**Domínio:** nome único, pago anualmente, vários TLD’s.

**Hospedagem:** espaço para armazenar arquivos, pago mensalmente, espaço, memória, recurso.

URL’s são compostas por: subdomínio (www), domínio (nome do site), TLD - top level domain (.com) e podem conter caminhos (/algumacoisa).

**Tipos de TLDS**;

**GTLD:** são TLDs genéricos sem indicação de país. Ex: .com, net, .gov, .io, .info, .online, .store

**ccTLD:** são TLDs com designação do país. Ex: .com.br, .edu.us,

**HTML:** é uma linguagem focada em conteúdo: textos, imagens, vídeo, tabelas

**CSS:** é uma linguagem focada em design: cores, sombras, tamanho, posicionamento

**JavaScript:** é uma linguagem focada em interações: menus, animações, pop ups, validações.

**Estrutura básica do HTML**

<!DOCTYPE html>

<html lang =”pt-br”>

<head>

<meta charset=”utf-8”>

<meta name=”viewport” content=”widht=device-widht,initial-scale=1.0”>

<title>Document</title>

</head>

<body>

<h1>Olá, mundo!</h1>

</body>

</html>

Para adicionar comentários no código: <!-- -->

Para adicionar simbos “<” e “>” no parágrafo: &lt; e &gt;

**Utilizando links externos no seu site:** para navegar dentro do seu próprio site, não é recomendado utilizar a tag **target=\_”blank”**, esta, deverá ser utilizada quando o usuário for para outro site, junto com a tag **rel=”external”.**

**CSS:**

**Mudar cor de fundo:** background color

**Mudar cor da fonte:** color

**Harmonia das cores**

**Cores primárias:** amarelo, vermelho, azul

**Cores secundárias:** laranja, violeta, verde

**Cores terciárias:** amarelo-esverdeado, amarelo-alaranjado, vermelho-alaranjado, vermelho-arroxeado, azul-arroxeado, azul-esverdeado

**Paleta de cores para o site:** de 3 a 5 cores

**Cores complementares:** cores que mais se contrastam umas com as outras. O oposto da barra no círculo cromático.

**Cores análogas:** cores imediatamente ao lado entre sua paleta.

**Cores análogas e uma complementar:** paleta de cores vizinha + uma cor complementar a partir da cor principal.

**Cores análogas relacionadas:** paleta principal + uma cor ao lado e pula uma paleta tanto para direita como para esquerda.

**Cores intercaladas:** paleta principal pula a cor vizinha, próxima cor, pula a cor vizinha, escolhe a cor. Tanto para direita como para esquerda.

**Monocromia:** trabalhar com apenas uma paleta de cor, mexendo em sua saturação e brilho, criando um degradê.

**Tipografia**

Antes de serem criadas, todas as letras minúsculas são baseadas na letra x.

Modo safe de incluir fontes no texto do site: Font-family: arial, helvética, sans-serif ou times, xxxx, serif

**Download de fontes externas:** <https://www.dafont.com/pt/>

**Comando para adicionar fonte externa no Vs code:**@font -face { (Usar um @font-face para cada font externa utilizada)

Font-family: ;

Src: url (‘nome da fonte’) format (‘nome do formato’)

font-weight: ;

font-style: ;

}

Tipos de format ()

opentype (otf)

truetype (ttf)

embedded-opentype

truetype-att (apple advanced typography)

svg

**Alinhamento de textos:**

text-align: left, center, right, justify

text-indent: (ajusta o espaçamento do parágrafo)

**Atalhos úteis:**

duplicar trecho do código: shift+alt+seta pra baixo

h2>p\*5.capitulo$ - O símbolo \* representa a quantidade de itens que serão criados. O símbolo . fara a criação de 5 classes e o símbolo $ vai colocar uma ordem numérica nas classes capítulos

**Tipos de caixa**

**Box level:** div, h1-h6, p, main, header, nav, article, aside, footer, form, video

**Inline level:** span, a, code, small, strong, em, sup-sub, label, button, input, select

**Shorthand de border:** boder: (width, style, color);

**Shorthand de padding:** padding: (top, right, bottom, left); – se as quatro medidas forem iguais apenas um valor pode ser declarado. Declarando dois valores, o primeiro será para top e bottom e o segundo para right e left.

**Shorthand de margin:** margin: (top, right, bottom, left);

**Shorthand de outline:** outline: (width, style, color);

**Criando variaveis na CSS**

:root {

--cor0: (código da cor);

--cor1: (código da cor);

}

\* { (remove todas as configurações de margem e padding)

Margin: 0px;

Padding: 0px;

}

**Video responsivel**

div.video {

    background-color: var(--cor5);

    margin: 0px -20px 0px -20px;

    padding: 10px;

    padding-bottom: 58%;

    position: relative;

}

div.video > iframe {

    position: absolute;

    top: 5%;

    left: 5%;

    width: 90%;

    height: 90%;

}

**Configurações de flexbox**

**Display:** flex

**Flex-direction:** row, collun

**Justify-content:** flex-start, flex-end, center, space-between, space-around

**Align-itens:** flex-start, flex-end, center, stretch, baseline

**Flex-wrap:** nowrap, wrap

**Align-content:** flex-start, flex-end, center, space-between, space-around

**Gap:** alinha o espaço entre as linhas e as colunas

**Row-gap:** espaço entre as linhas

**Column-gap:** espaço entre as colunas

**Git-GitHub**

Git: repositório local. Faz versionamento (backup) dos códigos armazenados localmente.

GitHub: repositório remoto. Faz versionamento (push) dos códigos na nuvem.

**Tutorial de como acessar o Github em outro computador**

Menu File > accounts > acessar a opção superior (github.com) > antes de clonar qualquer repositório é necessário ativar o Git > Menu: File/options/git/save > no menu principal (área inicial), escolher qual dos seus repositórios será clonado > escolher o local e selecionar a pasta > trabalhar, comitar o arquivo e fazer push para o Github.

**Para sair e não deixar rastros**

Fechar todas os programas, menos o github desktop > Repository > Remove e mandar a pasta para a lixeira > Menu File > Options > Accounts > Sign out (deslogar seu usuário) > Menu lateral Git > apagar name e email > Salvar > esvaziar a lixeira.

**Aplicando e configurando imagem de fundo**

Body: 100%vh (configuração para visualizar a % da view port)

Background-image: url ( )

Background-size: Ex: (100% 100%)

Background-repeat:

Background-position: center center

Background-size: cover (utilizar quando a imagem não cobrir toda a tela)

**Alinhas conteúdo verticalmente**

Criar uma caixa (container) e uma caixa de (conteúdo)

Aplicar no container o valor de **position: relative** e no conteúdo **position: absolute**

A partir desta configuração, o **conteúdo** passa a ganhar configurações nos valores **Top** e **Left**. Ex: top: 50% ~ left:50%)

Para alinhar verticalmente, aplicar o em **conteúdo** o valor **transform: translate (-50%, -50%)**

Hostnet

Hostinger