

# Construa um CSS mágico com variáveis nativas





Provavelmente, ao construir um website, você já deve ter reparado que muitos dos valores nos arquivos CSS são dados que se repetem constantemente, não é mesmo? Por exemplo, a paleta de cores que mantém o padrão visual da página é reutilizada em inúmeros pontos do código. Sendo assim, fazer a manutenção do projeto e alterar esses valores manualmente pode se tornar uma tarefa trabalhosa e passiva de erros, ainda mais em aplicações de grande escala.

Mas então, como podemos melhorar essa situação?

Uma boa alternativa é utilizar as **Variáveis no CSS** como aliadas!

Se você ficou interessado e deseja descobrir como fazer isso, vem comigo que nesse artigo eu te ensinarei mais sobre esse assunto detalhadamente.

## **Mas primeiro, o que são Variáveis de CSS?**

As variáveis de CSS, que também são chamadas de propriedades customizadas, trazem grandes benefícios e vão funcionar de forma bem semelhante às variáveis de linguagens de programação, ou seja, sua maior função é armazenar valores que serão utilizados posteriormente no código.

Elas permitem criar valores para todas as propriedades de CSS, sendo bastante utilizadas para gerenciar cores, fontes, valores de animações etc.

```
* CSS
1 :root {
2   --font-family-primary: 'Comfortaa', cursive;
3   --font-size-primary: 0.8rem;
4   --line-height-primary: 1.2rem;
5   --primary-color: #06113C;
6   --secondary-color: #FF8C32;
7   --tertiary-color: #DDDDDD;
8   --spacing-sm: 8px;
9   --spacing-lg: calc(var(--spacing-sm)*2)
10  --bg-primary: linear-gradient(45deg, var(--primary-color), var(--tertiary-color));
11 }
12
```

## Como utilizar as Variáveis de CSS?

As variáveis de CSS tem acesso ao DOM, o que permite que sejam declaradas globalmente ou localmente. Além disso, possuem uma sintaxe característica que já tivemos um pequeno spoiler na imagem acima. Vamos entender melhor como funciona?

### Sintaxe

A declaração ou criação de uma variável no arquivo CSS é composta por 4 partes, veja a seguir:

- 1º: Dois hífen (--)
- 2º: Nomeação da variável
- 3º: Dois pontos (:)
- 4º: Valor da variável

É importante destacar que alguns caracteres não são permitidos na nomeação, como por exemplo: ; # } ] ) etc.

```
* CSS
1 :root {
2   --global-primary-color: #FDAF75;
3 }
4
```

A aplicação das variáveis dentro de um elemento HTML em uma determinada propriedade CSS, é feita por meio da associação entre a função var() e o nome da variável, que é passada como parâmetro dentro dos parênteses. Dessa forma:

```
body {
  background: var(--global-primary-color);
}
```

### Declaração global

Fazemos isso declarando a variável dentro da pseudo classe `:root`. Esse seletor corresponde a raiz, é o mesmo que `<html>` em si, só que possui uma especificidade maior. Por meio do `root`, estamos disponibilizando elas globalmente, podendo ser utilizadas em qualquer lugar do documento.

## Declaração local

Para isso, declaramos a variável dentro do seletor que irá utilizá-la. Assim, ficarão restritas ao escopo da onde elas foram declaradas, sendo herdadas apenas pelos elementos filhos.

Lista de Compras

- Tomate
- Abóbora
- Abacate

```
HTML
1
2 <span>Lista de Compras</span>
3 <ul>
4   <li>Tomate</li>
5   <li>Abóbora</li>
6   <li>Abacate</li>
7 </ul>
8
CSS
1 ul {
2   --color-text-list: darkcyan;
3 }
4
5 ul li {
6   color: var(--color-text-list)
7 }
8
9 span {
10  color: var(--color-text-list) /*não irá aplicar*/
11 }
12
```

## De onde surgiu?

As variáveis são um recurso que durante muitos anos foi bastante solicitado. Isso porque, anteriormente, essa solução só era possível por meio de pré-processadores, como Sass, Stylus ou através de CSS in JS. Esses artifícios atuam como grandes parceiros, mas ainda sim havia algumas lacunas que faltavam ser preenchidas e que somente foram contempladas com as variáveis no CSS.

O principal ponto a se destacar é que as variáveis de pré processadores são estáticas e necessitam da etapa de compilação, ou seja, todas as funções, mixins e interpolações que ocorrem no Sass retornam CSS puro, para serem interpretadas pelo navegador.

Já as variáveis de CSS funcionam em tempo de execução, são dinâmicas e funcionam de forma independente das tecnologias usadas, atuando em qualquer contexto, sem precisar de etapas adicionais.

Além disso, ocorrem outras limitações com as variáveis de pré-processamento, como por exemplo a impossibilidade de uso de variáveis por meio do `@extend` nos medias query, a falta de aplicação da herança e efeito cascata nos elementos e como não há acesso ao DOM, não é possível serem controladas via JavaScript.

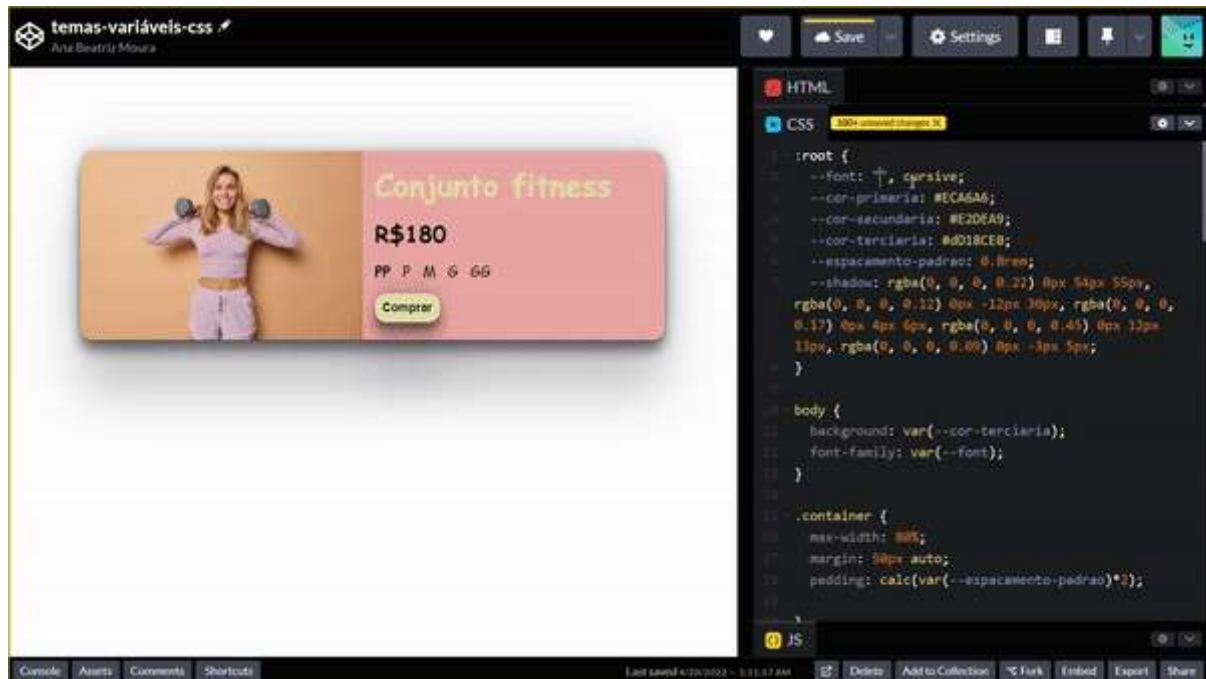
## E quais são as principais vantagens no uso?

### Código claro

Afinal, ler `--cor-principal-bordas` é muito mais semântico do que ler `#B4E197`, onde rapidamente conseguimos entender do que se trata.

## Favorece a manutenção

É possível alterarmos a folha de estilo em muitos lugares, apenas alterando o valor da variável em um único lugar, o que otimiza o tempo de trabalho e evita que possíveis erros de escrita aconteçam.



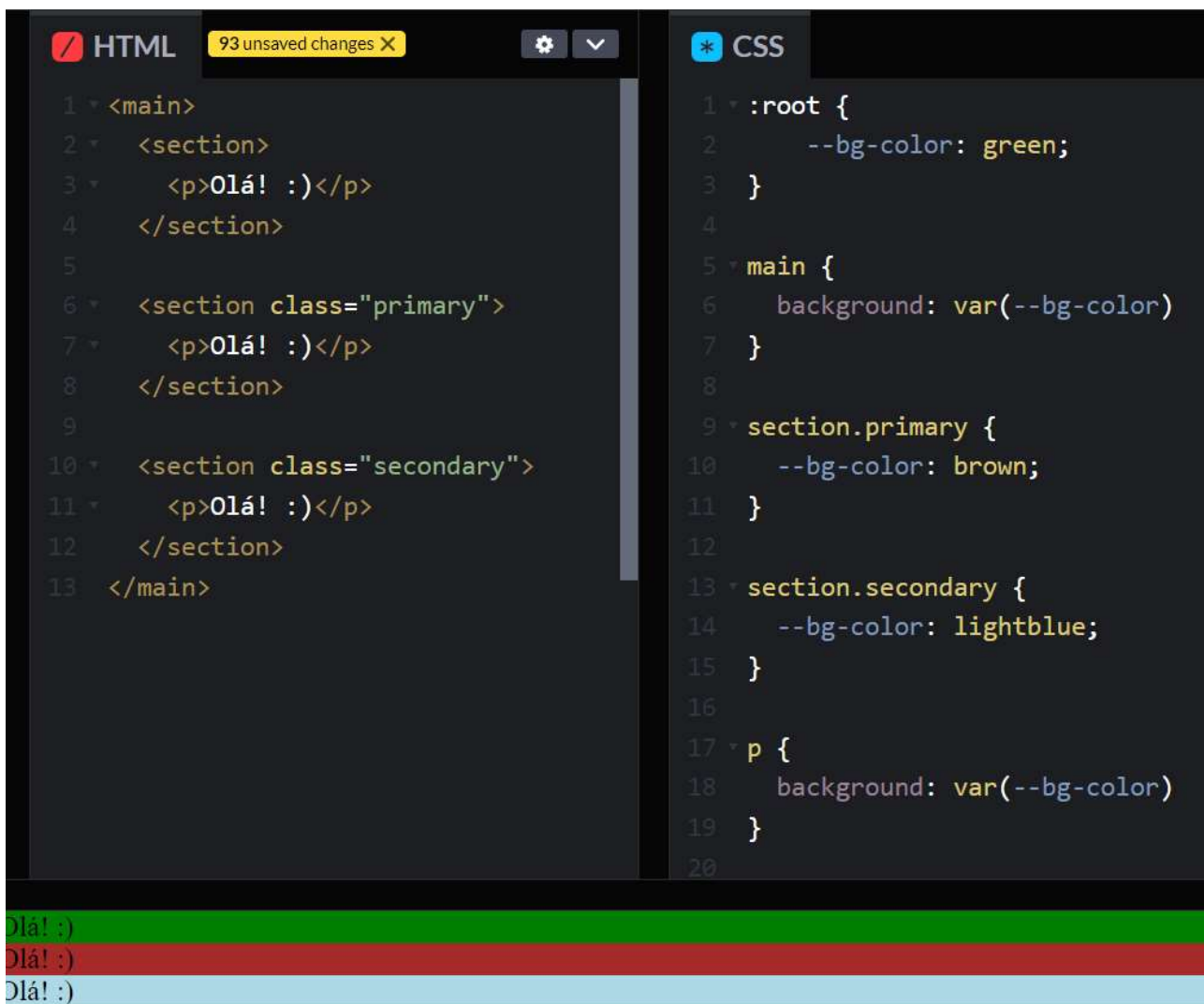
## Criação de temas

Com as variáveis, você pode criar e editar facilmente seus guias de estilos ou construir de maneira mais facilitada temáticas, um exemplo clássico disso é na construção de light/dark mode e como as variáveis de CSS são dinâmicas e tem acesso ao DOM, é possível ter um controle maior por meio da manipulação via Java Script, veja um exemplo prático nesse [Alura+ de Dark Mode](#).

## Conteúdo extra

### Cascata CSS

São beneficiadas pela cascata do CSS, onde podemos atribuir um novo valor a uma variável, de modo que os elementos filhos irão herdar as características do seu elemento pai mais próximo. Veja o exemplo:



Note que apesar o main estar utilizando a variável `--bg-color` com o valor `green`, quem ditou a cor de fundo dos parágrafos são as sections que possuíam outros valores atribuídos para essa variável, com exceção do parágrafo que não possui elemento pai section com valor definido, que neste caso, tomou como valor de background o aplicado para o main.

## Relação com o @media query

Atualmente, ainda não podemos utilizar variáveis como medidas de resolução de tela no @media query, dessa forma:

```
:root {
  --sm: 640px;
}

@media (max-width: var(--sm)) {
  ...
}
```

Ainda é necessário utilizar medidas fixas, mas dentro dos media queries é possível aplicar as variáveis, sem que haja problema algum.

```
:root {
  --color-text-desktop: red;
  --color-text-mobile: blue;
}

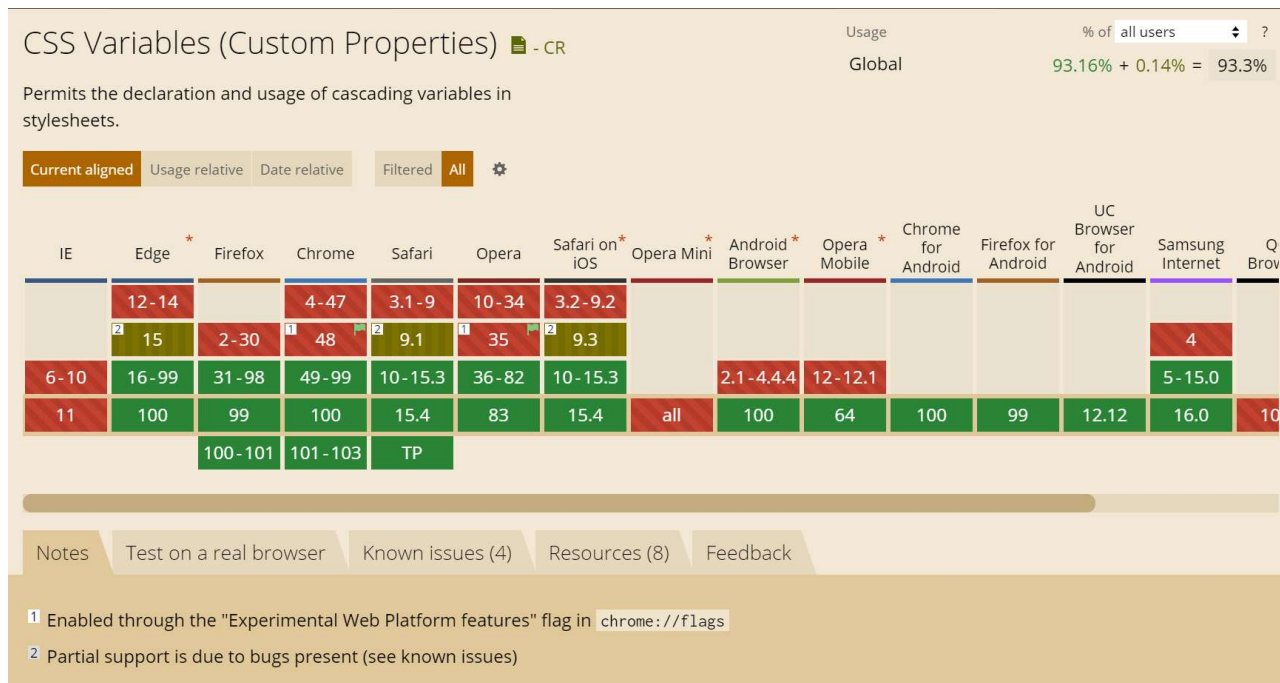
h1 {
  color: var(--color-text-desktop)
}
```



```
@media (max-width: 640px) {
  h1 {
    color: var(--color-text-mobile)
  }
}
```

## Suporte ao navegador

Segundo o site [Can I use](#), em relação ao suporte aos navegadores, as variáveis são compatíveis em praticamente todos os navegadores modernos de maneira satisfatória.



## Agora é com você!

Neste artigo, nós aprendemos sobre o poder do uso das variáveis no CSS em um projeto, entendemos melhor sobre os seus conceitos, vantagens e aplicação.

Para aprofundar o seu mergulho ainda mais no [desenvolvimento front-end](#), enriquecer seu portfólio e aplicar as variáveis de CSS em diversos projetos incríveis, não deixe de realizar os cursos da [formação front-end](#).

Agora me conta, você já utilizava as variáveis em sua rotina de desenvolvimento? Se sim, te convido a contar para nós aqui no fórum ou no discord da Alura as suas impressões da prática com elas. E se você nunca tinha ouvido falar ou se este conteúdo te ajudou de alguma forma, não deixe de dar o seu feedback, sua opinião é muito valiosa para nós e nos ajudar a evoluir e te proporcionar cada vez mais uma melhor experiência de aprendizado.

Até a próxima e bons estudos!