**Guia de tags HTML5**

**Tag inicial:**<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

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

<title>Document</title>

</head>

<body>

</body>

</html>

**Tags :**

Linha para separação: <hr>

Titulos: <h1>, <h2>, <h3>, <h4>, <h5>, <h6>

Parágrafos: <p>

Quebra de linha: <br>  
Sinal de menor e maior (< e >) em HTML: &lt; e &gt;

Símbolos:  
 &reg; ®

    &copy;

    &trade;

  &pound;

  &yen

   &cent;

  &delta;

  &Delta;

  &uarr;

**Curso HTML e CSS Anotações das aulas do curso em vídeo**

**Endereços importantes:**

Cursos - <https://github.com/gustavoguanabara>

Repositório - [https://gustavoguanabara.github.io](https://gustavoguanabara.github.io/)

Materiais PDF - <https://github.com/gustavoguanabara/html-css/tree/master/aulas-pdf>

**Aula 1 – Historia da internet**

História da Internet

Veio da Arpanet, para proteger centros militares americanos durante a guerra fria.

**Protocolo:** forma de comunicação mundial entre as redes de internet. O TCP/IP é o mais utilizado mundialmente hoje.

**HTML –** “HyperText MarkUp Language” – Linguagem de Marcação de Hipertexto.

Criado por Tim Berners-Lee.

**WWW – World Wide Web:** Sub rede dentro da rede de internet especializada em HTTP que é um protocolo de comunicação utilizado para sistemas de informação de hipermídia, distribuídos e colaborativos.

**Aula 2 – Como funciona a internet, domínio e hospedagem.**

Código Binário

**Binary Digit** = Bit

Cada caractere do código binário representa 1 Bit

A cada 8 dígitos temos 1 Byte, ou seja: 01000001 = Byte

Código MultiByte UTF/8 = Representa cada caractere por uma sequência de 1 a 4 Bytes

B Maiúsculo = BYTES, representa armazenamento;

b Minúsculo = BITS Representa transmissão, como a velocidade da internet.

|  |
| --- |
| 8 Bit = 1 Byte |
| 1024 Bytes = 1KiloByte |
| 1024 KiloByte = 1 MegaByte |
| 1024 MegaByte = 1 GigaByte |
| 1024 GigaByte = 1 TeraByte |
| 1024 TeraByte = PetaByte |
| 1024 PetaByte = 1 ExaByte |
| 1024 ExaByte = 1 Zeta Byte |
| 1024 ZetaByte = 1 YotaByte |

Para o site ser acessado é preciso:

**Domínio =** Nome único, ex: Jw. É pago anualmente, e é necessário ter um TLD.

**Hospedagem =** Local onde vai estar armazenado os arquivos, pago mensalmente. Como qualquer hospedagem, deve ser escolhido sabedoria.

Somente os Ips lançados no servidor de rede da internet podem ser acessados por outros computadores.

Ip= Rotulo numérico de identificação que utiliza um protocolo conectado e utilizado na internet, passando pela DNS para então chegar ao cliente pelo nome de domínio.



**DNS =** É como uma “agenda eletrônica” que localiza nomes de domínio desejados pelo cliente, para que ele não precise digitar o Ip (endereço do protocolo de internet) do site diretamente.

**URL =** “Uniform Resourse Locator” Endereço especifico de rede no qual se encontra um recurso informático. O famoso link.

**TLD =** Um dos componentes de endereços da internet, ex: .Org; .Com; .Io, etc.

**Aula 3 – Introdução HTML e CSS**

HTML e CSS não são considerados linguagem de programação.

**HTML – “HyperText MarkUp Language” – Linguagem de Marcação de Hipertexto.**

Focado no conteúdo, sendo: texto, imagens ou vídeos, etc.

**CSS – “Cascating Style Sheets” – Folhas de Estilo em Cascata**.

Focado no design, sendo: cores, sombras, tamanhos, pensionamento, etc.

Adicional, mas importante: JavaScript – Linguagem de programação que trabalha as interações entre o HTML e o CSS, Sendo: Menus, animações, popups, validações, etc. Deve ser estudado separadamente, mas em conjunto com o HTML e o CSS.



Conteúdo HTML

Exemplos de Tag:



Estilo em Css





**Aula 4 – Front-end, Back-End e full Stack**

Front-End = Client-Side: Corresponde à parte em que os usuários conseguem ver e interagir, como cores, fontes, menus, imagens entre outras funcionalidades.

Back-End = Server-Side: resume-se a tudo o que está por trás do site, que o usuário não tem acesso direto e não consegue interagir

Full-Stack = “O Bombril”, que se qualifica em Front-End e Back-End ao mesmo tempo.



**CAPÍTULO 6  
Detalhes sobre imagens...**  
  
**Tipos: JPEG, PNG E GIF  
  
JPEG**  Caso você esteja visando a compactação dos pixels para nãos deixar o arquivo da imagem muito grande.

**PNG**  Caso você queira um fundo transparente e com a qualidade melhor.

**GIF**  Formato compactado, mas não tanto quanto o JPEG.E tem imagem com animação :D

**Favicon :D**

**Sites para conseguir sua Favicon**

**IconArchive:** Ícones prontos.  
**Favicon.cc:** Desenhar seu próprio favcon.  
**Favicon.io:** Outras opções legais, como texto e emojis .