As imagens para realização do exercício estão no Teams, aula Site Responsivo.

Vamos separar a programação em dois blocos. Um bloco será específico para layouts de desktops. Outro para dispositivos móveis.

Uma página responsiva deve ter alguns blocos, que vão definir, podemos dizer, sua flexibilidade quanto ao ambiente que está sendo aberto

Observe que na montagem do cabeçalho do html:

<!DOCTYPE html>

<html>

<head>

    <title>Projeto Loja de Informática</title>

    <link rel="stylesheet" type="text/css" href="responsividade.css">

      <meta name="viewport" content="width=device-width, initial-scale=1">

</head>

A tag meta name define a configuração, onde será ajustada a escala do ambiente que se esteja trabalhando.

Estilos para Desktop

header {

  background-color: #05763b;

  color: #fff;

  padding: 20px;

  display: flex;

  justify-content: space-between;

  align-items: center;

}

.faixa {

  background-color: #115b03;

  color: #fff;

  padding: 5px;

}

O elemento header define o fundo como uma cor escura, a cor do texto como branco, o padding (preenchimento) como 20 pixels e define o display como flex, com o conteúdo justificado com espaço entre e alinhado verticalmente ao centro.

Foram introduzidas faixas acima e abaixo do cabeçalho da página, para dar o efeito abaixo:

Texto

Descrição gerada automaticamente

Vamos fazer uma alteração no código, para que a frase fique responsiva, quando alterarmos o tamanho da tela

Para isto, vamos mudar o tamanho da fonte de px para em e alterar o parâmetro px para vw (viewport width).

Traduzir este recurso é meio complicado. Pense que é a visão que você tem, do tamanho aplicado, em relação ao tamanho do navegador. Portanto, quando reduzir o campo de visão, a fonte também será ajustada.

Modifique o código para:

header {

  background-color: #05763b;

  color: #fff;

  padding: 20px;

  display: flex;

  justify-content: space-between;

  align-items: center;

  font-size: 1.2vw;

}

Modifique somente o valor deste ponto do projeto e teste no navegador

Resultado sendo positivo, daqui por diante você ajustará seu projeto, onde houver necessidade.

Use com atenção pois chega um ponto que não é possível ler o conteúdo.

Tags ul e li

Recordando: a tag ul é utilizada para controlar todos os elementos, de uma lista não ordenada.

A tag li é usada para criar uma lista de itens.

nav ul {

  display: flex;

}

nav li {

  margin-right: 20px;

  font-size: 1.2vw;

}

nav li:last-child {

  margin-right: 0;

}

nav a {

  color: #fff;

  text-decoration: none;

}

nav ul { display: flex; }: Esta regra define que a lista dentro da tag Nav deve ser um display tipo flex. Isso permitirá que os itens dentro da lista (elementos filhos) sejam dispostos horizontalmente um ao lado do outro em uma única linha.

nav li { margin-right: 1.2vw; }: Esta regra define que os itens dentro da lista (li) devem ter uma margem à direita de 1.2 em, ajustando conforme a tela for reduzida.

nav li:last-child { margin-right: 0; }: Esta regra devine que o último item dentro da lista deve ter uma margem à direita de 0. Isso garante que não haverá espaço extra à direita do último item da lista.

nav a { color: #fff; text-decoration: none; }: Esta regra define a aparência dos links contidos dentro da lista. A regra define que o texto dos links será branco (#FFF) e que não terá decoração de texto (como sublinhado).

**Exercício**

Crie um cabeçalho simples, que contenha os elementos trabalhos e que tenha como resposta a configuração abaixo:



Resposta da codificação

HTML

<!DOCTYPE html>

<html lang="pt-br">

<head>

  <meta charset="utf-8">

    <title>Projeto Loja de Informática</title>

    <link rel="stylesheet" type="text/css" href="responsividade.css">

      <meta name="viewport" content="width=device-width, initial-scale=1">

</head>

     <div class ="faixa"> </div>

    <body>

      <header>

        <h1>AllinOne - Tudo em Tecnologia para você</h1>

        <nav>

          <ul>

            <li><a href="#">Área Técnica</a></li>

            <li><a href="#">Artigos</a></li>

            <li><a href="#">Dúvidas</a></li>

          </ul>

        </nav>

      </header>

      <div class ="faixa"> </div>

</body>

</html>

CSS

/\* Definição do cabeçalho \*/

header {

  background-color: #05763b;

  color: #fff;

  padding: 20px;

  display: flex;

  justify-content: space-between;

  align-items: center;

  font-size: 1.2vw;

}

.faixa {

  background-color: #115b03;

  color: #fff;

  padding: 5px;

}

nav ul {

  display: flex;

}

nav li {

  margin-right: 20px;

  font-size: 1.2vw;

}

nav li:last-child {

  margin-right: 0;

}

nav a {

  color: #fff;

  text-decoration: none;

}

/\* fim da montagem do cabeçalho \*/

Na codificação acima, os textos seriam reduzidos ao mínimo, tornando difícil algumas leituras. Precisamos deixar eles responsivos, mas que reduzam até um limite

Altere os códigos para:

header {

  background-color: #05763b;

  color: #fff;

  padding: 20px;

  display: flex;

  justify-content: space-between;

  align-items: center;

  font-size: 1.2vw;

  font-size: min(16px, 2vw);

}

av li {

  margin-right: 20px;

  font-size: 1.2vw;

    font-size: 1.2vw;

    font-size: min(16px, 2vw);

}

**font-size: 1.2vw;**

Recordando o que foi tratado no início do documento, a unidade vw (viewport width) (largura da viewport) se ajusta automaticamente com o tamanho da tela do dispositivo. O tamanho da fonte será 1.2% do tamanho da largura da viewport.

**font-size: min(16px, 2vw);**

O código define um tamanho mínimo e máximo para a fonte. A função min() do CSS seleciona o valor mínimo entre os dois valores que são passados como argumentos.

Nesse caso, o valor mínimo será 16px ou 2vw, o que for menor. Isso significa que, se o tamanho da tela for muito pequeno, o tamanho da fonte será limitado a 16px para garantir que o texto permaneça legível.

Caso contrário, o tamanho da fonte será responsivo e se ajustará de acordo com a largura da viewport, até um máximo de 2vw.

As duas linhas são combinada (responsiva e adaptável) com uma limitação de tamanho mínimo para garantir que o texto permaneça legível em todos os tamanhos de tela.

css

Copy code

main {

max-width: 800px;

margin: 0 auto;

padding: 20px;

}

Esta regra CSS é para o elemento main, que é o conteúdo principal da página. Ela define uma largura máxima de 800 pixels, centraliza-o horizontalmente usando margem automática (margin: 0 auto;) e define o preenchimento como 20 pixels.

css

Copy code

/\* Estilos para dispositivos móveis \*/

@media (max-width: 768px) {

header {

flex-direction: column;

}

nav ul {

flex-direction: column;

margin: 0;

padding: 0;

}

nav li {

margin-right: 0;

margin-bottom: 10px;

}

main {

padding: 10px;

}

}

Este é um conjunto de regras CSS dentro de uma media query (@media (max-width: 768px)), que define estilos para dispositivos móveis. Quando a largura da tela é de no máximo 768 pixels, essas regras são aplicadas.

Dentro dessa media query, temos regras que modificam o layout da página, para torná-lo mais adequado para dispositivos móveis. O header passa a ter direção de coluna (flex-direction: column;), e o nav ul também. Os itens da lista (nav li) perdem a margem à direita e ganham uma margem inferior de 10 pixels. O main perde um pouco de padding para economizar espaço.

Espero que isso esclareça cada parte do código CSS que você postou!