

**INSTITUTO  
FEDERAL**  
Santa Catarina

# Programação para Internet I



# Tópicos desta aula

- Propriedades CSS personalizadas
- Outras funções CSS

**CSS**



# 1/2

## Propriedades CSS personalizadas

**CSS**



# Especificações

- *Custom properties* (propriedades personalizadas) ainda está em “*W3C Candidate Recommendation Draft*” porém já é aceita por todos navegadores atualizados (menos IE, claro).
  - W3C Candidate Recommendation Draft, 11 November 2021
  - <https://www.w3.org/TR/css-variables-1/#defining-variables>
- Documentação e referência principal:
  - [https://developer.mozilla.org/pt-BR/docs/Web/CSS/Using\\_CSS\\_custom\\_properties](https://developer.mozilla.org/pt-BR/docs/Web/CSS/Using_CSS_custom_properties)

# Propriedades personalizadas

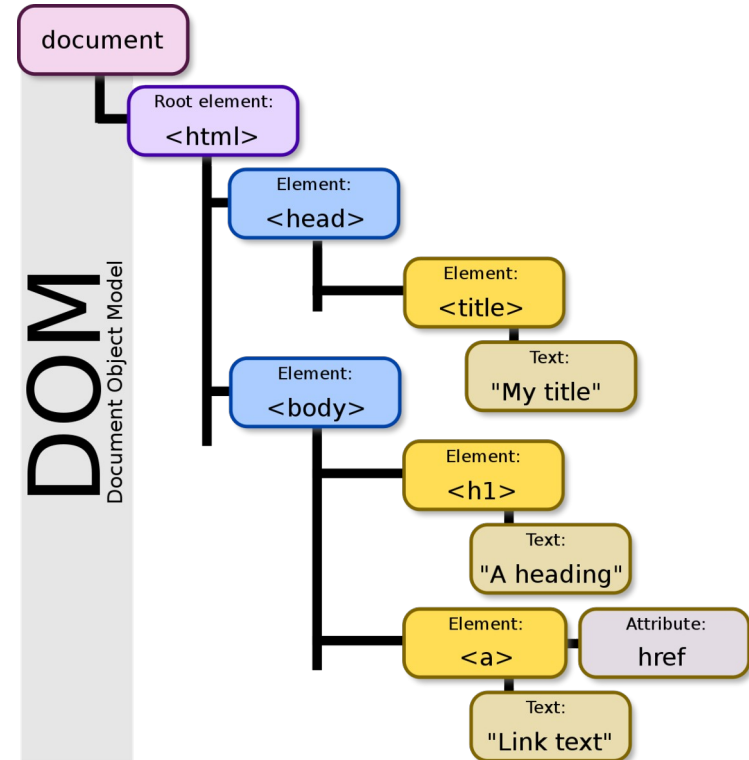
- Websites complexos tem uma quantidade muito grande de CSS, com uma quantidade de repetição de valores muito frequente.
- Por exemplo, a mesma cor pode ser usada em centenas de lugares diferentes, requerindo uma pesquisa global e substituição caso esta cor necessite ser trocada.
- Variáveis CSS permite um valor ser guardado em um lugar, então ser referenciado em muitos outros lugares.
- Um benefício adicional é a semântica dos identificadores. Por exemplo `--cor-principal-texto` é mais facil de ser entendido do que `#00ff00`

# Propriedades personalizadas

- A função `var()` é usada para inserir o valor de uma variável CSS.
- As variáveis CSS têm acesso ao DOM (Document Object Model), o que significa que você pode criar variáveis com escopo local ou global, alterar as variáveis com JavaScript e alterar as variáveis com base em consultas de mídia.
- Uma boa maneira de usar variáveis CSS é quando se trata das cores do seu design. Em vez de copiar e colar as mesmas cores repetidamente, você pode colocá-las em variáveis.

# O que é DOM?

- Quando uma página é carregada, o navegador cria um Document Object Model da página, que é uma árvore de objetos.
- Representa a página de forma que os programas possam alterar a estrutura do documento, alterar o estilo e conteúdo.
- O DOM representa o documento com nós e objetos, dessa forma, as linguagens de programação podem se conectar à página.
- O DOM é uma representação orientada a objetos da página da web.





# Utilizando propriedades personalizadas

```
elemento {  
  --cor-bg-principal: brown;  
  --cor-texto-principal: #D3D3D3;  
}
```

```
elemento {  
  background-color: var(--cor-bg-principal);  
  color: var(--cor-texto-principal);  
}
```



# Utilizando propriedades personalizadas

```
:root {  
  --main-bg-color: brown;  
}  
.classe1 {  
  color: white;  
  background-color: var(--main-bg-color);  
  margin: 10px;  
}  
.classe2 {  
  color: white;  
  background-color: var(--main-bg-color);  
  width: 100px;  
}
```

- Para utilizar em outros blocos, utilize a função **var()**.
- A variável pode ser declarada na pseudo-classe **:root**, que representa a raiz da árvore DOM.
- É o mesmo que o seletor html porém com mais especificidade.
- Propriedades personalizadas não funcionam como variáveis globais. Estão acessíveis somente ao elemento e seus descendentes diretos, como qualquer css.

# Valores de fallback em variáveis

```
:root {  
  --main-bg-color: brown;  
}  
  
.classe2 {  
  color: white;  
  background-color: var(--bg-color, black);  
  width: 100px;  
}
```

- A variável sempre deve começar com o prefixo --\*
- O segundo parâmetro de var() é um valor de fallback.
- Se a variável não for encontrada ou estiver inacessível, o segundo valor será utilizado.

# Treinando...



# Treinando

- A partir do exercício 3 (pizzaria), concentre no início do documento CSS todos valores de cores do site através de **propriedades personalizadas**, utilizando uma paleta limitada.
- Utilize essas propriedades no restante do documento através da função `var()`.
- Dicas de site de paletas de cores:
  - <https://www.toptal.com/designers/colourcode/quad-color-builder>
  - <https://colors.co/palettes/trending>
  - <https://paletton.com>

## 2/2

# Outras funções CSS

Fonte principal:  
[https://developer.mozilla.org/en-US/docs/Web/CSS/CSS\\_Functions](https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Functions)



# Funções CSS

- Como a maioria das outras linguagens, CSS possui funções.
- Já utilizamos algumas sem perceber.
- Veja o exemplo abaixo. `Url()` é uma função.

```
body {  
    background-image: url("fundo.png");  
}
```

# Funções CSS

- Principais categorias:
  - Funções de transformação
  - Funções matemáticas
  - Funções de filtros
  - Funções de cores
  - Funções de imagens
  - Funções de contagem
  - Funções de forma
  - Funções de referência
  - Funções de GRID
- Cada categoria pode ter dezenas de funções.
- Várias funções ainda são experimentais.
- Nesta aula veremos alguns exemplos.

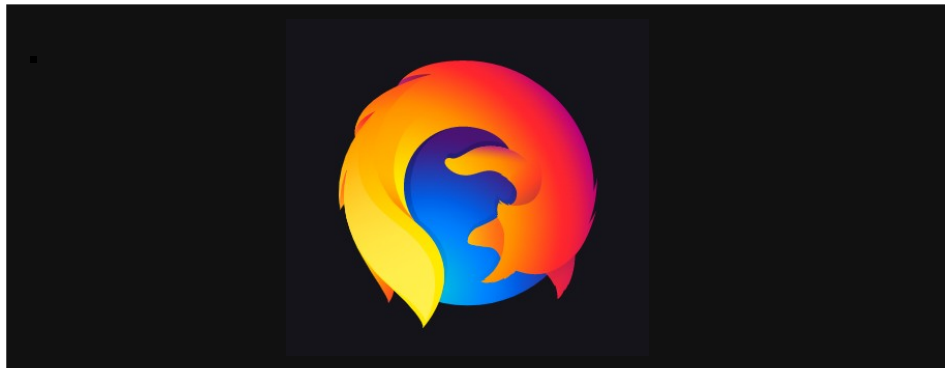




# Funções de transformação

# Funções de transformação

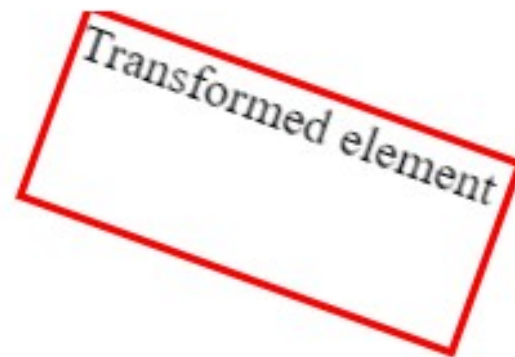
```
img{  
  transform: rotate(0.5turn);  
}
```



- Aplica-se a propriedade CSS **transform**.
- Representa uma transformação que afeta a aparência de um elemento.
- As funções de transformação podem girar, redimensionar, distorcer ou mover um elemento no espaço 2D ou 3D.

# Funções de transformação

```
div {  
  border: solid red;  
  transform: translate(30px, 20px) rotate(20deg);  
  width: 140px;  
  height: 60px;  
}
```



# Funções de transformação

- matrix()
- matrix3d()
- perspective()
- rotate()
- rotate3d()
- rotateX()
- rotateY()
- translate()
- translate3d()
- translateX()
- translateY()
- translateZ()
- rotateZ()
- scale()
- scale3d()
- scaleX()
- scaleY()
- scaleZ()
- skew()
- skewX()
- skewY()



# Funções de matemática

# Funções de matemática

```
#div1 {  
  background-color: yellow;  
  height: 100px;  
  width: max(50%, 300px);  
}
```

- Aplica-se a propriedades numéricas
- O exemplo ao lado define o tamanho da div para o maior valor encontrado: 50% ou 300px.
- Obviamente depende do tamanho da tela do dispositivo.
- Exemplo: Se a tela for maior que 600px, a div terá largura variável de 50%

# Funções de matemática

```
#div1 {  
  background-color: yellow;  
  height: 100px;  
  width: calc(100% - 80px);  
}
```

- Neste exemplo a div sempre terá uma largura fluída de 80% menos 80px;



# Funções de matemática

- `calc()`
- `clamp()`
- `max()`
- `min()`

# Funções de filtro

# Funções de filtros

- Os filtros são comumente usados para ajustar a renderização de uma imagem, um plano de fundo ou efeito gráfico.
- Pode ser utilizado nas propriedades **filter** e **backdrop-filter**
- Exemplo:

```
#myImg {  
  filter:blur(5px);  
}
```



# Funções de filtros

- Os filtros são comumente usados para ajustar a renderização de uma imagem, um plano de fundo ou uma borda.
- Exemplo:

```
#myImg {  
  filter: drop-shadow(8px 8px 10px gray);  
}
```



# Funções de filtros

- Os filtros são comumente usados para ajustar a renderização de uma imagem, um plano de fundo ou uma borda.
- Exemplo:

```
#myImg {  
  filter: grayscale(100%);  
}
```



# Funções de filtros

- É possível combinar filtros em uma declaração:

```
.do-more-things {  
  filter: blur(20px) grayscale(20%);  
}
```

# Outros Filtros

- blur()
- brightness()
- contrast()
- drop-shadow()
- grayscale()
- hue-rotate()
- invert()
- opacity()
- saturate()
- sepia()
- url() -para filtros SVG





# Funções de cores

# Funções de cores

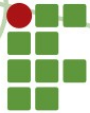
```
<style>
  div {
    background-color: rgba(255, 255, 255, 0.3);
    box-shadow: 0 10px 20px rgba(0, 0, 0, .5);
    width: 150px;
    padding: 20px;
    backdrop-filter: blur( 6.5px );
  }
</style>
<div>
  Lorem ipsum dolor sit amet consectetur
  adipisicing elit.
</div>
```

- A mais utilizada é a `rgba()`, que define cor para red, green, blue e alpha (transparência). Os valores das cores vão de 0 a 255 e o da transparência de 0 a 1.
- Outras propriedades são:
  - `rgb()`
  - `hsl()`
  - `hsla()`.
- O exemplo ao lado faz um efeito de vidro fosco combinando sombras, filtros e transparências.

# Funções de cores

```
<style>
  div {
    background-color: rgba(255, 255, 255, 0.3);
    box-shadow: 0 10px 20px rgba(0, 0, 0, .5);
    width: 150px;
    padding: 20px;
    backdrop-filter: blur( 6.5px );
  }
</style>
<div>
  Lorem ipsum dolor sit amet consectetur
  adipisicing elit.
</div>
```





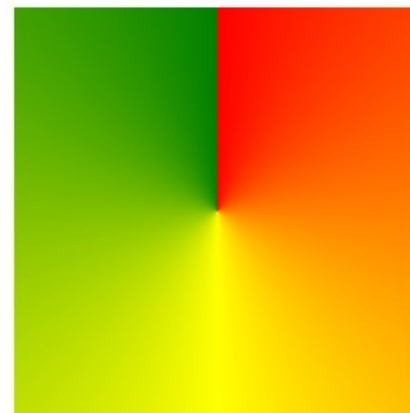
**INSTITUTO  
FEDERAL**  
Santa Catarina

# Funções de imagens

# Funções de cores

```
div {  
  height: 200px;  
  width: 200px;  
  /* For browsers that do not support gradients */  
  background-color: red;  
  background-image: conic-gradient(red, yellow, green);  
}
```

- Essas funções podem ser usadas sempre que uma `<image>` for válida como valor para uma propriedade.
- A maior parte das funções de imagens não foi implementada ainda nos navegadores.



# Outros funções de imagem

- conic-gradient()
- linear-gradient()
- radial-gradient()
- repeating-linear-gradient()
- repeating-radial-gradient()
- repeating-conic-gradient()
- cross-fade()
- element()
- paint()



**INSTITUTO  
FEDERAL**  
Santa Catarina

# Funções de contagem



# Funções de contagem

```
<style>
body {
  counter-reset: section;
}
h2::before {
  counter-increment: section;
  content: "Section " counter(section) ": ";
}
</style>
<h1>Using CSS Counters</h1>
<h2>HTML Tutorial</h2>
<h2>CSS Tutorial</h2>
```

- As funções de contador geralmente são usadas com a propriedade content, embora, em teoria, possam ser usadas sempre que uma <string> for suportada.

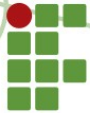
## Using CSS Counters

Section 1: HTML Tutorial

Section 2: CSS Tutorial

# Outros funções de contagem

- counter()
- counters()
- Symbols() - não implementado



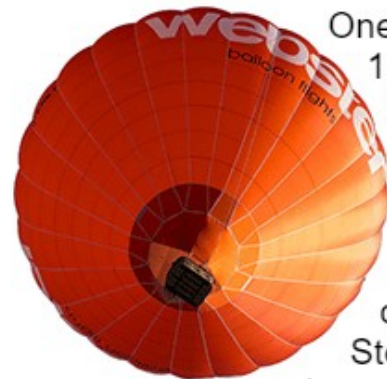
**INSTITUTO  
FEDERAL**  
Santa Catarina

# Funções de forma

# Funções de forma

```
img {  
  float: left;  
  shape-outside: circle(50%);  
}
```

- As funções de forma podem ser usadas como valores para o tipo de dados <basic-shape>, que é usado nas propriedades:
  - clip-path
  - offset-path
  - shape-outside.



One November night in the year 1782, so the story runs, two brothers sat over their winter fire in the little French town of Annonay, watching the grey smoke-wreaths from the hearth curl up the wide chimney. Their names were Stephen and Joseph Montgolfier, they were papermakers by trade, and were noted as possessing thoughtful minds and a deep interest in all scientific knowledge and new discovery. Before that night—a memorable night, as it was to prove—hundreds of millions of people had watched the rising smoke-wreaths of their fires without drawing any special inspiration from the fact.

# Outros funções de forma

- `circle()`
- `ellipse()`
- `inset()`
- `polygon()`
- `path()`



# Funções de referência

# Funções de referência

attr()	Usado para recuperar o valor de um atributo do elemento selecionado e usá-lo na folha de estilo.
env()	Usado para inserir o valor de uma variável de ambiente definida pelo ambiente do usuário. Não implementada totalmente
url()	Usada para incluir um arquivo.
var()	Usado para inserir um valor de uma propriedade personalizada em vez de qualquer parte de um valor de outra propriedade.

# Treinando...





# Treinando

- Para finalizar o projeto do exercício 03 (pizzaria), demonstre a utilização de 5 funções (além da já utilizada `var()`) no projeto
- Você pode adicionar novos elementos e conteúdos no seu HTML

