ANOTAÇÕES HTML CSS

Modem = aparelho que faz o processo de modulação e demodulação do sinal da internet, ajustando assim a comunicação entre cliente e servidor.

TLD = Top Level Domain (ex:.com, .net) GTLD (Generic Top Level Domain) ex: .gov, .edu; CCTLD (Country Code Top Level Domain) ex: .br, .uk, .us, .tv

Formatos de arquivos de imagens: JPG e PNG os mais comuns. Tamanhos ideais para background: 1920:1080, 1500:1000, 1200:800

Tamanhos para logo: entre 400 e 200 de largura (altura compatível)

Gerando ícones de favoritos

Site: favicon.io ou favicon.cc (se quiser criar manualmente um ícone)

Envelopando parte do código em uma TAG: Atalho Ctrl + Shift + p

Ex: (texto sem tag ou na tag errada) opção wrap with abreviation

**Apenas para o VSCode**: alt+l+o: abre a página com atualização em tempo real

**BUSCAR SEO (Search Engine Optimization) para melhorar a qualidade do site e ser encontrado pelo Google mais facilmente.**

Tag <picture>:

Utiliza-se a foto padrão como inicio com a tag <img src>, acima dessa tag em ordem do menor para o maior ou vice-versa, usa-se a tag <source>, melhor dizendo, source media type. Ex:

<picture>

        <source *media*="(max-width: 750px)" *srcset*="imagens/foto-p.png" *type*="image/png">

        <source *media*="(max-width: 1050px)" *srcset*="imagens/foto-m.png" *type*="image/png">

        <img *src*="imagens/foto-g.png" *alt*="imagem flexivel">

    </picture>

Em CSS vc pode utilizar o @charset “UTF8” caso acentuação esteja bugando, assim como colocado no html.

Tamanho de fontes:

Medidas absolutas

**cm**, **mm**, **in** (polegada), **px**, **pt** (ponto), **pc** (paica). Ponto (pt) e paica (pc) são usados para fontes impressas, pouco ou nunca usados em telas.

Medidas relativas

**em** (medida relativa ao tamanho atual da fonte segundo o tamanho do M maiúsculo), **ex** (relativo à altura x de uma fonte), **rem** (relativo ao root, fonte ligada ao body), **vw** (view width, quer dizer a largura da view port – largura da sua tela, seja qual for – medido em %), **vh** (altura da view port) e **%**.

Recomendações da W3C: usar **px** como absoluto e **em** como relativo, apenas.

**Shorthand** (comando usado para simplificar estilização de comandos (não somente fontes) formado por: font-style, font-weight, font-size e font-family) Ex: font: italic bolder 3em ‘Arial’, sans\_serif;

**Tipos de format (usados em fontes externas)**

Opentype (otf)

Truetype (ttf)

Embbed-opentype

Truetype-aat (Apple Advanced Typography

Sgv

Seletores em CSS

Tudo que em HTML for “id”, em CSS é “#” (sem aspas)

Tudo que em HTML for “class” em CSS é “.”

Pseudo Classes, é representado por “:”. Ex: p:hover {} Sempre está relacionado a uma ID ou Classe.

Pseudo Elementos, é representado por “::”. Ex: p::after {}

Children é representado no css como “>”.

Box em HTML

Height: altura

Width: largura

Border: borda

Padding: preenchimento ou acolchoamento (serve pra abrir espaço interno entre elementos)

Margin: margem (serve para abrir espaço externo entre os elementos)

Outline: Contorno ou traçado (como o nome diz, ele contorna a borda, caso precise de um adorno extra).

**Tipos de box**

Box-level: Sempre inicia uma linha abaixo da box anterior, ocupa sempre a largura total da tela e pula pra próxima linha pra continuar o conteúdo. Ex: a tag <div>

Inline-level: Ele continua na mesma linha do conteúdo e não quebra linha para baixo, ele continua alinhado com o conteúdo na mesma linha. Ocupa apenas o espaço necessário. Ex: a tag <span>

PS: user agent = é o navegador. As configurações padrão do seu navegador.

Configurações de caixa de borda e preenchimento que precisam ser configuradas obrigatoriamente no sentido horário (top, right, bottom e left nessa ordem). Ex border-top, border-right, etc

Para centralizar uma caixa box-level basta configurar o margin: auto.

Na shorthand da box, colocar um único valor vale pra todos os lados. Colocando 2 valores o 1 valor vale para top e bottom e o segundo para right e left.

Nota: usando tag com sinal de filho (>) pode-se fazer uma configuração a parte de outra tag igual sem parentesco. (dica: colocando apenas o comando sem a sentença de filho “>” ele aplica aquele paramento em qualquer nível sendo ‘ex: main p’) Ex:

article > aside{

    background-color: lightgray;

}

aside{

    background-color: red;

Box-shadow (sombras) shorthand com comandos de npx, npx, npx, npx e cor sendo o 1 comando deslocamento horizontal, o 2 comando deslocamento vertical, o 3 embaçamento, o 4 comando espalhamento e por último a cor da sombra. Pra colocar a sombra de forma inversa (dar sensação de profundidade) coloca-se a palavra **inset** antes do comando. (obs: sem virgulas) Ex:

nav{

    background-color: rgb(75, 74, 74);

    box-shadow: 2px 2px 5px 2px black;

}

Colocando seletor personalizado em listas: comando: list-style-type: \’codigo do emoji’;

PS: o código 00A0 é um espaço para melhorar o visual e descolar o símbolo da frase.

TABELAS EM HTML

“ver exercício 023”

Tr > linha da tabela

Th > usado para títulos com semântica

Td > usado para dados comuns

Conceito de escopo (scope) para dar mais semântica à tabela, diga-se: scope:col para coluna, scope:row para linha. Aplicáveis em TH apenas. Exsitem 4 tipos de escopo:

Row: para linhas

Col: para colunas

Rowgroup: para conjunto de linhas

Colgroup: para conjunto de colunas

MESCLAGEM DE TABELAS

Colspan e rowspan São os comandos de expansão de colunas e linhas da tabela, determinando quantos espaços aquela linha ou coluna deve ocupar. Ver vídeo 3 modulo, cap 21, aula 08 para retirar dúvidas.

DESTAQUE DE COLUNAS EM TABELAS

Utiliza-se “colgroup” como linha de comando dentro da tabela, identificando todas as colunas existentes na tabela, em seguida cria-se um estilo sendo class ou ID para aquela que deseja destacar. Ex:

<table>

<colgroup>

<col class=“coluna 1”>

<col class=“coluna 2”>

<col class=“coluna 3”>

<col class=“coluna 4”>

<colgroup>

<table>

Em seguida, cria-se o style para a classe (ou id) identificada. **Span** pode ser utilizado para expandir o destaque.

Responsividade de tabelas, basta colocar a tabela em uma **div**, em seguida usar um estilo de **overflow, x para horizontal e y para vertical.**

**IFRAMES**

Iframes são usados como pequenas janelas inline no seu código, mostrando algum conteúdo que interesse naquele momento. Nem todo navegador reconhece o iframe de forma eficaz, então dica é colocar uma ancora (a href) dentro do iframe para caso aconteça.

O tamanho padrão de um iframe é 300w x 150h (w de width e h de height, largura e altura respectivamente). Pode ser configurado normalmente em CSS. Existe um parâmetro chamado scolling, (opções de auto, yes e no) que permite ou não rolar o conteúdo (caso seja grande o suficiente) dentro do iframe, mas navegadores ainda estão se adaptando a esse comando e pode acontecer de não funcionar em determinadas situações devido ao navegador em si. Ex:

<iframe *src*="https://carlysonpatrick.github.io" *frameborder*="0" *scolling*="no">

</iframe>

Formularios em HTML

<form> e </form>. Utilizar sempre a tag label para linkar um campo de preenchimento com sua descrição, é obrigatório. Existem 2 métodos de envio de formulários o método GET e o POST.

Metodo GET: comporta apenas 3 mil bytes (3 mil caracteres), usada para formulários com informações não criticas ou secretas. Não comporta envio de fotos.

Metodo POST: O oposto do GET. Campos sensíveis (senhas, etc.)

Comandos interessantes em form:

Required – para campos obrigatórios

Min e maxlength – tamanho mínimo e máximo de caracteres definidos

Size – determina o tamanho da caixa do form para caber aquele numero de caracteres e cria uma rolagem lateral

Placeholder – mostra caracteres dentro da caixa para preencher (tipo nomes)

Autocomplete: on e off, liga e desliga a opção de auto completar.

Revisar capitulo 24 episodio 10 que contem exemplo de JS. Ex 025 form 010.