Foto preta e branca de onda

Descrição gerada automaticamente com confiança média

ENXURRADA DE BITS

Programação

Web

professor

Diego Simões

Imagem em preto e branco de vaso de flor

Descrição gerada automaticamente

Ana Clara Alves



ÍNDICE

ÍNDICE

[Aula 0 - Apresentação 7](#_Toc75186766)

[Sobre o programa 7](#_Toc75186767)

[Objetivos do programa 7](#_Toc75186768)

[Responsabilidades dos alunos 7](#_Toc75186769)

[Matérias 7](#_Toc75186770)

[Distribuição de pontos 8](#_Toc75186771)

[O que precisaremos? 8](#_Toc75186772)

[Aula 01 - HTML 9](#_Toc75186773)

[Definição de Web 9](#_Toc75186774)

[O que compões uma página Web 9](#_Toc75186775)

[O que são linguagens de Programação Web 9](#_Toc75186776)

[HTML 9](#_Toc75186777)

[Passo a passo para programar em HTML 10](#_Toc75186778)

[Passo a passo para visualizar seu site no navegador 10](#_Toc75186779)

[Anatomia de Tag’s 10](#_Toc75186780)

[Tag’s básicas 10](#_Toc75186781)

[Tag <html> 10](#_Toc75186782)

[Tag <head> 11](#_Toc75186783)

[Tag<body> 11](#_Toc75186784)

[Tag <img> 11](#_Toc75186785)

[Tag <a> 11](#_Toc75186786)

[Tag’s de texto e decoração 12](#_Toc75186787)

[Aula 02 - CSS 13](#_Toc75186788)

[O que é CSS? 13](#_Toc75186789)

[Definindo estilos 13](#_Toc75186790)

[Importando um arquivo CSS 13](#_Toc75186791)

[Anatomia CSS 13](#_Toc75186792)

[Propriedades básicas 13](#_Toc75186793)

[background-color 13](#_Toc75186794)

[color 14](#_Toc75186795)

[font-family 14](#_Toc75186796)

[font-size 14](#_Toc75186797)

[text-align 14](#_Toc75186798)

[letter-spacing & word-spacing 15](#_Toc75186799)

[background-image 15](#_Toc75186800)

[width & height 15](#_Toc75186801)

[float 15](#_Toc75186802)

[Box Model 16](#_Toc75186803)

[Content 16](#_Toc75186804)

[Padding 16](#_Toc75186805)

[Border 16](#_Toc75186806)

[Margin 16](#_Toc75186807)

[Aula 03 - GitHub 17](#_Toc75186808)

[Git 17](#_Toc75186809)

[GitHub 17](#_Toc75186810)

[Hospedagem de sites 17](#_Toc75186811)

[Hospedagem 17](#_Toc75186812)

[Domínio 17](#_Toc75186813)

[GitHub Pages 18](#_Toc75186814)

[Publicando sites no GitHub Pages 18](#_Toc75186815)

[Aula 04 - .class e #id, <div> e <span> 19](#_Toc75186816)

[Anatomia CSS 19](#_Toc75186817)

[Seletor de tag - Estilização 19](#_Toc75186818)

[Seletor de tag ID 19](#_Toc75186819)

[Seletor de tag CLASS 19](#_Toc75186820)

[Em qual caso usar cada uma? 19](#_Toc75186821)

[Seletor de tag - Organização 19](#_Toc75186822)

[Agrupador de tag DIV 20](#_Toc75186823)

[Agrupador de tag SPAN 20](#_Toc75186824)

[Em qual caso usar cada uma? 20](#_Toc75186825)

[Conteúdo adicional 20](#_Toc75186826)

[Disposição do texto – INLINE & BLOCK 20](#_Toc75186827)

[Aula 05 - Listas e Tag’s semânticas 21](#_Toc75186828)

[O que é uma lista para o HTML? 21](#_Toc75186829)

[Lista ordenada 21](#_Toc75186830)

[Lista não-ordenada 21](#_Toc75186831)

[Estilização de listas 21](#_Toc75186832)

[list-style 21](#_Toc75186833)

[list-style-type 22](#_Toc75186834)

[Tag’s semânticas 22](#_Toc75186835)

[Tag <header> 23](#_Toc75186836)

[Tag <nav> 23](#_Toc75186837)

[Tag <main> 23](#_Toc75186838)

[Tag <section> 23](#_Toc75186839)

[Tag <article> 23](#_Toc75186840)

[Tag <aside> 23](#_Toc75186841)

[Tag <footer> 23](#_Toc75186842)

[Aula 06 - Estados CSS e animações 24](#_Toc75186843)

[Pseudoclasses 24](#_Toc75186844)

[Sintaxe das pseudoclasses 24](#_Toc75186845)

[Algumas pseudoclasses 24](#_Toc75186846)

[:hover 24](#_Toc75186847)

[:focus 24](#_Toc75186848)

[:active 24](#_Toc75186849)

[:link 24](#_Toc75186850)

[:visited 24](#_Toc75186851)

[Pseudoelementos 25](#_Toc75186852)

[Sintaxe dos pseudoelementos 25](#_Toc75186853)

[Alguns pseudoelementos 25](#_Toc75186854)

[::before 25](#_Toc75186855)

[::after 25](#_Toc75186856)

[::placeholder 25](#_Toc75186857)

[::selection 25](#_Toc75186858)

[Animações 25](#_Toc75186859)

[Transition 25](#_Toc75186860)

[Algumas propriedades aplicadas ao transition 26](#_Toc75186861)

[Sintaxe da propriedade transition 26](#_Toc75186862)

[Transform 26](#_Toc75186863)

[Animation 26](#_Toc75186864)

[Algumas propriedades aplicadas ao animation 26](#_Toc75186865)

[Sintaxe da propriedade animation 27](#_Toc75186866)

[Aula 07 - JavaScript 28](#_Toc75186867)

[Implementando JS no código HTML 28](#_Toc75186868)

[Implementação com código interno 28](#_Toc75186869)

[Implementação com código externo 28](#_Toc75186870)

[Variáveis no JavaScript 29](#_Toc75186871)

[Tipos de variáveis 29](#_Toc75186872)

[Number 29](#_Toc75186873)

[String 29](#_Toc75186874)

[Boolean 29](#_Toc75186875)

[Operações em JavaScript 30](#_Toc75186876)

[Interpolação 30](#_Toc75186877)

[Função alert() 30](#_Toc75186878)

[Função console.log() 30](#_Toc75186879)

[Função prompt() 30](#_Toc75186880)

[Aula 08 - Estruturas de decisão e repetição 31](#_Toc75186881)

[Comandos de Decisão 31](#_Toc75186882)

[Comandos de Repetição 31](#_Toc75186883)

[Aula 09 - Funções 32](#_Toc75186884)

[Aula 10 - Integração do JavaScript ao HTML 33](#_Toc75186885)

[Aula 11 - Entradas, botões e eventos 34](#_Toc75186886)

[Aula 12 – Despedida 35](#_Toc75186887)

Desenho preto e branco

Descrição gerada automaticamente com confiança médiaDesenho preto e branco

Descrição gerada automaticamente com confiança médiaDesenho preto e branco

Descrição gerada automaticamente com confiança médiaDesenho preto e branco

Descrição gerada automaticamente com confiança médiaDesenho preto e branco

Descrição gerada automaticamente com confiança média

# Aula 0 - Apresentação

## Sobre o programa

* Totalmente gratuito
* Será ministrado aos sábados: Turma M1 - 13:00 até 14:30
* Monitorias semanais no servidor do Discord
* Objetivo: O básico das linguagens utilizadas para fazer **páginas web**

## Objetivos do programa

* Levar a comunidade externa o aprendizado de lógica e linguagem de programação
* Incentivar o pensamento lógico, despertando curiosidade e vontade de aprender
* Inserção na área da tecnologia
* Ajudar os alunos a desenvolver facilidade e bom desempenho em disciplinas escolares(matemática, física etc.)

## Responsabilidades dos alunos

* Ser frequente(se necessário, justificar suas faltas para o professor com antecedência)
* Esclarecer qualquer dúvida com professores ou monitores
* Fazer as atividades propostas
* Atingir pontuação mínima de 60 pontos

## Matérias

* HTML
* CSS
* Flor cor de rosa

  Descrição gerada automaticamenteComo publicar um site no GitHub
* As tag’s class, id, div e span
* Lista e tag’s semânticas
* Estados CSS e animações
* JavaScript
* Comandos de decisão e estruturas de repetição
* Funções
* Entradas, botões e eventos
* Intefração do JavaScript ao HTML

## Distribuição de pontos

* Atividades em sala: 20 pontos
* Atividades em casa (Simpsons): 20 pontos
* Site “Jornal do Bairro”: 20 pontos
* Site final: 40 pontos

## O que precisaremos?

Criar um email

↓

Acessar o Google Classroom

↓

Acessar o Discord

↓

Baixar o VSCode

# Aula 01 - HTML

## Definição de Web

**Word Wide Web**, WWW, traduzido como “Teia de comunicações mundial”

1ª definição: É o que estamos vendo e interagindo quando abrimos um navegador(Chrome, Firefox etc.)

2ª definição: Um conjunto de documentos de hipermídia interconectados, tipicamente acessados pela Internet. Esses documentos são as páginas web.

* **Hipermídia:** conteúdo digital que envolve texto, imagens, sons, vídeos etc.
* **Interconectado:** um documento que contém “links” (referências) a outros(daí a ideia de teia 🕸)

## O que compões uma página Web

* Um arquivo no formato HTML(ex.: arquivo.html)
* Arquivo CSS(ex.: estilo.css)
* Arquivos JavaScript(ex.: interação.js)
* Imagens(.png, .jpg, .gif)
* Vídeos
* Etc.

## O que são linguagens de Programação Web

* Flor cor de rosa

  Descrição gerada automaticamenteSão linguagens de programação específicas para o desenvolvimento de sites e aplicações que rodam na internet

HTML: Responsável pelo conteúdo e estrutura da página(texto, imagens, links)

CSS: Responsável pela estilização da página(cores, fonte, tamanhos e detalhes)

JavaScript: Responsável pelo comportamento dinâmico da página(pop-up, cálculos e efeitos de movimento)

## HTML

* Linguagem de Marcação de Hipertexto, composto por **tag’s**(normalmente fechadas);
* O código é visível quando **CTRL+U** é apertado, e para depurar **F12** é o melhor atalho.

## Passo a passo para programar em HTML

Escolher um editor de texto(**VSCode**)

↓

Criar um arquivo com extensão **.html**

↓

Abrir o arquivo no programa escolhido

↓

Adicionar as tag’s essenciais(html; head; title; body)

↓

Começar a programar seu site

## Passo a passo para visualizar seu site no navegador

Abra seus arquivos

↓

Procure seu arquivo **.html** salvado

↓

Clique com o botão direito do mouse em cima do arquivo

↓

Clique em abrir com o navegador

## Anatomia de Tag’s

Elemento

| Tag de abertura <⋅⋅⋅>

| | Conteúdo

| Tag de fechamento </⋅⋅⋅>

## Tag’s básicas

### Tag <html>

Define que o tipo do documento é HTML, envolve todas as outras tag’s, (exceto a tag especial <!DOCTYPE html>), dentro dela, deve existir as seguintes tag’s, nesta ordem:

<head>

...

</head>

<body>

...

</body>

### Tag <head>

Originado do inglês: “head=cabeça”; contém metainformação sobre este arquivo html, por exemplo:

* Codificação (*encoding*) da página;
* Título da página no topo do navegador (obrigatório);
* Estilização da página;
* Inclusão de arquivos (**.css**, **.js**) e o ícone da página.

### Tag<body>

Originado do inglês: “body=corpo”; contém todo o conteúdo da página, como:

* a tag de **parágrafos** (<p>...</p>);
* as tag’s de **títulos e subtítulos** (<h1>...</h1> [h1,h2,h3,h4,h5 e h6])
* a tag de imagem (<img>), etc.

endereço de imagem

(interno/externo)

### Tag <img>

↓

↑

↑

↓

tag de autofechamento

tag de abertura

<img src=”image.png”/>

atributo

* **Externa:** Aponta para algo (imagem) de um outro site. Quase sempre começa com http://

Ex.: http://google.com

* **Interna:** Aponta para algo da própria página/site. Pode ser um caminho **relativo** ou **absoluto**.

Ex.: imagens/gato.png (**relativo**)

/arquivos/cachorros.zip (**absoluto**)

endereço

(interno/externo)

### Tag <a>

atributo

↑

tag de fechamento

↑

<a href”https://www.google.com”> Texto </a>

↓

↓

tag de abertura

* **Externa:** Para fora da página, começa com http://

Ex.: http://google.com

* **Interna:** Para algo hospedado no próprio computador

Ex.: exemplo.html

### Tag’s de texto e decoração

* Negrito <strong>...</strong>
* *Itálico* <em>...</em>
* Sublinhado <ins>...</ins>
* ~~Tachado~~ <del>...</del>
* Grifado <mark>...</mark>
* Quebra de linha <br>
* Cria uma linha horizontal <hr>

# Aula 02 - CSS

## O que é CSS?

CSS é a sigla para **C**ascading **S**tyle **S**heets (folhas de estilo em cascatas), uma linguagem para especificar como os elementos de um site serão apresentados para o usuário(de forma visual), alterando a aparência do site.

## Definindo estilos

Uma maneira simples de adicionar CSS na página é por meio do elemento <style>...</style>, que ao ser colocado em umas das tag’s principais(<head> ou <body>; dando preferência a tag <head>) altera a estilização exibida do código.

## Importando um arquivo CSS

A maneira **correta** de adicionar estilos na página é por meio da importação de um arquivo com extensão *.css*(exemplo: **style.css**). Esse arquivo de estilização é considerado o correto por permitir uma divisão mais organizada do código, além de poder ser utilizado em várias páginas e ser mais intuitivo.

## Anatomia CSS

→

Seletor: Seleciona o elemento que vai ter a estilização dentro das **chaves**({})

h1

Propriedade: É a especificação do que será aplicado

{

Valor: São as características da propriedade aplicada

↱

color: orange;

→

text-align: center;

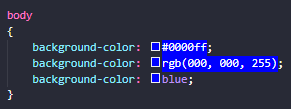
↓

}

Declaração: É o conjunto da propriedade aplicada e eu valor.

## Propriedades básicas

### background-color

Responsável por definir a cor de fundo, as cores podem ser escritas em RGB, hexadecimal e ainda contém suporte para algumas cores por extenso(0000FF[hexadecimal]/000 000 255[RGB]/Blue[extenso])

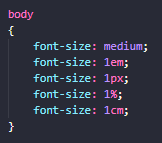
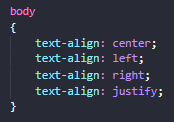
### color

Define a cor do texto, as cores seguem a regra padrão do formato(RGB/HEX/Color)

### font-family

É a propriedade usada para definir a fonte do texto.

### font-size

Responsável por definir o tamanho do texto. No exemplo é mostrado unidades de medidas, as mais conhecidas são **px**(pixel), **em**(referência de tamanho a fonte sendo usada), **%**(porcentagem[em relação ao elemento “pai”])**cm**(centímetros) ou da forma mais básica(usada apenas na fonte) **medium**, **large**, **small**(média, grande, pequeno etc.).

### text-align

Essa propriedade alinha o texto conforme o desejado, **centralizado**, **justificado**, **alinhado à esquerda** ou **alinhado à direita**.

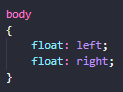
### letter-spacing & word-spacing

Define o espaçamento entre caracteres(letter) e/ou entre palavras(word).

### background-image

É a propriedade que define uma imagem como fundo da página.

### width & height

Definem a largura(width) de determinado elemento e/ou altura(height).

### float

Essa propriedade tira o elemento do fluxo normal e a coloca flutuando, fazendo com que outros elementos se posicionem ao seu redor.

## What is the CSS Box Model · Front End web development - coding leadBox Model

### Content

É o conteúdo, qualquer elemento colocado na página HTML, como links, imagens, blocos, texto etc.

### Padding

Define o espaçamento entre o conteúdo(contente) e a borda(border), fazendo o preenchimento do bloco.

### Border

Define a borda do elemento e suas características: cor, espessura e tipo de borda

### Margin

Define o espaço a partir da borda até o próximo elemento, fazendo a margem do elemento

# Aula 03 - GitHub

## Git

Git é um sistema de controle de versão de arquivos, ele é gratuito e de código aberto

Utilidade

* Controle de versão: consiste basicamente em um sistema que permite registrar alterações feitas no desenvolvimento de um software;
* Os arquivos do projeto são mantidos em um repositório, salvando as diferentes versões do software;
* Possibilita um trabalho em equipe de qualidade [principal];
* Permite testar novas funcionalidades e decidir se quer realmente implantá-las;

## GitHub

É um serviço Web que utiliza Git, sendo a maior plataforma de hospedagem de código-fonte atualmente e utilizado por diversas empresas da área de desenvolvimento

Para que utilizaremos o GitHub no Enxurrada de Bits?

* Manter um portfólio pessoal dos projetos que desenvolvemos;
* Hospedar gratuitamente os nossos sites de forma que qualquer pessoa com internet possa acessá-los.

## Hospedagem de sites

Um site é um conjunto de arquivos que pode ser baixado e lido por navegadores/browsers.

### Hospedagem

* É o espaço que seu site vai ocupar na internet e é por meio deste serviço que você armazena os arquivos que mantém seu site no ar.
* É difícil manter computadores ativos e por isso existem empresas fornecem esses serviços por diversos valores.

### Domínio

* É o endereço utilizado para acessar o seu site (ex.: www.meusite.com.br)
* Os domínios são únicos e precisam ser comprados

## GitHub Pages

* Serviço gratuito oferecido pelo GitHub de hospedagem de sites(aceita vários formatos: **.html**, **.css**, **.javascript**, **.php** etc.)
* Disponibilizar sites estáticos (não há interação com banco de dados[armazenamento de informações])

### Publicando sites no GitHub Pages

1ª Maneira

* Crie um repositório chamado usuário.github.io
* Envie seus arquivos para o repositório (tem que ter um index.html)
* Seu site estará em usuário.github.io

2ª Maneira (Muito mais legal :D)

* Crie um repositório com um nome qualquer
* Crie um branch chamado gh-pages
* Envie seus arquivos para essa branch (tem que ter um index.html);
* Seu site estará em usuário.github.io/nome-do-repositório

# Aula 04 - .class e #id, <div> e <span>

## Anatomia CSS

Seletor: Seleciona o elemento que vai ter a estilização dentro das **chaves**({})

→

h1

↱

Propriedade: É a especificação do que será aplicado

{

Valor: São as características da propriedade aplicada

→

color: orange;

text-align: center;

↓

}

Declaração: É o conjunto da propriedade aplicada e eu valor.

### Seletor de tag - Estilização

As regras (propriedades escritas dentro das chaves) se aplicarão a todos os elementos da tag que dá nome ao seletor. Ex.: Todo texto que utilizar da tag <h1>será estilizado conforme escrito no documento CSS

**Mas como estilizar ó alguns elementos da página?**

Usando os atributos universais HTML <class> e <id>.

#### Seletor de tag ID

<h1 id=”nome\_da\_id”>...</h1>

Ao usar o seletor no CSS, usa-se o símbolo #(hashtag) e as regras(propriedades) serão aplicadas apenas nos elementos que possuem o atributo **id**.

#### Seletor de tag CLASS

<h1 class=”nome\_da\_class”>...</h1>

Ao usar o seletor no CSS, usa-se o símbolo .(ponto) e as regras(propriedades) serão aplicadas apenas nos elementos que possuem o atributo **class**.

#### Em qual caso usar cada uma?

* Use <id> se a tag for única e exclusivas no seu site, como um título, o menu principal ou o logotipo
* Use <class> se a tag for utilizada para ‘agrupar’ a estilização de conteúdo do seu site que são comuns, nas partes que se repetem, como botões, links, listas ou até mesmo posts.

### Seletor de tag - Organização

Durante a estilização do HTML são atribuídas propriedades aos elementos, para isso é utilizado o CSS. Mas e quando precisamos agrupar elementos com o intuito de separar e organizar o código para estilizá-lo? Ou precisamos criar uma forma 2D sólida sem utilizar imagens? Talvez desejemos atribuir uma característica específica a uma única palavra de forma que não é possível utilizar apenas HTML.

**Mas como seria possível organizar o código dessa forma?**

Usando as tag’s universais HTML <div> e <span>.

São tag’s que só agrupam elementos sem interferir no código.

#### Agrupador de tag DIV

<div class=”ex1”>

<h1> ... </h1>

<p> ... </p>

</div>

Ao usar a tag *div* é possível criar uma “**quebra**” no código, de modo que ao atribuir propriedades da classe dessa *div*, todos os elementos dentro dela receberão as mesmas características. Ao criar uma *div* vazia também é possível criar formas sólidas 2D, como círculos, retângulos etc.

#### Agrupador de tag SPAN

<span> Frase genérica </span>

Ao usar a tag *span*, todos os elementos dentro dela receberão as mesmas propriedades, assim como na div.

#### Em qual caso usar cada uma?

* Use <div> se for atribuir várias propriedades em mais uma tag, a tag <div>pode ser utilizada em textos, tabela, imagens, links etc. para agrupá-las e organizar a estilização destas.
* Use <span> se caso queira atribuir propriedades a texto, como uma única palavra ou frase.

### Conteúdo adicional

#### Disposição do texto – INLINE & BLOCK

##### Elementos BLOCK

* Ocupam toda a largura disponível
* São dispostos um abaixo do outro
* Permite alteração nas suas dimensões (width, height)

##### Elementos INLINE

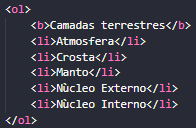
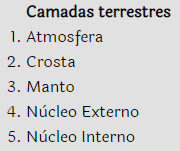
* Ocupam a largura necessária para o elemento
* São dispostos um à direita do outro
* Não permite alteração nas suas dimensões

# Aula 05 - Listas e Tag’s semânticas

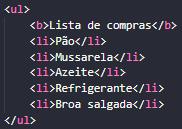
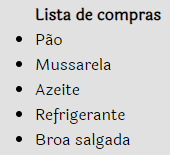
## O que é uma lista para o HTML?

Para o HTML uma lista é uma forma de apresentar mais de um valor de forma linear, dentro dessa lista os valores podem ser apresentados de forma ordenada ou não. Na construção de uma lista dentro do HTML são usadas as tag’s <li>, <ol> e/ou <ul>.

### Lista ordenada

Uma lista ordenada é aquela que é posta de forma sequencial, a **lista ordenada** é identificada pela tag <ol> que é **o**rdered **l**ist.

### Lista não-ordenada

Uma lista ordenada é aquela que é posta de forma sequencial, a **lista não-ordenada** é identificada pela tag <ul> que é **u**nordered **l**ist.

### Estilização de listas

#### list-style

Propriedade que modifica os marcadores laterais da lista, modificando o tipo de numeração/marcação etc.

##### list-style-type: especifica o tipo de marcador do item da lista;

##### list-style-image: especifica uma imagem como marcador da lista;

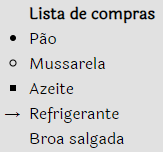
##### list-style-position: especifica a posição dos marcadores da lista;

#### list-style-type

##### <ol> - **o**rdered **l**ist

Podem ser utilizados números decimais (que podem ser precedidos por um 0), letras alfabéticas (maiúsculas ou minúsculas), numerais romanos (maiúsculas ou minúsculas), letras gregas, armênias, georgianas e de outros alfabetos.

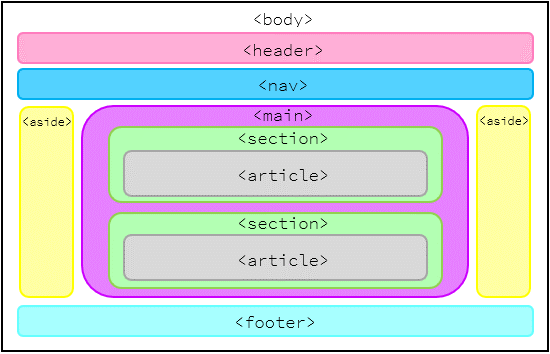
##### <ul> - **u**rdered **l**ist



Podem ser utilizados símbolos como círculos, disco, quadrados e qualquer outros símbolos da tabela ASCII, além de imagens ou até mesmo nada.

## Tag’s semânticas

São tag’s que organizam o código com a finalidade de diminuir a quantidade de div’s, essas tag’s indicam as áreas principais do site determinando para os navegadores a importância do conteúdo. Elas são:



<body>

<section>

<article>

<article>

<section>

<header>

<nav>

<main>

<aside>

<aside>

<footer>

### Tag <header>

Cabeçalho da página com todas as informações principais, como redes sociais e meios de contato.

### Tag <nav>

Dados de navegação da página, também chamado de menu, responsável pelos links que levam para as páginas secundárias.

### Tag <main>

Essa parte contém toda a informação principal da página.

### Tag <section>

Utilizada para fazer grandes separações no código, quando o tema abordado muda.

### Tag <article>

Tag utilizada para organizar os tópicos principais da página.

### Tag <aside>

Nessa área tem-se os elementos à parte do principal, como tópicos de outra página(ainda do site principal) e informações como números de visualização e anúncios.

### Tag <footer>

Rodapé da página, contém outras informações de contato, link do mapa da página e direitos autorais.

# Aula 06 - Estados CSS e animações

## Pseudoclasses

Pseudoclasses são códigos adicionador a determinados seletores para especificar um estado especial desse elemento, podendo acontecer depois de certo tempo ou depois de alguma interação do usuário.

Essas pseudoclasses permitem adicionar estilos aos elementos selecionados dependendo do histórico de navegação, cursor ou até do conteúdo.

### Sintaxe das pseudoclasses

→ valor

link:hover

{

text-decoration: underline;

}

↱

seletor

propriedade

↓

pseudoclasse

↳

### Algumas pseudoclasses

#### :hover

As propriedades dentro das chaves serão aplicadas no momento em que o cursor(ponteiro do mouse) estará sobre o elemento.

#### :focus

As propriedades dentro das chaves serão aplicadas no momento em que o elemento recebe “foco”, como no momento em que ele for clicado.

#### :active

As propriedades dentro das chaves serão aplicadas no momento em que o usuário está clicando no elemento.

#### :link

As propriedades dentro das chaves serão aplicadas enquanto o elemento não for clicado, em seu estado inicial.

#### :visited

As propriedades dentro das chaves serão aplicadas após ter “visitado” o elemento, normalmente aplicado à links.

## Pseudoelementos

Pseudoelementos são utilizados em elementos de forma específica, a fim de adicionar propriedade em decorrência do tempo ou de determinada ação, estilizando de maneira própria.

### Sintaxe dos pseudoelementos

→ valor

link::after

{

text-decoration: underline;

}

pseudoelemento

seletor

↓

propriedade

↱

↳

### Alguns pseudoelementos

#### ::before

As propriedades dentro das chaves serão adicionadas antes do elemento.

#### ::after

As propriedades dentro das chaves serão adicionadas depois do elemento.

#### ::placeholder

As propriedades dentro das chaves serão adicionadas à uma tag input indicando em qual caixa de entrada de informação o valor será inserido.

#### ::selection

As propriedades dentro das chaves serão adicionadas depois de determinada ação, como clicar e arrastar.

## Animações

Recurso natural do CSS que permite que sejam adicionadas transições e animações sem utilizar de outras extensões.

### Transition

Propriedade que permite adicionar uma transição entre dois estados de um determinado elemento. A função do transition é geralmente associada à seletores de estado, como o hover.

#### Algumas propriedades aplicadas ao transition

##### transition-property

Define quais propriedades do elemento com *transition* serão aplicadas a transição

##### transition-duration

Define o tempo da transição

##### transition-timing-function

Define a função de interpolação da transição

##### transition-delay

Define o delay(atraso) até a transição ocorrer

#### Sintaxe da propriedade transition

↱

duração

↳

propriedade CSS

↱

delay

função de interpolação

h3

{

transition: margin-left 2s ease 1s;

}

↳

### Transform

É a propriedade que “transforma” o elemento em que foi aplicado, podendo transpor, rotacionar, aumentar/diminuir etc.

### Animation

Animation é uma propriedade CSS responsável por “animar” textos e imagens(como também outros elementos) usando keyframes(quadros de animação) e seus respectivos parâmetros.

Os quadros de animação(@keyframes) definem o valor da propriedade que variará ao longo da animação, enquanto a animação em si(animation) configura o tempo, ordem e sequência de quadros que serão animados.

#### Algumas propriedades aplicadas ao animation

##### animation-name

É o nome que será dado a animação, que @keyframes utilizará.

##### animation-duration

Duração da animação.

##### animation-delay

Tempo que demorará para a animação iniciar depois do evento(delay).

##### animation-direction

Direção em que a animação ocorrerá.

##### animation-iteration-count

Contagem de quantas vezes a animação será rotacionada.

##### animation-timing-function

Função de interpolação.

##### animation-play-state

Define o estado da animação no momento, como parada ou rodando.

#### Sintaxe da propriedade animation

→ estado inicial

→ estado intermediário

o estado intermediário varia de 1% até 99%

→ estado final

@keyframes animation\_name

{

0%

{

scale: 1;

}

50%

{

scale: 1.5;

}

100%

{

scale: 1;

}

}

nome da animação

↱

função de interpolação

↳

img

{

animation: animation\_name ease;

}

# Aula 07 - JavaScript

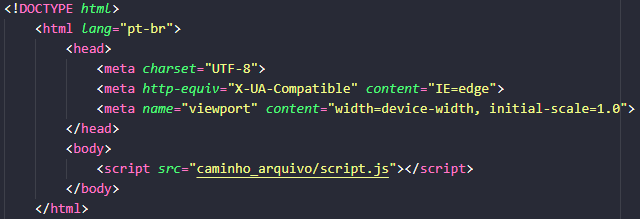
JavaScript(JS) é uma das linguagens que podem ser utilizadas juntamente com HTML com JS é possível mostrar mensagens, informações, fazer verificação e mudar a estética conforme o comportamento da página.

## Implementando JS no código HTML

### Implementação com código interno

Para implementar o JavaScript dentro do HTML sem a adição de um arquivo externo é preciso adicionar a tag <script> dentro da tag principal <body>, entretanto isso não é recomendado, o ideal é separar os arquivos e importá-los para não ferir o princípio de separação das responsabilidades.

### Implementação com código externo

Para implementar o JavaScript dentro do HTML importando um arquivo usa-se o mesmo princípio de importação do CSS, mas dessa vez usando <script scr=“arquivo.js”></script> que pode ser implementado tanto dentro da <head> ou da <body> dependendo apenas se o programador deseja que o código seja executado no início ou no fim da página.

## Variáveis no JavaScript

Variáveis são espaços na memória que são utilizados para armazenar dados. Durante a programação em JavaScript usa-se let para valores que ser alterados e const para valores imutáveis, para atribuir valores é usado o sinal de igualdade, ele é chamado de operador de atribuição.

### Tipos de variáveis

#### Number

São variáveis que armazenam caracteres numéricos, podendo ser:

##### Números inteiros

Dentro do programa podem ser convertidos usando parseInt(int do conjunto dos números inteiros), os números inteiros são todos que vão de –∞ até +∞.

Exemplo:

let nome = -12; const nome2 = 6;

##### Números racionais

Dentro do programa podem ser convertidos usando parseFloat(float de floating-point do conjunto dos números racionais), os números racionais são todos que vão de –∞ até +∞ incluindo números com vírgulas e dízimas que são periódicas.

Exemplo:

let nome = -1.333; const nome2 = 0,125;

#### String

São variáveis que armazenam caracteres textos, sinais e números. Nesse tipo de dado o valor é atribuído usando aspas duplas ou aspas simples.

Exemplo:

let nome = “Abc”; const nome2 = ‘Abc’;

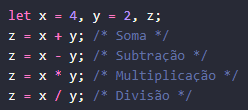
#### Boolean

São variáveis que só tem dois tipos, true(verdadeiro) ou false(falso), podem ser aplicados a operações e condições.

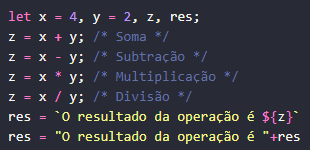
Exemplo:

let nome = false; const nome2 = true;

## Operações em JavaScript

São símbolos que são utilizados como operadores aritméticos, são usados para cálculos matemáticos dentro do JavaScript.

## Interpolação

É a inserção de variáveis em meio ao programa, para fazer a interpolação é necessário usar ` `(acentos graves), colocando a variável com ${var}, ou então usando “ ”(aspas duplas) e adicionando a variável depois com +var.

## Função alert()

A função alert() abre uma caixa de texto na tela, exibindo uma mensagem popup.

## Função console.log()

A função console.log() exibe uma mensagem no console(tela onde é possível adicionar códigos) do navegador.

## Função prompt()

A função prompt() abre uma caixa de texto na tela onde é possível que o usuário insira texto, no caso o que foi solicitado. O prompt também pode receber valores que serão inseridos em alguma variável.

# Aula 08 - Estruturas de decisão e repetição

## Comandos de Decisão

É o algoritmo usado para que ao longo da execução do código ele tome uma decisão conforme os parâmetros que direcionarão o “caminho” que o código tomará.

Exemplo:

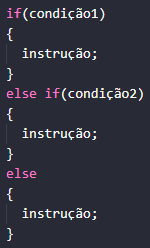
Maria tem 5 reais e está na escola.

Ela pode comprar um lanche, ir embora de ônibus ou não fazer nada.

**Se** ela comprar um lanche, ela vai para casa andando;

**Se não** comprar o lanche ela fica com fome e vai para casa de ônibus;

**Se não fizer nada**, ela guarda seu dinheiro.

Na linguagem da programação isso é escrito em if(se[uma condição]), if else(se senão[tirando a condição principal, uma segunda condição]) e else(se não[em nenhum dos casos]).

No caso de que seja necessário usar mais de uma condição no mesmo if, podem ser utilizados os operadores lógicos && - AND e || - OR, AND é usado quando todas as condições impostas são verdadeiras, OR é usado quando qualquer uma das condições forem verdadeiras.

## Comandos de Repetição

É o algoritmo usado para que ao decorrer do código seja executada uma instrução até que a condição imposta seja verdadeira.

Exemplo:

Maria está na escola com notas baixas e não deseja repetir de ano.

Ela vai continuar estudando **até que** sua média seja maior que 70%.

**Quando** sua média for acima ela parará de estudar e sua mãe lhe comprará um sorvete.

Na linguagem da programação isso pode ser escrito em while(enquanto[a condição for verdadeira]), for(para[determinada condição]).

No caso de que seja necessário usar mais de uma condição, podem ser utilizados os operadores lógicos && - AND e || - OR, AND é usado quando todas as condições impostas são verdadeiras, OR é usado quando qualquer uma das condições forem verdadeiras.

# Aula 09 - Funções

São 6 páginas

# Aula 10 - Integração do JavaScript ao HTML

São 16 páginas

# Aula 11 - Entradas, botões e eventos

São 11 páginas

# Aula 12 – Despedida