



# DevLinks

Por: Pedro Farias

▼ Conteúdo

▼ HTML e CSS

▼ Conceitos CSS - Cascata e Especificidade

- Cascata → Reflete a todas as tags que não estejam identificadas e leva em consideração a última modificação
- Especificidade → Uso de # id, . class para identificação na tag.

▼ CSS Conceitos - Box Model (Caixa)

Cada tag possui um Formato de caixa

Elementos da caixa:

Padding -> Preenchimento interno

Border -> Borda

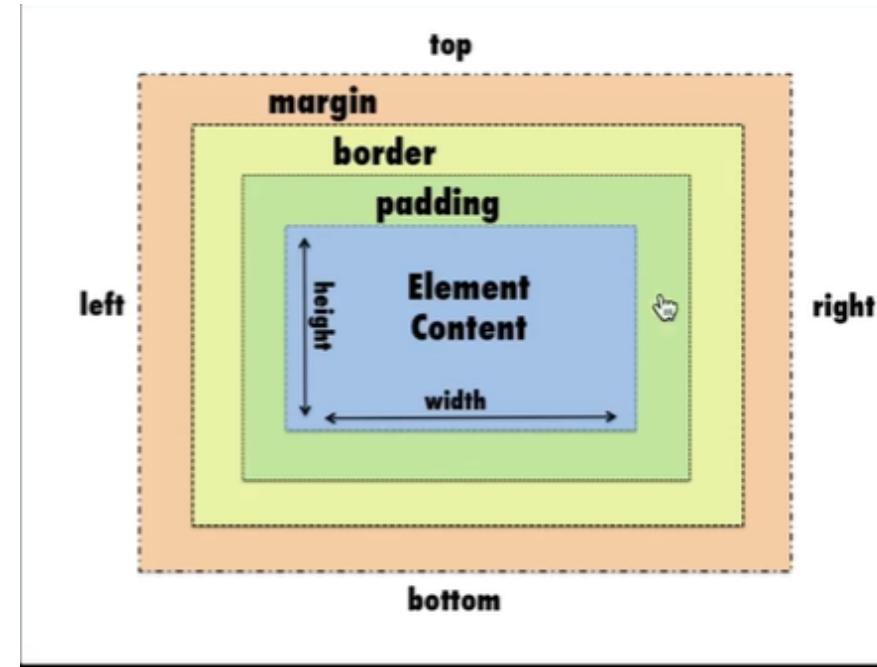
margin -> Espaçamento externo

Por padrão a maioria das tags tem um comportamento padrão, são display block, ou seja,

colocam uma caixa abaixo da outra.

div -> por padrão é display block

span -> por padrão é display inline



## ▼ Configurando o VSCode

### ▼ Extensões

- Live Server
- Omni Theme → Tema inspirado pela Rocketseat
- Material Icon Theme → ícones em geral
- Prettier → Formata (Indent) sozinho ao salvar (ctrl + s)

## ▼ Iniciando o projeto e o HTML

- Emmet → Ferramenta inteligente no VSCode - Autocompleta
- ! + Enter → Base do seu HTML
- Tag Head → Configurações da página
- Tag Body → Parte visível

## ▼ Adicionando o arquivo CSS

```
<link rel="stylesheet" href="./style.css">
```

- href → em que lugar no mundo está o css
- rel → Relacionamento
- ./ → Está na própria pasta

## ▼ Aplicando o background no projeto

### ▼ Uso do background

#### ▼ Criando um propriedade por vez

```
background-image: url('./assets/bg-mobile.jpg');  
background-repeat: no-repeat;  
background-position: center top;  
background-size: cover;
```

#### ▼ Atalho para todas as propriedades

- background: color image repeat position/size

```
background: url('./assets/bg-mobile.jpg') no-repeat top center/cover;
```

```
background-image: url('./assets/bg-mobile.jpg');
```

- url( ) → Um função

```
background-size: cover;
```

- cover → Cobre todo o espaço visível

## ▼ Adicionando a imagem de perfil

### ▼ Estilizando a imagem de Perfil

- No css geralmente não precisa definir a altura (height)

### ▼ Display none inline e block e alinhamento de elementos inline

- ▼ elemento Block → Ele entrega toda a largura de tela para o elemento. Forçando todos os elementos depois dele ficarem abaixo. É possível utilizar margin: auto.



- ▼ elemento inline → Permite outros elementos serem adicionados ao seu lado. Elementos inline (como img) não é possível utilizar margin: auto.

- CSS Trick (Pesquisar na web) → Truques de css

- ▼ Entendendo margin e padding

- ▼ margin/padding: x

- Se tenho uma única propriedade será aplicado para os 4 lados

- ▼ margin/padding: x x

- Ele pega a 1<sup>a</sup> propriedade para Top e Bottom (Cima e Baixo)
    - Ele pega a 2<sup>a</sup> propriedade para Right e Left (Direita e Esquerda)

- ▼ margin/padding: x x x

- Ele pega a 1<sup>a</sup> propriedade para Top (Cima)
    - Ele pega a 2<sup>a</sup> propriedade para Right e Left (Direita e Esquerda)
    - Ele pega a 3<sup>a</sup> propriedade para Bottom (Baixo)

- ▼ margin/padding: x x x x

- Ele pega as 4 propriedades, na ordem como o relógio: Top, Right, Bottom e Left

## ▼ Criando lista em HTML

```
<a href="https://github.com/pedrob14"  
target="_blank">Ver meu portfólio</a>
```

- href → Em que lugar do mundo está o link
- target → Se abre na mesma ou em uma nova página
- \_blank → Abre o link em uma nova aba

## ▼ Alinhando elementos com flex

```
display: flex;  
justify-content: center;  
Ou  
display: block;  
text-align: center
```

## ▼ Aplicando propriedades modernas

- border-radius → Arredondamento de borda

## ▼ Ajustando fontes dos links

### ▼ text-decoration

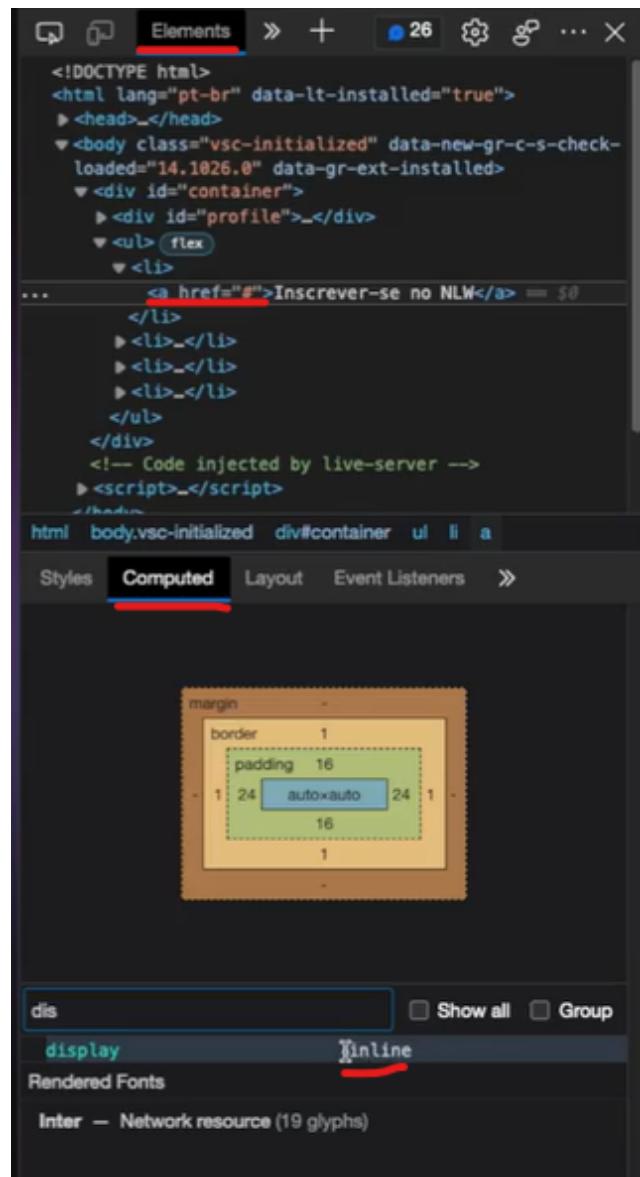
- none → *Retira o sublinhado do link/texto*
- underline → Sublinhado abaixo do link/texto
- overline → Sublinhado acima do link/texto
- line-through → Sublinhado no meio do link/texto

## ▼ Alterando propriedades com pseudo-selector hover

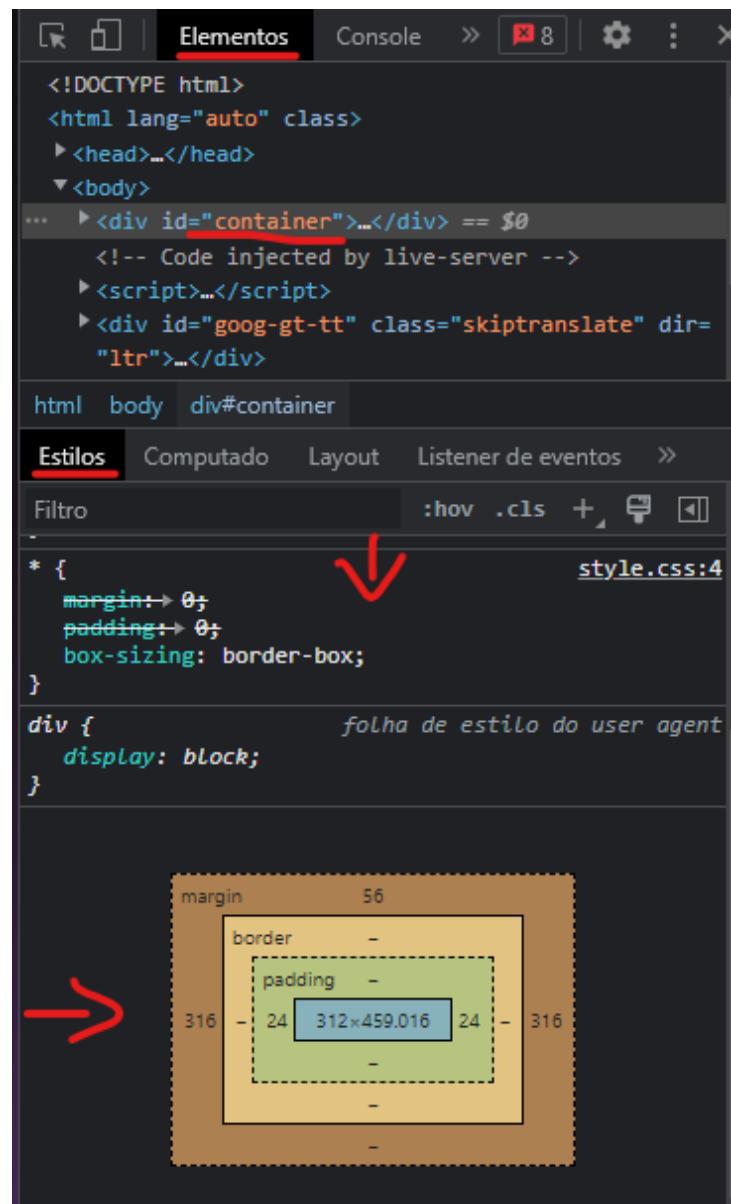
- hover → Ao passar o mouse por cima

## ▼ Analisando o código com a ferramenta do desenvolver DevToo

### ▼ Verificando o display padrão de cada tag pelo inspecionar



▼ Verificando o css no box model



## ▼ Adicionando ícones

- <https://ionic.io/ionicons>

## ▼ Javascript

### ▼ Algoritmos, Lógica de programação e Sintaxe

- Programação

É ensinar o computador

- Algoritmos

Sequência de passos, conjunto de regras

- Lógica de programação

Maneira de pensar

- Sintaxe

Maneira correta de escrever

## ▼ Introdução ao JavaScript

### O que é?

- Linguagem de programação

Interpretada e executada pelos navegadores

- A inteligência da tríade

HTML é a estrutura, CSS é a beleza e JS é a inteligência

- Não é JAVA

Apesar do nome ser semelhante, são linguagens diferentes

---

### Por que JS?

- Aplicativos

Para WEB, Desktop (Electron) e Mobile (React Native)

- Empresas famosas

Instagram, Google, Netflix, TikTok ...

- Moderna e Viva

Comunidade e linguagem que cresce cada vez mais

---

### Instruções e sintaxe

Toda linguagem é escrita com esses 2 princípios

- **Instruções** (declarações)

Ordens ao computador

- **Sintaxe**

Maneira correta de escrever

```
const mensagem = "Bom te ver aqui! "
alert(mensagem + (10 * 100) + " abraços")
// Bom te ver aqui! 1000 abraços
```



Existem **palavras reservadas** da linguagem.

Elas são responsáveis em dar significado a diversas instruções.

## Executando JavaScript

Podemos executar diretamente no Navegador

- Ferramenta **DevTools**  
Apertando o atalho **F12** no seu teclado
- Plataformas online  
[fronteditor.dev](http://fronteditor.dev)  
[codepen.io](http://codepen.io)
- Projeto no computador

```
<!-- em um arquivo index.html -->
<!-- executa js direto no arquivo .html -->
<script>alert('Olá!')</script>
<!-- importa um arquivo .js no seu .html -->
<script src=".script.js"></script>
```

## Projeto no computador

### ▼ Variáveis e Tipos de dados

- Variáveis

Uma caixinha onde guardamos um tipo de dado para usar mais tarde.

- Tipos de dados

Informações que podem ser em `textos`, `números`, `booleanos` (valores lógicos: verdadeiro ou falso) ou dados mais `estruturados`

## Variáveis e Tipos de dados

```
// declaro e atribuo valor
let boasVindas = 'Fala, Dev!';

// reatribuo valor
boasVindas = `Fala, Dev! Mayk Brito na área, que bom te ver aqui!`;

// constante não pode mudar o valor
const serHumano = true;
serHumano = false // Erro!

// declaro
let idade

// atribuo valor
idade = 40
```

## ▼ Funções

- Agrupamento de código
- Reuso de código
- Mini programas dentro do programa maior
- Toda linguagem oferece muitas opções

```
// usando uma função
alert('Fala, Dev!')

// criando uma função
function alert(text) {
    return text
}
```

## ▼ Objetos

### Tudo é objeto

- Atributos

São as propriedades de um objeto

- Métodos

São as funcionalidades de um objeto

```
// criando um objeto
const celular = {
  cor: 'preto',
  ligar: function() {}
}

// usando um objeto
celular.cor // preto
celular.ligar() // executa função
```

```
// criando um objeto
const person = {
  age: 18,
  drive: function() {}
}

// usando um objeto
person.drive()
```

## Document

Posso controlar minha página, meu documento HTML, pelo JavaScript, através do objeto `document`

```
document // objeto que existe em todo navegador
.querySelector('a') // seleciona a tag a
.click() // dá a ordem de clicar na tag a
```

### ▼ Adicionando lógica e condicional ao projeto

- Criando uma condicional e uma função que já realiza esta condicional

```
function toggleMode() {
  const html = document.documentElement

  /*if(html.classList.contains("light")) {
    html.classList.remove('light')
  } else {
    html.classList.add('light')
  } ou*/
  
  //Dentro da função toogle há esse conceito de add e remover.
  html.classList.toggle('light')
}
```

## ▼ Git e Github

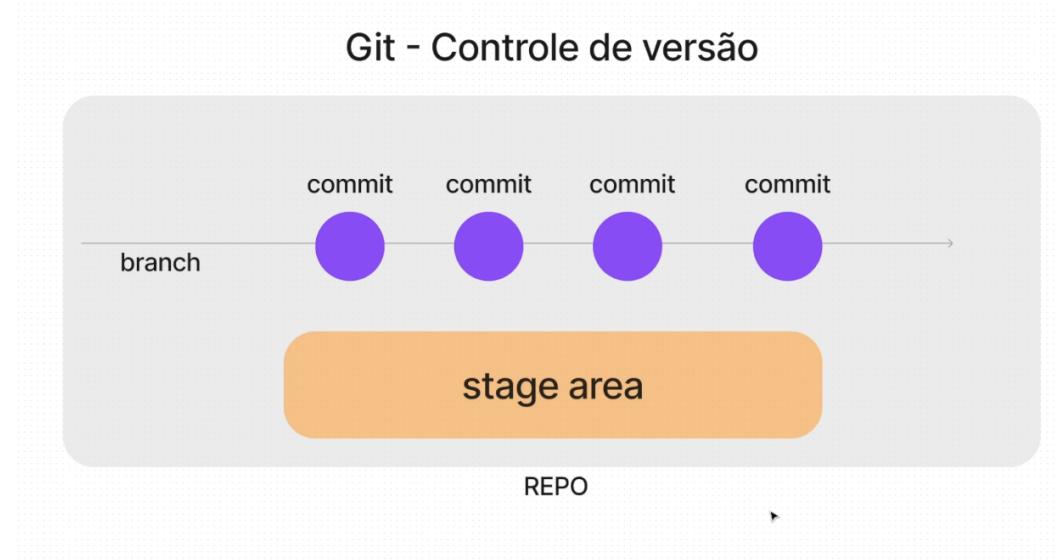
### ▼ Conceito

- Essas duas tecnologias estão presentes em todo o processo de programação, pois é com elas que você consegue ter um histórico do seu projeto.
- Você irá conhecer o que é o Git e os comandos fundamentais dele. Saberá como funciona o Github e como os dois se conectam.
- E o mais incrível, irá criar um link para compartilhar seu projeto com o mundo, divulgando esse link do seu projeto para qualquer pessoa!

### ▼ Introdução ao Git



Lembre de instalar o Git na sua máquina seguindo [essa documentação](#)



### O que é Git?

- Controle de versão
- Histórico do projeto
- Linha do tempo

## Conceitos básicos

- Repository

Local onde ficará o histórico do seu projeto

- Branch

Linha do tempo

- Commit

Pontos na história

- Stage

Preparação do que será enviado para o ponto na história

---

## Comandos

```
// inicia o git (repositório) no seu projeto  
git init  
  
// adiciona todos os arquivos modificados, ao stage  
git add .  
  
// cria e descreve um ponto na história  
git commit -m "message here"  
  
// envia alterações para o repositório remoto  
git push
```

### ▼ Iniciando git no projeto e criando o primeiro commit

- Repository

Local onde ficará o histórico do seu projeto

- Branch

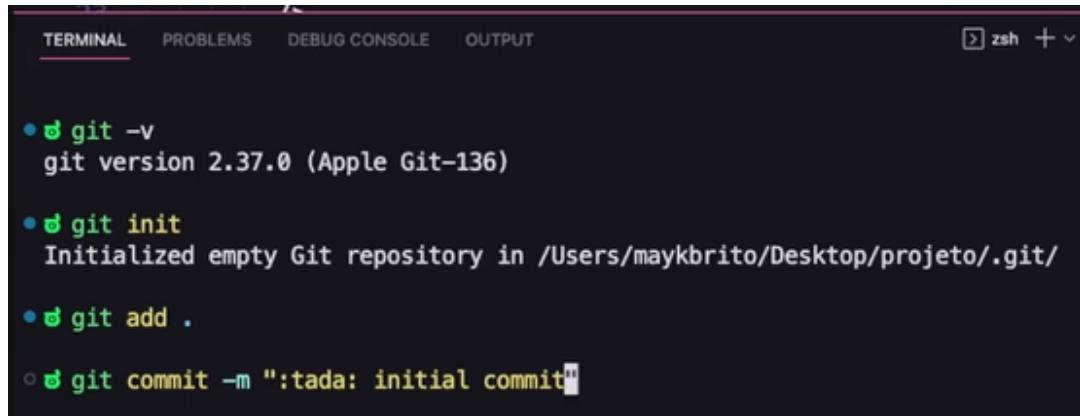
Linha do tempo

- Commit

Pontos na história

- Stage

Preparação do que será enviado para o ponto na história



A screenshot of a terminal window titled 'zsh'. The tabs at the top are 'TERMINAL', 'PROBLEMS', 'DEBUG CONSOLE', and 'OUTPUT'. The terminal content shows a sequence of Git commands:

- ⌘ git -v  
git version 2.37.0 (Apple Git-136)
- ⌘ git init  
Initialized empty Git repository in /Users/maykbrito/Desktop/projeto/.git/
- ⌘ git add .
- ⌘ git commit -m ":tada: initial commit"

## ▼ Adicionando projeto ao Github

### O que é ?

- Plataforma online para colocar seus códigos através do Git
- Trabalhar em diversos projetos Profissional para colaborar em projetos ou trabalhar com times
- Perfil para mostrar seu trabalho Portfólio

## ▼ mudando CSS com media queries

- Uso de at-rule

**At-rules** são instruções CSS que instruem o CSS sobre como se comportar. Eles começam com um sinal de arroba, '@'  
<https://developer.mozilla.org/en-US/docs/Web/CSS/At-rule#:~:text=At-rules%20are%20CSS%20statements,CSS%20block%2C%20whichever%20comes%20first>.

## ▼ Animações com CSS

- Uso de at-rule

**At-rules** são instruções CSS que instruem o CSS sobre como se comportar. Eles começam com um sinal de arroba, '@'  
<https://developer.mozilla.org/en-US/docs/Web/CSS/At-rule#:~:text=At-rules%20are%20CSS%20statements,CSS%20block%2C%20whichever%20comes%20first>.

▼ Acessar um editor de código online

<https://www.fronteditor.dev/>

▼ Pegar uma imagem aleatória no mundo

<https://source.unsplash.com/random>