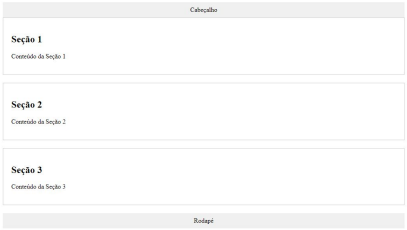
CEP - UP - UTFPR - PUCPr - UFPR – 13/05/2024 - 12:08:57.0
Prof Dr P Kantek (pkantek@gmail.com)
Organizando o laytou com CSS VIVXh53a, V: 1.06 76001 CARLOS EDUARDO ANTAO
24FRO107 - 1 entregar ate 16/maio _____ / / _____

Organizando o layout com CSS

Como se sabe o CSS permite que um layout seja construído para apresentar os conteúdos na hora de mostrar uma página.

1 coluna

Vamos começar com um layout simples, contendo header, footer e um bloco principal entre eles. Este bloco terá 3 sessões, devidamente empilhadas na vertical, como se pode ver em



Neste caso, o CSS será

```
/* Estilo geral do container */
.container {
    display: flex;
    flex-direction: column; /* Organiza os elementos na vertical */
    width: 100%; /* Ocupa a largura total da tela */
    max-width: 1000px; /* Largura máxima de 1000px */
    margin: 0 auto; /* Centraliza o container na tela */
    padding: 20px;            }
/* Estilos para header e footer */
header, footer {
    background-color: #f0f0f0;
    padding: 10px;
    text-align: center;       }
/* Estilos para as seções */
.secao {
    flex: 1 0 auto; /* Ocupa espaço disponível igualmente */
    padding: 20px;
    border: 1px solid #ccc;
    margin-bottom: 20px;      }
```

O HTML que o acessa é

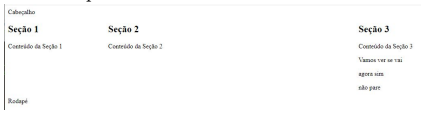
```
<!DOCTYPE html>
<html lang="pt-BR">
<head>
    <meta charset="UTF-8">
    <title>Sistema de Grade Básico</title>
    <link rel="stylesheet" href="stycols.css">
</head>
<body>
    <section class="container">
        <header>Cabeçalho</header>
        <main>
            <section class="secao">
                <h2>Seção 1</h2>
                <p>Conteúdo da Seção 1</p>
            </section>
            <section class="secao">
                <h2>Seção 2</h2>
                <p>Conteúdo da Seção 2</p>
            </section>
            <section class="secao">
                <h2>Seção 3</h2>
                <p>Conteúdo da Seção 3</p>
            </section>
        </main>
        <footer>Rodapé</footer>
    </section>
</body>
</html>
```

Valem as seguintes observações

- O container é definido com display: flex e flex-direction: column para organizar os elementos na vertical.
- A propriedade width: 100% faz com que o container ocupe a largura total da tela.
- max-width: 1000px define uma largura máxima de 1000px para o container, garantindo que ele não fique muito largo em telas maiores.
- margin: 0 auto centraliza o container na tela horizontalmente.
- As seções usam flex: 1 0 auto para ocupar espaço disponível igualmente dentro do container.
- A propriedade padding adiciona espaço interno às seções.
- A borda e a margem inferior estilizam as seções visualmente.

Usando 3 colunas sem grid

Agora vai-se usar um esquema mais simples (sem usar o seletor grid) que é indicado para páginas mais simples. Eis o resultado



Eis o CSS

```
#container {
  width: 100%; /* Define a largura total do container */ }
#coluna1 {
  width: 20%;
  float: left;
  box-sizing: border-box; background-color:#c4d5f6; }
#coluna2 {
  width: 50%;
  float: left;
  box-sizing: border-box; background-color:#a4d3b3; }
#coluna3 {
  width: 30%;
  float: left;
  box-sizing: border-box; background-color:#f1e4a3; }
```

Eis o HTML

```
<!DOCTYPE html>
<html lang="pt-BR">
<head>
    <meta charset="UTF-8">
    <title>Sistema de Grade Básico</title>
    <link rel="stylesheet" href="h53b.css">
</head>  <body>
    <header>Cabeçalho</header>
    <main>
   <section id="container">
         <section id="coluna1">
         <h2>Seção 1</h2>
         <p>Conteúdo da Seção 1</p>
         </section>
         <section id="coluna2">
         <h2>Seção 2</h2>
         <p>Conteúdo da Seção 2</p>
         </section>
         <section id="coluna3">
         <h2>Seção 3</h2>
         <p>Conteúdo da Seção 3</p>
          Vamos ver se vai <p>
          agora sim <p>não pare
         </section>
   </section>
      </main>
      <footer>Rodapé</footer>
</body>  </html>
```

Usando 3 colunas com grid

Com o resultado

Seja o CSS

```
#container {
  display: grid;
  /* Cria 3 colunas com largura igual -- 1fr=1 unid. frac.*/
  grid-template-columns: 1fr 2fr 1.5fr;
  gap: 20px; /* Define o espaçamento entre as colunas */  }
#coluna1{grid-column:1; background-color:#bbaacc;}
#coluna2{grid-column:2; background-color:#ccb4a2;}
#coluna3{grid-column:3; background-color:#d4d3a8;}
```

E o HTML

```
<!DOCTYPE html>
<html lang="pt-BR">
<head>
    <meta charset="UTF-8">
    <title>Sistema de Grade Básico</title>
    <link rel="stylesheet" href="h53a.css">
</head>
<body>
    <section id="container">
        <header>Cabeçalho</header>
        <main>
            <section id="coluna1">
                <h2>Seção 1</h2>
                <p>Conteúdo da Seção 1</p>
            </section>
            <section id="coluna2">
                <h2>Seção 2</h2>
                <p>Conteúdo da Seção 2</p>
            </section>
            <section id="coluna3">
                <h2>Seção 3</h2>
                <p>Conteúdo da Seção 3</p>
           Vamos ver se vai <p>
           agora sim <p>não pare
            </section>
        </main>
        <footer>Rodapé</footer>
    </section>  </body>  </html>
```

☞ Para você fazer

Vamos ao exercício da aula.
Construa um HTML único para esta aula. Deve ser um site com o tema

`O transporte ferroviário no Brasil`

e contendo as seguintes partes:

- um cabeçalho contendo 3 parágrafos explicando o que este exercício vai tratar (sites com 3 blocos flexíveis)
- um bloco `main` no qual estarão os conteúdos.
- o primeiro bloco deve conter a história do tema
- o segundo bloco deve conter o estado atual do tema
- o terceiro bloco é para você especular sobre o futuro do tema
- o rodapé deve conter a sua identificação (nome, turma, curso e colégio).

Depois deste HTML feito e funcionando, é hora de construir 3 arquivos CSS, a saber:

1. Mostre os 3 blocos na vertical.
2. Os blocos devem aparecer lado a lado, com os 3 blocos de mesmo tamanho
3. O primeiro e terceiro bloco devem ocupar 25% do espaço cada um, enquanto o bloco central deve ocupar os 50% restantes.

Use ou não o selector grid. Fica a seu critério. Note que o bloco do meio deve conter pelo menos 3 fotos. Se quiser colocar fotos nos outros blocos fique à vontade.

O ponto mais importante deste exercício é que o HTML deve ser o **mesmo** para os 3 exercícios.

Mostre estas atividades e os arquivos que você criou para o professor ANTES de devolver esta folha.

Avaliação

(preenchido pelo professor)

107-76001 - ga/ a

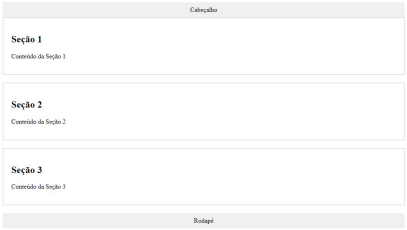
CEP - UP - UTFPR - PUCPr - UFPR – 13/05/2024 - 12:08:57.0
Prof Dr P Kantek (pkantek@gmail.com)
Organizando o laytou com CSS VIVXh53a, V: 1.06 76199 DANILO DEPETRIS SOARES
24FRO107 - 2 entregar ate 16/maio _____ / / _____

Organizando o layout com CSS

Como se sabe o CSS permite que um layout seja construído para apresentar os conteúdos na hora de mostrar uma página.

1 coluna

Vamos começar com um layout simples, contendo header, footer e um bloco principal entre eles. Este bloco terá 3 sessões, devidamente empilhadas na vertical, como se pode ver em



Neste caso, o CSS será

```
/* Estilo geral do container */
.container {
    display: flex;
    flex-direction: column; /* Organiza os elementos na vertical */
    width: 100%; /* Ocupa a largura total da tela */
    max-width: 1000px; /* Largura máxima de 1000px */
    margin: 0 auto; /* Centraliza o container na tela */
    padding: 20px;            }
/* Estilos para header e footer */
header, footer {
    background-color: #f0f0f0;
    padding: 10px;
    text-align: center;       }
/* Estilos para as seções */
.secao {
    flex: 1 0 auto; /* Ocupa espaço disponível igualmente */
    padding: 20px;
    border: 1px solid #ccc;
    margin-bottom: 20px;      }
```

O HTML que o acessa é

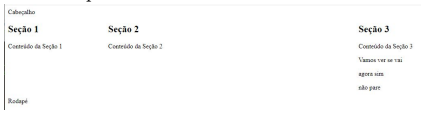
```
<!DOCTYPE html>
<html lang="pt-BR">
<head>
    <meta charset="UTF-8">
    <title>Sistema de Grade Básico</title>
    <link rel="stylesheet" href="stycols.css">
</head>
<body>
    <section class="container">
        <header>Cabeçalho</header>
        <main>
            <section class="secao">
                <h2>Seção 1</h2>
                <p>Conteúdo da Seção 1</p>
            </section>
            <section class="secao">
                <h2>Seção 2</h2>
                <p>Conteúdo da Seção 2</p>
            </section>
            <section class="secao">
                <h2>Seção 3</h2>
                <p>Conteúdo da Seção 3</p>
            </section>
        </main>
        <footer>Rodapé</footer>
    </section>
</body>
</html>
```

Valem as seguintes observações

- O container é definido com display: flex e flex-direction: column para organizar os elementos na vertical.
- A propriedade width: 100% faz com que o container ocupe a largura total da tela.
- max-width: 1000px define uma largura máxima de 1000px para o container, garantindo que ele não fique muito largo em telas maiores.
- margin: 0 auto centraliza o container na tela horizontalmente.
- As seções usam flex: 1 0 auto para ocupar espaço disponível igualmente dentro do container.
- A propriedade padding adiciona espaço interno às seções.
- A borda e a margem inferior estilizam as seções visualmente.

Usando 3 colunas sem grid

Agora vai-se usar um esquema mais simples (sem usar o seletor grid) que é indicado para páginas mais simples. Eis o resultado



Eis o CSS

```
#container {
  width: 100%; /* Define a largura total do container */ }
#coluna1 {
  width: 20%;
  float: left;
  box-sizing: border-box; background-color:#c4d5f6; }
#coluna2 {
  width: 50%;
  float: left;
  box-sizing: border-box; background-color:#a4d3b3; }
#coluna3 {
  width: 30%;
  float: left;
  box-sizing: border-box; background-color:#f1e4a3; }
```

Eis o HTML

```
<!DOCTYPE html>
<html lang="pt-BR">
<head>
    <meta charset="UTF-8">
    <title>Sistema de Grade Básico</title>
    <link rel="stylesheet" href="h53b.css">
</head>  <body>
    <header>Cabeçalho</header>
    <main>
   <section id="container">
         <section id="coluna1">
         <h2>Seção 1</h2>
         <p>Conteúdo da Seção 1</p>
         </section>
         <section id="coluna2">
         <h2>Seção 2</h2>
         <p>Conteúdo da Seção 2</p>
         </section>
         <section id="coluna3">
         <h2>Seção 3</h2>
         <p>Conteúdo da Seção 3</p>
          Vamos ver se vai <p>
          agora sim <p>não pare
         </section>
   </section>
      </main>
      <footer>Rodapé</footer>
</body>  </html>
```

Usando 3 colunas com grid

Com o resultado

Seja o CSS

```
#container {
  display: grid;
  /* Cria 3 colunas com largura igual -- 1fr=1 unid. frac.*/
  grid-template-columns: 1fr 2fr 1.5fr;
  gap: 20px; /* Define o espaçamento entre as colunas */  }
#coluna1{grid-column:1; background-color:#bbaacc;}
#coluna2{grid-column:2; background-color:#ccb4a2;}
#coluna3{grid-column:3; background-color:#d4d3a8;}
```

E o HTML

```
<!DOCTYPE html>
<html lang="pt-BR">
<head>
    <meta charset="UTF-8">
    <title>Sistema de Grade Básico</title>
    <link rel="stylesheet" href="h53a.css">
</head>
<body>
    <section id="container">
        <header>Cabeçalho</header>
        <main>
            <section id="coluna1">
                <h2>Seção 1</h2>
                <p>Conteúdo da Seção 1</p>
            </section>
            <section id="coluna2">
                <h2>Seção 2</h2>
                <p>Conteúdo da Seção 2</p>
            </section>
            <section id="coluna3">
                <h2>Seção 3</h2>
                <p>Conteúdo da Seção 3</p>
           Vamos ver se vai <p>
           agora sim <p>não pare
            </section>
        </main>
        <footer>Rodapé</footer>
    </section>  </body>  </html>
```

☞ Para você fazer

Vamos ao exercício da aula.
Construa um HTML único para esta aula. Deve ser um site com o tema

`O Teatro Guaira em Curitiba`

e contendo as seguintes partes:

- um cabeçalho contendo 3 parágrafos explicando o que este exercício vai tratar (sites com 3 blocos flexíveis)
- um bloco `main` no qual estarão os conteúdos.
- o primeiro bloco deve conter a história do tema
- o segundo bloco deve conter o estado atual do tema
- o terceiro bloco é para você especular sobre o futuro do tema
- o rodapé deve conter a sua identificação (nome, turma, curso e colégio).

Depois deste HTML feito e funcionando, é hora de construir 3 arquivos CSS, a saber:

1. Mostre os 3 blocos na vertical.
2. Os blocos devem aparecer lado a lado, com os 3 blocos de mesmo tamanho
3. O primeiro e terceiro bloco devem ocupar 25% do espaço cada um, enquanto o bloco central deve ocupar os 50% restantes.

Use ou não o selector grid. Fica a seu critério. Note que o bloco do meio deve conter pelo menos 3 fotos. Se quiser colocar fotos nos outros blocos fique à vontade.

O ponto mais importante deste exercício é que o HTML deve ser o **mesmo** para os 3 exercícios.

Mostre estas atividades e os arquivos que você criou para o professor ANTES de devolver esta folha.

Avaliação

(preenchido pelo professor)

107-76199 - ga/ a

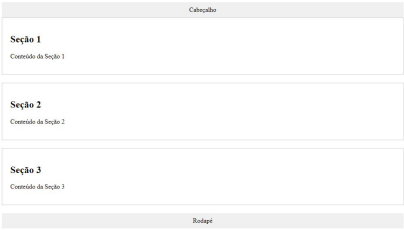
CEP - UP - UTFPR - PUCPr - UFPR – 13/05/2024 - 12:08:57.0
Prof Dr P Kantek (pkantek@gmail.com)
Organizando o laytou com CSS VIVXh53a, V: 1.06 76018 EDUARDO CAVALHEIRO DIAS
24FRO107 - 3 entregar ate 16/maio _____ / / _____

Organizando o layout com CSS

Como se sabe o CSS permite que um layout seja construído para apresentar os conteúdos na hora de mostrar uma página.

1 coluna

Vamos começar com um layout simples, contendo header, footer e um bloco principal entre eles. Este bloco terá 3 sessões, devidamente empilhadas na vertical, como se pode ver em



Neste caso, o CSS será

```
/* Estilo geral do container */
.container {
    display: flex;
    flex-direction: column; /* Organiza os elementos na vertical */
    width: 100%; /* Ocupa a largura total da tela */
    max-width: 1000px; /* Largura máxima de 1000px */
    margin: 0 auto; /* Centraliza o container na tela */
    padding: 20px;            }
/* Estilos para header e footer */
header, footer {
    background-color: #f0f0f0;
    padding: 10px;
    text-align: center;       }
/* Estilos para as seções */
.secao {
    flex: 1 0 auto; /* Ocupa espaço disponível igualmente */
    padding: 20px;
    border: 1px solid #ccc;
    margin-bottom: 20px;      }
```

O HTML que o acessa é

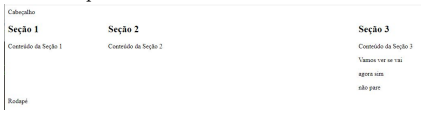
```
<!DOCTYPE html>
<html lang="pt-BR">
<head>
    <meta charset="UTF-8">
    <title>Sistema de Grade Básico</title>
    <link rel="stylesheet" href="stycols.css">
</head>
<body>
    <section class="container">
        <header>Cabeçalho</header>
        <main>
            <section class="secao">
                <h2>Seção 1</h2>
                <p>Conteúdo da Seção 1</p>
            </section>
            <section class="secao">
                <h2>Seção 2</h2>
                <p>Conteúdo da Seção 2</p>
            </section>
            <section class="secao">
                <h2>Seção 3</h2>
                <p>Conteúdo da Seção 3</p>
            </section>
        </main>
        <footer>Rodapé</footer>
    </section>
</body>
</html>
```

Valem as seguintes observações

- O container é definido com display: flex e flex-direction: column para organizar os elementos na vertical.
- A propriedade width: 100% faz com que o container ocupe a largura total da tela.
- max-width: 1000px define uma largura máxima de 1000px para o container, garantindo que ele não fique muito largo em telas maiores.
- margin: 0 auto centraliza o container na tela horizontalmente.
- As seções usam flex: 1 0 auto para ocupar espaço disponível igualmente dentro do container.
- A propriedade padding adiciona espaço interno às seções.
- A borda e a margem inferior estilizam as seções visualmente.

Usando 3 colunas sem grid

Agora vai-se usar um esquema mais simples (sem usar o seletor grid) que é indicado para páginas mais simples. Eis o resultado



Eis o CSS

```
#container {
  width: 100%; /* Define a largura total do container */ }
#coluna1 {
  width: 20%;
  float: left;
  box-sizing: border-box; background-color:#c4d5f6; }
#coluna2 {
  width: 50%;
  float: left;
  box-sizing: border-box; background-color:#a4d3b3; }
#coluna3 {
  width: 30%;
  float: left;
  box-sizing: border-box; background-color:#f1e4a3; }
```

Eis o HTML

```
<!DOCTYPE html>
<html lang="pt-BR">
<head>
    <meta charset="UTF-8">
    <title>Sistema de Grade Básico</title>
    <link rel="stylesheet" href="h53b.css">
</head>  <body>
    <header>Cabeçalho</header>
    <main>
   <section id="container">
         <section id="coluna1">
         <h2>Seção 1</h2>
         <p>Conteúdo da Seção 1</p>
         </section>
         <section id="coluna2">
         <h2>Seção 2</h2>
         <p>Conteúdo da Seção 2</p>
         </section>
         <section id="coluna3">
         <h2>Seção 3</h2>
         <p>Conteúdo da Seção 3</p>
          Vamos ver se vai <p>
          agora sim <p>não pare
         </section>
   </section>
      </main>
      <footer>Rodapé</footer>
</body>  </html>
```

Usando 3 colunas com grid

Com o resultado

Seja o CSS

```
#container {
  display: grid;
  /* Cria 3 colunas com largura igual -- 1fr=1 unid. frac.*/
  grid-template-columns: 1fr 2fr 1.5fr;
  gap: 20px; /* Define o espaçamento entre as colunas */  }
#coluna1{grid-column:1; background-color:#bbaacc;}
#coluna2{grid-column:2; background-color:#ccb4a2;}
#coluna3{grid-column:3; background-color:#d4d3a8;}
```

E o HTML

```
<!DOCTYPE html>
<html lang="pt-BR">
<head>
    <meta charset="UTF-8">
    <title>Sistema de Grade Básico</title>
    <link rel="stylesheet" href="h53a.css">
</head>
<body>
    <section id="container">
        <header>Cabeçalho</header>
        <main>
            <section id="coluna1">
                <h2>Seção 1</h2>
                <p>Conteúdo da Seção 1</p>
            </section>
            <section id="coluna2">
                <h2>Seção 2</h2>
                <p>Conteúdo da Seção 2</p>
            </section>
            <section id="coluna3">
                <h2>Seção 3</h2>
                <p>Conteúdo da Seção 3</p>
           Vamos ver se vai <p>
           agora sim <p>não pare
            </section>
        </main>
        <footer>Rodapé</footer>
    </section>  </body>  </html>
```

☞ Para você fazer

Vamos ao exercício da aula.
Construa um HTML único para esta aula. Deve ser um site com o tema

`A indústria do livro no mundo`

e contendo as seguintes partes:

- um cabeçalho contendo 3 parágrafos explicando o que este exercício vai tratar (sites com 3 blocos flexíveis)
- um bloco `main` no qual estarão os conteúdos.
- o primeiro bloco deve conter a história do tema
- o segundo bloco deve conter o estado atual do tema
- o terceiro bloco é para você especular sobre o futuro do tema
- o rodapé deve conter a sua identificação (nome, turma, curso e colégio).

Depois deste HTML feito e funcionando, é hora de construir 3 arquivos CSS, a saber:

1. Mostre os 3 blocos na vertical.
2. Os blocos devem aparecer lado a lado, com os 3 blocos de mesmo tamanho
3. O primeiro e terceiro bloco devem ocupar 25% do espaço cada um, enquanto o bloco central deve ocupar os 50% restantes.

Use ou não o selector grid. Fica a seu critério. Note que o bloco do meio deve conter pelo menos 3 fotos. Se quiser colocar fotos nos outros blocos fique à vontade.

O ponto mais importante deste exercício é que o HTML deve ser o **mesmo** para os 3 exercícios.

Mostre estas atividades e os arquivos que você criou para o professor ANTES de devolver esta folha.

Avaliação

(preenchido pelo professor)

107-76018 - ga/ a

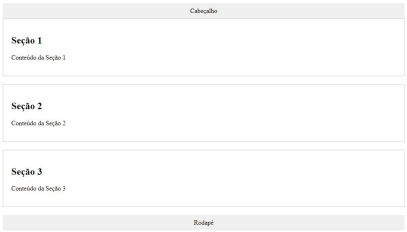
CEP - UP - UTFPR - PUCPr - UFPR – 13/05/2024 - 12:08:57.0
Prof Dr P Kantek (pkantek@gmail.com)
Organizando o laytou com CSS VIVXh53a, V: 1.06 76025 ERIC CORDEIRO 24FRO107 - 4
entregar ate 16/maio _____ / _____ / _____

Organizando o layout com CSS

Como se sabe o CSS permite que um layout seja construído para apresentar os conteúdos na hora de mostrar uma página.

1 coluna

Vamos começar com um layout simples, contendo header, footer e um bloco principal entre eles. Este bloco terá 3 sessões, devidamente empilhadas na vertical, como se pode ver em



Neste caso, o CSS será

```
/* Estilo geral do container */
.container {
    display: flex;
    flex-direction: column; /* Organiza os elementos na vertical */
    width: 100%; /* Ocupa a largura total da tela */
    max-width: 1000px; /* Largura máxima de 1000px */
    margin: 0 auto; /* Centraliza o container na tela */
    padding: 20px;            }
/* Estilos para header e footer */
header, footer {
    background-color: #f0f0f0;
    padding: 10px;
    text-align: center;       }
/* Estilos para as seções */
.secao {
    flex: 1 0 auto; /* Ocupa espaço disponível igualmente */
    padding: 20px;
    border: 1px solid #ccc;
    margin-bottom: 20px;      }
```

O HTML que o acessa é

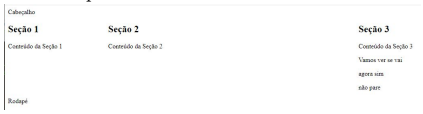
```
<!DOCTYPE html>
<html lang="pt-BR">
<head>
    <meta charset="UTF-8">
    <title>Sistema de Grade Básico</title>
    <link rel="stylesheet" href="stycols.css">
</head>
<body>
    <section class="container">
        <header>Cabeçalho</header>
        <main>
            <section class="secao">
                <h2>Seção 1</h2>
                <p>Conteúdo da Seção 1</p>
            </section>
            <section class="secao">
                <h2>Seção 2</h2>
                <p>Conteúdo da Seção 2</p>
            </section>
            <section class="secao">
                <h2>Seção 3</h2>
                <p>Conteúdo da Seção 3</p>
            </section>
        </main>
        <footer>Rodapé</footer>
    </section>
</body>
</html>
```

Valem as seguintes observações

- O container é definido com display: flex e flex-direction: column para organizar os elementos na vertical.
- A propriedade width: 100% faz com que o container ocupe a largura total da tela.
- max-width: 1000px define uma largura máxima de 1000px para o container, garantindo que ele não fique muito largo em telas maiores.
- margin: 0 auto centraliza o container na tela horizontalmente.
- As seções usam flex: 1 0 auto para ocupar espaço disponível igualmente dentro do container.
- A propriedade padding adiciona espaço interno às seções.
- A borda e a margem inferior estilizam as seções visualmente.

Usando 3 colunas sem grid

Agora vai-se usar um esquema mais simples (sem usar o seletor grid) que é indicado para páginas mais simples. Eis o resultado



Eis o CSS

```
#container {
  width: 100%; /* Define a largura total do container */ }
#coluna1 {
  width: 20%;
  float: left;
  box-sizing: border-box; background-color:#c4d5f6; }
#coluna2 {
  width: 50%;
  float: left;
  box-sizing: border-box; background-color:#a4d3b3; }
#coluna3 {
  width: 30%;
  float: left;
  box-sizing: border-box; background-color:#f1e4a3; }
```

Eis o HTML

```
<!DOCTYPE html>
<html lang="pt-BR">
<head>
    <meta charset="UTF-8">
    <title>Sistema de Grade Básico</title>
    <link rel="stylesheet" href="h53b.css">
</head>  <body>
    <header>Cabeçalho</header>
    <main>
   <section id="container">
         <section id="coluna1">
         <h2>Seção 1</h2>
         <p>Conteúdo da Seção 1</p>
         </section>
         <section id="coluna2">
         <h2>Seção 2</h2>
         <p>Conteúdo da Seção 2</p>
         </section>
         <section id="coluna3">
         <h2>Seção 3</h2>
         <p>Conteúdo da Seção 3</p>
          Vamos ver se vai <p>
          agora sim <p>não pare
         </section>
   </section>
      </main>
      <footer>Rodapé</footer>
</body>  </html>
```

Usando 3 colunas com grid

Com o resultado

Seja o CSS

```
#container {
  display: grid;
  /* Cria 3 colunas com largura igual -- 1fr=1 unid. frac.*/
  grid-template-columns: 1fr 2fr 1.5fr;
  gap: 20px; /* Define o espaçamento entre as colunas */  }
#coluna1{grid-column:1; background-color:#bbaacc;}
#coluna2{grid-column:2; background-color:#ccb4a2;}
#coluna3{grid-column:3; background-color:#d4d3a8;}
```

E o HTML

```
<!DOCTYPE html>
<html lang="pt-BR">
<head>
    <meta charset="UTF-8">
    <title>Sistema de Grade Básico</title>
    <link rel="stylesheet" href="h53a.css">
</head>
<body>
    <section id="container">
        <header>Cabeçalho</header>
        <main>
            <section id="coluna1">
                <h2>Seção 1</h2>
                <p>Conteúdo da Seção 1</p>
            </section>
            <section id="coluna2">
                <h2>Seção 2</h2>
                <p>Conteúdo da Seção 2</p>
            </section>
            <section id="coluna3">
                <h2>Seção 3</h2>
                <p>Conteúdo da Seção 3</p>
           Vamos ver se vai <p>
           agora sim <p>não pare
            </section>
        </main>
        <footer>Rodapé</footer>
    </section>  </body>  </html>
```

☞ Para você fazer

Vamos ao exercício da aula.
Construa um HTML único para esta aula. Deve ser um site com o tema

`O bairro do Portão em Curitiba`

e contendo as seguintes partes:

- um cabeçalho contendo 3 parágrafos explicando o que este exercício vai tratar (sites com 3 blocos flexíveis)
- um bloco `main` no qual estarão os conteúdos.
- o primeiro bloco deve conter a história do tema
- o segundo bloco deve conter o estado atual do tema
- o terceiro bloco é para você especular sobre o futuro do tema
- o rodapé deve conter a sua identificação (nome, turma, curso e colégio).

Depois deste HTML feito e funcionando, é hora de construir 3 arquivos CSS, a saber:

1. Mostre os 3 blocos na vertical.
2. Os blocos devem aparecer lado a lado, com os 3 blocos de mesmo tamanho
3. O primeiro e terceiro bloco devem ocupar 25% do espaço cada um, enquanto o bloco central deve ocupar os 50% restantes.

Use ou não o selector grid. Fica a seu critério. Note que o bloco do meio deve conter pelo menos 3 fotos. Se quiser colocar fotos nos outros blocos fique à vontade.

O ponto mais importante deste exercício é que o HTML deve ser o **mesmo** para os 3 exercícios.

Mostre estas atividades e os arquivos que você criou para o professor ANTES de devolver esta folha.

Avaliação

(preenchido pelo professor)

107-76025 - ga/ a

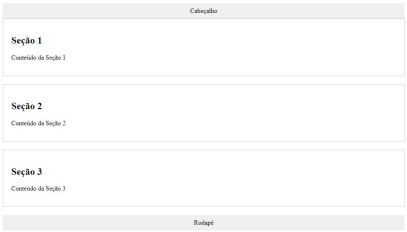
CEP - UP - UTFPR - PUCPr - UFPR – 13/05/2024 - 12:08:57.0
Prof Dr P Kantek (pkantek@gmail.com)
Organizando o laytou com CSS VIVXh53a, V: 1.06 76032 FELIPE EIJI KANO 24FRO107 - 5 entregar ate 16/maio _____ / _____ / _____

Organizando o layout com CSS

Como se sabe o CSS permite que um layout seja construído para apresentar os conteúdos na hora de mostrar uma página.

1 coluna

Vamos começar com um layout simples, contendo header, footer e um bloco principal entre eles. Este bloco terá 3 sessões, devidamente empilhadas na vertical, como se pode ver em



Neste caso, o CSS será

```
/* Estilo geral do container */
.container {
    display: flex;
    flex-direction: column; /* Organiza os elementos na vertical */
    width: 100%; /* Ocupa a largura total da tela */
    max-width: 1000px; /* Largura máxima de 1000px */
    margin: 0 auto; /* Centraliza o container na tela */
    padding: 20px;            }
/* Estilos para header e footer */
header, footer {
    background-color: #f0f0f0;
    padding: 10px;
    text-align: center;       }
/* Estilos para as seções */
.secao {
    flex: 1 0 auto; /* Ocupa espaço disponível igualmente */
    padding: 20px;
    border: 1px solid #ccc;
    margin-bottom: 20px;      }
```

O HTML que o acessa é

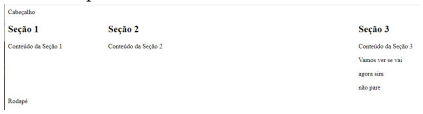
```
<!DOCTYPE html>
<html lang="pt-BR">
<head>
    <meta charset="UTF-8">
    <title>Sistema de Grade Básico</title>
    <link rel="stylesheet" href="stycols.css">
</head>
<body>
    <section class="container">
        <header>Cabeçalho</header>
        <main>
            <section class="secao">
                <h2>Seção 1</h2>
                <p>Conteúdo da Seção 1</p>
            </section>
            <section class="secao">
                <h2>Seção 2</h2>
                <p>Conteúdo da Seção 2</p>
            </section>
            <section class="secao">
                <h2>Seção 3</h2>
                <p>Conteúdo da Seção 3</p>
            </section>
        </main>
        <footer>Rodapé</footer>
    </section>
</body>
</html>
```

Valem as seguintes observações

- O container é definido com display: flex e flex-direction: column para organizar os elementos na vertical.
- A propriedade width: 100% faz com que o container ocupe a largura total da tela.
- max-width: 1000px define uma largura máxima de 1000px para o container, garantindo que ele não fique muito largo em telas maiores.
- margin: 0 auto centraliza o container na tela horizontalmente.
- As seções usam flex: 1 0 auto para ocupar espaço disponível igualmente dentro do container.
- A propriedade padding adiciona espaço interno às seções.
- A borda e a margem inferior estilizam as seções visualmente.

Usando 3 colunas sem grid

Agora vai-se usar um esquema mais simples (sem usar o seletor grid) que é indicado para páginas mais simples. Eis o resultado



Eis o CSS

```
#container {
  width: 100%; /* Define a largura total do container */ }
#coluna1 {
  width: 20%;
  float: left;
  box-sizing: border-box; background-color:#c4d5f6; }
#coluna2 {
  width: 50%;
  float: left;
  box-sizing: border-box; background-color:#a4d3b3; }
#coluna3 {
  width: 30%;
  float: left;
  box-sizing: border-box; background-color:#f1e4a3; }
```

Eis o HTML

```
<!DOCTYPE html>
<html lang="pt-BR">
<head>
    <meta charset="UTF-8">
    <title>Sistema de Grade Básico</title>
    <link rel="stylesheet" href="h53b.css">
</head>  <body>
    <header>Cabeçalho</header>
    <main>
   <section id="container">
         <section id="coluna1">
         <h2>Seção 1</h2>
         <p>Conteúdo da Seção 1</p>
         </section>
         <section id="coluna2">
         <h2>Seção 2</h2>
         <p>Conteúdo da Seção 2</p>
         </section>
         <section id="coluna3">
         <h2>Seção 3</h2>
         <p>Conteúdo da Seção 3</p>
          Vamos ver se vai <p>
          agora sim <p>não pare
         </section>
   </section>
      </main>
      <footer>Rodapé</footer>
</body>  </html>
```

Usando 3 colunas com grid

Com o resultado

Seja o CSS

```
#container {
  display: grid;
  /* Cria 3 colunas com largura igual -- 1fr=1 unid. frac.*/
  grid-template-columns: 1fr 2fr 1.5fr;
  gap: 20px; /* Define o espaçamento entre as colunas */  }
#coluna1{grid-column:1; background-color:#bbaacc;}
#coluna2{grid-column:2; background-color:#ccb4a2;}
#coluna3{grid-column:3; background-color:#d4d3a8;}
```

E o HTML

```
<!DOCTYPE html>
<html lang="pt-BR">
<head>
    <meta charset="UTF-8">
    <title>Sistema de Grade Básico</title>
    <link rel="stylesheet" href="h53a.css">
</head>
<body>
    <section id="container">
        <header>Cabeçalho</header>
        <main>
            <section id="coluna1">
                <h2>Seção 1</h2>
                <p>Conteúdo da Seção 1</p>
            </section>
            <section id="coluna2">
                <h2>Seção 2</h2>
                <p>Conteúdo da Seção 2</p>
            </section>
            <section id="coluna3">
                <h2>Seção 3</h2>
                <p>Conteúdo da Seção 3</p>
           Vamos ver se vai <p>
           agora sim <p>não pare
            </section>
        </main>
        <footer>Rodapé</footer>
    </section>  </body>  </html>
```

☞ Para você fazer

Vamos ao exercício da aula.
Construa um HTML único para esta aula. Deve ser um site com o tema

`A internet e o protocolo HTML`

e contendo as seguintes partes:

- um cabeçalho contendo 3 parágrafos explicando o que este exercício vai tratar (sites com 3 blocos flexíveis)
- um bloco `main` no qual estarão os conteúdos.
- o primeiro bloco deve conter a história do tema
- o segundo bloco deve conter o estado atual do tema
- o terceiro bloco é para você especular sobre o futuro do tema
- o rodapé deve conter a sua identificação (nome, turma, curso e colégio).

Depois deste HTML feito e funcionando, é hora de construir 3 arquivos CSS, a saber:

1. Mostre os 3 blocos na vertical.
2. Os blocos devem aparecer lado a lado, com os 3 blocos de mesmo tamanho
3. O primeiro e terceiro bloco devem ocupar 25% do espaço cada um, enquanto o bloco central deve ocupar os 50% restantes.

Use ou não o selector grid. Fica a seu critério. Note que o bloco do meio deve conter pelo menos 3 fotos. Se quiser colocar fotos nos outros blocos fique à vontade.

O ponto mais importante deste exercício é que o HTML deve ser o **mesmo** para os 3 exercícios.

Mostre estas atividades e os arquivos que você criou para o professor ANTES de devolver esta folha.

Avaliação

(preenchido pelo professor)

107-76032 - ga/ a

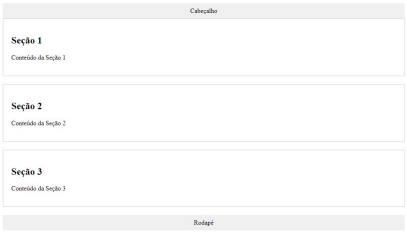
CEP - UP - UTFPR - PUCPr - UFPR – 13/05/2024 - 12:08:57.0
Prof Dr P Kantek (pkantek@gmail.com)
Organizando o laytou com CSS VIVXh53a, V: 1.06 76049 FELIPE JESMIEL LEITE 24FRO107 - 6 entregar ate 16/maio _____ / _____ / _____

Organizando o layout com CSS

Como se sabe o CSS permite que um layout seja construído para apresentar os conteúdos na hora de mostrar uma página.

1 coluna

Vamos começar com um layout simples, contendo header, footer e um bloco principal entre eles. Este bloco terá 3 sessões, devidamente empilhadas na vertical, como se pode ver em



Neste caso, o CSS será

```
/* Estilo geral do container */
.container {
    display: flex;
    flex-direction: column; /* Organiza os elementos na vertical */
    width: 100%; /* Ocupa a largura total da tela */
    max-width: 1000px; /* Largura máxima de 1000px */
    margin: 0 auto; /* Centraliza o container na tela */
    padding: 20px;            }
/* Estilos para header e footer */
header, footer {
    background-color: #f0f0f0;
    padding: 10px;
    text-align: center;       }
/* Estilos para as seções */
.secao {
    flex: 1 0 auto; /* Ocupa espaço disponível igualmente */
    padding: 20px;
    border: 1px solid #ccc;
    margin-bottom: 20px;      }
```

O HTML que o acessa é

```
<!DOCTYPE html>
<html lang="pt-BR">
<head>
    <meta charset="UTF-8">
    <title>Sistema de Grade Básico</title>
    <link rel="stylesheet" href="stycols.css">
</head>
<body>
    <section class="container">
        <header>Cabeçalho</header>
        <main>
            <section class="secao">
                <h2>Seção 1</h2>
                <p>Conteúdo da Seção 1</p>
            </section>
            <section class="secao">
                <h2>Seção 2</h2>
                <p>Conteúdo da Seção 2</p>
            </section>
            <section class="secao">
                <h2>Seção 3</h2>
                <p>Conteúdo da Seção 3</p>
            </section>
        </main>
        <footer>Rodapé</footer>
    </section>
</body>
</html>
```

Valem as seguintes observações

- O container é definido com display: flex e flex-direction: column para organizar os elementos na vertical.
- A propriedade width: 100% faz com que o container ocupe a largura total da tela.
- max-width: 1000px define uma largura máxima de 1000px para o container, garantindo que ele não fique muito largo em telas maiores.
- margin: 0 auto centraliza o container na tela horizontalmente.
- As seções usam flex: 1 0 auto para ocupar espaço disponível igualmente dentro do container.
- A propriedade padding adiciona espaço interno às seções.
- A borda e a margem inferior estilizam as seções visualmente.

Usando 3 colunas sem grid

Agora vai-se usar um esquema mais simples (sem usar o seletor grid) que é indicado para páginas mais simples. Eis o resultado

Eis o CSS

```
#container {
  width: 100%; /* Define a largura total do container */ }
#coluna1 {
  width: 20%;
  float: left;
  box-sizing: border-box; background-color:#c4d5f6; }
#coluna2 {
  width: 50%;
  float: left;
  box-sizing: border-box; background-color:#a4d3b3; }
#coluna3 {
  width: 30%;
  float: left;
  box-sizing: border-box; background-color:#f1e4a3; }
```

Eis o HTML

```
<!DOCTYPE html>
<html lang="pt-BR">
<head>
    <meta charset="UTF-8">
    <title>Sistema de Grade Básico</title>
    <link rel="stylesheet" href="h53b.css">
</head>  <body>
    <header>Cabeçalho</header>
    <main>
   <section id="container">
         <section id="coluna1">
         <h2>Seção 1</h2>
         <p>Conteúdo da Seção 1</p>
         </section>
         <section id="coluna2">
         <h2>Seção 2</h2>
         <p>Conteúdo da Seção 2</p>
         </section>
         <section id="coluna3">
         <h2>Seção 3</h2>
         <p>Conteúdo da Seção 3</p>
          Vamos ver se vai <p>
          agora sim <p>não pare
         </section>
   </section>
      </main>
      <footer>Rodapé</footer>
</body>  </html>
```

Usando 3 colunas com grid

Com o resultado

Seja o CSS

```
#container {
  display: grid;
  /* Cria 3 colunas com largura igual -- 1fr=1 unid. frac.*/
  grid-template-columns: 1fr 2fr 1.5fr;
  gap: 20px; /* Define o espaçamento entre as colunas */  }
#coluna1{grid-column:1; background-color:#bbaacc;}
#coluna2{grid-column:2; background-color:#ccb4a2;}
#coluna3{grid-column:3; background-color:#d4d3a8;}
```

E o HTML

```
<!DOCTYPE html>
<html lang="pt-BR">
<head>
    <meta charset="UTF-8">
    <title>Sistema de Grade Básico</title>
    <link rel="stylesheet" href="h53a.css">
</head>
<body>
    <section id="container">
        <header>Cabeçalho</header>
        <main>
            <section id="coluna1">
                <h2>Seção 1</h2>
                <p>Conteúdo da Seção 1</p>
            </section>
            <section id="coluna2">
                <h2>Seção 2</h2>
                <p>Conteúdo da Seção 2</p>
            </section>
            <section id="coluna3">
                <h2>Seção 3</h2>
                <p>Conteúdo da Seção 3</p>
          Vamos ver se vai <p>
          agora sim <p>não pare
           </section>
        </main>
        <footer>Rodapé</footer>
    </section>  </body>  </html>
```

☞ Para você fazer

Vamos ao exercício da aula.
Construa um HTML único para esta aula. Deve ser um site com o tema

`Uma caminhada pelo centro de Curitiba`

e contendo as seguintes partes:

- um cabeçalho contendo 3 parágrafos explicando o que este exercício vai tratar (sites com 3 blocos flexíveis)
- um bloco `main` no qual estarão os conteúdos.
- o primeiro bloco deve conter a história do tema
- o segundo bloco deve conter o estado atual do tema
- o terceiro bloco é para você especular sobre o futuro do tema
- o rodapé deve conter a sua identificação (nome, turma, curso e colégio).

Depois deste HTML feito e funcionando, é hora de construir 3 arquivos CSS, a saber:

1. Mostre os 3 blocos na vertical.
2. Os blocos devem aparecer lado a lado, com os 3 blocos de mesmo tamanho
3. O primeiro e terceiro bloco devem ocupar 25% do espaço cada um, enquanto o bloco central deve ocupar os 50% restantes.

Use ou não o selector grid. Fica a seu critério. Note que o bloco do meio deve conter pelo menos 3 fotos. Se quiser colocar fotos nos outros blocos fique à vontade.

O ponto mais importante deste exercício é que o HTML deve ser o **mesmo** para os 3 exercícios.

Mostre estas atividades e os arquivos que você criou para o professor ANTES de devolver esta folha.

Avaliação

(preenchido pelo professor)

107-76049 - ga/ a

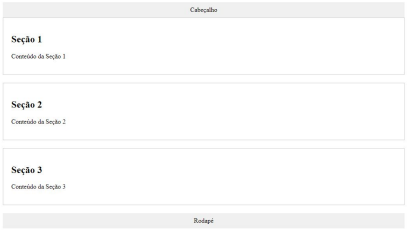
CEP - UP - UTFPR - PUCPr - UFPR – 13/05/2024 - 12:08:57.0
Prof Dr P Kantek (pkantek@gmail.com)
Organizando o laytou com CSS VIVXh53a, V: 1.06 76056 GABRIELA BERNARDO PASQUALOTT 24FRO107 - 7 entregar ate 16/maio _____ / _____ / _____

Organizando o layout com CSS

Como se sabe o CSS permite que um layout seja construído para apresentar os conteúdos na hora de mostrar uma página.

1 coluna

Vamos começar com um layout simples, contendo header, footer e um bloco principal entre eles. Este bloco terá 3 sessões, devidamente empilhadas na vertical, como se pode ver em



Neste caso, o CSS será

```
/* Estilo geral do container */
.container {
    display: flex;
    flex-direction: column; /* Organiza os elementos na vertical */
    width: 100%; /* Ocupa a largura total da tela */
    max-width: 1000px; /* Largura máxima de 1000px */
    margin: 0 auto; /* Centraliza o container na tela */
    padding: 20px;            }
/* Estilos para header e footer */
header, footer {
    background-color: #f0f0f0;
    padding: 10px;
    text-align: center;       }
/* Estilos para as seções */
.secao {
    flex: 1 0 auto; /* Ocupa espaço disponível igualmente */
    padding: 20px;
    border: 1px solid #ccc;
    margin-bottom: 20px;      }
```

O HTML que o acessa é

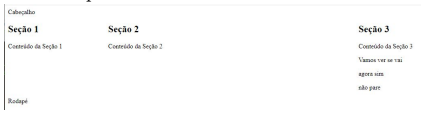
```
<!DOCTYPE html>
<html lang="pt-BR">
<head>
    <meta charset="UTF-8">
    <title>Sistema de Grade Básico</title>
    <link rel="stylesheet" href="stycols.css">
</head>
<body>
    <section class="container">
        <header>Cabeçalho</header>
        <main>
            <section class="secao">
                <h2>Seção 1</h2>
                <p>Conteúdo da Seção 1</p>
            </section>
            <section class="secao">
                <h2>Seção 2</h2>
                <p>Conteúdo da Seção 2</p>
            </section>
            <section class="secao">
                <h2>Seção 3</h2>
                <p>Conteúdo da Seção 3</p>
            </section>
        </main>
        <footer>Rodapé</footer>
    </section>
</body>
</html>
```

Valem as seguintes observações

- O container é definido com display: flex e flex-direction: column para organizar os elementos na vertical.
- A propriedade width: 100% faz com que o container ocupe a largura total da tela.
- max-width: 1000px define uma largura máxima de 1000px para o container, garantindo que ele não fique muito largo em telas maiores.
- margin: 0 auto centraliza o container na tela horizontalmente.
- As seções usam flex: 1 0 auto para ocupar espaço disponível igualmente dentro do container.
- A propriedade padding adiciona espaço interno às seções.
- A borda e a margem inferior estilizam as seções visualmente.

Usando 3 colunas sem grid

Agora vai-se usar um esquema mais simples (sem usar o seletor grid) que é indicado para páginas mais simples. Eis o resultado



Eis o CSS

```
#container {
  width: 100%; /* Define a largura total do container */ }
#coluna1 {
  width: 20%;
  float: left;
  box-sizing: border-box; background-color:#c4d5f6; }
#coluna2 {
  width: 50%;
  float: left;
  box-sizing: border-box; background-color:#a4d3b3; }
#coluna3 {
  width: 30%;
  float: left;
  box-sizing: border-box; background-color:#f1e4a3; }
```

Eis o HTML

```
<!DOCTYPE html>
<html lang="pt-BR">
<head>
    <meta charset="UTF-8">
    <title>Sistema de Grade Básico</title>
    <link rel="stylesheet" href="h53b.css">
</head>  <body>
    <header>Cabeçalho</header>
    <main>
   <section id="container">
         <section id="coluna1">
         <h2>Seção 1</h2>
         <p>Conteúdo da Seção 1</p>
         </section>
         <section id="coluna2">
         <h2>Seção 2</h2>
         <p>Conteúdo da Seção 2</p>
         </section>
         <section id="coluna3">
         <h2>Seção 3</h2>
         <p>Conteúdo da Seção 3</p>
          Vamos ver se vai <p>
          agora sim <p>não pare
         </section>
   </section>
      </main>
      <footer>Rodapé</footer>
</body>  </html>
```

Usando 3 colunas com grid

Com o resultado

Seja o CSS

```
#container {
  display: grid;
  /* Cria 3 colunas com largura igual -- 1fr=1 unid. frac.*/
  grid-template-columns: 1fr 2fr 1.5fr;
  gap: 20px; /* Define o espaçamento entre as colunas */  }
#coluna1{grid-column:1; background-color:#bbaacc;}
#coluna2{grid-column:2; background-color:#ccb4a2;}
#coluna3{grid-column:3; background-color:#d4d3a8;}
```

E o HTML

```
<!DOCTYPE html>
<html lang="pt-BR">
<head>
    <meta charset="UTF-8">
    <title>Sistema de Grade Básico</title>
    <link rel="stylesheet" href="h53a.css">
</head>
<body>
    <section id="container">
        <header>Cabeçalho</header>
        <main>
            <section id="coluna1">
                <h2>Seção 1</h2>
                <p>Conteúdo da Seção 1</p>
            </section>
            <section id="coluna2">
                <h2>Seção 2</h2>
                <p>Conteúdo da Seção 2</p>
            </section>
            <section id="coluna3">
                <h2>Seção 3</h2>
                <p>Conteúdo da Seção 3</p>
           Vamos ver se vai <p>
           agora sim <p>não pare
            </section>
        </main>
        <footer>Rodapé</footer>
    </section>  </body>  </html>
```

☞ Para você fazer

Vamos ao exercício da aula.
Construa um HTML único para esta aula. Deve ser um site com o tema

`A indústria da música`

e contendo as seguintes partes:

- um cabeçalho contendo 3 parágrafos explicando o que este exercício vai tratar (sites com 3 blocos flexíveis)
- um bloco `main` no qual estarão os conteúdos.
- o primeiro bloco deve conter a história do tema
- o segundo bloco deve conter o estado atual do tema
- o terceiro bloco é para você especular sobre o futuro do tema
- o rodapé deve conter a sua identificação (nome, turma, curso e colégio).

Depois deste HTML feito e funcionando, é hora de construir 3 arquivos CSS, a saber:

1. Mostre os 3 blocos na vertical.
2. Os blocos devem aparecer lado a lado, com os 3 blocos de mesmo tamanho
3. O primeiro e terceiro bloco devem ocupar 25% do espaço cada um, enquanto o bloco central deve ocupar os 50% restantes.

Use ou não o selector grid. Fica a seu critério. Note que o bloco do meio deve conter pelo menos 3 fotos. Se quiser colocar fotos nos outros blocos fique à vontade.

O ponto mais importante deste exercício é que o HTML deve ser o **mesmo** para os 3 exercícios.

Mostre estas atividades e os arquivos que você criou para o professor ANTES de devolver esta folha.

Avaliação

(preenchido pelo professor)

107-76056 - ga/ a

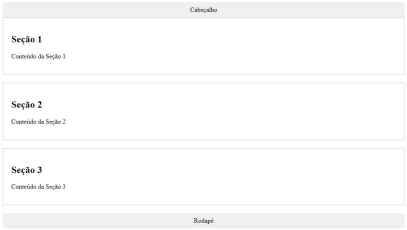
CEP - UP - UTFPR - PUCPr - UFPR – 13/05/2024 - 12:08:57.0
Prof Dr P Kantek (pkantek@gmail.com)
Organizando o laytou com CSS VIVXh53a, V: 1.06 76063 GUILHERME VITOR DA SILVA
24FRO107 - 8 entregar ate 16/maio _____ / / _____

Organizando o layout com CSS

Como se sabe o CSS permite que um layout seja construído para apresentar os conteúdos na hora de mostrar uma página.

1 coluna

Vamos começar com um layout simples, contendo header, footer e um bloco principal entre eles. Este bloco terá 3 sessões, devidamente empilhadas na vertical, como se pode ver em



Neste caso, o CSS será

```
/* Estilo geral do container */
.container {
    display: flex;
    flex-direction: column; /* Organiza os elementos na vertical */
    width: 100%; /* Ocupa a largura total da tela */
    max-width: 1000px; /* Largura máxima de 1000px */
    margin: 0 auto; /* Centraliza o container na tela */
    padding: 20px;            }
/* Estilos para header e footer */
header, footer {
    background-color: #f0f0f0;
    padding: 10px;
    text-align: center;       }
/* Estilos para as seções */
.secao {
    flex: 1 0 auto; /* Ocupa espaço disponível igualmente */
    padding: 20px;
    border: 1px solid #ccc;
    margin-bottom: 20px;      }
```

O HTML que o acessa é

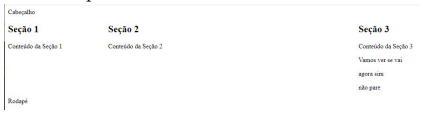
```
<!DOCTYPE html>
<html lang="pt-BR">
<head>
    <meta charset="UTF-8">
    <title>Sistema de Grade Básico</title>
    <link rel="stylesheet" href="stycols.css">
</head>
<body>
    <section class="container">
        <header>Cabeçalho</header>
        <main>
            <section class="secao">
                <h2>Seção 1</h2>
                <p>Conteúdo da Seção 1</p>
            </section>
            <section class="secao">
                <h2>Seção 2</h2>
                <p>Conteúdo da Seção 2</p>
            </section>
            <section class="secao">
                <h2>Seção 3</h2>
                <p>Conteúdo da Seção 3</p>
            </section>
        </main>
        <footer>Rodapé</footer>
    </section>
</body>
</html>
```

Valem as seguintes observações

- O container é definido com display: flex e flex-direction: column para organizar os elementos na vertical.
- A propriedade width: 100% faz com que o container ocupe a largura total da tela.
- max-width: 1000px define uma largura máxima de 1000px para o container, garantindo que ele não fique muito largo em telas maiores.
- margin: 0 auto centraliza o container na tela horizontalmente.
- As seções usam flex: 1 0 auto para ocupar espaço disponível igualmente dentro do container.
- A propriedade padding adiciona espaço interno às seções.
- A borda e a margem inferior estilizam as seções visualmente.

Usando 3 colunas sem grid

Agora vai-se usar um esquema mais simples (sem usar o seletor grid) que é indicado para páginas mais simples. Eis o resultado



Eis o CSS

```
#container {
  width: 100%; /* Define a largura total do container */ }
#coluna1 {
  width: 20%;
  float: left;
  box-sizing: border-box; background-color:#c4d5f6; }
#coluna2 {
  width: 50%;
  float: left;
  box-sizing: border-box; background-color:#a4d3b3; }
#coluna3 {
  width: 30%;
  float: left;
  box-sizing: border-box; background-color:#f1e4a3; }
```

Eis o HTML

```
<!DOCTYPE html>
<html lang="pt-BR">
<head>
    <meta charset="UTF-8">
    <title>Sistema de Grade Básico</title>
    <link rel="stylesheet" href="h53b.css">
</head>  <body>
    <header>Cabeçalho</header>
    <main>
   <section id="container">
         <section id="coluna1">
         <h2>Seção 1</h2>
         <p>Conteúdo da Seção 1</p>
         </section>
         <section id="coluna2">
         <h2>Seção 2</h2>
         <p>Conteúdo da Seção 2</p>
         </section>
         <section id="coluna3">
         <h2>Seção 3</h2>
         <p>Conteúdo da Seção 3</p>
          Vamos ver se vai <p>
          agora sim <p>não pare
         </section>
   </section>
      </main>
      <footer>Rodapé</footer>
</body>  </html>
```

Usando 3 colunas com grid

Com o resultado

Seja o CSS

```
#container {
  display: grid;
  /* Cria 3 colunas com largura igual -- 1fr=1 unid. frac.*/
  grid-template-columns: 1fr 2fr 1.5fr;
  gap: 20px; /* Define o espaçamento entre as colunas */  }
#coluna1{grid-column:1; background-color:#bbaacc;}
#coluna2{grid-column:2; background-color:#ccb4a2;}
#coluna3{grid-column:3; background-color:#d4d3a8;}
```

E o HTML

```
<!DOCTYPE html>
<html lang="pt-BR">
<head>
    <meta charset="UTF-8">
    <title>Sistema de Grade Básico</title>
    <link rel="stylesheet" href="h53a.css">
</head>
<body>
    <section id="container">
        <header>Cabeçalho</header>
        <main>
            <section id="coluna1">
                <h2>Seção 1</h2>
                <p>Conteúdo da Seção 1</p>
            </section>
            <section id="coluna2">
                <h2>Seção 2</h2>
                <p>Conteúdo da Seção 2</p>
            </section>
            <section id="coluna3">
                <h2>Seção 3</h2>
                <p>Conteúdo da Seção 3</p>
           Vamos ver se vai <p>
           agora sim <p>não pare
            </section>
        </main>
        <footer>Rodapé</footer>
    </section>  </body>  </html>
```

☞ Para você fazer

Vamos ao exercício da aula.
Construa um HTML único para esta aula. Deve ser um site com o tema

`A praça Tiradentes de Curitiba`

e contendo as seguintes partes:

- um cabeçalho contendo 3 parágrafos explicando o que este exercício vai tratar (sites com 3 blocos flexíveis)
- um bloco `main` no qual estarão os conteúdos.
- o primeiro bloco deve conter a história do tema
- o segundo bloco deve conter o estado atual do tema
- o terceiro bloco é para você especular sobre o futuro do tema
- o rodapé deve conter a sua identificação (nome, turma, curso e colégio).

Depois deste HTML feito e funcionando, é hora de construir 3 arquivos CSS, a saber:

1. Mostre os 3 blocos na vertical.
2. Os blocos devem aparecer lado a lado, com os 3 blocos de mesmo tamanho
3. O primeiro e terceiro bloco devem ocupar 25% do espaço cada um, enquanto o bloco central deve ocupar os 50% restantes.

Use ou não o selector grid. Fica a seu critério. Note que o bloco do meio deve conter pelo menos 3 fotos. Se quiser colocar fotos nos outros blocos fique à vontade.

O ponto mais importante deste exercício é que o HTML deve ser o **mesmo** para os 3 exercícios.

Mostre estas atividades e os arquivos que você criou para o professor ANTES de devolver esta folha.

Avaliação

(preenchido pelo professor)

107-76063 - ga/ a

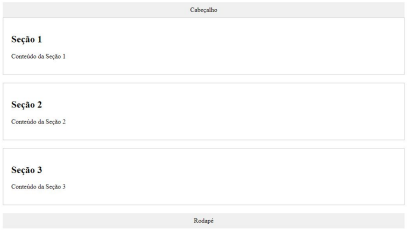
CEP - UP - UTFPR - PUCPr - UFPR – 13/05/2024 - 12:08:57.0
Prof Dr P Kantek (pkantek@gmail.com)
Organizando o laytou com CSS VIVXh53a, V: 1.06 76087 LAURA PACHECO DELATTRE
24FRO107 - 9 entregar ate 16/maio _____ / / _____

Organizando o layout com CSS

Como se sabe o CSS permite que um layout seja construído para apresentar os conteúdos na hora de mostrar uma página.

1 coluna

Vamos começar com um layout simples, contendo header, footer e um bloco principal entre eles. Este bloco terá 3 sessões, devidamente empilhadas na vertical, como se pode ver em



Neste caso, o CSS será

```
/* Estilo geral do container */
.container {
    display: flex;
    flex-direction: column; /* Organiza os elementos na vertical */
    width: 100%; /* Ocupa a largura total da tela */
    max-width: 1000px; /* Largura máxima de 1000px */
    margin: 0 auto; /* Centraliza o container na tela */
    padding: 20px;            }
/* Estilos para header e footer */
header, footer {
    background-color: #f0f0f0;
    padding: 10px;
    text-align: center;       }
/* Estilos para as seções */
.secao {
    flex: 1 0 auto; /* Ocupa espaço disponível igualmente */
    padding: 20px;
    border: 1px solid #ccc;
    margin-bottom: 20px;      }
```

O HTML que o acessa é

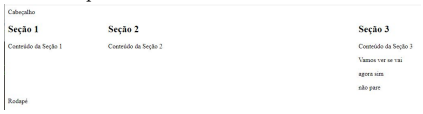
```
<!DOCTYPE html>
<html lang="pt-BR">
<head>
    <meta charset="UTF-8">
    <title>Sistema de Grade Básico</title>
    <link rel="stylesheet" href="stycols.css">
</head>
<body>
    <section class="container">
        <header>Cabeçalho</header>
        <main>
            <section class="secao">
                <h2>Seção 1</h2>
                <p>Conteúdo da Seção 1</p>
            </section>
            <section class="secao">
                <h2>Seção 2</h2>
                <p>Conteúdo da Seção 2</p>
            </section>
            <section class="secao">
                <h2>Seção 3</h2>
                <p>Conteúdo da Seção 3</p>
            </section>
        </main>
        <footer>Rodapé</footer>
    </section>
</body>
</html>
```

Valem as seguintes observações

- O container é definido com display: flex e flex-direction: column para organizar os elementos na vertical.
- A propriedade width: 100% faz com que o container ocupe a largura total da tela.
- max-width: 1000px define uma largura máxima de 1000px para o container, garantindo que ele não fique muito largo em telas maiores.
- margin: 0 auto centraliza o container na tela horizontalmente.
- As seções usam flex: 1 0 auto para ocupar espaço disponível igualmente dentro do container.
- A propriedade padding adiciona espaço interno às seções.
- A borda e a margem inferior estilizam as seções visualmente.

Usando 3 colunas sem grid

Agora vai-se usar um esquema mais simples (sem usar o seletor grid) que é indicado para páginas mais simples. Eis o resultado



Eis o CSS

```
#container {
  width: 100%; /* Define a largura total do container */ }
#coluna1 {
  width: 20%;
  float: left;
  box-sizing: border-box; background-color:#c4d5f6; }
#coluna2 {
  width: 50%;
  float: left;
  box-sizing: border-box; background-color:#a4d3b3; }
#coluna3 {
  width: 30%;
  float: left;
  box-sizing: border-box; background-color:#f1e4a3; }
```

Eis o HTML

```
<!DOCTYPE html>
<html lang="pt-BR">
<head>
    <meta charset="UTF-8">
    <title>Sistema de Grade Básico</title>
    <link rel="stylesheet" href="h53b.css">
</head>  <body>
    <header>Cabeçalho</header>
    <main>
   <section id="container">
         <section id="coluna1">
         <h2>Seção 1</h2>
         <p>Conteúdo da Seção 1</p>
         </section>
         <section id="coluna2">
         <h2>Seção 2</h2>
         <p>Conteúdo da Seção 2</p>
         </section>
         <section id="coluna3">
         <h2>Seção 3</h2>
         <p>Conteúdo da Seção 3</p>
          Vamos ver se vai <p>
          agora sim <p>não pare
         </section>
   </section>
      </main>
      <footer>Rodapé</footer>
</body>  </html>
```

Usando 3 colunas com grid

Com o resultado

Seja o CSS

```
#container {
  display: grid;
  /* Cria 3 colunas com largura igual -- 1fr=1 unid. frac.*/
  grid-template-columns: 1fr 2fr 1.5fr;
  gap: 20px; /* Define o espaçamento entre as colunas */  }
#coluna1{grid-column:1; background-color:#bbaacc;}
#coluna2{grid-column:2; background-color:#ccb4a2;}
#coluna3{grid-column:3; background-color:#d4d3a8;}
```

E o HTML

```
<!DOCTYPE html>
<html lang="pt-BR">
<head>
    <meta charset="UTF-8">
    <title>Sistema de Grade Básico</title>
    <link rel="stylesheet" href="h53a.css">
</head>
<body>
    <section id="container">
        <header>Cabeçalho</header>
        <main>
            <section id="coluna1">
                <h2>Seção 1</h2>
                <p>Conteúdo da Seção 1</p>
            </section>
            <section id="coluna2">
                <h2>Seção 2</h2>
                <p>Conteúdo da Seção 2</p>
            </section>
            <section id="coluna3">
                <h2>Seção 3</h2>
                <p>Conteúdo da Seção 3</p>
           Vamos ver se vai <p>
           agora sim <p>não pare
            </section>
        </main>
        <footer>Rodapé</footer>
    </section>  </body>  </html>
```

☞ Para você fazer

Vamos ao exercício da aula.
Construa um HTML único para esta aula. Deve ser um site com o tema

`O bairro do Portão em Curitiba`

e contendo as seguintes partes:

- um cabeçalho contendo 3 parágrafos explicando o que este exercício vai tratar (sites com 3 blocos flexíveis)
- um bloco `main` no qual estarão os conteúdos.
- o primeiro bloco deve conter a história do tema
- o segundo bloco deve conter o estado atual do tema
- o terceiro bloco é para você especular sobre o futuro do tema
- o rodapé deve conter a sua identificação (nome, turma, curso e colégio).

Depois deste HTML feito e funcionando, é hora de construir 3 arquivos CSS, a saber:

1. Mostre os 3 blocos na vertical.
2. Os blocos devem aparecer lado a lado, com os 3 blocos de mesmo tamanho
3. O primeiro e terceiro bloco devem ocupar 25% do espaço cada um, enquanto o bloco central deve ocupar os 50% restantes.

Use ou não o selector grid. Fica a seu critério. Note que o bloco do meio deve conter pelo menos 3 fotos. Se quiser colocar fotos nos outros blocos fique à vontade.

O ponto mais importante deste exercício é que o HTML deve ser o **mesmo** para os 3 exercícios.

Mostre estas atividades e os arquivos que você criou para o professor ANTES de devolver esta folha.

Avaliação

(preenchido pelo professor)

107-76087 - ga/ a

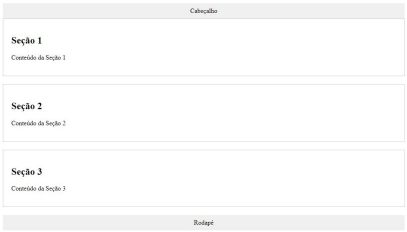
CEP - UP - UTFPR - PUCPr - UFPR – 13/05/2024 - 12:08:57.0
Prof Dr P Kantek (pkantek@gmail.com)
Organizando o laytou com CSS VIVXh53a, V: 1.06 76106 LUIZ MIGUEL OLINEK 24FRO107 - 10 entregar ate 16/maio _____ / _____ / _____

Organizando o layout com CSS

Como se sabe o CSS permite que um layout seja construído para apresentar os conteúdos na hora de mostrar uma página.

1 coluna

Vamos começar com um layout simples, contendo header, footer e um bloco principal entre eles. Este bloco terá 3 sessões, devidamente empilhadas na vertical, como se pode ver em



Neste caso, o CSS será

```
/* Estilo geral do container */
.container {
    display: flex;
    flex-direction: column; /* Organiza os elementos na vertical */
    width: 100%; /* Ocupa a largura total da tela */
    max-width: 1000px; /* Largura máxima de 1000px */
    margin: 0 auto; /* Centraliza o container na tela */
    padding: 20px;            }
/* Estilos para header e footer */
header, footer {
    background-color: #f0f0f0;
    padding: 10px;
    text-align: center;       }
/* Estilos para as seções */
.secao {
    flex: 1 0 auto; /* Ocupa espaço disponível igualmente */
    padding: 20px;
    border: 1px solid #ccc;
    margin-bottom: 20px;      }
```

O HTML que o acessa é

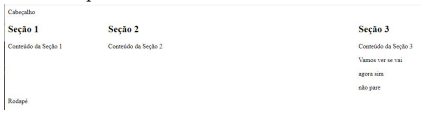
```
<!DOCTYPE html>
<html lang="pt-BR">
<head>
    <meta charset="UTF-8">
    <title>Sistema de Grade Básico</title>
    <link rel="stylesheet" href="stycols.css">
</head>
<body>
    <section class="container">
        <header>Cabeçalho</header>
        <main>
            <section class="secao">
                <h2>Seção 1</h2>
                <p>Conteúdo da Seção 1</p>
            </section>
            <section class="secao">
                <h2>Seção 2</h2>
                <p>Conteúdo da Seção 2</p>
            </section>
            <section class="secao">
                <h2>Seção 3</h2>
                <p>Conteúdo da Seção 3</p>
            </section>
        </main>
        <footer>Rodapé</footer>
    </section>
</body>
</html>
```

Valem as seguintes observações

- O container é definido com display: flex e flex-direction: column para organizar os elementos na vertical.
- A propriedade width: 100% faz com que o container ocupe a largura total da tela.
- max-width: 1000px define uma largura máxima de 1000px para o container, garantindo que ele não fique muito largo em telas maiores.
- margin: 0 auto centraliza o container na tela horizontalmente.
- As seções usam flex: 1 0 auto para ocupar espaço disponível igualmente dentro do container.
- A propriedade padding adiciona espaço interno às seções.
- A borda e a margem inferior estilizam as seções visualmente.

Usando 3 colunas sem grid

Agora vai-se usar um esquema mais simples (sem usar o seletor grid) que é indicado para páginas mais simples. Eis o resultado



Eis o CSS

```
#container {
  width: 100%; /* Define a largura total do container */ }
#coluna1 {
  width: 20%;
  float: left;
  box-sizing: border-box; background-color:#c4d5f6; }
#coluna2 {
  width: 50%;
  float: left;
  box-sizing: border-box; background-color:#a4d3b3; }
#coluna3 {
  width: 30%;
  float: left;
  box-sizing: border-box; background-color:#f1e4a3; }
```

Eis o HTML

```
<!DOCTYPE html>
<html lang="pt-BR">
<head>
    <meta charset="UTF-8">
    <title>Sistema de Grade Básico</title>
    <link rel="stylesheet" href="h53b.css">
</head>  <body>
    <header>Cabeçalho</header>
    <main>
   <section id="container">
         <section id="coluna1">
         <h2>Seção 1</h2>
         <p>Conteúdo da Seção 1</p>
         </section>
         <section id="coluna2">
         <h2>Seção 2</h2>
         <p>Conteúdo da Seção 2</p>
         </section>
         <section id="coluna3">
         <h2>Seção 3</h2>
         <p>Conteúdo da Seção 3</p>
          Vamos ver se vai <p>
          agora sim <p>não pare
         </section>
   </section>
      </main>
      <footer>Rodapé</footer>
</body>  </html>
```

Usando 3 colunas com grid

Com o resultado

Seja o CSS

```
#container {
  display: grid;
  /* Cria 3 colunas com largura igual -- 1fr=1 unid. frac.*/
  grid-template-columns: 1fr 2fr 1.5fr;
  gap: 20px; /* Define o espaçamento entre as colunas */  }
#coluna1{grid-column:1; background-color:#bbaacc;}
#coluna2{grid-column:2; background-color:#ccb4a2;}
#coluna3{grid-column:3; background-color:#d4d3a8;}
```

E o HTML

```
<!DOCTYPE html>
<html lang="pt-BR">
<head>
    <meta charset="UTF-8">
    <title>Sistema de Grade Básico</title>
    <link rel="stylesheet" href="h53a.css">
</head>
<body>
    <section id="container">
        <header>Cabeçalho</header>
        <main>
            <section id="coluna1">
                <h2>Seção 1</h2>
                <p>Conteúdo da Seção 1</p>
            </section>
            <section id="coluna2">
                <h2>Seção 2</h2>
                <p>Conteúdo da Seção 2</p>
            </section>
            <section id="coluna3">
                <h2>Seção 3</h2>
                <p>Conteúdo da Seção 3</p>
          Vamos ver se vai <p>
          agora sim <p>não pare
            </section>
        </main>
        <footer>Rodapé</footer>
    </section>  </body>  </html>
```

☞ Para você fazer

Vamos ao exercício da aula.
Construa um HTML único para esta aula. Deve ser um site com o tema

`O bairro do Bacacheri em Curitiba`

e contendo as seguintes partes:

- um cabeçalho contendo 3 parágrafos explicando o que este exercício vai tratar (sites com 3 blocos flexíveis)
- um bloco `main` no qual estarão os conteúdos.
- o primeiro bloco deve conter a história do tema
- o segundo bloco deve conter o estado atual do tema
- o terceiro bloco é para você especular sobre o futuro do tema
- o rodapé deve conter a sua identificação (nome, turma, curso e colégio).

Depois deste HTML feito e funcionando, é hora de construir 3 arquivos CSS, a saber:

1. Mostre os 3 blocos na vertical.
2. Os blocos devem aparecer lado a lado, com os 3 blocos de mesmo tamanho
3. O primeiro e terceiro bloco devem ocupar 25% do espaço cada um, enquanto o bloco central deve ocupar os 50% restantes.

Use ou não o selector grid. Fica a seu critério. Note que o bloco do meio deve conter pelo menos 3 fotos. Se quiser colocar fotos nos outros blocos fique à vontade.

O ponto mais importante deste exercício é que o HTML deve ser o **mesmo** para os 3 exercícios.

Mostre estas atividades e os arquivos que você criou para o professor ANTES de devolver esta folha.

Avaliação

(preenchido pelo professor)

107-76106 - ga/ a

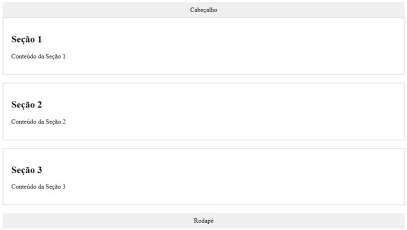
CEP - UP - UTFPR - PUCPr - UFPR – 13/05/2024 - 12:08:57.0
Prof Dr P Kantek (pkantek@gmail.com)
Organizando o laytou com CSS VIVXh53a, V: 1.06 76113 NICOLE DA SILVA DE AZEVEDO
24FRO107 - 11 entregar ate 16/maio _____ / / _____

Organizando o layout com CSS

Como se sabe o CSS permite que um layout seja construído para apresentar os conteúdos na hora de mostrar uma página.

1 coluna

Vamos começar com um layout simples, contendo header, footer e um bloco principal entre eles. Este bloco terá 3 sessões, devidamente empilhadas na vertical, como se pode ver em



Neste caso, o CSS será

```
/* Estilo geral do container */
.container {
    display: flex;
    flex-direction: column; /* Organiza os elementos na vertical */
    width: 100%; /* Ocupa a largura total da tela */
    max-width: 1000px; /* Largura máxima de 1000px */
    margin: 0 auto; /* Centraliza o container na tela */
    padding: 20px;            }
/* Estilos para header e footer */
header, footer {
    background-color: #f0f0f0;
    padding: 10px;
    text-align: center;       }
/* Estilos para as seções */
.secao {
    flex: 1 0 auto; /* Ocupa espaço disponível igualmente */
    padding: 20px;
    border: 1px solid #ccc;
    margin-bottom: 20px;      }
```

O HTML que o acessa é

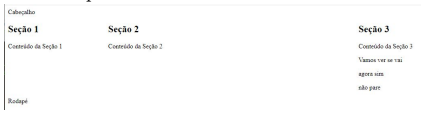
```
<!DOCTYPE html>
<html lang="pt-BR">
<head>
    <meta charset="UTF-8">
    <title>Sistema de Grade Básico</title>
    <link rel="stylesheet" href="stycols.css">
</head>
<body>
    <section class="container">
        <header>Cabeçalho</header>
        <main>
            <section class="secao">
                <h2>Seção 1</h2>
                <p>Conteúdo da Seção 1</p>
            </section>
            <section class="secao">
                <h2>Seção 2</h2>
                <p>Conteúdo da Seção 2</p>
            </section>
            <section class="secao">
                <h2>Seção 3</h2>
                <p>Conteúdo da Seção 3</p>
            </section>
        </main>
        <footer>Rodapé</footer>
    </section>
</body>
</html>
```

Valem as seguintes observações

- O container é definido com display: flex e flex-direction: column para organizar os elementos na vertical.
- A propriedade width: 100% faz com que o container ocupe a largura total da tela.
- max-width: 1000px define uma largura máxima de 1000px para o container, garantindo que ele não fique muito largo em telas maiores.
- margin: 0 auto centraliza o container na tela horizontalmente.
- As seções usam flex: 1 0 auto para ocupar espaço disponível igualmente dentro do container.
- A propriedade padding adiciona espaço interno às seções.
- A borda e a margem inferior estilizam as seções visualmente.

Usando 3 colunas sem grid

Agora vai-se usar um esquema mais simples (sem usar o seletor grid) que é indicado para páginas mais simples. Eis o resultado



Eis o CSS

```
#container {
  width: 100%; /* Define a largura total do container */ }
#coluna1 {
  width: 20%;
  float: left;
  box-sizing: border-box; background-color:#c4d5f6; }
#coluna2 {
  width: 50%;
  float: left;
  box-sizing: border-box; background-color:#a4d3b3; }
#coluna3 {
  width: 30%;
  float: left;
  box-sizing: border-box; background-color:#f1e4a3; }
```

Eis o HTML

```
<!DOCTYPE html>
<html lang="pt-BR">
<head>
    <meta charset="UTF-8">
    <title>Sistema de Grade Básico</title>
    <link rel="stylesheet" href="h53b.css">
</head>  <body>
    <header>Cabeçalho</header>
    <main>
   <section id="container">
         <section id="coluna1">
         <h2>Seção 1</h2>
         <p>Conteúdo da Seção 1</p>
         </section>
         <section id="coluna2">
         <h2>Seção 2</h2>
         <p>Conteúdo da Seção 2</p>
         </section>
         <section id="coluna3">
         <h2>Seção 3</h2>
         <p>Conteúdo da Seção 3</p>
          Vamos ver se vai <p>
          agora sim <p>não pare
         </section>
   </section>
      </main>
      <footer>Rodapé</footer>
</body>  </html>
```

Usando 3 colunas com grid

Com o resultado

Seja o CSS

```
#container {
  display: grid;
  /* Cria 3 colunas com largura igual -- 1fr=1 unid. frac.*/
  grid-template-columns: 1fr 2fr 1.5fr;
  gap: 20px; /* Define o espaçamento entre as colunas */  }
#coluna1{grid-column:1; background-color:#bbaacc;}
#coluna2{grid-column:2; background-color:#ccb4a2;}
#coluna3{grid-column:3; background-color:#d4d3a8;}
```

E o HTML

```
<!DOCTYPE html>
<html lang="pt-BR">
<head>
    <meta charset="UTF-8">
    <title>Sistema de Grade Básico</title>
    <link rel="stylesheet" href="h53a.css">
</head>
<body>
    <section id="container">
        <header>Cabeçalho</header>
        <main>
            <section id="coluna1">
                <h2>Seção 1</h2>
                <p>Conteúdo da Seção 1</p>
            </section>
            <section id="coluna2">
                <h2>Seção 2</h2>
                <p>Conteúdo da Seção 2</p>
            </section>
            <section id="coluna3">
                <h2>Seção 3</h2>
                <p>Conteúdo da Seção 3</p>
           Vamos ver se vai <p>
           agora sim <p>não pare
            </section>
        </main>
        <footer>Rodapé</footer>
    </section>  </body>  </html>
```

☞ Para você fazer

Vamos ao exercício da aula.
Construa um HTML único para esta aula. Deve ser um site com o tema

`O bairro do Portão em Curitiba`

e contendo as seguintes partes:

- um cabeçalho contendo 3 parágrafos explicando o que este exercício vai tratar (sites com 3 blocos flexíveis)
- um bloco `main` no qual estarão os conteúdos.
- o primeiro bloco deve conter a história do tema
- o segundo bloco deve conter o estado atual do tema
- o terceiro bloco é para você especular sobre o futuro do tema
- o rodapé deve conter a sua identificação (nome, turma, curso e colégio).

Depois deste HTML feito e funcionando, é hora de construir 3 arquivos CSS, a saber:

1. Mostre os 3 blocos na vertical.
2. Os blocos devem aparecer lado a lado, com os 3 blocos de mesmo tamanho
3. O primeiro e terceiro bloco devem ocupar 25% do espaço cada um, enquanto o bloco central deve ocupar os 50% restantes.

Use ou não o selector grid. Fica a seu critério. Note que o bloco do meio deve conter pelo menos 3 fotos. Se quiser colocar fotos nos outros blocos fique à vontade.

O ponto mais importante deste exercício é que o HTML deve ser o **mesmo** para os 3 exercícios.

Mostre estas atividades e os arquivos que você criou para o professor ANTES de devolver esta folha.

Avaliação

(preenchido pelo professor)

107-76113 - ga/ a

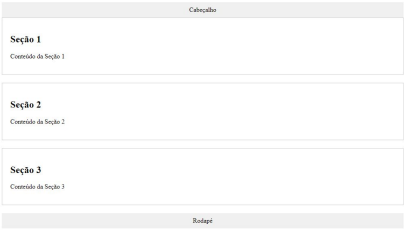
CEP - UP - UTFPR - PUCPr - UFPR – 13/05/2024 - 12:08:57.0
Prof Dr P Kantek (pkantek@gmail.com)
Organizando o laytou com CSS VIVXh53a, V: 1.06 76120 PAULO JOSE FREIRE CORREA
24FRO107 - 12 entregar ate 16/maio _____ / / _____

Organizando o layout com CSS

Como se sabe o CSS permite que um layout seja construído para apresentar os conteúdos na hora de mostrar uma página.

1 coluna

Vamos começar com um layout simples, contendo header, footer e um bloco principal entre eles. Este bloco terá 3 sessões, devidamente empilhadas na vertical, como se pode ver em



Neste caso, o CSS será

```
/* Estilo geral do container */
.container {
    display: flex;
    flex-direction: column; /* Organiza os elementos na vertical */
    width: 100%; /* Ocupa a largura total da tela */
    max-width: 1000px; /* Largura máxima de 1000px */
    margin: 0 auto; /* Centraliza o container na tela */
    padding: 20px;            }
/* Estilos para header e footer */
header, footer {
    background-color: #f0f0f0;
    padding: 10px;
    text-align: center;       }
/* Estilos para as seções */
.secao {
    flex: 1 0 auto; /* Ocupa espaço disponível igualmente */
    padding: 20px;
    border: 1px solid #ccc;
    margin-bottom: 20px;      }
```

O HTML que o acessa é

```
<!DOCTYPE html>
<html lang="pt-BR">
<head>
    <meta charset="UTF-8">
    <title>Sistema de Grade Básico</title>
    <link rel="stylesheet" href="stycols.css">
</head>
<body>
    <section class="container">
        <header>Cabeçalho</header>
        <main>
            <section class="secao">
                <h2>Seção 1</h2>
                <p>Conteúdo da Seção 1</p>
            </section>
            <section class="secao">
                <h2>Seção 2</h2>
                <p>Conteúdo da Seção 2</p>
            </section>
            <section class="secao">
                <h2>Seção 3</h2>
                <p>Conteúdo da Seção 3</p>
            </section>
        </main>
        <footer>Rodapé</footer>
    </section>
</body>
</html>
```

Valem as seguintes observações

- O container é definido com display: flex e flex-direction: column para organizar os elementos na vertical.
- A propriedade width: 100% faz com que o container ocupe a largura total da tela.
- max-width: 1000px define uma largura máxima de 1000px para o container, garantindo que ele não fique muito largo em telas maiores.
- margin: 0 auto centraliza o container na tela horizontalmente.
- As seções usam flex: 1 0 auto para ocupar espaço disponível igualmente dentro do container.
- A propriedade padding adiciona espaço interno às seções.
- A borda e a margem inferior estilizam as seções visualmente.

Usando 3 colunas sem grid

Agora vai-se usar um esquema mais simples (sem usar o seletor grid) que é indicado para páginas mais simples. Eis o resultado

Eis o CSS

```
#container {
  width: 100%; /* Define a largura total do container */ }
#coluna1 {
  width: 20%;
  float: left;
  box-sizing: border-box; background-color:#c4d5f6; }
#coluna2 {
  width: 50%;
  float: left;
  box-sizing: border-box; background-color:#a4d3b3; }
#coluna3 {
  width: 30%;
  float: left;
  box-sizing: border-box; background-color:#f1e4a3; }
```

Eis o HTML

```
<!DOCTYPE html>
<html lang="pt-BR">
<head>
    <meta charset="UTF-8">
    <title>Sistema de Grade Básico</title>
    <link rel="stylesheet" href="h53b.css">
</head>  <body>
    <header>Cabeçalho</header>
    <main>
   <section id="container">
         <section id="coluna1">
         <h2>Seção 1</h2>
         <p>Conteúdo da Seção 1</p>
         </section>
         <section id="coluna2">
         <h2>Seção 2</h2>
         <p>Conteúdo da Seção 2</p>
         </section>
         <section id="coluna3">
         <h2>Seção 3</h2>
         <p>Conteúdo da Seção 3</p>
          Vamos ver se vai <p>
          agora sim <p>não pare
         </section>
   </section>
      </main>
      <footer>Rodapé</footer>
</body>  </html>
```

Usando 3 colunas com grid

Com o resultado

Seja o CSS

```
#container {
  display: grid;
  /* Cria 3 colunas com largura igual -- 1fr=1 unid. frac.*/
  grid-template-columns: 1fr 2fr 1.5fr;
  gap: 20px; /* Define o espaçamento entre as colunas */  }
#coluna1{grid-column:1; background-color:#bbaacc;}
#coluna2{grid-column:2; background-color:#ccb4a2;}
#coluna3{grid-column:3; background-color:#d4d3a8;}
```

E o HTML

```
<!DOCTYPE html>
<html lang="pt-BR">
<head>
    <meta charset="UTF-8">
    <title>Sistema de Grade Básico</title>
    <link rel="stylesheet" href="h53a.css">
</head>
<body>
    <section id="container">
        <header>Cabeçalho</header>
        <main>
            <section id="coluna1">
                <h2>Seção 1</h2>
                <p>Conteúdo da Seção 1</p>
            </section>
            <section id="coluna2">
                <h2>Seção 2</h2>
                <p>Conteúdo da Seção 2</p>
            </section>
            <section id="coluna3">
                <h2>Seção 3</h2>
                <p>Conteúdo da Seção 3</p>
           Vamos ver se vai <p>
           agora sim <p>não pare
            </section>
        </main>
        <footer>Rodapé</footer>
    </section>  </body>  </html>
```

☞ Para você fazer

Vamos ao exercício da aula.
Construa um HTML único para esta aula. Deve ser um site com o tema

`Uma caminhada pelo centro de Curitiba`

e contendo as seguintes partes:

- um cabeçalho contendo 3 parágrafos explicando o que este exercício vai tratar (sites com 3 blocos flexíveis)
- um bloco `main` no qual estarão os conteúdos.
- o primeiro bloco deve conter a história do tema
- o segundo bloco deve conter o estado atual do tema
- o terceiro bloco é para você especular sobre o futuro do tema
- o rodapé deve conter a sua identificação (nome, turma, curso e colégio).

Depois deste HTML feito e funcionando, é hora de construir 3 arquivos CSS, a saber:

1. Mostre os 3 blocos na vertical.
2. Os blocos devem aparecer lado a lado, com os 3 blocos de mesmo tamanho
3. O primeiro e terceiro bloco devem ocupar 25% do espaço cada um, enquanto o bloco central deve ocupar os 50% restantes.

Use ou não o selector grid. Fica a seu critério. Note que o bloco do meio deve conter pelo menos 3 fotos. Se quiser colocar fotos nos outros blocos fique à vontade.

O ponto mais importante deste exercício é que o HTML deve ser o **mesmo** para os 3 exercícios.

Mostre estas atividades e os arquivos que você criou para o professor ANTES de devolver esta folha.

Avaliação

(preenchido pelo professor)

107-76120 - ga/ a

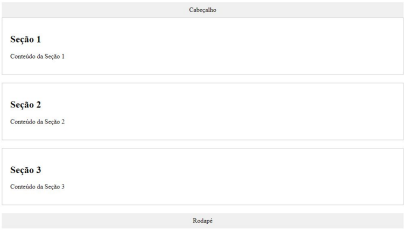
CEP - UP - UTFPR - PUCPr - UFPR – 13/05/2024 - 12:08:57.0
Prof Dr P Kantek (pkantek@gmail.com)
Organizando o laytou com CSS VIVXh53a, V: 1.06 76137 PEDRO HENRIQUE PIEKARSKI DE 24FRO107 - 13 entregar ate 16/maio _____ / / _____

Organizando o layout com CSS

Como se sabe o CSS permite que um layout seja construído para apresentar os conteúdos na hora de mostrar uma página.

1 coluna

Vamos começar com um layout simples, contendo header, footer e um bloco principal entre eles. Este bloco terá 3 sessões, devidamente empilhadas na vertical, como se pode ver em



Neste caso, o CSS será

```
/* Estilo geral do container */
.container {
    display: flex;
    flex-direction: column; /* Organiza os elementos na vertical */
    width: 100%; /* Ocupa a largura total da tela */
    max-width: 1000px; /* Largura máxima de 1000px */
    margin: 0 auto; /* Centraliza o container na tela */
    padding: 20px;            }
/* Estilos para header e footer */
header, footer {
    background-color: #f0f0f0;
    padding: 10px;
    text-align: center;       }
/* Estilos para as seções */
.secao {
    flex: 1 0 auto; /* Ocupa espaço disponível igualmente */
    padding: 20px;
    border: 1px solid #ccc;
    margin-bottom: 20px;      }
```

O HTML que o acessa é

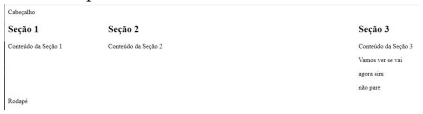
```
<!DOCTYPE html>
<html lang="pt-BR">
<head>
    <meta charset="UTF-8">
    <title>Sistema de Grade Básico</title>
    <link rel="stylesheet" href="stycols.css">
</head>
<body>
    <section class="container">
        <header>Cabeçalho</header>
        <main>
            <section class="secao">
                <h2>Seção 1</h2>
                <p>Conteúdo da Seção 1</p>
            </section>
            <section class="secao">
                <h2>Seção 2</h2>
                <p>Conteúdo da Seção 2</p>
            </section>
            <section class="secao">
                <h2>Seção 3</h2>
                <p>Conteúdo da Seção 3</p>
            </section>
        </main>
        <footer>Rodapé</footer>
    </section>
</body>
</html>
```

Valem as seguintes observações

- O container é definido com display: flex e flex-direction: column para organizar os elementos na vertical.
- A propriedade width: 100% faz com que o container ocupe a largura total da tela.
- max-width: 1000px define uma largura máxima de 1000px para o container, garantindo que ele não fique muito largo em telas maiores.
- margin: 0 auto centraliza o container na tela horizontalmente.
- As seções usam flex: 1 0 auto para ocupar espaço disponível igualmente dentro do container.
- A propriedade padding adiciona espaço interno às seções.
- A borda e a margem inferior estilizam as seções visualmente.

Usando 3 colunas sem grid

Agora vai-se usar um esquema mais simples (sem usar o seletor grid) que é indicado para páginas mais simples. Eis o resultado



Eis o CSS

```
#container {
  width: 100%; /* Define a largura total do container */ }
#coluna1 {
  width: 20%;
  float: left;
  box-sizing: border-box; background-color:#c4d5f6; }
#coluna2 {
  width: 50%;
  float: left;
  box-sizing: border-box; background-color:#a4d3b3; }
#coluna3 {
  width: 30%;
  float: left;
  box-sizing: border-box; background-color:#f1e4a3; }
```

Eis o HTML

```
<!DOCTYPE html>
<html lang="pt-BR">
<head>
    <meta charset="UTF-8">
    <title>Sistema de Grade Básico</title>
    <link rel="stylesheet" href="h53b.css">
</head>  <body>
    <header>Cabeçalho</header>
    <main>
   <section id="container">
         <section id="coluna1">
         <h2>Seção 1</h2>
         <p>Conteúdo da Seção 1</p>
         </section>
         <section id="coluna2">
         <h2>Seção 2</h2>
         <p>Conteúdo da Seção 2</p>
         </section>
         <section id="coluna3">
         <h2>Seção 3</h2>
         <p>Conteúdo da Seção 3</p>
          Vamos ver se vai <p>
          agora sim <p>não pare
         </section>
   </section>
      </main>
      <footer>Rodapé</footer>
</body>  </html>
```

Usando 3 colunas com grid

Com o resultado

Seja o CSS

```
#container {
  display: grid;
  /* Cria 3 colunas com largura igual -- 1fr=1 unid. frac.*/
  grid-template-columns: 1fr 2fr 1.5fr;
  gap: 20px; /* Define o espaçamento entre as colunas */  }
#coluna1{grid-column:1; background-color:#bbaacc;}
#coluna2{grid-column:2; background-color:#ccb4a2;}
#coluna3{grid-column:3; background-color:#d4d3a8;}
```

E o HTML

```
<!DOCTYPE html>
<html lang="pt-BR">
<head>
    <meta charset="UTF-8">
    <title>Sistema de Grade Básico</title>
    <link rel="stylesheet" href="h53a.css">
</head>
<body>
    <section id="container">
        <header>Cabeçalho</header>
        <main>
            <section id="coluna1">
                <h2>Seção 1</h2>
                <p>Conteúdo da Seção 1</p>
            </section>
            <section id="coluna2">
                <h2>Seção 2</h2>
                <p>Conteúdo da Seção 2</p>
            </section>
            <section id="coluna3">
                <h2>Seção 3</h2>
                <p>Conteúdo da Seção 3</p>
           Vamos ver se vai <p>
           agora sim <p>não pare
            </section>
        </main>
        <footer>Rodapé</footer>
    </section>  </body>  </html>
```

☞ Para você fazer

Vamos ao exercício da aula.
Construa um HTML único para esta aula. Deve ser um site com o tema

`A indústria da música`

e contendo as seguintes partes:

- um cabeçalho contendo 3 parágrafos explicando o que este exercício vai tratar (sites com 3 blocos flexíveis)
- um bloco `main` no qual estarão os conteúdos.
- o primeiro bloco deve conter a história do tema
- o segundo bloco deve conter o estado atual do tema
- o terceiro bloco é para você especular sobre o futuro do tema
- o rodapé deve conter a sua identificação (nome, turma, curso e colégio).

Depois deste HTML feito e funcionando, é hora de construir 3 arquivos CSS, a saber:

1. Mostre os 3 blocos na vertical.
2. Os blocos devem aparecer lado a lado, com os 3 blocos de mesmo tamanho
3. O primeiro e terceiro bloco devem ocupar 25% do espaço cada um, enquanto o bloco central deve ocupar os 50% restantes.

Use ou não o selector grid. Fica a seu critério. Note que o bloco do meio deve conter pelo menos 3 fotos. Se quiser colocar fotos nos outros blocos fique à vontade.

O ponto mais importante deste exercício é que o HTML deve ser o **mesmo** para os 3 exercícios.

Mostre estas atividades e os arquivos que você criou para o professor ANTES de devolver esta folha.

Avaliação

(preenchido pelo professor)

107-76137 - ga/ a

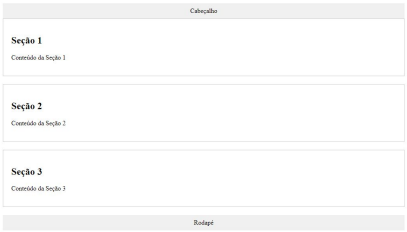
CEP - UP - UTFPR - PUCPr - UFPR – 13/05/2024 - 12:08:57.0
Prof Dr P Kantek (pkantek@gmail.com)
Organizando o laytou com CSS VIVXh53a, V: 1.06 76144 RAUL KNAPKI CUNHA 24FRO107 - 14 entregar ate 16/maio _____ / _____ / _____

Organizando o layout com CSS

Como se sabe o CSS permite que um layout seja construído para apresentar os conteúdos na hora de mostrar uma página.

1 coluna

Vamos começar com um layout simples, contendo header, footer e um bloco principal entre eles. Este bloco terá 3 sessões, devidamente empilhadas na vertical, como se pode ver em



Neste caso, o CSS será

```
/* Estilo geral do container */
.container {
    display: flex;
    flex-direction: column; /* Organiza os elementos na vertical */
    width: 100%; /* Ocupa a largura total da tela */
    max-width: 1000px; /* Largura máxima de 1000px */
    margin: 0 auto; /* Centraliza o container na tela */
    padding: 20px;            }
/* Estilos para header e footer */
header, footer {
    background-color: #f0f0f0;
    padding: 10px;
    text-align: center;       }
/* Estilos para as seções */
.secao {
    flex: 1 0 auto; /* Ocupa espaço disponível igualmente */
    padding: 20px;
    border: 1px solid #ccc;
    margin-bottom: 20px;      }
```

O HTML que o acessa é

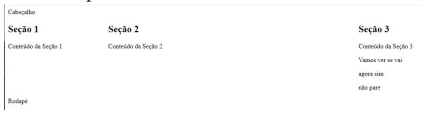
```
<!DOCTYPE html>
<html lang="pt-BR">
<head>
    <meta charset="UTF-8">
    <title>Sistema de Grade Básico</title>
    <link rel="stylesheet" href="stycols.css">
</head>
<body>
    <section class="container">
        <header>Cabeçalho</header>
        <main>
            <section class="secao">
                <h2>Seção 1</h2>
                <p>Conteúdo da Seção 1</p>
            </section>
            <section class="secao">
                <h2>Seção 2</h2>
                <p>Conteúdo da Seção 2</p>
            </section>
            <section class="secao">
                <h2>Seção 3</h2>
                <p>Conteúdo da Seção 3</p>
            </section>
        </main>
        <footer>Rodapé</footer>
    </section>
</body>
</html>
```

Valem as seguintes observações

- O container é definido com display: flex e flex-direction: column para organizar os elementos na vertical.
- A propriedade width: 100% faz com que o container ocupe a largura total da tela.
- max-width: 1000px define uma largura máxima de 1000px para o container, garantindo que ele não fique muito largo em telas maiores.
- margin: 0 auto centraliza o container na tela horizontalmente.
- As seções usam flex: 1 0 auto para ocupar espaço disponível igualmente dentro do container.
- A propriedade padding adiciona espaço interno às seções.
- A borda e a margem inferior estilizam as seções visualmente.

Usando 3 colunas sem grid

Agora vai-se usar um esquema mais simples (sem usar o seletor grid) que é indicado para páginas mais simples. Eis o resultado



Eis o CSS

```
#container {
  width: 100%; /* Define a largura total do container */ }
#coluna1 {
  width: 20%;
  float: left;
  box-sizing: border-box; background-color:#c4d5f6; }
#coluna2 {
  width: 50%;
  float: left;
  box-sizing: border-box; background-color:#a4d3b3; }
#coluna3 {
  width: 30%;
  float: left;
  box-sizing: border-box; background-color:#f1e4a3; }
```

Eis o HTML

```
<!DOCTYPE html>
<html lang="pt-BR">
<head>
    <meta charset="UTF-8">
    <title>Sistema de Grade Básico</title>
    <link rel="stylesheet" href="h53b.css">
</head>  <body>
    <header>Cabeçalho</header>
    <main>
   <section id="container">
         <section id="coluna1">
         <h2>Seção 1</h2>
         <p>Conteúdo da Seção 1</p>
         </section>
         <section id="coluna2">
         <h2>Seção 2</h2>
         <p>Conteúdo da Seção 2</p>
         </section>
         <section id="coluna3">
         <h2>Seção 3</h2>
         <p>Conteúdo da Seção 3</p>
          Vamos ver se vai <p>
          agora sim <p>não pare
         </section>
   </section>
      </main>
      <footer>Rodapé</footer>
</body>  </html>
```

Usando 3 colunas com grid

Com o resultado

Seja o CSS

```
#container {
  display: grid;
  /* Cria 3 colunas com largura igual -- 1fr=1 unid. frac.*/
  grid-template-columns: 1fr 2fr 1.5fr;
  gap: 20px; /* Define o espaçamento entre as colunas */  }
#coluna1{grid-column:1; background-color:#bbaacc;}
#coluna2{grid-column:2; background-color:#ccb4a2;}
#coluna3{grid-column:3; background-color:#d4d3a8;}
```

E o HTML

```
<!DOCTYPE html>
<html lang="pt-BR">
<head>
    <meta charset="UTF-8">
    <title>Sistema de Grade Básico</title>
    <link rel="stylesheet" href="h53a.css">
</head>
<body>
    <section id="container">
        <header>Cabeçalho</header>
        <main>
            <section id="coluna1">
                <h2>Seção 1</h2>
                <p>Conteúdo da Seção 1</p>
            </section>
            <section id="coluna2">
                <h2>Seção 2</h2>
                <p>Conteúdo da Seção 2</p>
            </section>
            <section id="coluna3">
                <h2>Seção 3</h2>
                <p>Conteúdo da Seção 3</p>
           Vamos ver se vai <p>
           agora sim <p>não pare
            </section>
        </main>
        <footer>Rodapé</footer>
    </section>  </body>  </html>
```

☞ Para você fazer

Vamos ao exercício da aula.
Construa um HTML único para esta aula. Deve ser um site com o tema

`Um carro esportivo da marca FERRARI`

e contendo as seguintes partes:

- um cabeçalho contendo 3 parágrafos explicando o que este exercício vai tratar (sites com 3 blocos flexíveis)
- um bloco `main` no qual estarão os conteúdos.
- o primeiro bloco deve conter a história do tema
- o segundo bloco deve conter o estado atual do tema
- o terceiro bloco é para você especular sobre o futuro do tema
- o rodapé deve conter a sua identificação (nome, turma, curso e colégio).

Depois deste HTML feito e funcionando, é hora de construir 3 arquivos CSS, a saber:

1. Mostre os 3 blocos na vertical.
2. Os blocos devem aparecer lado a lado, com os 3 blocos de mesmo tamanho
3. O primeiro e terceiro bloco devem ocupar 25% do espaço cada um, enquanto o bloco central deve ocupar os 50% restantes.

Use ou não o selector grid. Fica a seu critério. Note que o bloco do meio deve conter pelo menos 3 fotos. Se quiser colocar fotos nos outros blocos fique à vontade.

O ponto mais importante deste exercício é que o HTML deve ser o **mesmo** para os 3 exercícios.

Mostre estas atividades e os arquivos que você criou para o professor ANTES de devolver esta folha.

Avaliação

(preenchido pelo professor)

107-76144 - ga/ a

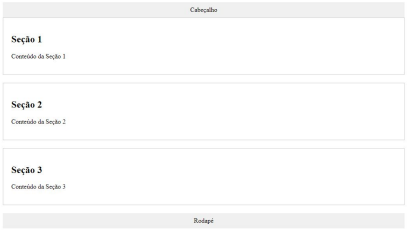
CEP - UP - UTFPR - PUCPr - UFPR – 13/05/2024 - 12:08:57.0
Prof Dr P Kantek (pkantek@gmail.com)
Organizando o laytou com CSS VIVXh53a, V: 1.06 76168 VICTOR BRONDANI DOS SANTOS
24FRO107 - 15 entregar ate 16/maio _____ / / _____

Organizando o layout com CSS

Como se sabe o CSS permite que um layout seja construído para apresentar os conteúdos na hora de mostrar uma página.

1 coluna

Vamos começar com um layout simples, contendo header, footer e um bloco principal entre eles. Este bloco terá 3 sessões, devidamente empilhadas na vertical, como se pode ver em



Neste caso, o CSS será

```
/* Estilo geral do container */
.container {
    display: flex;
    flex-direction: column; /* Organiza os elementos na vertical */
    width: 100%; /* Ocupa a largura total da tela */
    max-width: 1000px; /* Largura máxima de 1000px */
    margin: 0 auto; /* Centraliza o container na tela */
    padding: 20px;            }
/* Estilos para header e footer */
header, footer {
    background-color: #f0f0f0;
    padding: 10px;
    text-align: center;       }
/* Estilos para as seções */
.secao {
    flex: 1 0 auto; /* Ocupa espaço disponível igualmente */
    padding: 20px;
    border: 1px solid #ccc;
    margin-bottom: 20px;      }
```

O HTML que o acessa é

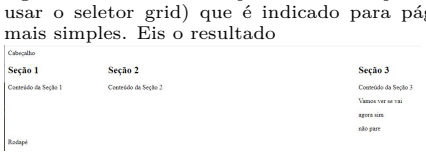
```
<!DOCTYPE html>
<html lang="pt-BR">
<head>
    <meta charset="UTF-8">
    <title>Sistema de Grade Básico</title>
    <link rel="stylesheet" href="stycols.css">
</head>
<body>
    <section class="container">
        <header>Cabeçalho</header>
        <main>
            <section class="secao">
                <h2>Seção 1</h2>
                <p>Conteúdo da Seção 1</p>
            </section>
            <section class="secao">
                <h2>Seção 2</h2>
                <p>Conteúdo da Seção 2</p>
            </section>
            <section class="secao">
                <h2>Seção 3</h2>
                <p>Conteúdo da Seção 3</p>
            </section>
        </main>
        <footer>Rodapé</footer>
    </section>
</body>
</html>
```

Valem as seguintes observações

- O container é definido com display: flex e flex-direction: column para organizar os elementos na vertical.
- A propriedade width: 100% faz com que o container ocupe a largura total da tela.
- max-width: 1000px define uma largura máxima de 1000px para o container, garantindo que ele não fique muito largo em telas maiores.
- margin: 0 auto centraliza o container na tela horizontalmente.
- As seções usam flex: 1 0 auto para ocupar espaço disponível igualmente dentro do container.
- A propriedade padding adiciona espaço interno às seções.
- A borda e a margem inferior estilizam as seções visualmente.

Usando 3 colunas sem grid

Agora vai-se usar um esquema mais simples (sem usar o seletor grid) que é indicado para páginas mais simples. Eis o resultado



Eis o CSS

```
#container {
  width: 100%; /* Define a largura total do container */ }
#coluna1 {
  width: 20%;
  float: left;
  box-sizing: border-box; background-color:#c4d5f6; }
#coluna2 {
  width: 50%;
  float: left;
  box-sizing: border-box; background-color:#a4d3b3; }
#coluna3 {
  width: 30%;
  float: left;
  box-sizing: border-box; background-color:#f1e4a3; }
```

Eis o HTML

```
<!DOCTYPE html>
<html lang="pt-BR">
<head>
    <meta charset="UTF-8">
    <title>Sistema de Grade Básico</title>
    <link rel="stylesheet" href="h53b.css">
</head>  <body>
    <header>Cabeçalho</header>
    <main>
   <section id="container">
         <section id="coluna1">
         <h2>Seção 1</h2>
         <p>Conteúdo da Seção 1</p>
         </section>
         <section id="coluna2">
         <h2>Seção 2</h2>
         <p>Conteúdo da Seção 2</p>
         </section>
         <section id="coluna3">
         <h2>Seção 3</h2>
         <p>Conteúdo da Seção 3</p>
          Vamos ver se vai <p>
          agora sim <p>não pare
         </section>
   </section>
      </main>
      <footer>Rodapé</footer>
</body>  </html>
```

Usando 3 colunas com grid

Com o resultado

Seja o CSS

```
#container {
  display: grid;
  /* Cria 3 colunas com largura igual -- 1fr=1 unid. frac.*/
  grid-template-columns: 1fr 2fr 1.5fr;
  gap: 20px; /* Define o espaçamento entre as colunas */  }
#coluna1{grid-column:1; background-color:#bbaacc;}
#coluna2{grid-column:2; background-color:#ccb4a2;}
#coluna3{grid-column:3; background-color:#d4d3a8;}
```

E o HTML

```
<!DOCTYPE html>
<html lang="pt-BR">
<head>
    <meta charset="UTF-8">
    <title>Sistema de Grade Básico</title>
    <link rel="stylesheet" href="h53a.css">
</head>
<body>
    <section id="container">
        <header>Cabeçalho</header>
        <main>
            <section id="coluna1">
                <h2>Seção 1</h2>
                <p>Conteúdo da Seção 1</p>
            </section>
            <section id="coluna2">
                <h2>Seção 2</h2>
                <p>Conteúdo da Seção 2</p>
            </section>
            <section id="coluna3">
                <h2>Seção 3</h2>
                <p>Conteúdo da Seção 3</p>
           Vamos ver se vai <p>
           agora sim <p>não pare
            </section>
        </main>
        <footer>Rodapé</footer>
    </section>  </body>  </html>
```

☞ Para você fazer

Vamos ao exercício da aula.
Construa um HTML único para esta aula. Deve ser um site com o tema

`A estrada da Graciosa (entre Curitiba e Morretes)`

e contendo as seguintes partes:

- um cabeçalho contendo 3 parágrafos explicando o que este exercício vai tratar (sites com 3 blocos flexíveis)
- um bloco `main` no qual estarão os conteúdos.
- o primeiro bloco deve conter a história do tema
- o segundo bloco deve conter o estado atual do tema
- o terceiro bloco é para você especular sobre o futuro do tema
- o rodapé deve conter a sua identificação (nome, turma, curso e colégio).

Depois deste HTML feito e funcionando, é hora de construir 3 arquivos CSS, a saber:

1. Mostre os 3 blocos na vertical.
2. Os blocos devem aparecer lado a lado, com os 3 blocos de mesmo tamanho
3. O primeiro e terceiro bloco devem ocupar 25% do espaço cada um, enquanto o bloco central deve ocupar os 50% restantes.

Use ou não o selector grid. Fica a seu critério. Note que o bloco do meio deve conter pelo menos 3 fotos. Se quiser colocar fotos nos outros blocos fique à vontade.

O ponto mais importante deste exercício é que o HTML deve ser o **mesmo** para os 3 exercícios.

Mostre estas atividades e os arquivos que você criou para o professor ANTES de devolver esta folha.

Avaliação

(preenchido pelo professor)

107-76168 - ga/ a

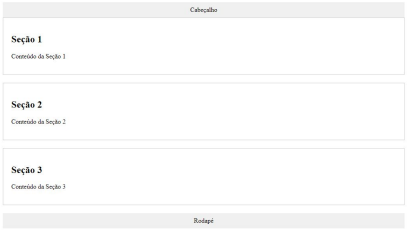
CEP - UP - UTFPR - PUCPr - UFPR – 13/05/2024 - 12:08:57.0
Prof Dr P Kantek (pkantek@gmail.com)
Organizando o laytou com CSS VIVXh53a, V: 1.06 76175 VICTOR HUGO DOS SANTOS DE CA 24FRO107 - 16 entregar ate 16/maio _____ / / _____

Organizando o layout com CSS

Como se sabe o CSS permite que um layout seja construído para apresentar os conteúdos na hora de mostrar uma página.

1 coluna

Vamos começar com um layout simples, contendo header, footer e um bloco principal entre eles. Este bloco terá 3 sessões, devidamente empilhadas na vertical, como se pode ver em



Neste caso, o CSS será

```
/* Estilo geral do container */
.container {
    display: flex;
    flex-direction: column; /* Organiza os elementos na vertical */
    width: 100%; /* Ocupa a largura total da tela */
    max-width: 1000px; /* Largura máxima de 1000px */
    margin: 0 auto; /* Centraliza o container na tela */
    padding: 20px;            }
/* Estilos para header e footer */
header, footer {
    background-color: #f0f0f0;
    padding: 10px;
    text-align: center;       }
/* Estilos para as seções */
.secao {
    flex: 1 0 auto; /* Ocupa espaço disponível igualmente */
    padding: 20px;
    border: 1px solid #ccc;
    margin-bottom: 20px;      }
```

O HTML que o acessa é

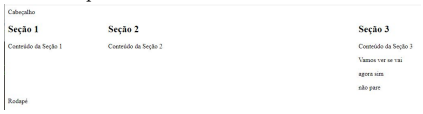
```
<!DOCTYPE html>
<html lang="pt-BR">
<head>
    <meta charset="UTF-8">
    <title>Sistema de Grade Básico</title>
    <link rel="stylesheet" href="stycols.css">
</head>
<body>
    <section class="container">
        <header>Cabeçalho</header>
        <main>
            <section class="secao">
                <h2>Seção 1</h2>
                <p>Conteúdo da Seção 1</p>
            </section>
            <section class="secao">
                <h2>Seção 2</h2>
                <p>Conteúdo da Seção 2</p>
            </section>
            <section class="secao">
                <h2>Seção 3</h2>
                <p>Conteúdo da Seção 3</p>
            </section>
        </main>
        <footer>Rodapé</footer>
    </section>
</body>
</html>
```

Valem as seguintes observações

- O container é definido com display: flex e flex-direction: column para organizar os elementos na vertical.
- A propriedade width: 100% faz com que o container ocupe a largura total da tela.
- max-width: 1000px define uma largura máxima de 1000px para o container, garantindo que ele não fique muito largo em telas maiores.
- margin: 0 auto centraliza o container na tela horizontalmente.
- As seções usam flex: 1 0 auto para ocupar espaço disponível igualmente dentro do container.
- A propriedade padding adiciona espaço interno às seções.
- A borda e a margem inferior estilizam as seções visualmente.

Usando 3 colunas sem grid

Agora vai-se usar um esquema mais simples (sem usar o seletor grid) que é indicado para páginas mais simples. Eis o resultado



Eis o CSS

```
#container {
  width: 100%; /* Define a largura total do container */ }
#coluna1 {
  width: 20%;
  float: left;
  box-sizing: border-box; background-color:#c4d5f6; }
#coluna2 {
  width: 50%;
  float: left;
  box-sizing: border-box; background-color:#a4d3b3; }
#coluna3 {
  width: 30%;
  float: left;
  box-sizing: border-box; background-color:#f1e4a3; }
```

Eis o HTML

```
<!DOCTYPE html>
<html lang="pt-BR">
<head>
    <meta charset="UTF-8">
    <title>Sistema de Grade Básico</title>
    <link rel="stylesheet" href="h53b.css">
</head>  <body>
    <header>Cabeçalho</header>
    <main>
   <section id="container">
         <section id="coluna1">
         <h2>Seção 1</h2>
         <p>Conteúdo da Seção 1</p>
         </section>
         <section id="coluna2">
         <h2>Seção 2</h2>
         <p>Conteúdo da Seção 2</p>
         </section>
         <section id="coluna3">
         <h2>Seção 3</h2>
         <p>Conteúdo da Seção 3</p>
          Vamos ver se vai <p>
          agora sim <p>não pare
         </section>
   </section>
      </main>
      <footer>Rodapé</footer>
</body>  </html>
```

Usando 3 colunas com grid

Com o resultado

Seja o CSS

```
#container {
  display: grid;
  /* Cria 3 colunas com largura igual -- 1fr=1 unid. frac.*/
  grid-template-columns: 1fr 2fr 1.5fr;
  gap: 20px; /* Define o espaçamento entre as colunas */  }
#coluna1{grid-column:1; background-color:#bbaacc;}
#coluna2{grid-column:2; background-color:#ccb4a2;}
#coluna3{grid-column:3; background-color:#d4d3a8;}
```

E o HTML

```
<!DOCTYPE html>
<html lang="pt-BR">
<head>
    <meta charset="UTF-8">
    <title>Sistema de Grade Básico</title>
    <link rel="stylesheet" href="h53a.css">
</head>
<body>
    <section id="container">
        <header>Cabeçalho</header>
        <main>
            <section id="coluna1">
                <h2>Seção 1</h2>
                <p>Conteúdo da Seção 1</p>
            </section>
            <section id="coluna2">
                <h2>Seção 2</h2>
                <p>Conteúdo da Seção 2</p>
            </section>
            <section id="coluna3">
                <h2>Seção 3</h2>
                <p>Conteúdo da Seção 3</p>
           Vamos ver se vai <p>
           agora sim <p>não pare
            </section>
        </main>
        <footer>Rodapé</footer>
    </section>  </body>  </html>
```

☞ Para você fazer

Vamos ao exercício da aula.
Construa um HTML único para esta aula. Deve ser um site com o tema

`A indústria da música`

e contendo as seguintes partes:

- um cabeçalho contendo 3 parágrafos explicando o que este exercício vai tratar (sites com 3 blocos flexíveis)
- um bloco `main` no qual estarão os conteúdos.
- o primeiro bloco deve conter a história do tema
- o segundo bloco deve conter o estado atual do tema
- o terceiro bloco é para você especular sobre o futuro do tema
- o rodapé deve conter a sua identificação (nome, turma, curso e colégio).

Depois deste HTML feito e funcionando, é hora de construir 3 arquivos CSS, a saber:

1. Mostre os 3 blocos na vertical.
2. Os blocos devem aparecer lado a lado, com os 3 blocos de mesmo tamanho
3. O primeiro e terceiro bloco devem ocupar 25% do espaço cada um, enquanto o bloco central deve ocupar os 50% restantes.

Use ou não o selector grid. Fica a seu critério. Note que o bloco do meio deve conter pelo menos 3 fotos. Se quiser colocar fotos nos outros blocos fique à vontade.

O ponto mais importante deste exercício é que o HTML deve ser o **mesmo** para os 3 exercícios.

Mostre estas atividades e os arquivos que você criou para o professor ANTES de devolver esta folha.

Avaliação

(preenchido pelo professor)

107-76175 - ga/ a

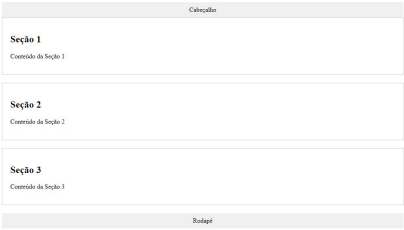
CEP - UP - UTFPR - PUCPr - UFPR – 13/05/2024 - 12:08:57.0
Prof Dr P Kantek (pkantek@gmail.com)
Organizando o laytou com CSS VIVXh53a, V: 1.06 76182 VITOR EDUARDO DE GOES FONTES 24FRO107 - 17 entregar ate 16/maio / _____ / _____

Organizando o layout com CSS

Como se sabe o CSS permite que um layout seja construído para apresentar os conteúdos na hora de mostrar uma página.

1 coluna

Vamos começar com um layout simples, contendo header, footer e um bloco principal entre eles. Este bloco terá 3 sessões, devidamente empilhadas na vertical, como se pode ver em



Neste caso, o CSS será

```
/* Estilo geral do container */
.container {
    display: flex;
    flex-direction: column; /* Organiza os elementos na vertical */
    width: 100%; /* Ocupa a largura total da tela */
    max-width: 1000px; /* Largura máxima de 1000px */
    margin: 0 auto; /* Centraliza o container na tela */
    padding: 20px;            }
/* Estilos para header e footer */
header, footer {
    background-color: #f0f0f0;
    padding: 10px;
    text-align: center;       }
/* Estilos para as seções */
.secao {
    flex: 1 0 auto; /* Ocupa espaço disponível igualmente */
    padding: 20px;
    border: 1px solid #ccc;
    margin-bottom: 20px;      }
```

O HTML que o acessa é

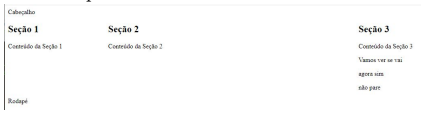
```
<!DOCTYPE html>
<html lang="pt-BR">
<head>
    <meta charset="UTF-8">
    <title>Sistema de Grade Básico</title>
    <link rel="stylesheet" href="stycols.css">
</head>
<body>
    <section class="container">
        <header>Cabeçalho</header>
        <main>
            <section class="secao">
                <h2>Seção 1</h2>
                <p>Conteúdo da Seção 1</p>
            </section>
            <section class="secao">
                <h2>Seção 2</h2>
                <p>Conteúdo da Seção 2</p>
            </section>
            <section class="secao">
                <h2>Seção 3</h2>
                <p>Conteúdo da Seção 3</p>
            </section>
        </main>
        <footer>Rodapé</footer>
    </section>
</body>
</html>
```

Valem as seguintes observações

- O container é definido com display: flex e flex-direction: column para organizar os elementos na vertical.
- A propriedade width: 100% faz com que o container ocupe a largura total da tela.
- max-width: 1000px define uma largura máxima de 1000px para o container, garantindo que ele não fique muito largo em telas maiores.
- margin: 0 auto centraliza o container na tela horizontalmente.
- As seções usam flex: 1 0 auto para ocupar espaço disponível igualmente dentro do container.
- A propriedade padding adiciona espaço interno às seções.
- A borda e a margem inferior estilizam as seções visualmente.

Usando 3 colunas sem grid

Agora vai-se usar um esquema mais simples (sem usar o seletor grid) que é indicado para páginas mais simples. Eis o resultado



Eis o CSS

```
#container {
  width: 100%; /* Define a largura total do container */ }
#coluna1 {
  width: 20%;
  float: left;
  box-sizing: border-box; background-color:#c4d5f6; }
#coluna2 {
  width: 50%;
  float: left;
  box-sizing: border-box; background-color:#a4d3b3; }
#coluna3 {
  width: 30%;
  float: left;
  box-sizing: border-box; background-color:#f1e4a3; }
```

Eis o HTML

```
<!DOCTYPE html>
<html lang="pt-BR">
<head>
    <meta charset="UTF-8">
    <title>Sistema de Grade Básico</title>
    <link rel="stylesheet" href="h53b.css">
</head>  <body>
    <header>Cabeçalho</header>
    <main>
   <section id="container">
         <section id="coluna1">
         <h2>Seção 1</h2>
         <p>Conteúdo da Seção 1</p>
         </section>
         <section id="coluna2">
         <h2>Seção 2</h2>
         <p>Conteúdo da Seção 2</p>
         </section>
         <section id="coluna3">
         <h2>Seção 3</h2>
         <p>Conteúdo da Seção 3</p>
          Vamos ver se vai <p>
          agora sim <p>não pare
         </section>
   </section>
      </main>
      <footer>Rodapé</footer>
</body>  </html>
```

Usando 3 colunas com grid

Com o resultado

Seja o CSS

```
#container {
  display: grid;
  /* Cria 3 colunas com largura igual -- 1fr=1 unid. frac.*/
  grid-template-columns: 1fr 2fr 1.5fr;
  gap: 20px; /* Define o espaçamento entre as colunas */  }
#coluna1{grid-column:1; background-color:#bbaacc;}
#coluna2{grid-column:2; background-color:#ccb4a2;}
#coluna3{grid-column:3; background-color:#d4d3a8;}
```

E o HTML

```
<!DOCTYPE html>
<html lang="pt-BR">
<head>
    <meta charset="UTF-8">
    <title>Sistema de Grade Básico</title>
    <link rel="stylesheet" href="h53a.css">
</head>
<body>
    <section id="container">
        <header>Cabeçalho</header>
        <main>
            <section id="coluna1">
                <h2>Seção 1</h2>
                <p>Conteúdo da Seção 1</p>
            </section>
            <section id="coluna2">
                <h2>Seção 2</h2>
                <p>Conteúdo da Seção 2</p>
            </section>
            <section id="coluna3">
                <h2>Seção 3</h2>
                <p>Conteúdo da Seção 3</p>
           Vamos ver se vai <p>
           agora sim <p>não pare
            </section>
        </main>
        <footer>Rodapé</footer>
    </section>  </body>  </html>
```

☞ Para você fazer

Vamos ao exercício da aula.
Construa um HTML único para esta aula. Deve ser um site com o tema

`O sistema de transporte coletivo de Curitiba`

e contendo as seguintes partes:

- um cabeçalho contendo 3 parágrafos explicando o que este exercício vai tratar (sites com 3 blocos flexíveis)
- um bloco `main` no qual estarão os conteúdos.
- o primeiro bloco deve conter a história do tema
- o segundo bloco deve conter o estado atual do tema
- o terceiro bloco é para você especular sobre o futuro do tema
- o rodapé deve conter a sua identificação (nome, turma, curso e colégio).

Depois deste HTML feito e funcionando, é hora de construir 3 arquivos CSS, a saber:

1. Mostre os 3 blocos na vertical.
2. Os blocos devem aparecer lado a lado, com os 3 blocos de mesmo tamanho
3. O primeiro e terceiro bloco devem ocupar 25% do espaço cada um, enquanto o bloco central deve ocupar os 50% restantes.

Use ou não o selector grid. Fica a seu critério. Note que o bloco do meio deve conter pelo menos 3 fotos. Se quiser colocar fotos nos outros blocos fique à vontade.

O ponto mais importante deste exercício é que o HTML deve ser o **mesmo** para os 3 exercícios.

Mostre estas atividades e os arquivos que você criou para o professor ANTES de devolver esta folha.

Avaliação

(preenchido pelo professor)

107-76182 - ga/ a