

HTML 5 & CSS 3

Módulo 5: Introdução a Layouts Responsivos



FORMULA
<WEB>
PRO-MAX

Introdução ao Módulo de Layouts Responsivo

Layouts responsivos em CSS são um aspecto essencial do desenvolvimento web, permitindo que sites se adaptem e sejam exibidos corretamente em vários dispositivos e tamanhos de tela. Essa abordagem garante que os usuários tenham uma experiência consistente e amigável, independentemente de acessarem o site em um computador, tablet ou dispositivo móvel. A teoria principal por trás dos layouts responsivos inclui consultas de mídia, unidades flexíveis e grades fluidas.

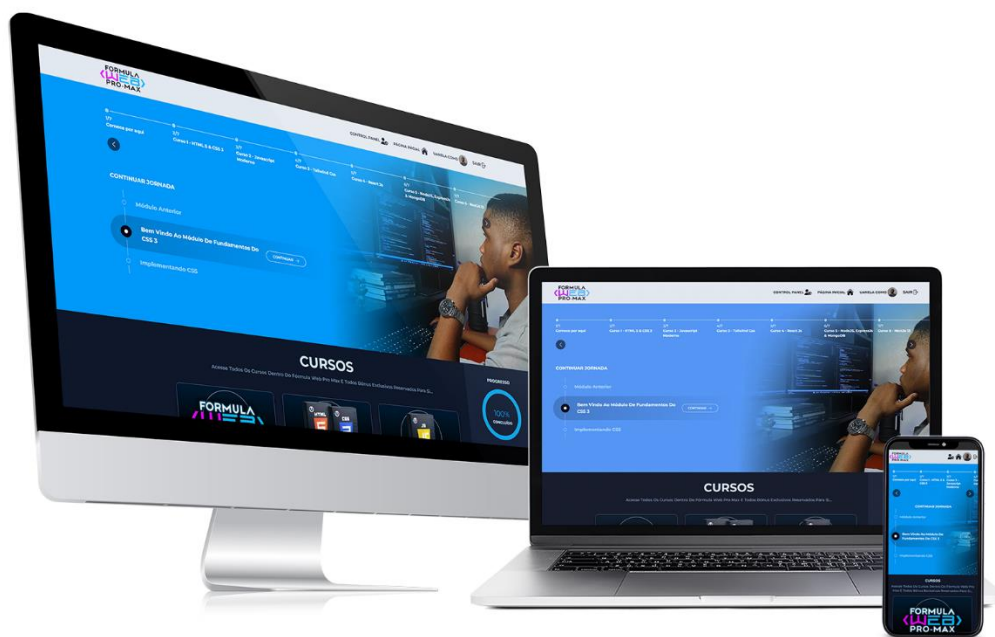


Figura 1: Exemplo de Layouts Responsivo (Portal Fórmula Web Pro Max)

Consultas de Mídia (media-queries):

Consultas de mídia são a base do design responsivo em CSS. Elas permitem aplicar diferentes estilos de CSS com base nas características do dispositivo do usuário, como tamanho da tela, resolução, orientação e outros. As consultas de mídia utilizam a regra `@media` e seguem a sintaxe:

```
@media (característica-de-mídia) {
```

```
/* Regras de CSS para a característica de mídia especificada */
```

```
}
```

Por exemplo, para direcionar dispositivos com uma largura máxima de 768px (comumente usada para tablets e dispositivos móveis), você pode usar a seguinte consulta de mídia:

```
@media (max-width: 768px) {  
    /* Regras de CSS para dispositivos com largura máxima de 768px ou menos */  
}
```

Grades Fluidas (fluid grids):

Um sistema de grade fluida é uma estrutura de layout que usa unidades relativas, como porcentagens, para definir as larguras das colunas em vez de valores fixos em pixels. Isso permite que o conteúdo se ajuste automaticamente de acordo com o tamanho da tela. O conceito é dividir a página em uma grade de colunas e usar porcentagens para determinar suas larguras.

Por exemplo, um layout de duas colunas pode ser definido da seguinte forma:

```
.container {  
    width: 100%; /* Garante que o contêiner ocupe toda a largura disponível */  
}  
  
.column {  
    float: left;  
    width: 50%; /* Cada coluna ocupa 50% da largura do contêiner */  
}
```

Unidades Flexíveis:

Para alcançar a responsividade, é crucial usar unidades flexíveis para dimensionar elementos. Duas unidades comumente usadas são em e rem. Essas unidades se adaptam ao tamanho da fonte do elemento pai ou do elemento raiz (rem). Ao usar essas unidades, os elementos podem se adaptar a mudanças no tamanho da fonte ou na largura da janela de visualização.

em: Representa o tamanho da fonte do elemento em relação ao tamanho da fonte do elemento pai. Por exemplo, se um elemento tem tamanho de fonte de 1.5em, ele terá 1,5 vezes o tamanho da fonte do elemento pai.

rem: Representa o tamanho da fonte do elemento raiz (html). Ele permite criar uma referência de dimensionamento consistente em todo o documento.

Mobile First Design

O design "mobile first" (primeiro para dispositivos móveis) é uma abordagem de desenvolvimento web onde o foco principal é criar a versão do site ou aplicativo para dispositivos móveis primeiro e, em seguida, expandir e adaptar o layout para dispositivos maiores, como tablets e desktops. Essa abordagem coloca a experiência do usuário em dispositivos móveis como prioridade, considerando a crescente quantidade de usuários que acessam a internet através de smartphones.

Meta Tag Viewport:

Para garantir a responsividade em dispositivos móveis, você deve incluir a meta tag viewport em seu HTML. Essa tag informa ao navegador como controlar as dimensões e a escala da página em vários dispositivos.

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

O width=device-width garante que a largura da página seja definida como a largura da tela do dispositivo, e o initial-scale=1.0 define o nível de zoom inicial como 100%.

Ao combinar consultas de mídia, grades fluidas, unidades flexíveis, mobile first design e a meta tag viewport, você pode criar layouts responsivos que se adaptam suavemente a diferentes tamanhos de tela e dispositivos, proporcionando uma experiência de usuário ideal em uma ampla variedade de plataformas.

Aprendendo Media Queries Do Zero

Consultas de mídia são uma parte fundamental do design responsivo em CSS, pois permitem aplicar estilos diferentes com base nas características do dispositivo. Elas são usadas para adaptar o layout e o design do site para diferentes tamanhos de tela e tipos de dispositivos. As consultas de mídia usam a regra @media e são escritas em CSS para definir diferentes estilos com base em condições específicas.

Exemplo 1: Consulta de mídia para dispositivos móveis com largura máxima de 768px.

```
/* Estilos padrão para todos os dispositivos */
```

```
body {  
    font-size: 16px;  
    color: #333;  
}
```

```
/* Estilos específicos para dispositivos móveis com largura máxima de 768px */
```

```
@media (max-width: 768px) {  
    body {  
        font-size: 14px;  
        color: #555;  
    }  
}
```

Neste exemplo, a consulta de mídia (max-width: 768px) é usada para aplicar estilos diferentes apenas quando a largura da tela é igual ou inferior a 768 pixels. Quando a largura da tela é maior que 768 pixels, os estilos definidos fora da consulta de mídia são aplicados. No entanto, quando a largura da tela é 768 pixels ou menos, os estilos dentro da consulta de mídia são aplicados, alterando o tamanho da fonte e a cor do texto.

Esse é apenas um exemplo de como as consultas de mídia podem ser usadas para tornar um site responsivo e adaptável a diferentes dispositivos. Com a utilização adequada de consultas de mídia, é possível criar layouts e designs versáteis que proporcionem uma experiência consistente e amigável para os usuários, independentemente do dispositivo que estão usando para acessar o site.

Alguns pontos de interrupção comuns que os desenvolvedores costumam usar em consultas de mídia para criar designs responsivos para vários dispositivos:

Dispositivos móveis:

320px: Geralmente usado para dispositivos móveis pequenos e smartphones mais antigos.

360px: Ponto de interrupção comum para smartphones modernos com tela um pouco maior.

414px: Usado para smartphones maiores e alguns phablets mais antigos.

Tablets:

768px: Ponto de interrupção comum para tablets no modo retrato.

1024px: Ponto de interrupção comum para tablets no modo paisagem.

Desktops e laptops:

1280px: Frequentemente usado para laptops pequenos e monitores de desktop.

1440px: Ponto de interrupção para monitores de desktop maiores.

1920px: Outro ponto de interrupção comum para telas de desktop grandes.

Lembrando que esses tamanhos não são padrões fixos, e os desenvolvedores podem ajustá-los com base em seus requisitos de design específicos. Além disso, com a constante evolução da web e da tecnologia, é essencial considerar uma variedade maior de pontos de interrupção e adotar uma abordagem "mobile-first" ao projetar layouts responsivos.

Conforme a web e a tecnologia continuam a evoluir, os desenvolvedores podem precisar adaptar os pontos de interrupção das consultas de mídia para acomodar novos dispositivos e resoluções de tela. Sempre teste seus designs em vários dispositivos e tamanhos de tela para garantir uma experiência do usuário ideal em diferentes plataformas.

Unidades Em & Rem

As unidades rem e em são unidades de medida em CSS que se baseiam no tamanho da fonte de elementos específicos para determinar o tamanho de outros elementos na página. Elas são úteis para criar layouts responsivos e escaláveis. Vamos explicar cada uma delas:

Unidade em:

A unidade em representa o tamanho da fonte de um elemento em relação ao tamanho da fonte do elemento pai. Seu valor é relativo à fonte do elemento pai, permitindo que os elementos se dimensionem de acordo com o contexto. Por exemplo, se um elemento tem o valor 1em para sua propriedade de font-size, ele terá o mesmo tamanho que a fonte do seu elemento pai.

/ Exemplo */*

```
body {  
    font-size: 16px; /* Definindo o tamanho da fonte do body como 16px */  
}  
  
h1 {  
    font-size: 2em; /* Tamanho da fonte é igual a 2 vezes o tamanho da fonte do elemento pai (16px * 2) = 32px */  
}
```

Unidade rem:

A unidade rem representa o tamanho da fonte do elemento raiz (geralmente o elemento html). Dessa forma, é possível criar uma base de dimensionamento consistente para toda a página. Por exemplo, se o tamanho da fonte do elemento raiz (html) for definido como 16px, um elemento com valor 2rem terá um tamanho de fonte igual a 32px, independentemente do tamanho da fonte do elemento pai.

```
/* Exemplo */  
html {  
    font-size: 16px; /* Definindo o tamanho da fonte do elemento raiz como 16px */  
}  
h1 {  
    font-size: 2rem; /* Tamanho da fonte é igual a 2 vezes o tamanho da fonte do  
    elemento raiz (16px * 2) = 32px */  
}
```

Comparação entre em e rem:

em: Relativo ao tamanho da fonte do elemento pai. Seu valor pode ser influenciado por tamanhos de fonte de elementos ancestrais.

rem: Relativo ao tamanho da fonte do elemento raiz (html). Seu valor é consistente em toda a página.

A escolha entre em e rem depende do contexto do projeto e das necessidades de dimensionamento. em é mais adequado quando se deseja dimensionar com base no tamanho da fonte do elemento pai, enquanto rem oferece uma abordagem mais consistente para criar layouts escaláveis em toda a página e é unidade mais utilizadas entre desenvolvedores web profissionais modernos.

Lembrando que, ao usar essas unidades, é importante considerar como as mudanças no tamanho da fonte podem afetar outros elementos do layout, e sempre realizar testes em vários dispositivos para garantir que o design permaneça legível e harmonioso.

Unidades Vh & Vw

Em desenvolvimento web, vh e vw são unidades de medida usadas em CSS (Cascading Style Sheets) para dimensionar elementos em relação à janela de visualização (viewport). A viewport é a área visível de uma página da web, que geralmente é a janela do navegador ou a área onde a página da web é exibida.

vh: Altura da Viewport

vh representa a altura da viewport e corresponde a 1/100 da altura da viewport. Por exemplo, se a altura da viewport for 800 pixels, 1vh será equivalente a 8 pixels ($800 * 1/100$). Essa unidade é útil para criar elementos dimensionados com base na altura da viewport. É frequentemente utilizada para criar seções de tela cheia ou elementos que se redimensionam proporcionalmente à altura da janela do navegador.

Exemplo:

```
div {  
    height: 50vh; /* Essa div ocupará 50% da altura da viewport */  
}
```

vw: Largura da Viewport

vw representa a largura da viewport e corresponde a 1/100 da largura da viewport. Por exemplo, se a largura da viewport for 1200 pixels, 1vw será equivalente a 12 pixels ($1200 * 1/100$). Semelhante ao vh, essa unidade é útil para criar elementos dimensionados com base na largura da viewport. É comumente utilizada para criar layouts responsivos que se adaptam a diferentes tamanhos de tela.

Exemplo:

```
div {  
    width: 25vw; /* Essa div ocupará 25% da largura da viewport */  
}
```

Ao utilizar vh e vw, você pode criar layouts responsivos e dinâmicos que se ajustam automaticamente com base nas dimensões da viewport, tornando suas páginas da web mais adaptáveis a diferentes dispositivos e tamanhos de tela. Lembre-se de que essas unidades são suportadas na maioria dos navegadores modernos, mas é sempre uma boa ideia testar o seu design em diferentes dispositivos para garantir a compatibilidade entre os navegadores.