Minicurso FrontEnd

Modelo de slide: por Chip Heath e Dan Heath / Google presentation

Como será o minicurso?

Será um minicurso prático, desenvolveremos uma página em HTML.

O objetivo do minicurso será ensinar o básico html, css e js - para construção de páginas web e alguns conceitos interessantes

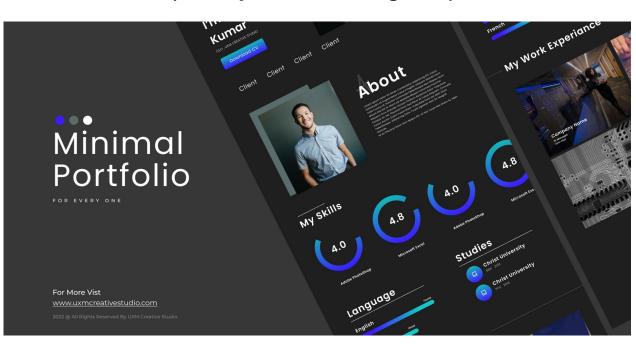


Como será o minicurso?

A página que desenvolvemos será a reprodução de um design disponível no

Figma Community.

<u>Link</u>





1. Vamos começar

Neste Link você encontrará a estruturação de pastas inicial que precisaremos.

Components

Nesta pasta, estão os Web Components da aplicação.

→ Styles

Nesta pasta, estão os estilos utilizados na main.

→ Assets

Nesta pasta, estão as imagens utilizadas.

Começaremos criando o arquivo **css global** do nosso site. Nele adicionaremos margins, paddings, variáveis do tema...



Dica

margins: espaçamentos externos ao elemento.

paddings: espaçamentos internos ao elemento.

variáveis CSS: criação prévia de, por exemplo, cores que serão usadas repetidamente no estilo. Cor primária, secundária do sistema..

Estilizaremos componentes e criaremos classes que possuem um padrão na página

Precisamos carregar as fontes e suas variações utilizadas. <u>Google Fonts</u>



Dica

Usaremos Poppins@300; Poppins; Poppins@600; Questrial.

<u>Link</u>



Dica de Unidades de CSS

em: proporção do tamanho informado pelo elemento-pai para font-size.

rem: proporção do tamanho informado pelo elemento-raí para font-size.

vh: proporção da altura do viewport..

vw: proporção da largura do viewport..

vmin: proporção do menor valor entre altura e largura do viewport.

vmax: proporção do menor valor entre altura e largura do viewport..

ex: proporção da altura de um caractere minúsculo da fonte utilizada..

ch: proporção para unidade de caractere / para o tamanho de uma string.

px: proporção para unidade de pixel..

fr: proporção da fração dos espaço disponível no container de grid.

https://desenvolvimentoparaweb.com/css/unidades-css-rem-vh-vw-vmin-vmax-ex-ch/



Dica de Propriedades CSS - Utilizadas

display: tipo de renderização / como os elementos filhos serão organizados na caixa.

flex-direction: a direção dos itens flexíveis, distribuição em coluna ou linha, por exemplo. Para display: flex.

justify-content: organização dos itens em linha/coluna, de acordo com flex-direction..

align-items: organização dos itens em coluna/linha, contrário a flex-direction.

grid-template-columns: número e largura das colunas. Para display: grid.

grid-template-rows: número e largura das linhas. Para display: grid.

gap/column-gap/row-gap: espaço entre linhas e colunas / espaço entre colunas / espaço entre linhas, respectivamente..

z-index: ordem na pilha de elementos. Maior z-index, mais a frente.

font-size: tamanho de uma fonte.

line-height: tamanho de cada linha de um parágrafo.

color: cor de uma fonte.



Dica de Propriedades CSS - Utilizadas

font-weight: Peso de uma fonte.

text-decoration: A decoração de um texto

border-radius: Arredondamento das bordas do elemento.

background: Propriedades do plano de fundo. Se declarado apenas um valor, se comporta como

background-color, define a cor de fundo.

width: Largura de um elemento.

overflow/overflow-x/overflow-y: o que deve acontecer se o conteúdo ultrapassar o tamanho da caixa toda / na horizontal / na vertical, respectivamente.

white-space: como o espaço em branco em um elemento é tratado.

height: Altura de um elemento.

border: Propriedades para a borda de um elemento, largura, estilo, cor..

font-family: qual fonte deverá ser usada.

position: o tipo de método de posicionamento usado para um elemento (static, relative, absolute, fixed, or sticky).



Dica de Propriedades CSS - Utilizados

::-webkit-scrollbar: Pseudoelemento de toda a barra de rolagem.

::-webkit-scrollbar-thumb: Pseudoelemento para a barra de rolagem arrastável.

:corner-present: Pseudo-classe, indica se as bordas da barra de rolagem estão presentes.

:horizontal: Pseudo-classe, qualquer item da barra de rolagem horizontal.

:single-button: Pseudo-classe, único botão no início ou final da barra de rolagem.

:increment: Pseudo-classe, botões ou barra de incremento, rolagem para a direita/rolagem para cima.

:decrement: Pseudo-classe, botões ou barra de decremento, rolagem para a esquerda/rolagem para baixo.

::before: Seletor, insere algo antes do conteúdo de cada elemento(s) selecionado(s).

::after: Seletor, insere algo depois do conteúdo de cada elemento(s) selecionado(s).

content: Seletor, insere algo depois do conteúdo de cada elemento(s) selecionado(s).

grid-template-areas: Organização das áreas, possivelmente nomeadas com grid-area, na caixa..

repeat(): Loop para organizar os itens na grid. Declaração: (quantidade por linha/coluna, tamanho).

https://www.w3schools.com/cssref/



Dica de Propriedades CSS - Utilizados

min-height/max-height: Tamanho mínimo e máximo da altura do elemento..

min-width/max-width: Tamanho mínimo e máximo da largurado elemento..

justify-items: O padrão para justify-selft de todos os itens da caixa. Justify-self define como a caixa é justificada/distribuída no container.

@keyframes: Regras para a animação de acordo com a porcentagem de mudança.

clip: Recorta um elemento em cima, direita, baixo, esquerda.

transform: Aplica uma transformação 2D ou 3D a um elemento.

animation: Propriedades de uma animação: nome, duração, temporizador, delay, iteração, direção, preenchimento, etsdao.

Utilizaremos WebComponents para alguns elementos. Leia mais sobre isso aqui:



Dica

Elementos personalizados: APIs JavaScript fornecidas pelo navegador que permitem conectar código (normalmente um JavaScriptclass) com elementos declarativos personalizados na marcação (por exemplo <my-custom-element>,).

Shadow DOM: Outro conjunto de APIs JavaScript para incorporar uma árvore DOM "sombra" que é renderizada separadamente do DOM principal. Normalmente, você criará uma instância do Shadow DOM para cada elemento personalizado que permitirá definir o escopo de estilos e nós DOM por elemento. Isso é crucial para encapsular elementos, pois evita colisões de estilo e de referência DOM.

A maioria das APIs disponíveis para interagir com componentes HTML personalizados estão disponíveis por meio da abstração **CustomElementRegistry**

Utilizaremos **WebComponents** para alguns elementos. Leia mais sobre isso aqui:



Base para um Elemento Customizável

```
class ClassName extends HTMLElement {
  constructor() {
    super();
    const shadowRoot = this.attachShadow({ mode: "open" });
    shadowRoot.appendChild(YourContainer);
}
customElements.define("my-component", ClassName );
```

Esse arquivo deve ser importado no final da página HTML index;



Para praticar!

Desenvolva a responsividade da página seguindo este guia: <u>link</u>

Desenvolva este outro portifólio: <u>link</u>

Explore desafios em <u>DevChallenge</u>

Hospede sua aplicação estática em Surge.sh

Compartilhe seu progresso nas redes







