



IN

INFINITY SCHOOL

VISUAL ART CREATIVE CENTER

AULA 07 - APROFUNDANDO EM CSS AVANÇADO

O QUE IREMOS APRENDER

01

REVISÃO DA AULA ANTERIOR

02

TRANSIÇÕES E ANIMAÇÕES

03

CSS VARIABLES

Revisão da Aula Anterior

INTRODUÇÃO AO GRID LAYOUT

PSEUDO ELEMENTOS

MEDIA QUERIES



TRANSIÇÕES E ANIMAÇÕES



Animações e transições no CSS criam efeitos visuais dinâmicos e interativos.

Transições: Suavizam mudanças de propriedades CSS, como a cor de um botão ao passar o mouse. São configuradas com a propriedade a ser alterada, duração e função de temporização.

Animações: Permitem sequências de movimentos e transformações com `@keyframes`. Podem ser configuradas para repetir, reverter e pausar, oferecendo controle detalhado sobre os elementos.

TRANSIÇÕES

As transições no CSS suavizam as mudanças de propriedades dos elementos, criando efeitos visuais agradáveis.

Com a propriedade **transition**, é possível definir a duração, o tipo de interpolação e o atraso para a mudança de uma ou mais propriedades CSS.

Elas tornam a experiência do usuário mais fluida, aplicando efeitos suaves em ações como passar o mouse sobre um botão ou mudar a cor de um link.



```
elemento {  
    transition: propriedade duração função-temporização;  
}
```

EXEMPLO DE TRANSIÇÃO

button: Define os estilos iniciais para o elemento <button>.

- **background-color: blue;** Define a cor de fundo inicial do botão como azul.
- **transition: background-color 0.5s ease;** Define uma transição suave de 0.5 segundos para a mudança de cor de fundo, com a função de temporização "ease" (início e fim suaves, meio rápido).
- **button: hover:** Define os estilos para o botão no estado de hover.
- **background-color: green;** Altera a cor de fundo do botão para verde quando o cursor do mouse está sobre ele.

Resultado: A cor de fundo do botão muda suavemente de azul para verde em 0.5 segundos ao passar o mouse sobre ele.

A code editor window with a dark background and light blue border. At the top left, there are three colored circles: red, yellow, and green. Below them is CSS code for a button transition. The code is as follows:

```
1  button {
2      background-color: blue;
3      transition: background-color 0.5s ease;
4  }
5
6  button:hover {
7      background-color: green;
8  }
```

ATIVIDADE PRÁTICA

Atividade 01

- Crie uma página HTML com um botão e uma caixa de texto (div).
- Aplique uma transição suave de 0.5 segundos para a cor de fundo do botão ao passar o mouse sobre ele.
- Aplique uma transição para aumentar o tamanho da caixa de texto ao passar o mouse sobre ela.

Objetivo:

Praticar o uso de transições em CSS para criar efeitos suaves ao interagir com elementos.

ANIMAÇÕES



```
elemento {  
    animation: nome-da-animacao duração função-temporização;  
}
```

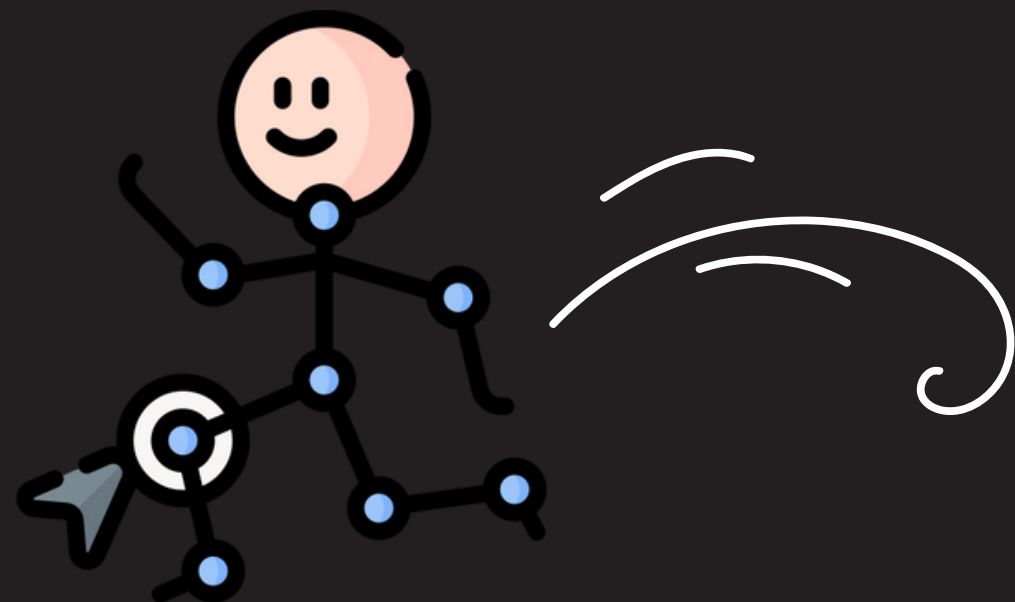
As animações em CSS criam efeitos visuais dinâmicos e interativos, melhorando a experiência do usuário.

Com a propriedade `animation` e a regra `@keyframes`, é possível definir como e quando um elemento se transforma ao longo do tempo.

Animações podem mover elementos, alterar cores, mudar tamanhos e muito mais, tornando a interface mais atraente e envolvente.

PROPRIEDADES COMUNS DE ANIMAÇÃO

- **animation-name:** Nome da animação definida com @keyframes.
- **animation-duration:** Duração da animação.
- **animation-timing-function:** Função de temporização (ex.: ease, linear).
- **animation-delay:** Atraso antes do início da animação.
- **animation-iteration-count:** Número de repetições da animação (ex.: infinite).
- **animation-direction:** Direção da animação (ex.: normal, reverse, alternate).



SINTAXE DE ANIMAÇÃO

```
1 @keyframes nome-da-animacao {
2   0% { propriedade: valor; }
3   100% { propriedade: valor; }
4 }
5
6 elemento {
7   animation: nome-da-animacao duração função-temporização;
8 }
```

@keyframes nome-da-animacao: Define a animação com o nome nome-da-animacao.

0% { propriedade: valor; }: Estado inicial da animação.

100% { propriedade: valor; }: Estado final da animação.

elemento: Seleciona o elemento que receberá a animação.

animation: Aplica a animação ao elemento.

nome-da-animacao: Nome da animação.

duração: Tempo de execução da animação (ex.: 2s).

função-temporização: Controla a velocidade da animação (ex.: ease, linear).

EXEMPLO PRÁTICO

```
1 @keyframes mover {  
2   0% { transform: translateX(0); }  
3   100% { transform: translateX(100px); }  
4 }  
5  
6 .div-movimento {  
7   animation: mover 2s ease-in-out;  
8 }  
9
```

A animação `mover` desloca o elemento `div-movimento` 100px à direita em 2 segundos, com a função de temporização "`ease-in-out`", que começa e termina lentamente, acelerando no meio.

ATIVIDADE PRÁTICA

Atividade 02

- Crie uma página HTML com uma caixa (div).
- Use a propriedade `@keyframes` para definir uma animação que mova a caixa da esquerda para a direita.
- Aplique a animação à caixa usando a propriedade `animation`, fazendo com que a animação dure 2 segundos e repita indefinidamente.

Objetivo:

Praticar a criação de animações simples usando CSS para fazer elementos se moverem e mudarem de aparência.

ATIVIDADE PRÁTICA

Atividade 03

Crie uma página HTML e siga os passos abaixo:

- Adicione um botão com a classe btn.
- Defina uma animação para a div usando @keyframes que altere a posição, a opacidade e a cor de fundo da div em diferentes pontos da animação.
- A animação deve durar 4 segundos e deve repetir infinitamente.

Objetivo:

Praticar a criação de animações complexas com múltiplos keyframes.

CSS VARIABLES

CSS Variables são valores reutilizáveis definidos no código CSS que podem ser aplicados em várias partes do estilo.

Elas permitem definir um valor, como uma cor ou tamanho, e usá-lo em várias regras CSS.

Isso facilita a manutenção do design, pois ao alterar a variável, todas as instâncias são atualizadas automaticamente.

São especialmente úteis em projetos grandes para garantir uniformidade e facilitar mudanças.

CRIANDO VARIÁVEIS NO CSS

Sintaxe:



- Definição: `--nome-da-variavel: valor;`
- Uso: `var(--nome-da-variavel);`

O código CSS ao lado usa variáveis para definir e reutilizar valores de estilo.

Em `:root`, as variáveis `--main-color` e `--padding` são definidas com uma cor azul e um espaçamento de 10px.

A classe `.elemento` aplica essas variáveis, definindo a cor de fundo como azul e o espaçamento interno como 10px, facilitando a manutenção e a consistência do design.



```
1 :root {
2     --main-color: #3498db;
3     --padding: 10px;
4 }
5
6 .elemento {
7     background-color: var(--main-color);
8     padding: var(--padding);
9 }
```

APLICAÇÕES DAS VARIÁVEIS NO CSS

Cores Temáticas:

- Definir cores principais, secundárias, e de fundo para todo o site, facilitando a criação de temas e a alteração de esquemas de cores.

Tamanhos de Fonte:

- Controlar tamanhos de fonte para diferentes elementos, garantindo consistência tipográfica em todo o site.

Espaçamentos:

- Definir margens e espaçamentos padrão para ser reutilizados em várias partes do layout.

Bordas e Raios de Bordas:

- Controlar estilos de borda e raios de borda de forma consistente

Transições e Animações:

- Definir durações padrão para transições e animações.

ATIVIDADE PRÁTICA

Atividade 04

Crie uma página HTML e Siga os passos abaixo:

Defina Variáveis Globais:

- Crie variáveis CSS para armazenar as cores principais do tema, como a cor de fundo, a cor do texto e a cor de destaque.
- Defina essas variáveis no seletor :root para que possam ser usadas em todo o documento.

Aplique as Variáveis:

- Utilize as variáveis CSS para definir o fundo da página, a cor do texto de cabeçalhos e parágrafos, e a cor de destaque para links e botões.

Objetivo:

Praticar a criação de variáveis CSS para definir e aplicar um esquema de cores temático a uma página web.

ATIVIDADE PRÁTICA

Atividade 05

Crie uma página HTML e siga os passos abaixo:

Defina Variáveis para Tamanhos:

- Crie variáveis CSS para armazenar os tamanhos de fontes e espaçamentos em diferentes tamanhos de tela (pequena, média e grande).
- Defina essas variáveis no seletor `:root`.

Aplice as Variáveis:

- Utilize as variáveis CSS para definir o tamanho da fonte dos cabeçalhos e o espaçamento entre elementos.
- Modifique as variáveis em um `@media` query para ajustar o layout em diferentes tamanhos de tela.

Objetivo:

Utilizar variáveis CSS para definir tamanhos e espaçamentos responsivos em uma página web.

SE LIGA NO CONTEÚDO DA PRÓXIMA AULA!

AULA 08 - HTML & CSS
REVISÃO GERAL E PROJETO FINAL

IN

INFINITY SCHOOL

VISUAL ART CREATIVE CENTER