**Módulos do curso:**

M01 -> Primeiros passos HTML e CSS -> Conceitos básicos, preparação do ambiente, semântica do HTML 5, textos, títulos, ligações, multidimia, estilos.

M02 -> Deixando as coisas mais bonitas -> Fundamentos de disign, psicologia das cores, tipografia, elementos CSS, modelo de caixas, wireframe, responsividade.

M03 -> colocando um protótipo no ar -> Versionamento de software, hospedagem de sites estáticos, tabelas.

M04 -> aprofundando conhecimentos -> quadros em linha, formulários, media queries mobile first.

M05 -> Novas Tecnologias -> Flexbox, Grid Layout, Projeto final.

**Como Funciona a internet:**

Representação de dados:

O computador como equipamento eletrônico funciona através de sinal, que baseados em ondas elétricas baseados em 0 e 1, são chamados de códigos binários, para que a gente tenha algum dado, precisamos de 8 bits ou declarado como 1 byte -> 1 caractere = letra A. o computador só entende essas ondas binarias ou ondas quadradas, que são linhas retinhas representadas por 0 e 1.

Código multibyte UTF-8:

Mesmo conceito ensinado na aula de medida de armazenamento: múltiplos de 210, essa representação de dados já está clara porque além de estudas já foram ensinadas por mim dentro de sala de aula, parando na medida de petabyte, já que não chegamos nem perto ainda de usar uma medida maior nos dias de hoje.

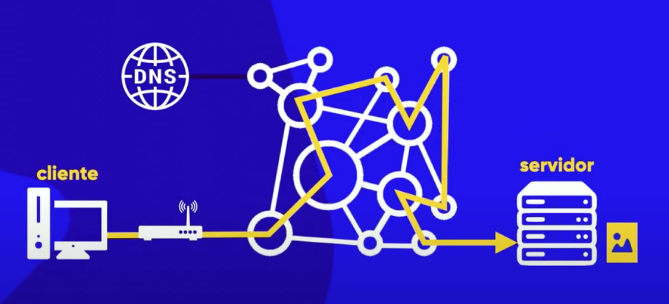
MB: Megabytes -> eles servem para medida de armazenamento

Mb: Megabits -> esse já mais utilizados para transmissão, como uma rede de internet.

Como nos conectamos a internet:

Começamos com o nosso dispositivo que pode ser declarado como cliente, acessamos o site que está alocado na internet, que podemos ter acesso por um modem ou roteador, todo esse conjunto me da o acesso a internet. Para que o acesso funcione bem, temos que entender que o aparelho que dá o acesso à internet, ele usa a modulação (que seria mudar a codificação do aparelho para a internet) e a demodulação (que muda a codificação da internet para o aparelho).

Servidor, essa e base de acesso para todos os arquivos que estão na internet, esse servidor guarda uma numeração do site que está ou que acessar, e também guarda a sua numeração, pois é, quem navega também tem uma numeração.

Servidor DNS: sistema de nome de domínio, são servidores especiais, quando você tenta acessar algo, primeiro ele passa pelo DNS que é quem administra essa transição, depois o DNS organiza e te mando para o caminho que você pediu. O roteamento pode seguir diversos caminhos diferentes, pode pegar um mais rápido se tiver tudo bem, se tiver congestionado ele pode ir por uma rota mais longa.

Está aí uma imagem que ilustra esse caminho 🡪

**Domínio de hospedagem**

Situação 1: Guardado no servidor

Quando um URL é digitado, ele manda a informação para o DNS que vai retornar para o cliente o numero do servidor e depois o seu dispositivo procura no servidor.

Situação 2: Guardado no meu PC

Aqui os arquivos criados ficaram no meu computador, se quero que amigos acessem site oque faço? Eles vão ficar sem ver.

Situação 3: guardado em outro PC

Meu amigo fez um site e quer me mostrar como faço? Não tem como ainda.

O HTML e CSS só funciona se estiver no servidor, a partir do momento que mando para um servidor, crio meu endereço pro site. Sempre que criar um site e quiser ele em funcionamento tem que ter em mente dois nomes:

**Domínio** -> Nome único – Pago anualmente – vários TLDs

**Hospedagem** -> Espaço para armazenar os arquivos – pago mensalmente – espaço, memória, recurso -> á hospedagem é um item delicado que tem de ser escolhido com bastante atenção.

Vamos entender uma **URL** 🡪

Ali ficaram dois exemplos com formatos diferentes. É importante lembra que o **http://** é um protocolo que não faz parte dessa leitura de **URL**.

**Diferença entre HTML, CSS e Java Script**

Como Funcionam a HTML e as CSS?

Não se programa nessas duas formas de criação de site, até porque nenhuma das duas são linguagens de programação, HTML e CSS são usados para desenvolver ou criar sites e nunca programar.

**HTML** 🡪 **H**YPER**T**EXT – **M**ARKUP – **L**ANGUAGE = linguagem de marcação para hipertexto.

**CSS** 🡪 **C**ASCADING – **S**TYLE – **S**HEETS = Folhas de estilo em cascada.

A linguagem **HTML** é a parte do conteúdo de um site, todo o conteúdo de um site é feito por essa linguagem, seja ele um texto, imagem, vídeos e tabelas entre outra, existem muitas possibilidades de conteúdo. A **CSS** tem foco total em design, cores, sombras, tamanhos e posicionamento. O **JavaScritp** que é uma linguagem de programação, pode trabalhar junto com essa outras ferramenta para interatividade do site, para dar mais vida, menus interativos, animações, popups e validações.

O conjunto das 3 linguagens que dão vida para um site bem feito e essenciais para um programador que quer ter domínio nessa área.

**Conteúdo em HTML**

O **<h1>** no inicio é abertura de tag

**<h1>exemplo de título<h1>**

O “exemplo de título é o conteúdo, e o **<h1>** no final fecha a tag.

**<p>Exemplo de parágrafo<p>**

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

A grande maioria das tags tem abertura e fechamento, algumas não tem, como mostra o exemplo da imagem ali em cima. A grande mais dessa tags tem parâmetro e valor. No exemplo **alt=** é um parâmetro e **Exemplo de foto** é um valor.

**Estilo CSS**

Anatomia do css 🡪

Estrutura básica de documento HTML

<!DOCTYPE html>

<html lang=”pt-br”>

<head>

<meta charset= “UTF-8”>

<meta name=”viewport”

Content=”width=device-width,

Initial-scale=1.0”>

<title>Document</title>

<head>

<body>

<h1>Olá, Mundo!</h1>

<body>

</html>

Como funciona a HTML e as CSS?

Depois de colocar a HTML e CSS, quem for fazer acesso, vai receber o dado visual de todas as linhas que foram digitadas no navegador.

Entendendo essa estrutura de HTML que foi digitada:

<!DOCTUPE html>

Serve para o navegador entender que é a linguagem de html mais recente, versão mais moderna.

<html lang=”pt-br>...</html>

Esse é o esqueleto principa da página.

<head>...</head>

Ali não se guarda conteúdo, e sim partes que que são feitas fora do html, links de arquivos de estilo do CSS, scripts (javaScritp) e outros.

<meta charset=”UTF-8”>

Define como as letras e símbolos serão entendidos, o UTF-8 permite acentos, cedilha e caracteres de várias línguas.

<meta name="viewport" content="width=device-width, initial-scale=1.0">

Para design responsivo, mostra o site proporcionalmente.

**Front-end, Back-end e Full Stack**

As tecnologias de HTML, CSS e javaScritp são cliente-side, o servidor ele não usa essa tecnologia, ele apenas de entrega o arquivo, e o seu navegador utiliza essa tecnologia.

Esse é o termo para o front-end, tecnologias que funcionam a partir do cliente, com as tecnologias citadas a cima.

A tecnologias do lado do servidor que são chamadas de server-side, que são chamados de beck-end quem é especializado nessa área. As linguagens Server-side são php, javaScript(a mesma linguagem do front-end, mas com tecnologias diferentes), C#, python, Ruby, java.

Quando essas duas áreas são bem estruturadas no estudo, ou seja quando você domina essas duas áreas temos o desenvolvedor full Stack. É uma área respeitada, mas exige muito estudo, muita pratica entre outros. Guanabara manda o recado de coração que é impossível chegar nesse nível de Full Stack.

**Começando HTML**

Acredito que acostumar com um editor de código, que no caso será usado o visual code, é essencial e fundamenta mental, é como se acostumar com um novo ambiente de trabalho, é preciso olhar e olhar, e manter aquele local/software em mente, até ter clareza do que ele faz.

Na primeira aula de introdução ao Visual code, vi que é uma área de fácil compreensão, criei algumas linhas, parágrafos e separações a mais, só pra não ficar naquela de copia e cola. A compressão total a área de trabalho do editor veio depois de ver vídeo, ler o PDF que foi criado como forma de estudo, e ler livro de introdução ao HTML.

Indo para parte prática 🡪 **<h1>** tag que inicia título como principal**. </h1>** tag que finaliza esse título, essas tags vão de **h1** até **h6**.

**<p>** e **</p>** tags que abrem e fecham parágrafos, dentro dessas tags com utilizar **<br>** para saltar uma linha dentro do site, **&lt;** e essa tag(na verdade não sei se é tag, mas serve para mostra uma kkkk.

**Exemplo** 🡪 tenho que criar um site onde mostrar exatamente a tag **<p>,** e eu não posso simplesmente digitar ela, então o **&lt;** vai ser usado no lugar dos **<>**.

**Vou fazer uma separa para ficar mais fácil:**

* **<h1> -> serve como tag de título**
* **<p> -> serve como tag de parágrafo**
* **<br> -> Serve dentro do parágrafo para pular linha, e não precisa ser fechada**
* **&lt; e &gt; 🡪 serve para mostrar tags dentro do parágrafo no site**

Símbolos e emojis no site

Esses emojis e símbolos são adicionados a partir de código que consigo adquirir em sites da internet, vi que pelo visual code da pra colocar alguns padrões por ele mesmo, mas usando o atalho “WIN +.” é possível adicionar os emojis de forma bem mais pratica.

**Imagens no site**

**Direitos autorais de imagem** 🡪 não é correto pegar imagens da internet e simplesmente colocar no meu site, é algo totalmente ilegal, vamos para fontes onde posso usar, onde tenho o direito de usar no meu site.

Aqui dei uma busca por fora e vi que é mais fácil utilizar sites que tem imagens que te dão a liberdade de usar em seu site, esses sites são:

PEXELS -> https://www.pexels.com/pt-br/

PIXABAY -> https://pixabay.com/pt/

UNSPLASH -> <https://unsplash.com/>

Exemplo de como usar se precisar de uma por site:

📸 Foto por [Nome do autor] via Unsplash  
ou  
Imagem: [Autor] – Pexels

💡 Exemplo real:

Foto por Andrea Piacquadio via Pexels

De forma mais fácil, posso simplesmente pedir para que o ChatGPT crie uma imagem, ou usar minha própria imagem, tudo isso se encaixa em uma realidade de direitos autorais para o site.

**Falando sobre formatos de imagem**:

O Gimp é o programa que vai ser usado para trabalhar com essas imagens que vão ser lançadas nos sites.

Ele é um programa mais limitado, mais é gratuito oque ajuda muito pra esse tipo de aprendizado, já tinha ele instalado então não é meu primeiro contato com o software.

O melhor formado para uma imagem de fundo grande é o JPG.

Uma imagem flutuando por cima com fundo transparente é o PNG.

Geralmente as imagens maiores vem no formato JPG que são imagens JPEG, esse formato de imagem não tem suporte a transparência.

Imagens com fundo transparente, como logos ou personagens sem fundos, esses vêm no formato PNG.

Já entendia sobre os formatos de imagens, mas não tinha essa clareza de tamanho e especificação de cada formato de imagem.

De forma individual posso lembrar dos sites que compactam as imagens, pode ajudar na otimização do site ter um site que carrega mais rápido, mais fluido.