**Conceitos Básicos**

**DNS** (Domain Name System) = Sistema de Nome de Domínio: liga o nome do site (domínio) ao IP do mesmo, ou seja, serve para identificar a qual domínio aquele IP pertence ou a qual IP aquele domínio pertence.

Duas coisas para que as pessoas possam acessar meu site:

* Um **DOMÍNIO**: nome que identifica o site.
  + Cada domínio é **único**, portanto, cada site tem um nome diferente;
  + Pago **anualmente**;
  + Mais barato que a hospedagem.
* Uma **HOSPEDAGEM**: local onde esse site é armazenado
  + Pago **mensalmente**;
  + **Data center:** local físico onde essa hospedagem acontece.

Ex.: **Data Center Ascenty**

**URL**(Uniform Resource Locator) = **Local de Recursos Uniforme**: é um endereço específico de cada site ou parte dele. Partes da URL:

* **Domínio**: junção entre nome e TLD;
* **Sub-domínio**: é o prefixo que acompanha o domínio.

**Ex.:** [***www****.google*.com](http://www.google.com), ***images****.google.com* etc.

* **TLD** (Top Level Domain): são sufixos do domínio.
  + **GTLD (Generic Top Level Domain): TLDs** sem a indicação do país.

Ex.: ***.com*** (comercial), ***.gov***(governamental) etc.

* + **ccTLD(Country Code Top Level Domain):** TLD que identifica o país daquele domínio.

Ex.: ***.com.br***, ***.edu.fr*** etc.

* **Caminho:** local específico dentro de um domínio.
* Protocolo: indica qual o protocolo usado.

Ex.: ***http://*** ou ***https://***



**HTML, CSS e Javascript**

* **HTML e CSS:** não são linguagens de programação;
* **Javascript:** é linguagem de programação.

A melhorar forma que uma pessoa que usa HTML e CSS deve dizer é: ‘**Desenvolvo em HTML e CSS’**.

**Definições**

**HTML (HyperText Markup Language): Linguagem de marcação de Hipertexto =** são textos que levam um ao outro, como se tivessem ligados

**CSS (Cascade Style Sheet):** folhas de estilo em cascata = são ‘folhas’ que modificam a característica uma da outra, como se fosse um efeito acumulativo.

HTML na prática é **Conteúdo**:

* Imagem;
* Texto;
* Vídeo;
* Tabela.
* Etc.

CSS na prática é o **Design**:

* Cores;
* Formatação de texto;
* Formatação de imagem;
* Estilo do site;
* Etc.

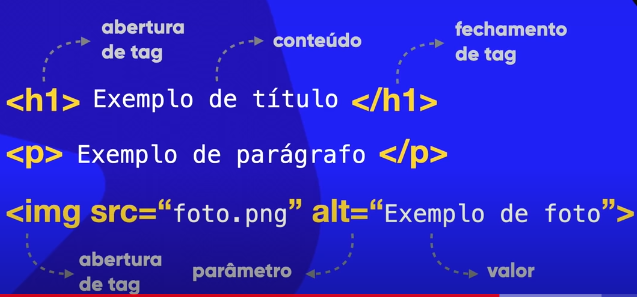
Javascript na prática é **Interatividade**:

* Menus;
* Animações;
* Pop-ups;
* Validações.

**Elementos HTML**

TAG em HTML: é quando algo é botado entre **< >** (colchetes angulares). Para cada conteúdo, temos uma tag diferente.

Ex.: **<h1>** exemplo de texto **</h1>**

* **Tudo entre ‘exemplo de texto’ é o conteúdo;**
* **<h1> e </h1> é o limite da tag: primeiro é a abertura da tag; o último é o fechamento da tag;**
* **Essa tag é para títulos**

Existem algumas tags que não têm fechamento.

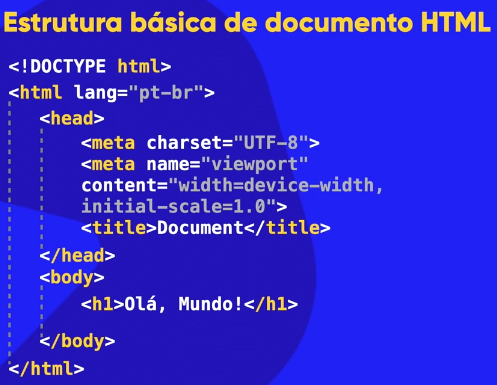
* Essas tags não têm conteúdo, por isso não precisam de fechamento

**O CSS dentro do CSS**

* **Geralmente se encontra dentro da tag styles do HTML;**
* **Seletor serve para modificar algum conteúdo de uma tag do HTML.**

**Ex.: o conteúdo da tag h1**

* **Sempre bote ; depois de cada declaração**

**Principais partes do documento HTML:**

* **<!DOCTYPE html>:** essa estrutura indica que o documento é HTML5;
* **<html>:** é onde ficará todo o conteúdo principal da página, também é chamado de **Root Element**(Elemento Raiz);
* **<title>:** o título da página que aparecerá na aba do navegador;
* **<head>:** onde ficará todo o conteúdo que não será visível para os visitantes da página, como descrição da página, nome do autor, o css utilizado, conjunto de caracteres etc.
* **<body>:** onde ficará todo conteúdo que será visível ao usuário, como textos, imagens, vídeos, áudios, etc.

**Meta tag(<meta>)**

Tags que descrevem o documento HTM, por isso se chamam meta(do grego, si próprio).

Características:

* Sempre ficam dentro de <head>;
* Como ficam em <head>, não serão exibidas na página;

Exemplos:

* **<meta charset=’utf-8’>:** especifica a codificação dos caracteres da página.

**Nota:** Utf-8 é o conjunto de caracteres universal, suportando todos os caracteres de qualquer linguagem, ou seja, essa página terá suporte a qualquer idioma.

**Nota 2 :** alguns navegadores corrigem a codificação automaticamente, como o Chrome, mas é bom declarar qual codificação que a página está usando por via das dúvidas.

* **<meta http-equiv=’refresh’ content=’30’>:** faz a página recarregar a cada 30 segundos.

A maioria dos elementos metas têm os atributos **name** e **content**. Onde:

* **Name:** nome do tipo de meta tag;
* **Content:** o conteúdo daquela meta tag.

Algumas meta-tags desse tipo:

**Ex. 1:** definindo o autor da página

**<meta name=’author’ content=’nome do autor’>**

**Ex. 2:** definindo uma descrição para página

**<meta name=’description’ content=’descrição da página’>**

**Ex. 4:** definindo as palavras-chave para os mecanismos de buscas

**<meta name=’keywords’ content=’HTML, CSS, Javascript’ >**

**Ex. 3:** fazendo a viewport da página ficar boa em qualquer dispositivo.

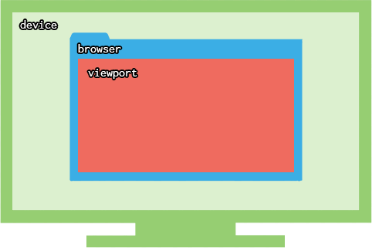
**<meta name=’viewport content=’width=device-width, initial-escale=1.0’>**

**Width=device-width ->** determina que a largura da página vai ficar de acordo com a largura do dispositivo(pc, celular tablet etc)

**Initial-scale=1.0** **->** define o zoom inicial da página. Valores vão de 0 a 5, sendo 1=100%

**Para saber todas as meta tags e suas características, acesse** [**meta tags**](https://www.w3schools.com/tags/tag_meta.asp)

**Viewport e seus Tipos**

**Viewport:** é a área retangular que está sendo exibida na tela. Em WEB, a viewport é o documento HTML que está sendo exibido na janela ou tela (caso esteja em tela cheia).

Existem dois tipos de viewport:

* A viewport visível é chamada de **Visual Viewport**.
* A parte não visível é chamada de **Layout Viewport**. Para visualizar o conteúdo que não está sendo visível, é preciso dar scroll na página.

**Nota:** Quando uma página é redimensionada(dando zoom, por exemplo), a parte que efetivamente sofre redimensionamento é a Visual Viewport.

**Alguns outros termos básicos**

**Front-End ou Cliente-Side:** são linguagens e tecnologias que são executadas na máquina do cliente.

**Ex.:** HTML, CSS e Javascript são executadas na máquina do cliente, pelo navegador, ou seja, tecnologia Front-End.

Um desenvolvedor front-end, em desenvolvimento e sites, está responsável pela parte interativa e visual do site, está preocupado com a experiência do usuário.

**Back-End ou Server-Side:** são linguagens e tecnologias que são executadas no servidor.

**Ex.:** Java, PHP, Python, Node.js, C# etc.

Um desenvolvedor back-end está responsável pelo gerenciamento de dados e com o código em si.

**Full-Stack:** quando envolve a parte de Front-End e de Back-End.