Disciplina de Desenvolvimento Web

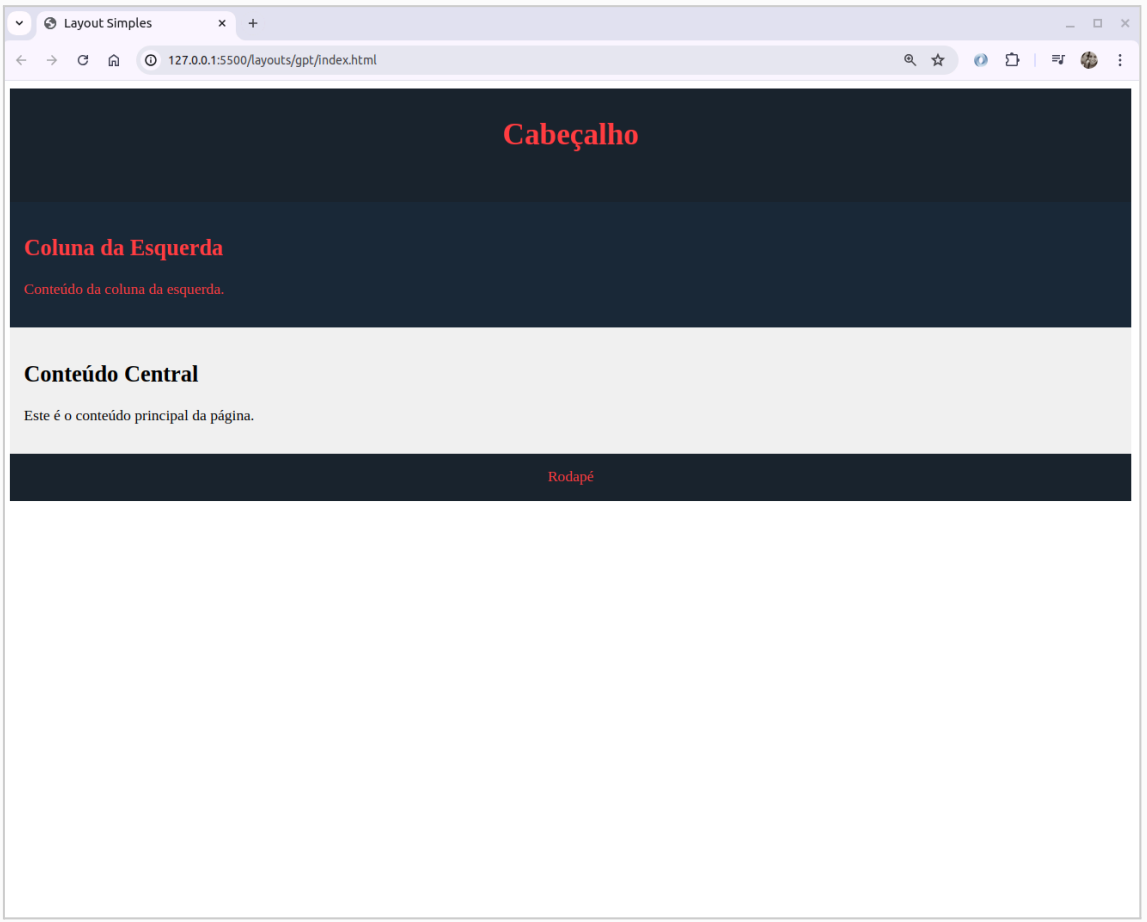
PROFESSOR JEFFERSON CHAVES

jefferson.chaves@ifpr.edu.br

ATIVIDADE AVALIATIVA

O objetivo desta atividade é formalizar o conceito de caixas, unidades de medidas e outros detalhes que envolvem o uso do HTML e CSS por meio de um passo a passo.

Você implementará o leaiute simples de página HTML e CSS com um cabeçalho, conteúdo central com uma coluna à esquerda e um rodapé. O resultado final deve ser parecer com isso:



Leiaute versão 1.0

Passo 0: Preparando o projeto

Crie uma pasta chamada atividade-layout. Dentro desta pasta crie um arquivo nomeado como **index.html** e outro como **style.css.**

Passo 1: Estrutura HTML

Primeiro, vamos criar a estrutura básica do HTML, inserindo no arquivo index.html o seguinte código:

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Leiaute Simples</title>
   <link rel="stylesheet" href="style.css">
</head>
<body>
   <header>
       <h1>Cabeçalho</h1>
   </header>
   <div class="container">
       <aside class="sidebar">
           <h2>Coluna da Esquerda</h2>
           <p>Conteúdo da coluna da esquerda.</p>
       </aside>
       <main class="main-content">
           <h2>Conteúdo Central</h2>
           <p>Este é o conteúdo principal da página.</p>
       </main>
   </div>
   <footer>
       <p>Rodapé</p>
   </footer>
</body>
</html>
```

Repare no uso das tags <header>, <aside>, <main>, <footer> e : todas elas são caixas e funcionam da mesma forma! Ocupam uma linha inteira, podem ter altura e largura e etc. O que as diferencia é o significado atribuído a cada uma.

- Poderíamos trocar todas elas por ? Sim! O funcionamento não seria afetado, embora o significado (semântica) diminuiria muito.
- Poderíamos trocar todas elas por <footer>? Sim! E o funcionamento não seria afetado, contudo não faria sentido algum.

Use o live server para testar sua página e ver o resultado.

Passo 2: Estilos CSS

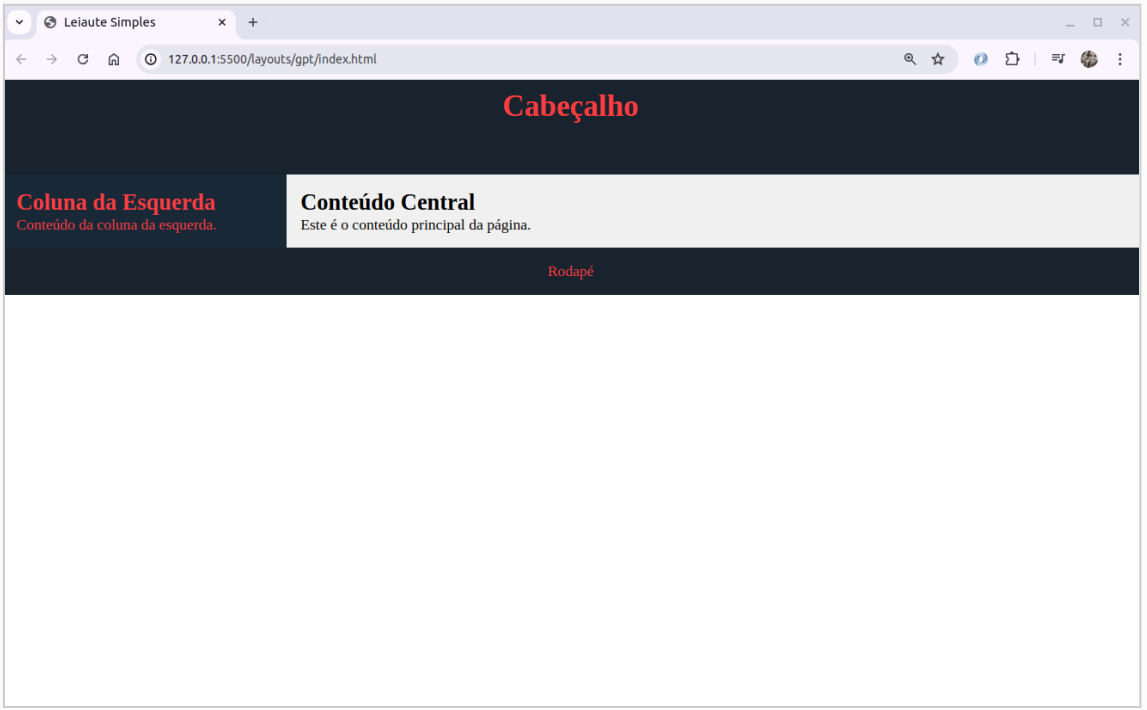
Agora, vamos adicionar estilos para posicionar os elementos adequadamente. Vamos aproveitar e colocar alguma cor nos elementos. Adicione o seguinte código ao arquivo style.css:

```
header {
    background-color: #19242f;
    color: #ff4043;
    text-align: center;
    padding: 10px 0;
}

.sidebar {
    background-color: #1c2a39;
    color: #ff4043;
    padding: 15px;
    box-sizing: border-box;
}

.main-content {
    background-color: #f1f1f1;
    color: #000;
    padding: 15px;
}

footer {
    background-color: #19242f;
    color: #ff4043;
    text-align: center;
    padding: 15px 0;
    height: 50px;
}
```

Use o live server para testar sua página e ver o resultado.

Logo se nota que as caixas não estão de acordo com o que gostaríamos. Isso porque todo elemento possui um comportamento padrão e caso seja desejado mudá-los é necessário explicitar esse novo comportamento por meio do CSS.

Vamos trabalhar na implementação da versão 2.0 da nossa página, que deve ficar assim:



Leiaute versão 2.0

O comportamento das caixas é ocupar horizontalmente toda a tela e verticalmente a altura do conteúdo que estiver dentro do conteúdo. Vamos alterar esse comportamento diminuindo o

tamanho das colunas para caberem lado a lado, adicione as seguintes propriedades ano nosso CSS:

```
.sidebar {
   width: 25%;
   . . .
}

.main-content {
   width: 75%;
   . . .
}
```

Abrindo um Parênteses

Aqui cabe um destaque. Estamos estilizando nossa barra lateral por meio da classe “.sidebar”. Observer um trecho do nosso HTML:

```
<aside class="sidebar">
    <h2>Coluna da Esquerda</h2>
    <p>Conteúdo da coluna da esquerda.</p>
</aside>
```

Poderíamos nos referir a esse elemento no CSS como:

```
aside {
}

.sidebar {
}

aside.sidebar {
}
```

A diferença entre eles é a especificidade. Por exemplo, a palavra “aside” em tradução livre significa “ao lado”. Essa

caixa é utilizada para indicar que o conteúdo ali trata-se de um conteúdo complementar ao principal.

Perceba que é possível que tenhamos outras caixas “aside” em um projeto HTML, de forma que utilizássemos diretamente esse elemento com 25% de largura, TODAS as caixas “aside” teriam esse comportamento.

Uma solução para estilizar somente esse elemento é identificá-lo de alguma forma. No HTML e CSS essa forma é utilizando os atributos id ou class.

```
<elemento id="meu-id" class="minha-classe"> conteúdo </elemento>
```

Um id é uma é uma identificação única para um elemento enquanto que uma classe é uma identificação para um conjunto, uma classe de elementos.

Assim, o seletor .sidebar identifica qualquer elemento dessa classe e seria mais adequado. Já o seletor aside.sidebar identificaria todos os elementos <aside> que possuírem a classe “.sidebar”. Isso é útil quando elementos diferentes usam a mesma classe.

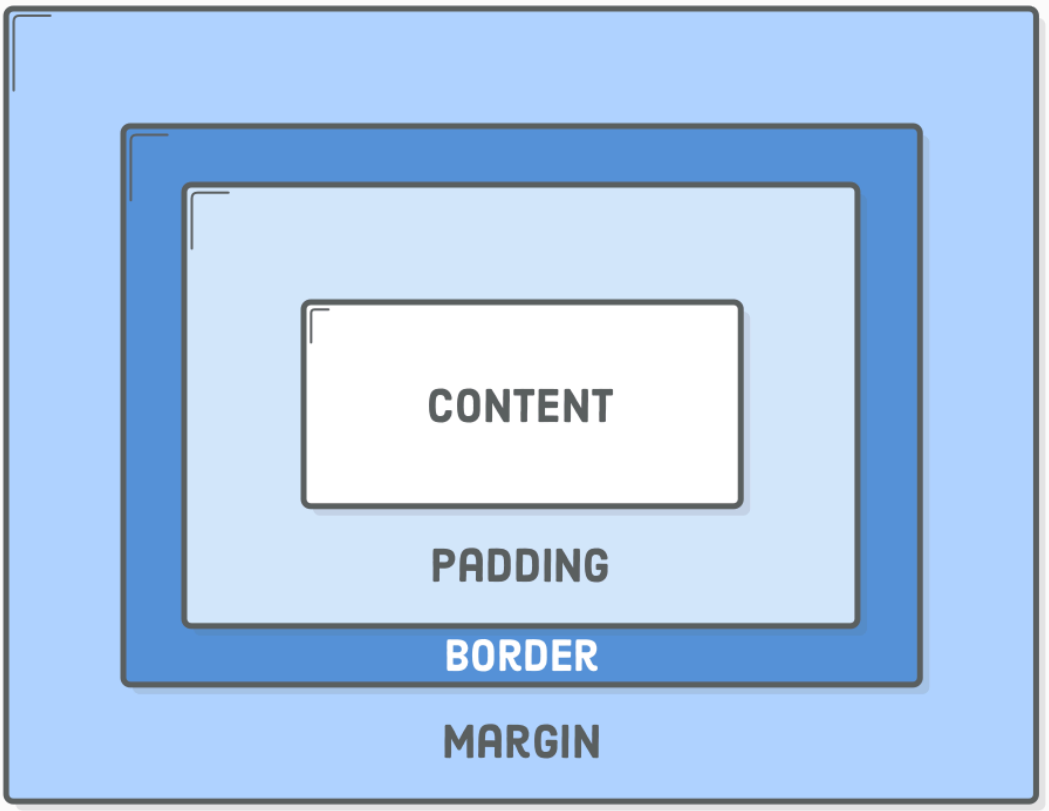
Fechando Parênteses

Mesmo após definir os tamanhos, as caixas lateral e central ainda não se organizaram como o esperado.

Passo 3: Ajustando o CSS

O modelo de caixa

As caixas CSS tem uma forma particular de calcular suas dimensões. Isso é chamado de modelo de caixa ou box-model. O box-model padrão diz que os atributos que determinam larguras e alturas devem ser somados.



Assim os atributos width, padding, margin e border são SOMADOS a largura do elemento. Observe nosso seletor .main-content, como exemplo:

```
.main-content {
   width: 75%;
   padding: 15px;
```

```
    . . .
}
```

Isso significa que a largura da caixa será calculada somando-se 75% do tamanho da tela + 15px de padding, fazendo com que as duas caixas não caibam lado a lado.

Esse modo de cálculo padrão é contra-intuitivo, sendo comum a quase todas as páginas alterarem esse cálculo, utilizando a propriedade “box-sizing” alterando o valor padrão de content-box para border-box.

```
.sidebar {
    box-sizing: border-box;
    width: 25%;
    padding: 15px;
    . . .
}

.main-content {
    box-sizing: border-box;
    width: 75%;
    padding: 15px;
    . . .
}
```

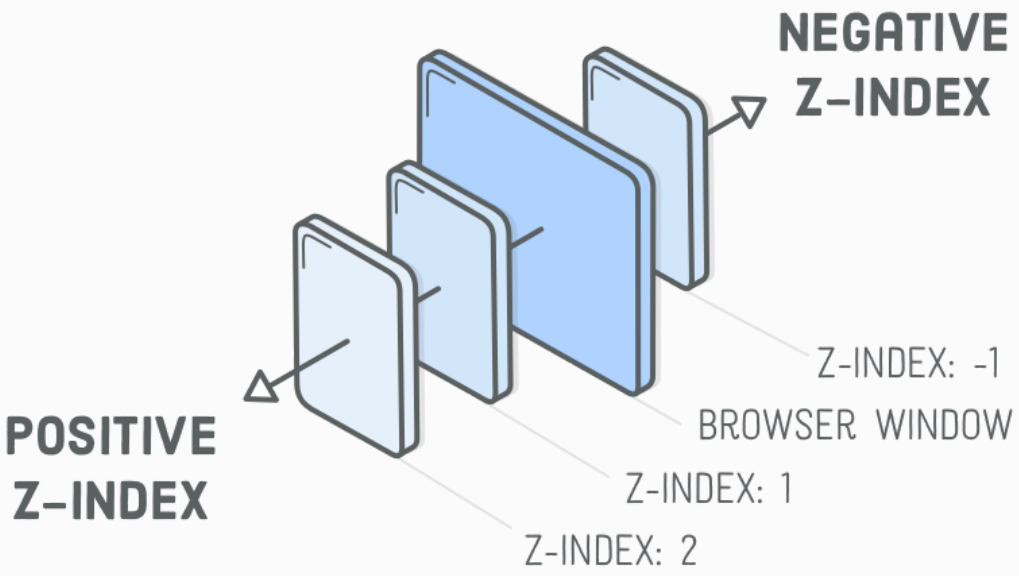
Saiba mais sobre box-sizing clicando aqui!

Passo 4: Ajustando o CSS

Os eixos do HTML e CSS

O layout ainda não está como esperado, mesmo com os tamanhos já ajustados. Isso ocorre porque o comportamento padrão das caixas é ocupar a linha toda!

Uma coisa que devemos saber é que o HTML possui eixos nas dimensões horizontal, vertical e de profundidade.



Podemos alterar o comportamento da nossa caixa, que faz com que ela ocupe a linha toda, fazendo-a flutuar. Ao flutuar, os elementos do tipo caixa, saem no nível 0 e vão para o nível 1, flutuando e se encaixando. Os principais valores são “right”, “left” e “none”. Adicione a propriedade “float” com o valor “left” para .sidebar e .main-content:

```
.sidebar {
   box-sizing: border-box;
   float: left;
   width: 25%;
   padding: 15px;
   . . .
}

.main-content {
   box-sizing: border-box;
   float: left;
   width: 75%;
   padding: 15px;
   . . .
}

footer {
   width: 100%;
   float: left;
   . . .
}
```

Agora sim! O conteúdo está como esperávamos! Mas, o conteúdo parece muito acumulado ao centro. Como tudo em HTML e CSS, existem muitas formas de se resolver esse problema. Uma delas é “esticar o conteúdo do centro”. Mas como fazer isso?

Passo 5: Ajustando o CSS

Calculando tamanhos para .sidebar e .main-content

Nesse caso, não seria funcional definir-se tamanho fixo, em pixels, para o centro. Isso faria com que telas maiores ou menores tivessem uma apresentação diferente daquela desejada.

Também não é possível, nesse caso, usar a unidade de medida percentual, já que essa medida é relativa e só funciona verticalmente, quanto uma caixa “pai” possui um tamanho conhecido. Por exemplo, ao definir uma caixa com altura de 50%, por exemplo, o CSS irá procurar saber 50% referente a que valor. Se não encontrar, essa propriedade é ignorada.

A solução ideal aqui é utilizar a unidade de medida “vh” ou view-height, que em tradução livre significa tamanho da tela (visível). Os valores possíveis vão de 0 a 100.

```
header {
   height: 100px;
   . . .
}

.sidebar {
   height: 100vh;
   . . .
}

.main-content {
   height: 100vh;
   . . .
}
```

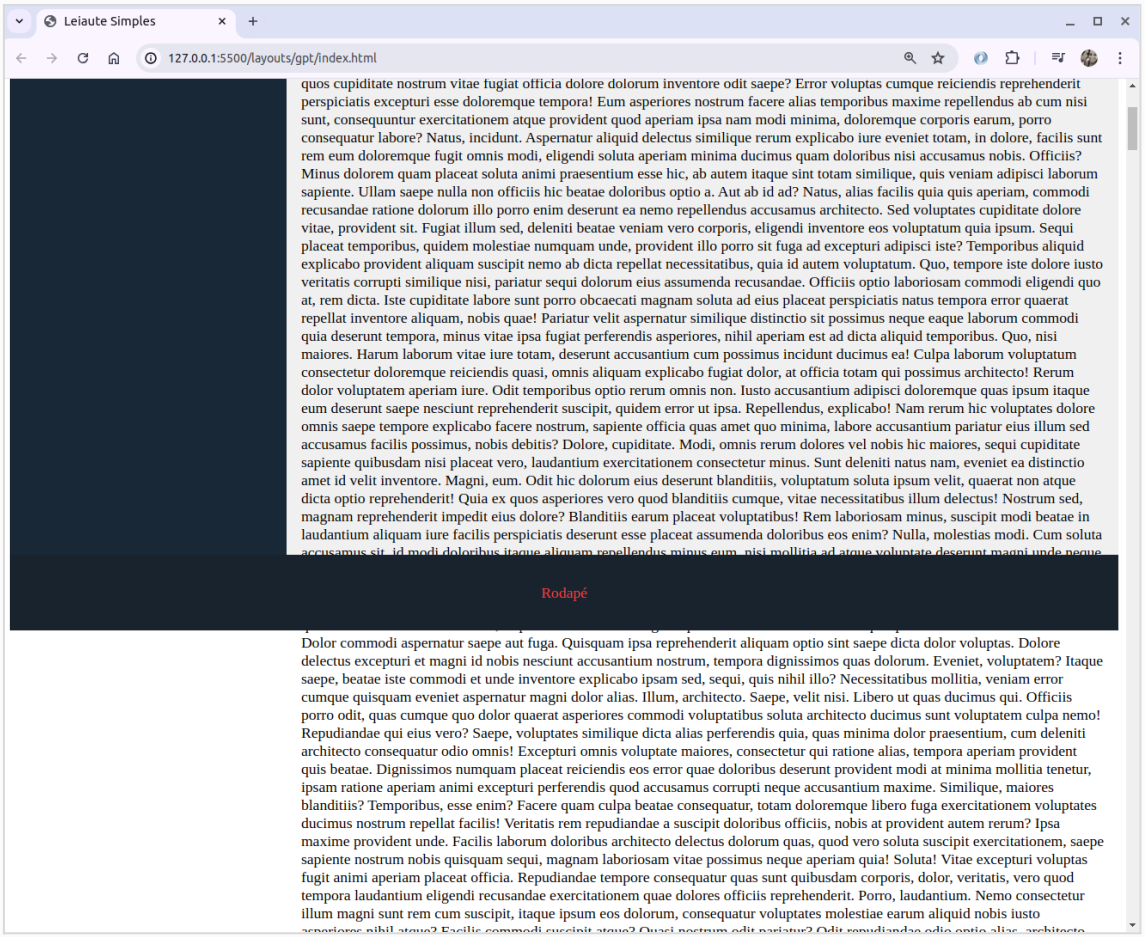
```
footer {
   height: 50px;
   . . .
}
```

Ainda temos um problema. O tamanho foi calculado com base no tamanho da tela e acabou ficando maior que o esperado. Isso porque já tinhamos cabeçalho e rodapé ocupando espaço na tela. Vamos usar uma função do CSS (sim, o CSS tem funções) que irá calcular esse tamanho adequadamente. O calculo para o tamanho do .main-content e da .sidebar dever ser feito obtendo-se o tamanho da tela, e subtraindo-se os tamanhos do header e footer.

100vh - 100px - 50px

```
.sidebar {
   height: calc(100vh-100px-50px);
   . . .
}

.main-content {
   height: calc(100vh-100px-50px);
   . . .
}
```

Ainda temos um último problema: determinamos que a altura será essa calculada. Contudo, o conteúdo dessas duas caixas pode aumentar, acontecendo o que chamamos de “transbordamento” ou “overflow”. Em seu HTML adicione um elemento lorem*500 e observe o resultado:



Faça o último ajuste:

```
.sidebar {
   min-height: calc(100vh-100px-50px);
   . . .
}

.main-content {
   min-height: calc(100vh-100px-50px);
   . . .
}
```

O leiaute está concluído!