

**Curso Técnico Superior Profissional em Tecnologias e Programação de Sistemas de Informação**

**Unidade Curricular:** Design Hipermédia

**1.º Ano/1.º Semestre**

**Docente:** Marco Miguel Olival Olim

**Data** 23/10/2017

---

**ESTE EXERCÍCIO PRETENDE ILUSTRAR AS DIFERENÇAS ENTRE AS DIFERENTES TÉCNICAS PARA CONCEBER LAYOUTS EM CSS**

---

- 1- Iniciar um ficheiro de html com a seguinte estrutura:

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <link rel="stylesheet" href="style-inlineBlock.css">
  <title>Exercicio 1: Layouts</title>
</head>

<body>
  <h1>Inline-Block</h1>
  <section id="inline-block">
    <div>Lorem ipsum dolor sit amet consectetur adipisicing elit. Aliquid non,
consequatur debitis eius doloremque dolor in voluptates labore. Vel provident laboriosam
ipsam eveniet deserunt repellat odio libero vero animi aut obcaecati quam, aspernatur
placeat excepturi similique pariatur odit. Nesciunt delectus sed dolorum quia officiis
nihil dicta deleniti eius magni ea.</div>
    <div>Necessitatibus quisquam repellat ratione doloremque beatae voluptas </div>
    <div>dignissimos velit nam aliquid, cumque tenetur vitae id at quaerat aut dolor
culpa nulla. A quia inventore dolorum id fugiat, iusto repellendus! Similique, obcaecati
eveniet excepturi quod, tempore ullam quos vel itaque incidunt
    deleniti molestiae ipsam a? Non?</div>
  </section>
```

- 2- Pode agora optar por criar diferentes secções ou então novos ficheiros html correspondentes às outras técnicas de Layout, nomeadamente “Float”, “FlexBox” e “Grid”.

- 3- Pretende-se criar três painéis de informação usando as diferentes técnicas:

Deleniti, molestiae nam. Neque labore numquam optio quod, atque eum, quae dolorum temporibus illum corporis laboriosam ea, fugiat maxime quos cumque impedit dolorem facilis fugit earum. Temporibus cumque numquam animi magnam nihil culpa illum illo porro dignissimos quos delectus natus reiciendis fuga voluptatibus iste dolorem, deserunt ab distinctio ullam obcaecati?

Enim unde est rem, soluta porro dicta, ea voluptate iusto quam possimus vero

quibusdam animi temporibus accusamus iste corporis a, officiis cupiditate error est quae molestias eligendi sit iusto impedit vitae. Sint similique porro minima provident incidunt ea doloribus odit. Adipisci sapiente dolorum neque maiores molestiae hic, laboriosam vero iste sint officia voluptate.

- 4- Crie o ficheiro CSS com os seguintes estilos:

```
body{
  background-color: #eee;
}
div{
  padding: 10px;
  background-color: white;
  border: solid 1px #ccc;
}
```

- 5- Inicialmente estes painéis de informação são apresentados na vertical com “display: block” aplicado por default, ocupando toda a largura do browser.
- 6- Para modificar a orientação para horizontal alteramos o display para inline-block, além de reduzir a largura dos painéis para 30%.

```
#inline-block div{

  display: inline-block;
  width: 30%;
}
```

- 7- O Inline-block é caracterizado por deixar um pequeno espaço entre os blocos, sem que exista opção para o eliminar de forma direta. Note também que se a informação for de tamanho diferente, os painéis deixam de ocupar o espaço superior.

## Inline-Block

Lorem ipsum dolor sit amet consectetur adipisicing elit. Aliquid non, consequatur debitis eius doloremque dolor in voluptates labore. Vel provident laboriosam ipsam eveniet deserunt repellat odio libero vero animi aut obcaecati quam, aspernatur placeat excepturi similique pariatur odit. Nesciunt delectus sed dolorum quia officiis nihil dicta deleniti eius magni ea.

Necessitatibus quisquam repellat ratione doloremque beatae voluptas sunt

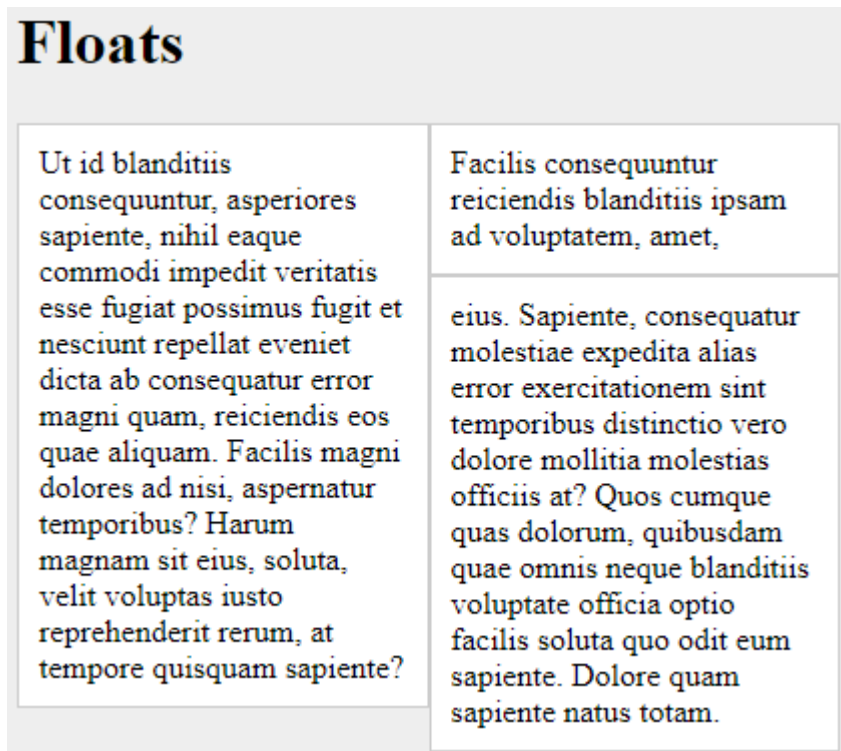
dignissimos velit nam aliquid, cumque tenetur vitae id at quaerat aut dolor culpa nulla. A quia inventore dolorum id fugiat, iusto repellendus! Similique, obcaecati eveniet excepturi quod, tempore ullam quos vel itaque incidunt deleniti molestiae ipsam a? Non?

Cofinanciado por:

- 8- Embora os floats não sejam nativamente uma solução para criação de Layouts, apresentam contudo resultados consistentes, como é possível constatar pelo bootstrap 3. Aplique este estilo no ficheiro ou secção correspondente:

```
#float div {  
  float:left;  
  width: 30%;  
}
```

- 9- Os “floats” resolvem o problema de alinhamento dos espaços de forma mais elegante, ocupando dinamicamente os espaços deixados vazios.



- 10- Os “floats” resolvem o problema de alinhamento dos espaços de forma mais elegante, ocupando dinamicamente os espaços deixados vazios. No entanto, se pretender evitar este colapsar de espaços nas secções seguintes tem de aplicar um “clear” entre elementos.

```
.limpar-floats{  
  clear: both;  
}
```

- 11- Para aplicar o flexbox, basta invocá-lo que este sobrepõe-se à utilização simultânea de floats, resultando numa solução alternativa para browsers antigos que não suportem flexbox:

```
#flexbox{  
  display:flex;  
}  
#flexbox div{  
  flex: 1;  
}
```

Cofinanciado por:



- 12- O FlexBox foi criado para ser uma solução robusta para Layouts. Calcula automaticamente o espaço disponível entre elementos, permite centrá-los horizontal/verticalmente e até permite alterar-lhes a ordem.

## FlexBox

Deleniti, molestiae nam. Neque labore numquam optio quod, atque eum, quae dolorum temporibus illum corporis laboriosam ea, fugiat maxime quos cumque impedit dolorem facilis fugit earum. Temporibus cumque numquam animi magnam nihil culpa illum illo porro dignissimos quos delectus natus reiciendis fuga voluptatibus iste dolorem, deserunt ab distinctio ullam obcaecati?

Enim unde est rem, soluta porro dicta, ea voluptate iusto quam possimus vero

Ducimus animi temporibus accusamus iste corporis a, officiis cupiditate error est quae molestias eligendi sit iusto impedit vitae. Sint similique porro minima provident incidunt ea doloribus odit. Adipisci sapiente dolorum neque maiores molestiae hic, laboriosam vero iste sint officia voluptate.

- 13- O “flex:1” estabelece uma relação de igual proporcionalidade entre elementos mas, tal como em exemplos anteriores, é possível definir uma proporção e solicitar ao flexbox que preencha o espaço remanescente

```
#flexbox{
  display:flex;
  justify-content: space-between;
}
#flexbox div{
  width: 30%;
}
```

- 14- Usando apenas “flex-direction: column-reverse” e “align-items: center” podemos alterar completamente a disposição do Layout

## FlexBox

Ducimus animi temporibus accusamus iste corporis a, officiis cupiditate error est quae molestias eligendi sit iusto impedit vitae. Sint similique porro minima provident incidunt ea doloribus odit. Adipisci sapiente dolorum neque maiores molestiae hic, laboriosam vero iste sint officia voluptate.

Enim unde est rem, soluta porro dicta, ea voluptate iusto quam possimus vero

Deleniti, molestiae nam. Neque labore numquam optio quod, atque eum, quae dolorum temporibus illum corporis

- 15- A maior limitação que se pode apontar ao Flexbox é este ser apenas um sistema uni-dimensional. No entanto ficou disponível recentemente um sistema bi-dimensional (que usa linhas e colunas) para criação de layouts – o Grid

```
#grid{
  display:grid;
  grid-template-columns: 1fr 1fr 1fr;
}
```

Cofinanciado por: