**HTML**

**<h1> -- abertura de tag**

**</h1> -- fechamento de tag**

**<img src="foto.png" alt="Exemplo de foto">**

**<img... -- abertura de tag**

**"src= e alt=" -- parâmetro**

**Conteúdo entre "" -- valor**

**Estilo em CSS**

**h1 {**

**font-family: Arial;**

**font-size: 20pt;**

**color: blue;**

**}**

**h1 -- Seletor**

**font-size: 20pt; -- declaração**

**declaração = propriedade + valor**

**font size = propriedade e blue = valor**

**h1 = título principal**

**h2 = secundário**

**h3 = ....**

**até h6**

**Não obrigatoriamente h1 é grande e h6 é pequeno**

**Podemos ter mais de um h1 por página**

**FAVICON - <link> - escolher favicon**

**ctrl + shift + P , abb , Wrap with abbreviation = para colocar uma tag no início e final de um trecho selecionado**

**--- Mais rápido: ctrl+shift+P, Enter, <tag desejada>, Enter**

**tag <pre> mantém a formatação**

**tag <code> muda os caracteres para um mono espaçado**

**Recomendado utilizar a tag <q> (quote) para citações, ao invés de apenas digitar ""**

**tag <blockquote> para citações complexas e maiores**

**- <blockquote cite="link"**

**tag <abbr> para inserir o significado de siglas. Ex: HTML**

**<bdo> para textos invertidos**

**- <bdo dir="rtl"> Texto </bdo>**

**INÚTIL AHAHAHA**

**<ol> ordered list**

**<li> itens da lista**

**</li> opcional a partir do HTML5**

**<ul> unordered list**

**<a> âncora - links externos**

**target="\_blank" - para abrir numa nova aba**

**target="\_self" - para abrir na mesma aba**

**rel="external" - para melhor indexação**

**rel="nofollow" - para links externos onde você não dá um "aval" (propagandas, etc)**

**rel="next" - para página secundária dentro do seu site**

**rel="prev" - para retornar à home**

**Quando for criar vários links ou uma lista, ao invés de colocar a tag várias vezes, é só escrever (tag\*número de vezes) e dar enter.**

**Por exemplo:**

**a\*5 = 5 links**

**li\*10 = 10 elementos da lista.**

**DOWNLOADS**

**<a href="endereço\_do\_arquivo" download="nome\_do\_arquivo"> texto a ser clicado aqui </a>**

**Para baixar direto sem antes abrir o arquivo, incluir também o parâmetro type="ler abaixo"**

**iana.org/assignments/media-types**

**IMAGEM DINÂNICA**

**tag <picture> - possibilia vários src (fontes)**

**Dentro dela usar a <source> (media type) pra cada imagem, da menor pra maior (lembrar de preencher com a extensão em type="image/..."**

**por ultimo usar a tag <img> para a maior imagem**

**Ex:**

**<picture>**

**<source media="(max-width: 750px)" srcset="imagens/foto-p.png" type="image/png">**

**<source media="(max-width: 1050px)" srcset="imagens/foto-m.png" type="image/png">**

**<img src="imagens/foto-g.png" alt="imagem flexivel">**

**</picture>**

**Mudar a ordem pode quebrar a função de imagem dinâmica**

**ÁUDIOS**

**Forma 1:**

**Tag <audio>, "controls" dentro da tag audio**

**- Se quiser que toque automaticamente a iniciar, utilizar "autoplay" dentro da tag <audio>**

**- Se quiser que o audio toque em looping, utilizar "loop" dentro da tag <audio>**

**- Compatível com .mp3 / .wav / .ogg (evitar o wav pois são mais pesados)**

**Forma 2:**

**Entre <audio> e </audio> utilizar a <source src="" type="audio/">**

**--- no caso de mp3 usar o type="audio/mpeg". wav e ogg não mudam**

**Dentro da <audio> usar preload="metadata" controls**

**VÍDEOS**

**Ex1:**

**<video src="video.mp4" width="800" poster="" controls autoplay loop></video>**

**- poster="" para utilizar capa no primeiro frame**

**Ex2:**

**<video width="320" height="240" controls>**

**<source src="movie.mp4" type="video/mp4">**

**<source src="movie.ogg" type="video/ogg">**

**Your browser does not support the video tag.**

**</video>**

**- Compatível com .mp4 / .webm (qualidade piorada) / .ogv**

**Programa HandBrake para conversão**

**CSS**

**INLINE - é feito dentro de cada tag, o que é trabalhoso e não recomendado. Deve ser utilizado em casos e ajustes bem pontuais.**

**Ex:**

**<body style="backgroud-color: black; font-family: arial"**

**<h1 style="color: white;">**

**INTERNOS - é utilizado com a tag <style> dentro da tag <head> e pode ser útil para sites mais simples, com uma ou poucas páginas.**

**Ex:**

**<style>**

**h1 {**

**background-color: navy;**

**}**

**</style>**

**EXTERNOS - o mais prático, utilizado para mudar os estilos de todas as páginas de uma vez. Utilizar a tag <link> (link:css) dentro da tag <head> para criar a página "nomedapagina.css"**

**--- Criar a @charset "UTF-8"; dentro da página de CSS**

**--- Escrever dentro da página de CSS da mesma forma que escreve dentro da tag <style> </style>**

**Colorzilla** para selecionares cores de sites

**@import** deve sempre ficar no início da tag <style> ou no início do documento CSS externo e FORA de um seletor {}

**FONT-FACE**

@font-face {

font-family: "nome que você quiser";

src: url(selecionar a pasta e arquivo);

}

Chrome Web Store

- **Fonts Ninja** - extensão para reconhecer fontes nos sites, tamanho, espaçamento, etc

**FONTES EM IMAGENS**

- WhatFontIs.com

- Fontsquirrel.com

- MyFonts.com

**ALINHAMENTO**

- Será considerado o alinhamento que mais específico (body, h1, etc)

text-indent: 30px; - para início de parágrafos (espaçamento)

**SELETORES**

**# = id**

**. = class**

**: = pseudo-class**

**:: = pseudo-element**

**> = children**

Em HTML é id – em CSS é # (utilizar em apenas um elemento)

Ex: <h1 id=”destaque”> / #destaque { } ou h1#destaque { }

Em HTML é class – em CSS é . (pode utilizar em vários elementos)

Ex: <h1 class=”destaque”> / .destaque { }

Obs: um elemento pode conter configurações de id e de class ao mesmo tempo, mas as do id sobrepõem-se às de class. Ex: <h1 id=”titulo” class=”destaque”>

Também é possível colocar mais de uma class em um mesmo elemento. Ex: <h1 class=”titulo destaque”>

**PSEUDO-CLASSES = :**

Uma ***pseudo-classe*** CSS é uma palavra-chave adicionada a seletores que especifica um estado especial do elemento selecionado. Por exemplo, [:hover](https://developer.mozilla.org/pt-BR/docs/Web/CSS/:hover) pode ser usado para alterar a cor de um botão quando o usuário passar o cursor sobre ele.

Ex:

HTML -- <h1 class=”titulo”> TESTE </h1>

CSS -- .titulo {

background-color: black;

}

.titulo:hover {

Background-color: yellor;

}

:visited – muda aparência de links já visitados/clicados

:active – muda a aparência com a ação do clique do mouse

**PSEUDO-ELEMENTOS em CSS = ::**

Um **pseudo-elemento** CSS é uma palavra-chave adicionada a um seletor que permite que você estilize uma parte específica do elemento selecionado. Por exemplo, o pseudo-elemento [::first-line](https://developer.mozilla.org/pt-BR/docs/Web/CSS/::first-line) aplica o estilo apenas na primeira linha de um parágrafo.

::before – adiciona algo antes do elemento selecionado

::after - adiciona algo após o elemento selecionado

Ex:

h1::before {

content: “»”;

}

h1::after {

content: “«”;

}

**MODELO DE CAIXAS**

Aninhamento – uma caixa dentro de outra

**Height** – altura

**Width** – largura

**Border** – colado no conteúdo

**Padding** – espaço entre border e o conteúdo

**Margin** – espaço para fora do conteúdo, após a borda

**Outline** – uma nova borda, fora do elemento e colado à borda

**Box-level** – quebra o conteúdo, preenche a largura inteira na próxima linha, quebra de novo e retoma o conteúdo. Exemplos: <div>, <h1> ~<h6>, <p>, <main>, <header>...

**Inline-level** – continua na mesma linha onde estava o conteúdo e também retoma o conteúdo na mesma linha. Exemplo: <span>, <a>, <code>, <small>, <strong>, <sup/ sub>...

**FAZER NO INÍCIO DO CSS**

Pra quem não quiser sofrer com erro de margins e paddings excessivos na página, só criar em todo projeto um: \* { margin: 0; padding: 0; border: 0; }

**SHORTHAND de estilização**: largura/espessura estilo cor

Ex: border: 5px solid red

**GROUPING TAGS** (tags de agrupamento)

**Header**

**Main** Todos são vistos como <**div**> mas os navegadores conseguem indexar melhor e dar mais relevância dessa forma

**Footer**

**Aside** - deve ser usada para criar uma seção em uma página web com conteúdo não necessariamente relacionado com o principal, algo totalmente separado do contexto. Um ótimo exemplo da utilização de <aside> é para criar uma barra lateral com propagandas, conhecida como sidebar. Além disso, pode ser usada para biografia do autor, grupos de elementos de navegação externa, pull quotes e outros conteúdos considerados separados do principal da página.

**Section** - A tag section é utilizada para marcar as seções de conteúdo de uma página. Com Esse elemento agrupamos de forma lógica nosso conteúdo, separando a informação em áreas diferentes. O principal objetivo é retirar essa responsabilidade das divs. Tendo como principal diferencial a navegação semântica com HTML 5.

Digamos que na sua página inicial você tenha três chamadas principais: Clientes, Serviços e Notícias. Aí um ótimo lugar para você usar a tag section. Cada uma dessas chamadas podem ser um elemento de seção. A tag pode ser usada em conjunto com o H1, H2, H3, H4, H5 e H6, elementos para indicar a estrutura do documento. Ele pode conter um header e um footer se necessário.

**BORDAS PERSONALIZADAS**

Utilizar normalmente o **border (solid ou transparent)** e em seguida:

Utilizar border-image-source: url(‘nomedoarquivo’);

border-image-slice: “número”;

border-image-repeat: repeat;

**SHORT HAND** – border-image: url(“nomedoarquivo”) 27 repeat;

**PLANEJAMENTO DE SITE**

**Mockflow.com** – criar conta gratuita

Fazer um rascunho, imaginar a diagramação do site que irá criar

Além do main e footer, embrar de utilizar:

Nav (menu) article, aside,

**Variáveis em CSS para facilitar a criação do projeto:**

:root {

(sempre começar com –)

--cor0: “cor”;

--cor1: “cor”;

--cor3: “cor”;

--fonte-padrao: Arial, Verdana, ...;

--fonte-destaque: “Bebas Neue”, cursive;

}

Ex:

h1 {

font-family: var(--fonte-titulo);

color: var(--cor1);

**Git** – repositório local, versionamento de software

**Github** – repositório remoto e também hospedagem

**Push** – enviar atualizações do local para o remoto

**GitHub em vários pcs**

- Fazer login no github desktop

- Depois vai em File > Options > Git- clica nos dois campos e depois em save

- Escolhe qual repositório quer clonar e depois clica em clonar

-

Criei o exercício 022

- DISPLAY: INLINE-BLOCK - Deixa tudo na mesma linha

- background-size: (px, %, etc)

- **contain**: exibirá a imagem toda, mas criando espaços dos lados ou acima/abaixo da tela, como nos filmes, dependendo das dimensões da tela;

- **cover**: exibirá a imagem toda, mas cortando partes da imagem de acordo com o tamanho da viewport;

- 100% (base/largura) 100% (altura): exibirá em todo o espaço selecionado, desconsiderando a proporcionalidade;

- background-repeat: no-repeat, repeat-x (linha), repeat-y (coluna)

- background-position: left top (padrão), left center, right bottom, center center, etc - **PRIMEIRO AS COLUNAS E DEPOIS AS LINHAS**

**ATENÇÃO**: no height utilizar 97vh (98, 99, etc), onde **vh** = a 1/100 da altura da viewport. Então, por exemplo, se a altura do navegador é 900px, 1vh equivale a 9px

**DICA:** Configurar cor de fundo do site para que o usuário já possa consumir o conteúdo antes da página carregar por completo, vídeos, imagens grandes, etc.

background-attachment: scroll (padrão, rola junto), fixed (fixo),

**SHORTHAND - BACKGROUND**: color > image > position >/> size > repeat > attachment

**ATALHO PARA COMENTÁRIOS: Selecione a palavra e depois pressione Shift + Alt + A**

**ALINHAMENTO VERTICAL**

Display: block

Margin: auto

**ALINHAMENTO VERTICAL**

position:

Quando A é pai de B, então colocar relative para o A e absolute para o B

**A = relative**

**B = absolute**

Ao utilizar o absolute num elemento, você libera a utilização dos elementos left e top. Ex:

Left: 200px; (%, etc)

Top: 200px;

display: flex; - Possui diversas funções (procurem pela documentação), uma delas é habilitar o manejamento do conteúdo

justify-content: center; - Alinhamento horizontal, ou seja, dispensa o "margin: auto"

align-items: center; - Alinhamento vertical

**font-variant: small-caps** – todas ficam em forma de caixa alta, mas as iniciais permanecem maiores.

**UTILIZAR MAIS ID JUNTO COM CLASS**

Ex:

<section class=”boximagem” id=”img01”>

<section class=”boximagem” id=”img02”>

.boximagem {

color: #FFF;

}

.boximagem#img01 {

Background-image: url (../imagens/img01.jpg)

}

.boximagem#img02 {

Background-image: url (../imagens/img02.jpg)

}

**TABELAS**

Tabelas ainda são úteis, dependendo da sua utilização. Utilizar tabelas para a criação da estrutura de sites, por exemplo, já caiu em desuso.

Hierarquia de tabelas (simples)

TABLE = tabela = <table>

TABLE ROW = linha de tabela = <tr>

TABLE HEADER = cabeçalho de tabela = <th>

<th **scope**=”row” ou “col”> para definir do que ele é título, da linha ou coluna. No caso de conjunto de linhas ou colunas, **rowgroup** ou **colgroup**.

TABLE DATA = dado de tabela = <td>

OBS: Não é obrigatório o fechamento dessas tags (</tr>, </td>, etc), porém requer cuidado e atenção. Caso não respeite a hierarquia correta, o conteúdo poderá aparecer antes ou depois da tabela.

**Border-collapse**: separete (padrão, um pouco separadas), colapse (juntas).

**Vertical-align**: alinhamento vertical para células da tabela – top, middle, bottom

**<CAPTION>**:utilizar antes da <thead>. É a legenda/título da tabela e aparece logo acima, mas fora da tabela

**EFEITO ZEBRADO**

tbody > tr:nth-child(even/odd/2n/2n-1) {

            background-color: lightgrey;

        }

**CONGELAMENTO DO TÍTULO/CABEÇALHO**

table {

            position: relative;

        }

thead {

            position: sticky;

            top: -1px;

        }

**MESCLAGEM DE CÉLULAS**

Existem expansões em forma de coluna e em forma de linha. Ex:

**Coluna**

<td colspan="2">B</td>

2 = número de colunas que B ocupará

**Linha**

<td rowspan="3">D</td>

3 = número de linhas que D ocupará

**AGRUPAMENTO DE COLUNAS**

É possível criar um grupo de colunas para facilitar a sua personalização em conjunto com **colgroup**. Ex:

<style>

        .cnome {

            background-color: aquamarine;

        }

        .cgrupo {

            background-color: blueviolet;

        }

    </style>

</head>

<body>

    <table>

        <colgroup> - tag para criar os agrupamentos

            <col class="cnome"> - usar um <col> para cada coluna

            <col class="cgrupo" span="3"> - span seleciona outras colunas

        </colgroup>

**TABELAS RESPONSIVAS**

Para que tabelas não criem o scroll em toda a página, é possível criar o scroll apenas para a própria tabela. Ex:

<style>

        #container {

            overflow-x: auto; - x para eixo horizontal e y para o vertical, ou utilizar apenas “overflow”.

        }

    </style>

</head>

<body>

    <div id="container"> - Envelopar a tabela numa div e classificar a div com uma id ou class.

        <table>

**IFRAMES**

Mostra um quadro com uma prévia do site desejado, sendo possível personalizar o quadro.

Recomenda-se utilizar um link do site “por trás”, dentro do iframe, para caso o navegador não suporte a função. Ex:

    <p>Acesando o site do

        <iframe src="https://www.cursoemvideo.com" frameborder="0">

            <a href="https://www.cursoemvideo.com" target="\_blank" rel="external">Curso em Vídeo</a>

        </iframe>

        para aprender a programar.</p>

O tamanho padrão dos iframes é de **300x150**, sendo possível personaliza-los dentro da própria tag ou via CSS\*.

Ex1:

<p>Acesando o site do

        <iframe src="https://www.cursoemvideo.com" frameborder="0" width="300" height="300">

            <a href="https://www.cursoemvideo.com" target="\_blank" rel="external">Curso em Vídeo</a>

        </iframe>

        para aprender a programar.</p>

    <p>O site do

Ex2:

<style>

        iframe {

            width: 200px;

            height: 200px;

        }

    </style>

\*O CSS é soberano e sempre prevalecerá sobre as demais personalizações.

Parâmetro **scrolling=”no”** faz com que suma a barra de rolagem na prévia do site

Para personalizar a borda do iframe, marcar como 0 dentro da tag no html e personalizar normalmente via CSS.