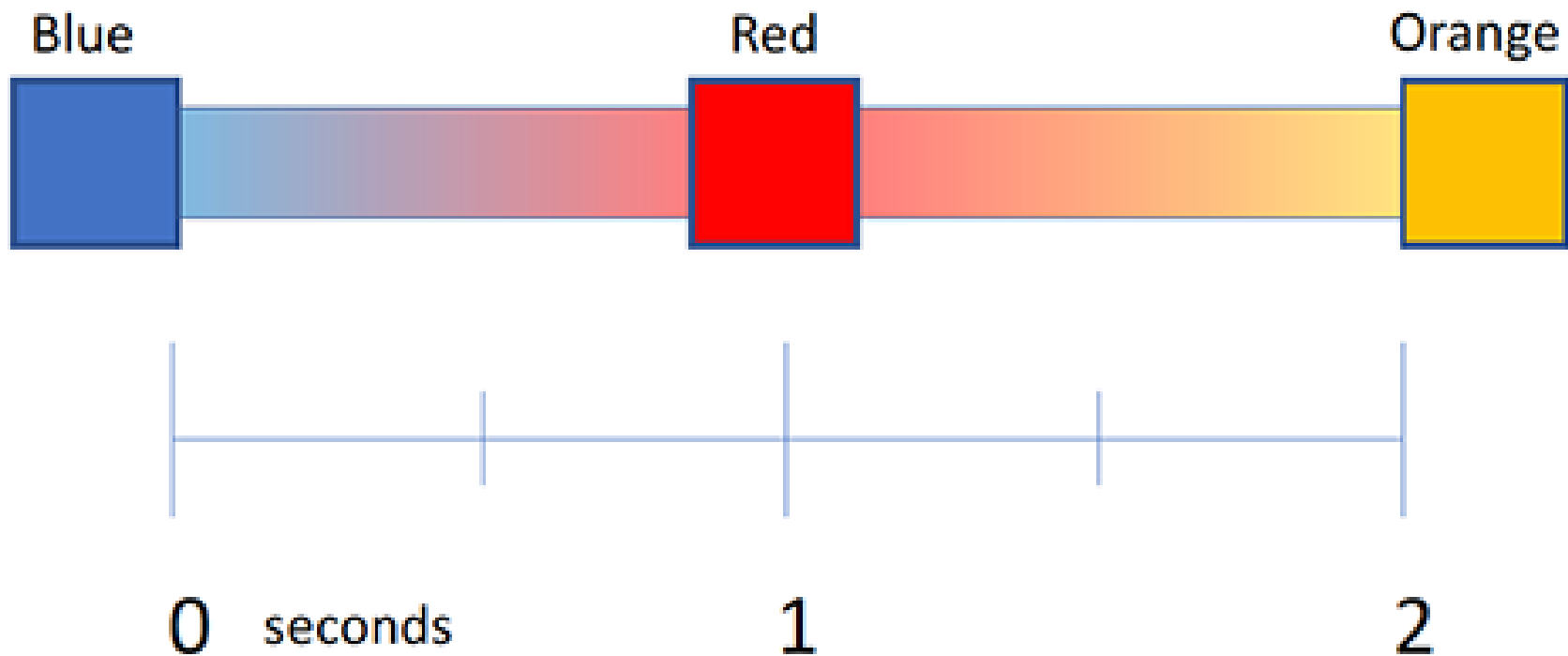


Animações





Animações



Animações

Primeiro vamos criar a diretiva **@keyframes** e nomear com o nome que desejarmos.

```
<style>
  p{
    font-size:20px;
  }

  h1, h2{
    color:#2b6a9a;
  }

  h2{
    font-size:28px;
  }

  @keyframes fadeIn{

  }

</style>
```

Animações

Dentro desta diretiva vamos criar os pontos de mudança, por exemplo:

```
@keyframes fadeIn{  
  0% { }  
  100% { }  
}
```

Animações

Dentro das chaves destes pontos colocamos as propriedades CSS que queremos animar e mudar. Exemplo:

```
@keyframes fadeIn{  
  0%{opacity:0}  
  100%{opacity:1}  
}
```

Animações

Agora basta “chamar” esta diretiva com a propriedade **animation** no elemento que desejamos animar e o tempo total desta animação em segundos:

```
h1, h2{  
    color:#2b6a9a;  
    animation: fadeIn 3s;  
}
```


Animações

Outras possibilidades em animações são os ***easings***

Com eles podemos deixar as animações mais dinâmicas e naturais, implementando efeitos.

Os *easings* fazem parte do que chamamos de ***timing functions***, e com isto podemos criar os efeitos de ***bounce*** (“quicar” elementos)


```
@keyframes bounceIn{  
  
  0%{  
    transform:scale(0.1);  
    opacity:0  
  }  
  
  60%{  
    transform:scale(1.2);  
    opacity:1  
  }  
  
  100%{  
    transform:scale(1);  
  }  
}
```

Animações

```
img{  
  animation: bounceIn 1s;  
}
```

Observação:

```
img{  
  animation: bounceIn 1s;  
}
```



Aqui usamos uma *shorthand* (forma abreviada) de aplicar a propriedade *animation*. Existem outras propriedades para ela.



Fecomércio RS




Observação:

```
img{  
  animation-name: bounceIn;  
  animation-duration: 1s;  
}
```

*Forma não abreviada do mesmo
código anterior*

Outras propriedade para o animation:

```
img{  
  animation-name: bounceIn;  
  animation-duration: 1s;  
  animation-timing-function: ease-in;  
}
```



Controla a distribuição de velocidade da animação

animation-timing-function:

ease-in: animação começa devagar;

ease-out: animação termina devagar;

ease-in-out: animação começa e termina devagar;

linear: padrão. Velocidade igual do início ao fim da animação.

CSS transition timing functions

linear

ease-in

ease

ease-in-out

ease-out

cubic-bezier

FIRE ALL!



Fecomércio RS



Shorthands

```
img{  
  animation: bounceIn 1s ease-in  
  infinite alternate;  
}
```

repete a
animação sempre
(loop)

repete a
animação do
início ao fim e do
fim ao início
alternadamente