**Introdução à Programação WEB**

**O que é programar?**

* Codar, Desenvolver, Programar

Resolver problemas do mundo real através das linhas de código

* Falar com o computador

Aprender linguagens para dar ordens ao computador

**Mercado**

* Vagas

[Mais de 790 mil até 2025 - Brasscom](https://brasscom.org.br/estudo-da-brasscom-aponta-demanda-de-797-mil-profissionais-de-tecnologia-ate-2025/)

* Global

Trabalhar em qualquer lugar do mundo

* Tempo de Experiência

Cerca de 2 anos

**Áreas de atuação e salários**

Podemos aprender programação em diversas áreas de atuação como, por exemplo:

* WEB
* Mobile
* Dados
* Inteligência artificial (A.I)
* Negócios
* Analista
* Internet das coisas (IoT) e outras.

<aside> 💡 O **salário** é de acordo com a vaga e a experiência da pessoa. Podem iniciar entre 2 a 3 salários mínimos e chegar a mais de 10, 20 😱

</aside>

**Programação WEB**

* Front-end

A parte visual das coisas, a interação com o usuário

* Back-end

A parte lógica das coisas, as regras de negócio

* Fullstack

Front e Back juntos

**Habilidades**

* Hard skills

Habilidades técnicas e conceitos da área

* Soft skills

Habilidades comportamentais (comunicação, resolução de problemas, gerenciamento de tempo e tarefas, trabalho em equipe …)

**Internet**

* Rede mundial de computadores
* O computadores estão conectados entre si
* Transferência de dados

**Programas e Arquivos**

* Programa e aplicativos (apps)
  + Navegadores
  + Mensagens
  + Redes sociais
* Arquivos
  + Foto, vídeo, textos, etc.
  + São interpretados por programas ou apps

**Front e Back-end**

Trocando informações

**Comunicação**

Traduzimos como front frente e back fundo. Existe uma comunicação entre essas duas pontas. Mas o que significa essas palavras?

<aside> 💡 Imagina uma farmácia onde você vai pedir um remédio para o atendimento

</aside>

**Cliente x Servidor**

* Cliente

Navegador (Browser)

* Servidor

Computador em algum lugar do mundo que tem os códigos

* Troca de dados (arquivos)

Cliente faz o pedido e Servidor escuta e responde ao pedido

Cliente é o front-end, servidor é o back-end

**Tecnologias Front-end**

* HTML

Linguagem de marcação de texto para estrutura os textos, criar links, imagens, etc...

* CSS

Linguagem de estilo para deixar o HTML bonito

* JavaScript

Linguagem de programação que funciona no Navegador

**Tecnologias Back-end**

* NodeJS

Rodar o JavaScript no computador

* SQL

Banco de dados para proteger os dados do seu programa

**HTML**

**O que é HTML?**

* Estruturar textos, criar links, imagens, vídeo, etc ...
* Hypertext Markup Language

Linguagem de marcação de texto

**Hypertext**

* Hipertexto
* Texto que contém links

**Markup**

* Marcação do texto
* Elemento HTML ou **tag**

Existem inúmeras tags e cada uma deles irá servir para um determinado propósito. Ex.: imagem, texto grande, link, parágrafo, etc...

**Sintaxe de uma tag**

Como escrevemos tags HTML?

<p>conteúdo</p>

<!-- Aqui vem um comentário -->

**Atributos**

Adicionam informações e/ou configurações à uma tag

Como escrevemos atributos?

**CSS**

**O que é CSS?**

* Estilos para o HTML
* Cascading Style Sheets

Folha de Estilo em Cascata

* Apresentação visual para o cliente

**Declaration**

* **Declaração**

Pedaço de código que irá ditar as propriedades e valores a serem aplicadas a um elemento HTML

* Como escrevemos ?

body {

background: black;

/\* color: green; Essa linha será ignorada \*/

}

**Cascading**

* **Cascata**

Quando há 2 (ou mais) declarações a última será mais relevante

body {

background: red;

}

body {

background: blue;

}

**Specificity**

* Especificidade

Cada seletor tem um peso e a soma dos pesos, será levada em conta para que determinada declaração seja mais específica

#id {

/\* peso 100 \*/

}

.class {

/\* peso 10 \*/

}

element {

/\* peso 1 \*/

}

*A cascata perde prioridade e é priorizada a especificidade da declaração*

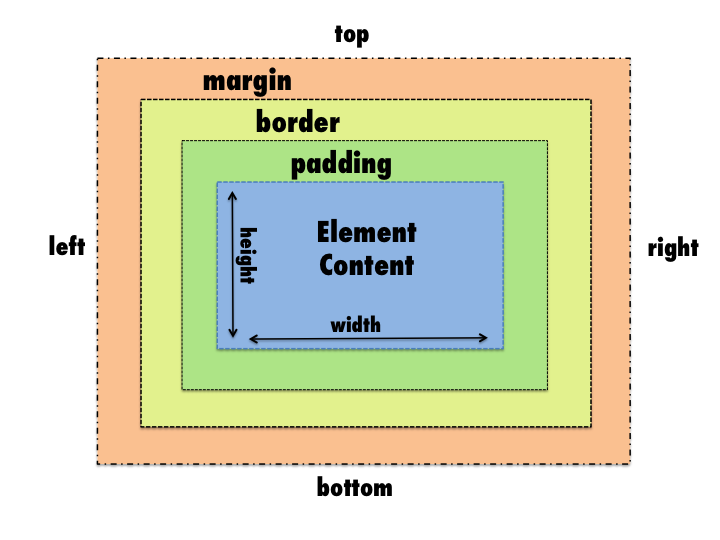
**Box Model**

* **Tudo são caixas**

Todos os elementos HTML serão considerados uma caixa, assim como uma caixa de papelão

* **Caixas possuem determinadas propriedades**

Conteúdo, Largura, Altura, Borda, Preenchimento (espaço interno), Espaçamento (espaço externo)



**JavaScript**

**Algoritmos e Lógica de programação**

**O que é programar?**

Programação nada mais é do que ensinar o computador

**Ensinar o computador**

* Algoritmos

Sequência de passos, conjunto de regras

* Lógica de programação

Maneira de pensar

* Sintaxe

Maneira correta de escrever

**JavaScript**

A linguagem da WEB

**O que é?**

* Linguagem de programação

Interpretada e executada pelos navegadores

* A inteligência da tríade

HTML é a estrutura, CSS é 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

<aside> 💡 Existem palavras reservadas da linguagem. Elas são responsáveis em dar significado a diversas instruções.

</aside>

**Executando JavaScript**

Podemos executar diretamente no Navegador

* Ferramenta DevTools

Apertando o atalho F12 no seu teclado

* Plataformas online

[fronteditor.dev](https://fronteditor.dev) [codepen.io](https://codepen.io)

* Projeto no computador

**Projeto no computador**

<!-- em um arquivo index.html -->

<!-- importamos um arquivo .js no .html -->

<script src="./script.js"></script>

**Variáveis e Tipos de dados**

Facilitando as informações

**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! Tudo beleza?!`;

// constante não pode mudar o valor

const serHumano = true;

serHumano = false // Erro!

**Funções**

Agrupando e reutilizando o código

**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**

É tudo sobre 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

**DOM**

Document Object Model

**Document Object Model - DOM**

É a modelagem dos nossos elementos HTML em Objeto JavaScript

**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

# NLWSetup – estudo de biblioteca

## NLWSetup

This class will be used in NLW #11 Setup and will help students  
to create the event's project by simplifying several JS concepts.

## Constructor

#### new NLWSetup(form)

Description:

* You must build the HTML structure like this
* <form>
* <div class="habits">
* <div class="habit" data-name="run">🏃🏽‍♂️</div>
* <div class="habit" data-name="water">💧</div>
* <div class="habit" data-name="food">🍎</div>
* </div>
* <div class="days"></div>

</form>

Example of usage

const form = document.querySelector('form')

const nlwSetup = new NLWSetup(form)

Source:

* [NLWSetup.js](https://maykbrito.github.io/libs/NLWSetup/documentation/NLWSetup.js.html), [line 5](https://maykbrito.github.io/libs/NLWSetup/documentation/NLWSetup.js.html" \l "line5)

##### **Parameters:**

| **Name** | **Type** | **Description** |
| --- | --- | --- |
| form | HTMLFormElement |  |

### Methods

#### addDay(date)

Description:

* Add a day to registered days and render the layout after that

example of usage

nlwSetup.addDay('31/12')

Source:

* [NLWSetup.js](https://maykbrito.github.io/libs/NLWSetup/documentation/NLWSetup.js.html), [line 225](https://maykbrito.github.io/libs/NLWSetup/documentation/NLWSetup.js.html" \l "line225)

##### **Parameters:**

| **Name** | **Type** | **Description** |
| --- | --- | --- |
| date | string | DD/MM   * DD day with 2 digits (01-31) * MM month with 2 digits (01-12) |

#### dayExists(date) → {boolean}

Description:

* Check if day already exists in the set of days

example of usage

nlwSetup.dayExists('31/12') // true or false

Source:

* [NLWSetup.js](https://maykbrito.github.io/libs/NLWSetup/documentation/NLWSetup.js.html), [line 210](https://maykbrito.github.io/libs/NLWSetup/documentation/NLWSetup.js.html" \l "line210)

##### **Parameters:**

| **Name** | **Type** | **Description** |
| --- | --- | --- |
| date | string | DD/MM   * DD day with 2 digits (01-31) * MM month with 2 digits (01-12) |

##### **Returns:**

Type

boolean

#### load()

Description:

* It will load the internal [data](https://maykbrito.github.io/libs/NLWSetup/documentation/NLWSetup.html#data) and [render the layout](https://maykbrito.github.io/libs/NLWSetup/documentation/NLWSetup.html#renderLayout)

Source:

* [NLWSetup.js](https://maykbrito.github.io/libs/NLWSetup/documentation/NLWSetup.js.html), [line 71](https://maykbrito.github.io/libs/NLWSetup/documentation/NLWSetup.js.html" \l "line71)

#### renderLayout()

Description:

* Will render the layout by checking the registered days of the habits data  
  Each day that was included in the habits data, will be checked in the input layout

The <div class="days"></div> container will be filled with html,  
repeating the .day container for every day registered

example

<div class="day">

<div>01/01</div>

<input type="checkbox" name="run" value="01/01"/>

<input type="checkbox" name="water" value="01/01"/>

<input type="checkbox" name="food" value="01/01"/>

</div>

Source:

* [NLWSetup.js](https://maykbrito.github.io/libs/NLWSetup/documentation/NLWSetup.js.html), [line 105](https://maykbrito.github.io/libs/NLWSetup/documentation/NLWSetup.js.html" \l "line105)

#### setData(data)

Description:

* The data to build the habits table

Example of usage

const data = {

run: ['01-01', '01-02', '01-06'],

water: ['01-04', '01-05'],

food: ['01-01', '01-03'],

}

nlwSetup.setData(data)

So

# Git

<aside> 💡 Lembre de instalar o Git na sua máquina seguindo [essa documentação](https://www.notion.so/Preparando-o-Ambiente-151ab9f547c0445a99d993ff39ef5048)

</aside>

## 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\*\*

# Github

<aside> 💡 Lembre de criar sua conta no Github seguindo [esse documento](https://www.notion.so/Preparando-o-Ambiente-151ab9f547c0445a99d993ff39ef5048)

</aside>

## 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

##### **Parameters:**

| **Name** | **Type** | **Description** |
| --- | --- | --- |
| data | object | Object that contains the data composed by   * key: The habit name as the data-name of the .habit container * value: Array of days to be checked as done   Each day must be like the format *MM-DD* where *MM*: is the month and *DD*: is the day  { run: ['01-01', '01-02', '01-06'] } |