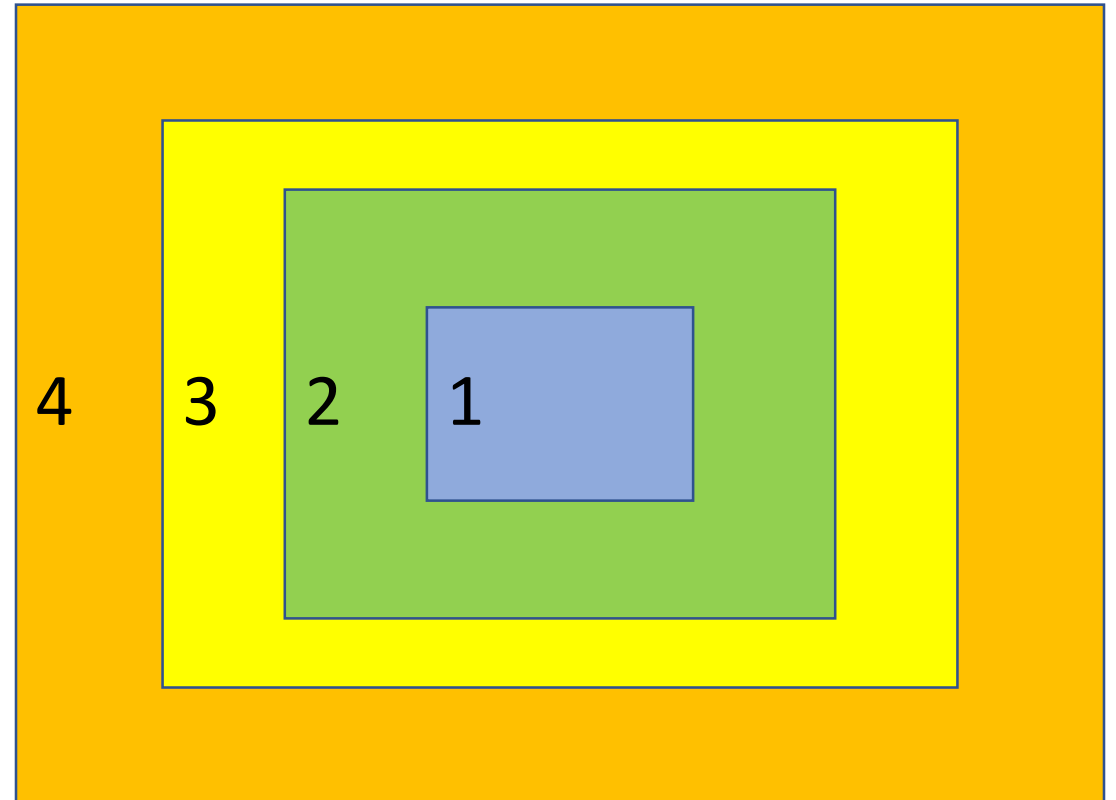


CSS Box Model + Positioning

Posicionamento e disposição de elementos na página

Box model

- Para o CSS, tudo são caixas (“box”).
- Cada caixa é constituída por:
 1. Conteúdo
 2. Padding
 3. Border
 4. Margem
- O tamanho da caixa depende do conteúdo, padding, e border.



Box model

Pergunta:

Dado o código à direita, quais são as dimensões do div com id “teste”?

1. Largura: 100px, altura: 50px;
2. Largura: 142px, altura: 92px;
3. Largura: 162px, altura: 112px;

```
<div id="teste">Olá mundo</div>
```

```
<style>
```

```
  #teste {
```

```
    width: 100px;
```

```
    height: 50px;
```

```
    padding: 20px;
```

```
    border: 1px solid black;
```

```
    margin: 10px;
```

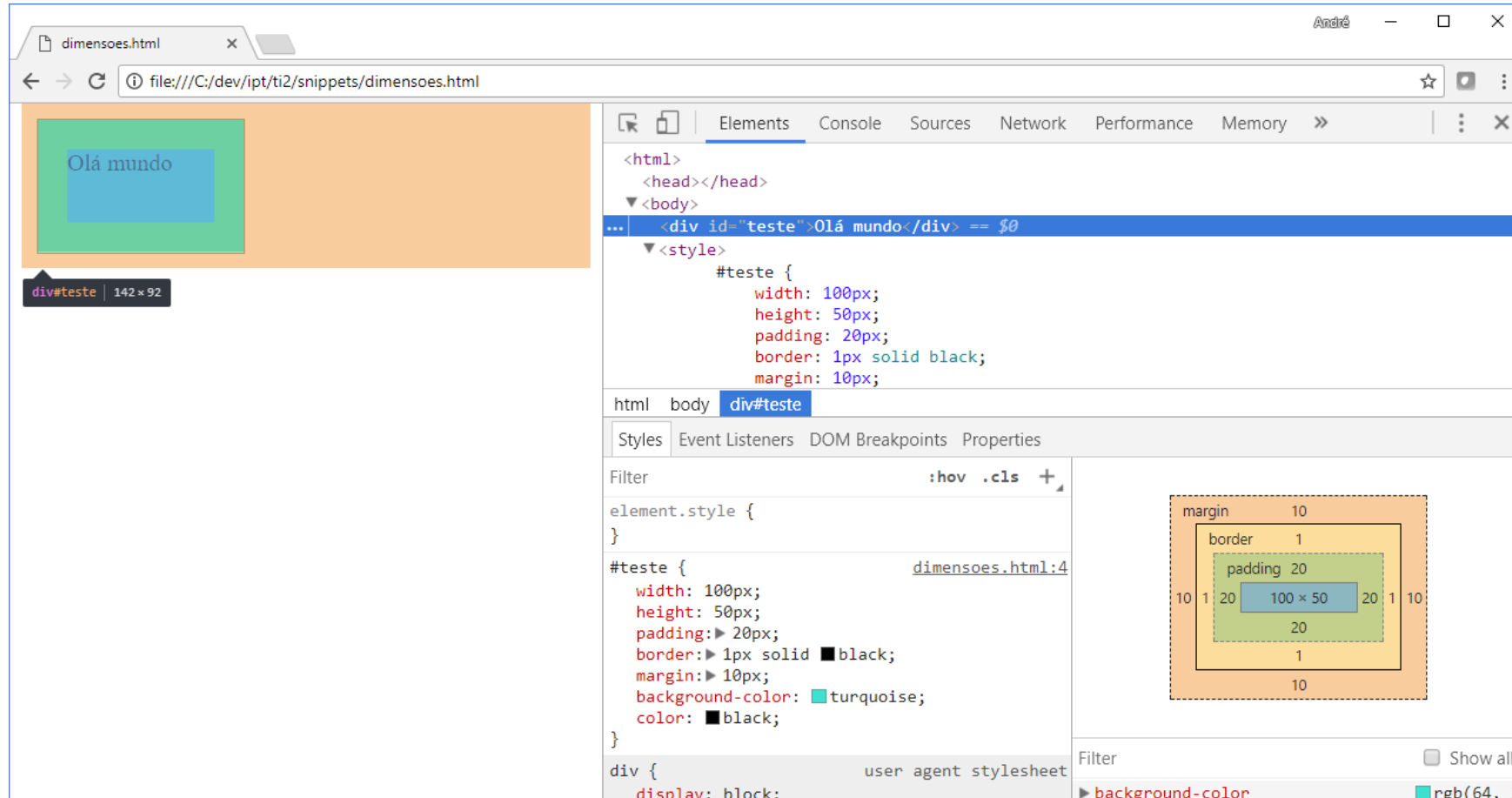
```
    background-color: turquoise;
```

```
    color: black;
```

```
  }
```

```
</style>
```

Resposta correta: 2



Resposta correta: 2!?

By default, exceto nos Internet Explorer(s) velhos em modo *quirks*, as dimensões de uma caixa são determinadas pelas seguintes fórmulas:

$$LarguraTotal = (Content)Width + PaddingLeft + PaddingRight + BorderLeft + BorderRight$$

$$AlturaTotal = (Content)Height + PaddingTop + PaddingBottom + BorderTop + BorderBottom$$

Box model

- Este comportamento, como é por vezes confuso e inesperado, faz com que as pessoas mudem este comportamento.
- A propriedade CSS “box-sizing” é usada para mudar este comportamento:
 - **content-box**: *Default*, exceto no caso do IE em modo *quirks*.
 - **border-box**: Este valor faz com que o “width/height” seja a largura/altura da caixa, e não do conteúdo.
- **Nota: Frameworks como o Bootstrap usam “box-sizing: border-box” by default!**

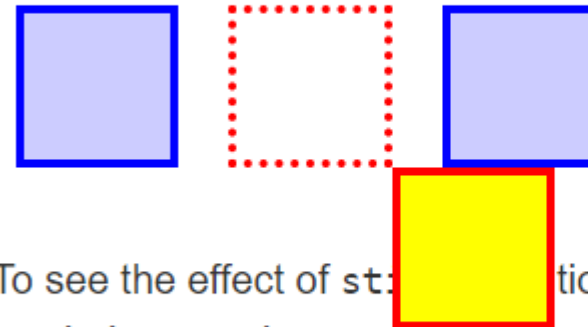
CSS: position

- O “position” permite definir a posição de um elemento.
- Existem 5 tipos de posicionamento:
 - **static** (Default, não faz nada)
 - **relative** (Relativo à posição original do elemento)
 - **absolute** (Relativo ao elemento “contentor”)
 - **fixed** (Relativo ao viewport / janela)
 - **sticky** (Misto entre o “fixed” e o “relative”)
- O deslocamento é controlado através das propriedades “**top**”, “**left**”, “**right**”, e “**bottom**”.

Position: relative

- Permite ajustar a posição dos elementos, relativamente à posição onde eles ficariam originalmente (determinado pelo browser).
- Não afeta o posicionamento de outros elementos no mesmo contentor.
- Nota: `bottom == -top`; `right == -left`.

In this demo you can control the `position` property for the yellow box.



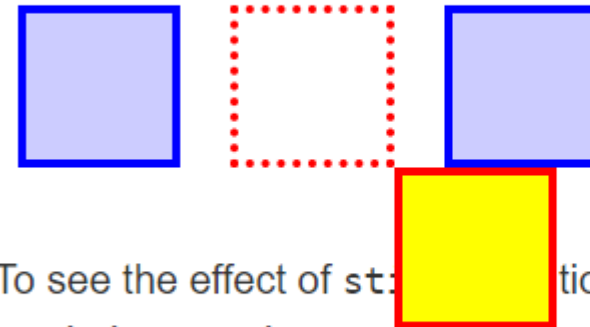
To see the effect of `position: sticky`, select the `position: sticky` option and scroll this container.

The element will scroll along with its container, until it is at the top of the container (or reaches the effect).

Position: relative

- Usos:
 - Quando queremos alterar ligeiramente o posicionamento de um elemento sem afetar os demais.
 - Quando queremos criar um “contentor”.
- Não é o mesmo que usar o “margin”!
 - O uso do “margin” afeta o posicionamento dos outros.

In this demo you can control the `position` property for the yellow box.



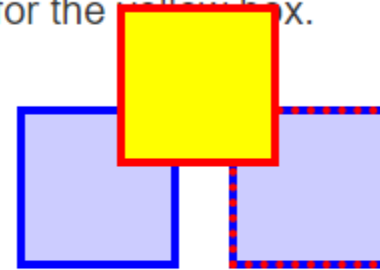
To see the effect of `position: sticky`, select the `position: sticky` option and scroll this container.

The element will scroll along with its container, until it is at the top of the container (or reaches the effect

Position: absolute

- Permite ajustar a posição dos elementos, relativamente ao seu contentor.
- Remove o elemento do fluxo, logo afeta o posicionamento de outros elementos no mesmo contentor.
- Pode ser usado para posicionamentos mais complicados.

In this demo you can control the `position` property for the yellow box.



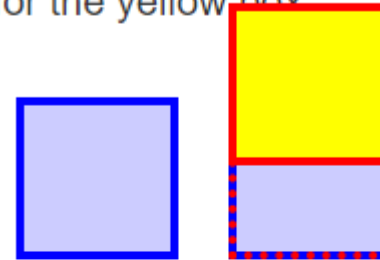
To see the effect of `sticky` positioning, select the `position: sticky` option and scroll this container.

Position: fixed

- Permite colocar um elemento num determinado local relativo ao *viewport* (“janela”).
- O elemento é removido do fluxo, por isso afeta o posicionamento de outros elementos!
- Pode ser usado para elementos que devem estar sempre presentes na janela (ex: barras de navegação, pop-ups, etc.).

Imagem e demo: [Mozilla Developer Network](#)

in the same way you can control the `position` property for the yellow box

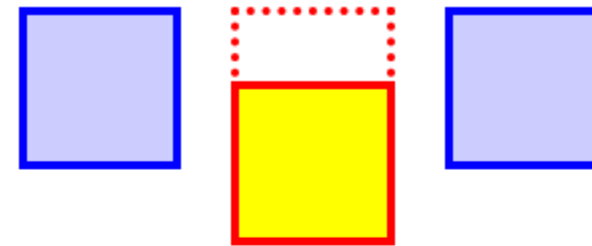


To see the effect of `sticky` positioning, select the `position: sticky` option and scroll this container.

The element will scroll along with its container, until it is at the top of the container (or reaches the offset

Position: sticky

- É um misto entre “relative”, e “fixed”:
 - Comporta-se como “relative” até um determinado limite (definido através do top, left, bottom, right).
 - Após esse limite, comporta-se como “fixed”, ficando “preso” na sua posição, mesmo se se fizer scroll.
- Diferença do “fixed”: os valores são relativos ao contentor.



To see the effect of `sticky` positioning, select the `position: sticky` option and scroll this container.

The element will scroll along with its container, until it is at the top of the container (or reaches the offset specified in `top`), and will then stop scrolling, so it stays visible.

Material de estudo

- CSS positioning: <https://developer.mozilla.org/en-US/docs/Web/CSS/position>
- CSS positioning examples: <https://developer.mozilla.org/samples/cssref/css-positioning.html>
- Contentores: [https://developer.mozilla.org/en-US/docs/Web/CSS/All About The Containing Block](https://developer.mozilla.org/en-US/docs/Web/CSS/All%20About%20The%20Containing%20Block)

Imagem: <https://deskjet.co/products/css-is-awesome-mug>

Obrigado!

André Carvalho

Email: afecarvalho@ipt.pt

