

*Desbravando e  
criando websites  
reais*

O desenvolvimento *Web* moderno  
que você precisa saber!



**PUCPR**  

---

**GRUPO MARISTA**



# Vinícius Godoy de Mendonça

---

- Na área de informática desde **1996**
- Atuou na **Siemens, Positivo Informática e Sinax**
- Mestre em **computação gráfica** pela PUCPR
- Outros cursos em que dou aula:
  - TDJ, BSI, BCC, BES
  - Apple Developer Academy
  - Especialização em AppDev, Vants e Drones, Tecn. Agronegócio
- Livro **Learning OpenCV** da Packt Publishing
- Moderador do **GUJ**, o maior fórum de Java do Brasil

# Rodrigo Fujioka

---

- Na área de informática desde 2004
- Atuou no PagSeguro, Anymarket, Senado Federal, Ministério da Justiça,
- Doutor em computação pelo Centro de Informática da UFPE
- Especialista em Gestão de Projetos pela FGV.
- Certificado SCJP, Azuze, Safe5, Kanban KMPI e Mamanagement3.0
- Autor do livro: Arquitetura de Referência para Gerenciamento de Ambientes Virtuais 3D
- Geek e Pai de meninas





# Seja bem-vindo!

Bem vindo ao curso “***Desbravando e criando websites reais***”. Neste encontro, o que iremos aprender:














- ***Como reagir a passagem do mouse***
- ***Como criar diversos tipos de animações***

# Este curso

Este curso é uma iniciativa da PUCPR para que você aprenda um pouco mais sobre a área de informática

- Alta empregabilidade
- Bons salários
- Possibilidade de trabalhar com tecnologia de ponta



CURSO	MODALIDADE	TIPO DE GRADUAÇÃO
✓ Análise e Desenvolvimento de Sistemas	 EAD	Tecnólogo
✓ Big Data e Inteligência Analítica	 EAD	Tecnólogo
✓ CiberSegurança <b>NEW</b>	 PRESENCIAL	Bacharelado
✓ Ciência da Computação	 PRESENCIAL	Bacharelado
✓ Ciência de Dados	 EAD	Tecnólogo
✓ Engenharia de Software	  HÍBRIDO PRESENCIAL	Bacharelado
✓ Gestão da Tecnologia da Informação	 EAD	Tecnólogo
✓ Inteligência Artificial Aplicada	 HÍBRIDO	Tecnólogo
✓ Jogos Digitais	<b>NEW</b>   Virtual/Ao Vivo PRESENCIAL	Tecnólogo Bacharelado <b>NEW</b>
✓ Segurança da Informação	 PRESENCIAL	Tecnólogo
✓ Sistemas de Informação	 PRESENCIAL	Bacharelado



QUER SABER +?

# HTML de exemplo

HTML

```
<h1>Animação</h1>
<p>
  <span class="nome">Seu nome</span>
</p>
```

CSS

```
.nome {
  display: block;
  background-color: red;
  color: white;
  border: 3px solid black;
  font-weight: bold;
  font-size: xx-large;
  vertical-align: middle;
  text-align: center;
  padding: 10px;
  margin: 10px;
  width: 200px;
}
```

# Propriedades legais

## Fundo em gradiente:

- background-image: linear-gradient(*direction*, *cor1*, *cor2*, *cor3*);
- A direção pode ser to right, to bottom ou um ângulo

## Sombras

- text-shadow: distX distY cor
- box-shadow: distx disty cor



Até o HTML 5, animações só poderiam ser feitas através de **programação**.

Agora, podemos utilizar o **CSS** para isso!

# Reagindo ao mouse

Praticamente todos os elementos css **pseudo-classes**. Elas são utilizadas para definir **estilo** em um elemento em **situações especiais**. Por exemplo:

- Quando o mouse **passar por cima** (hover), ou **clicar** (active)
- Para diferenciar o estilo de **links visitados** (visited) de **não visitados** (unvisited)
- Para destacar somente o **primeiro ou último elementos** de um grupo, etc (first-child, last-child).

Experimente adicionar  
o seguinte css  
ao seu arquivo

CSS

```
.nome:hover {  
    background-color: green;  
}  
  
.nome:active {  
    background-color: yellow;  
}
```

# Controlando as transições

Não precisamos que as transições ocorram imediatamente

- `transition-property`: Define a qual propriedade aplicaremos a transição
- `transition-duration`: tempo que levará a transição
- `transition-delay`: tempo que o css aguardará antes da transição ocorrer
- `transition-time-function`: função que descreve o comportamento da transição: linear (padrão), ease, ease-in, ease-out, ease-in-out e steps

Os tempos são especificados em **segundos**.

# Transições suaves

Vamos modificar o css  
do hover

```
.nome:hover {  
  background-color: green;  
  transition-duration: 1s;  
}
```

# Atividade

Altere a página final da aula passada para ter elementos animados. Quem não tiver sua própria página, pode utilizar a feita por nós:

[https://github.com/rodrigofujioka/puc-cursos-livres/blob/master/desbravando-criacao-sites/modulo\\_css\\_basico/internal\\_layout.html](https://github.com/rodrigofujioka/puc-cursos-livres/blob/master/desbravando-criacao-sites/modulo_css_basico/internal_layout.html)

Use as propriedades e transições que você aprendeu até aqui



# Transformando objetos

- A propriedade transform permite transformar objetos de maneira mais radical:
  - **scale**: Altera o tamanho do elemento. `scale(2)` dobra o tamanho. Pode ser aplicado separadamente com `scaleX` e `scaleY`
  - **skew**: Inclina o objeto para a direita em um paralelogramo. Também pode ser aplicado separadamente. Ex: `skew(25deg)`
  - **translate**: Move o elemento em x e y. Ex: `translate(100,50);`
  - **rotate**: Rotaciona um elemento para direita. Ex: `rotate(50deg)`. Também podemos usar `rotateX`, `rotateY` e `rotateZ`

Você **pode** aplicar várias transformações ao mesmo tempo com espaço

# Transformando objetos

## Que tal modificar o css do active?

```
.nome:active {  
  color: black;  
  background-color: yellow;  
  transform: rotate(180deg) scale(1.5);  
}
```

# Atividade

Você conseguiria desenhar a Terra e a Lua girando em torno da terra?

Experimente trocar a ordem das transformações até chegar nesse efeito! 😊

# Sequências de animação

- A tag **animation** permite definir uma animação complexa, formada por vários passos diferentes. Ela possui várias sub-propriedades:
  - **animation-delay**: Tempo para iniciar a animação
  - **animation-direction**: Define se a animação deve voltar ao ponto inicial, ou executar de trás para frente quando terminar (normal, reverse, alternate e alternate-reverse)
  - **animation-iteration-count**: Configura o número de repetições. Você pode usar infinite para repetir sempre.
  - **animation-timing-function**: Função do tempo da animação
  - **animation-name**: Define o **nome da regra** da animação

# Regras de animação

- As regras de animação são criadas através do comando @keyframes
- Você deve inserir um keyframe a cada percentual de tempo de execução, de 0 até 100%. Em cada keyframe, pode haver uma transição diferente. Os nomes from e to também podem significar 0 e 100%.

Dica o comando animation pode ser usado para simplificar:

animation nome duração função delay count direction

Vamos montar um exemplo em que o nome anda um pouco, gira, anda mais um pouco e depois volta?

```
.nome {  
  display: block;  
  background-color: red;  
  color: white;  
  border: 3px solid black;  
  font-weight: bold;  
  font-size: xx-large;  
  vertical-align: middle;  
  text-align: center;  
  padding: 10px;  
  margin: 10px;  
  width: 200px;  
  animation-duration: 2s;  
  animation-iteration-count: infinite;  
  animation-name: nomeAndando;  
  animation-direction: alternate;  
}
```

```
@keyframes nomeAndando {  
  from {  
    transform: translate(0px,0px);  
  }  
  
  25% {  
    transform: translate(500px,0px);  
  }  
  
  75% {  
    transform: translate(500px,0px) rotate(360deg);  
  }  
  
  to {  
    transform: translate(1000px, 0px) rotate(360deg);  
  }  
}
```



# Atividade

- Utilizando os comandos de animação, delay e imagens, monte um pequeno desenho animado.
- Em seguida, compartilhe com os seus colegas

Esperamos  
que tenha  
gostado!



# Referências

- [https://developer.mozilla.org/pt-BR/docs/Web/CSS/CSS Transitions/Using CSS transitions](https://developer.mozilla.org/pt-BR/docs/Web/CSS/CSS_Transitions/Using_CSS_transitions)
- <https://css-tricks.com/almanac/properties/a/animation/>
- <https://css-tricks.com/almanac/properties/t/transform/>