**Nome:** Guilherme Fabiano Terra da Silva

1. **Seção 1**

* Introdução ao conteúdo do curso;
* HTML
* CSS
* JavaScript
* React
* Introdução a a TypeScript

1. **Seção 2**
2. **HTML**

* Não é linguagem de programação;
* É linguagem de marcação;
* Pois não possui if else, loops e outras capacidades padrões das linguagens de programação;
* É o esqueleto / estrutura do site;

1. **CSS**

* É uma linguagem de estilização;
* Faz o polimento do HTML;

1. **IDE usada no curso**

* VS Code;
* Instalação e configuração;

1. **Início HTML**
2. **Anatomia de tags**

* Tags possuem propósitos;
* Tags são envolvidas com <> e fechamento </>;
* Toda tag possui um nome e pode possuir atributos;
* Dentro da tag vai o conteúdo;

1. **Estrutura básica**

* DOCTYPE: Declara versão do HTML;
* Html: Para envolver todo html;
* Head: Para configurações;
* Body: Para elementos visíveis;

1. **Títulos**

* h1 a h6;

1. **Parágrafos**

* p;

1. **Quebra de linha (Não se usa mais)**

* br;

1. **Linha horizontal (Não se usa mais)**

* hr;

1. **Comentários**

* <!--Texto -->;

1. **Links**

* **Tag:** a;
* **Direcionar para link:** Atributo – href=”link”;
* **Abrir em uma nova aba:** Atributo – target=”\_blank”;

1. **Imagens**

* **Tag:** img;
* **Selecionar imagem:** Atributo – src=”local/caminho/link”;

1. **Listas não ordenadas**

* <ul> </ul>;
* Dentro da tag ul se coloca as linhas <li> </li>;

1. **Listas ordenadas**

* <ol> </ol>;
* Dentro da tag ol se coloca as linhas <li> </li>;

1. **Tabela**

* <table> </table>;
* Dentro de table vai tr, e dentro dos tr’s vai um th e vários td’s;
* **Linhas:** <tr> </tr>;
* **Colunas:** <th> </th> e <td> </td>;
* Th é só para primeira linha;

1. **Início CSS**
2. **Como adicionar?**

* Inline (Com atributo style);
* Internal (Na tag head);
* External (Em arquivo externo);

1. **Estrutura padrão**

* Nome da propriedade: Valor;

1. **ID**

* #nomeDoID;
* É único no site, não pode ter mais de 1 coisa com o mesmo id;
* Diferente do class (.nomeDaClasse) que é usada várias vezes;

1. **Cores**

* Os mais comuns são: Hexadecimal, RGB e RGBA;
* Hexadecimal é o padrão;
* Hexadecimal é um código no formato #xxxxxx ou #xxx (X: 0 a F);
* Os valores do RGB vão de 0 a 255 para cada tonalidade;
* Os valores de RGBA são 0 a 255 para RGB e 0 a 1 para A;
* A é a opacidade / nível de visibilidade;

1. **Color e Background-color**

* Regra: cor (em Hex, RGB, RGBA);
* RGBA é útil para deixar apenas um elemento com transárência;
* Um outro jeito de aplicar transparência é usando:
* opacity: (0 a 1);

1. **Auxiliar uso de imagem**

* Background-position;
* Background-size;

1. **Dimensões**

* Width;
* Height;

1. **Padding**

* **Todas as direções:** padding: valor;
* **Direção específica:** padding-direção: valor;
* **Todas com valores diferentes:** padding: topo direita inferior esquerda;
* **Laterais e extremidades sup e inf:** padding: laterais supEinf;

1. **Box-sizing: Border-box**

* Faz com que o tamanho do padding, seja incluso no tamanho máx do width da div;
* Exemplo: Se vc coloca uma div de width: 100px e coloca um padding de 25px, a div final fica com 150px (100px inicial + 25px de cada lado de padding). Com o Box-sizing: Border-box o resultado final é 100px, sendo 50px de padding e 50px para o conteúdo;

1. **Alinhamento de texto**

* Text-align: valor;
* Center, right;

1. **Decoração de textos**

* Text-decoration: underline;
* Text-decoration: line-through;
* EX: texto
* EX: ~~texto~~

1. **Transformação de textos**

* Não se escreve texto do HTML em UPPERCASE;
* O CSS se encarrega de transformar;
* Text-transform: uppercase;
* Text-transform: lowercase;

1. **Espaçamento de letras**

* Letter-spacing: Npx;

1. **Fontes**

* Font-family: nomeDaFonte;
* Sans serif é mais comum;

1. **Estilo da fonte**

* Font-style: nomeDoEstilo;
* Italic, oblique e etc;

1. **Grossura de letras**

* Font-weight: valor;
* Valor vai de 100 a 900;

1. **Tamanho da fonte**

* Font-size: valor;
* Normalmente é 16px;
* Tem múltiplas unidades;

1. **Display**

* Relacionada ao comportamento dos blocos / estruturas;
* Podem ser block ou inline;
* **Block:** Ocupa linha toda;
* **Inline:** Fica um do lado do outro;
* Display: inline-block;
* Display: block;

1. **Escondendo elementos com Display**

* Para esconder basta usar none no Display;
* Display: none;

1. **Posições**

* Relative, fixed, absolute, stick e etc;
* Position: nomeDaPosicao;
* A posição padrão é static;
* **Posição relative:** Move o elemento pela tela. Inicialmente não muda nada, mas depois de usar é possível aplicar top/left/right/bottom: valor; para desloca-lo da posição atual para outra;
* **Posição absolute:** Move o elemento pela tela. Inicialmente não muda nada, mas depois de usar é possível aplicar top/left/right/bottom: valor; para desloca-lo dentro de um container que possui posição relative. Se não existir o container ele leva em consideração o body;
* **Posição fixed:** Trava o elemento na tela, não deixando sair nunca, independentemente do scroll. É obrigatório aplicar o top/left/right/bottom: valor, se não ele nãi fica travado;
* **Posição stick:** É que nem o fixed, mas ele só fica fixo na tela a partir do momento que o scroll passa dele, de forma que ele ficaria para cima do site. Se o stick estiver mais para baixo ele não fica fixo na tela;

1. **Z-Index**

* Determina quem vai ficar na frente de quem, caso ocorra colisão;
* Z-index: valor;
* Não usar valores absurdos, somente o necessário;