

Posicionamento com CSS

Desenvolvimento Web I

Ricardo de la Rocha Ladeira

{ricardo.ladeira@ifc.edu.br}

6 de dezembro de 2020

Introdução

- ▶ O **posicionamento** do conteúdo em *websites* pode ser determinante para a permanência dos usuários na página.
- ▶ Tamanho da tela, localização dos elementos e adaptação/responsividade são alguns dos fatores que devem ser considerados.

Introdução

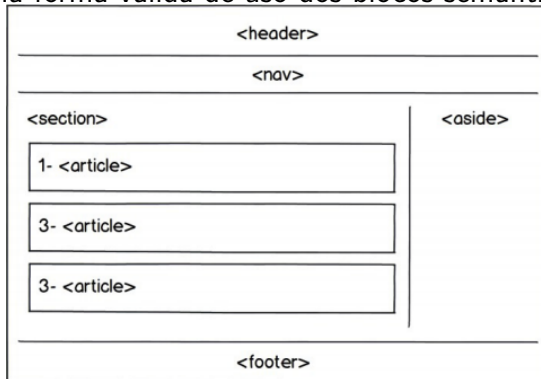
Posicionamento de Blocos Semânticos

- ▶ Os blocos semânticos utilizados para agrupar elementos normalmente são posicionadas da seguinte forma:
 - ▶ **header** na parte superior da página;
 - ▶ Menu de navegação (**nav**) abaixo da apresentação da página (do **header**). Porém, há páginas que apresentam menus de navegação também à esquerda e até mesmo acima do **header**.
 - ▶ O rodapé (**footer**) fica na parte inferior da página.
 - ▶ Conteúdos marcados com a tag **aside** devem aparecer na lateral da página.
 - ▶ O conteúdo principal (**main**) costuma estar entre o centro e a parte esquerda da página.
 - ▶ O conteúdo principal pode ser dividido em **sections** e também pode conter **articles**.

Introdução

Posicionamento de Blocos Semânticos

- Uma forma válida de uso dos blocos semânticos:



Introdução

- ▶ Por mais que possamos alterar a disposição dos elementos, a linguagem HTML define um **posicionamento** padrão para eles, seguindo **de cima para baixo**, e **da esquerda para a direita**.
- ▶ Considerando os eixos x e y , sendo
 - ▶ x_0 a coordenada do eixo horizontal
 - ▶ y_0 a coordenada o eixo vertical
 - ▶ p um ponto representado pelas coordenadas (x_0, y_0)
- ▶ Podemos dizer que o ponto $p = (0, 0)$ é representado pelo ponto mais acima e mais à esquerda do corpo da página.
- ▶ Se o desenvolvedor deseja alterar este posicionamento padrão, é necessário utilizar CSS.

Posicionamento com CSS

- ▶ Algumas propriedades CSS que permitem manipular o posicionamento são:

- ▶ `width`
- ▶ `height`
- ▶ `position`
- ▶ `float`
- ▶ `top`
- ▶ `bottom`
- ▶ `left`
- ▶ `right`

- ▶ `display`
- ▶ `text-align`
- ▶ `padding`
- ▶ `margin`
- ▶ `overflow`
- ▶ `z-index`
- ▶ `clear`
- ▶ ...

Posicionamento

width

- ▶ Define a largura do bloco.

- ▶ Exemplos:

```
#abc { width: 10%; }  
#def { width: 50%; }  
#ghi { width: 44px; }  
div { width: 4em; }
```

- ▶ Porém, só definindo a largura, mas sem conteúdo, nossa `div` não aparece na página. Por quê?
- ▶ Exemplo: `width.html`.

Posicionamento

height

- ▶ A propriedade `height` ajuda a resolver o nosso problema.
- ▶ Define a altura do elemento (em %, cm, em, px...).
- ▶ Exemplo: `height.html`.

Posicionamento

position

- ▶ Propriedade que define o tipo de posicionamento do elemento.
- ▶ Alguns valores possíveis:
 - ▶ **absolute** (**absoluto**)
 - ▶ **relative** (**relativo**)
 - ▶ **fixed** (fixo)
 - ▶ **static** (estático)
 - ▶ **sticky** (“grudento”/“pegajoso”)

Posicionamento

position: absolute

- ▶ **Absoluto**: definido com base na posição do elemento pai.
- ▶ Ver exemplo [absolute.html](#).
- ▶ Como seriam posicionados os elementos no exemplo abaixo?

CSS:

```
img, p {  
  position: absolute;  
  top: 0px;  
  left: 0px;  
}
```

HTML:

```
  
<p>Elementos posicionados no mesmo lugar!</p>
```

Posicionamento

`position: relative`

- ▶ **Relativo:** definido com base na posição que o elemento naturalmente ocuparia.
- ▶ Ver exemplo [relative.html](#).
- ▶ Como seriam posicionados os elementos no exemplo abaixo?

CSS:

```
p {  
  position: relative;  
  top: 100px;  
  left: 100px;  
}
```

HTML:

```
  
<p>Em que local o p sera posicionado?</p>
```

Posicionamento

`position: relative`

- ▶ Agora comente o código CSS e veja o posicionamento natural do parágrafo!

CSS:

```
p {  
  position: relative;  
  top: 100px;  
  left: 100px;  
}
```

HTML:

```
  
<p>Em que local o p sera posicionado?</p>
```

Posicionamento

`position:` `static`, `fixed` e `sticky`

- ▶ `static`: é o valor padrão. Não pode ser reposicionado com as propriedades `top`, `bottom`, `left` e `right`. Posiciona o elemento de acordo com o fluxo natural da página.
- ▶ `fixed`: é posicionado de acordo com a visualização da janela. Permanece no mesmo local mesmo se a janela for redimensionada. Usa-se `top`, `bottom`, `left` e `right` para posicionar o elemento.
- ▶ `sticky`: alterna entre os valores `relative` e `fixed`. A posição do elemento é relativa enquanto ele não precisa da barra de rolagem (*scroll*). A partir do momento que utiliza a barra de rolagem, estabelece uma posição fixa para o elemento. Ver exemplo “**Sociologia**”.

Posicionamento

float

- ▶ A propriedade `float` indica que o elemento poderá flutuar, à direita ou à esquerda.
- ▶ seletor { `float: left;` }
- ▶ É muito usado para posicionamento de blocos.

Posicionamento

float

- ▶ `float` é uma das propriedades **mais importantes** para posicionamento.
- ▶ O elemento que flutua **não altera** o posicionamento dos elementos **anteriores**, mas **altera** o posicionamento dos elementos **posteriores**.
- ▶ Como? Os elementos posteriores passam a flutuar em volta dele.

Posicionamento

float

- Como seriam posicionados os elementos no exemplo abaixo?

```

<p style="text-align:justify">
```

0 paragrafo vai para a esquerda, pois a imagem flutua para a direita, e o paragrafo flutua em volta dela.

```
</p>
```

- Exemplo disponível: [Float.html](#).

Posicionamento

Ferramentas

- ▶ Ferramentas *online* para posicionamento, estilização e criação de layouts:
 - ▶ <https://css-grid-layout-generator.pw/>
 - ▶ <https://griddy.io/>
 - ▶ <https://cssgrid-generator.netlify.app/>
 - ▶ <http://www.blended-html.com/>
 - ▶ <https://csscreator.com/version2/pagelayout.php>
 - ▶ <https://www.cssportal.com/layout-generator/>
 - ▶ <https://www.generateit.net/layout-generator/>
 - ▶ <https://www.htmlcodegenerator-tools.com/2019/11/html-css-responsive-grid-layout-generator-blogger.html>
 - ▶ <http://www.initializr.com/>
 - ▶ <https://www.layoutit.com/pt>
 - ▶ <https://www.pagecolumn.com/>
 - ▶ <https://vue-grid-generator.netlify.app/>
 - ▶ <https://www.webdesignrankings.com/resources/csslayoutgenerator/>

Exercícios

1. Altere o código do arquivo `Posicionamento.html` para que o bloco amarelo fique à esquerda do bloco verde.
2. Crie exemplo(s) utilizando as propriedades citadas na página 6 deste arquivo.

Referências

- ▶ W3C. CSS border property. Disponível em: <https://www.w3schools.com/cssref/pr_border.asp>. Acesso em: 25 jun. 2018.
- ▶ W3C. CSS Padding. Disponível em: <https://www.w3schools.com/css/css_padding.asp>. Acesso em: 25 jun. 2018.
- ▶ W3C. CSS Layout - float and clear. Disponível em: <http://www.w3schools.com/css/css_float.asp>. Acesso em: 26 jan. 2017.
- ▶ W3C. CSS Layout - The position Property. Disponível em: <http://www.w3schools.com/css/css_positioning.asp>. Acesso em: 30 jan. 2017.

Posicionamento com CSS

Desenvolvimento Web I

Ricardo de la Rocha Ladeira

{ricardo.ladeira@ifc.edu.br}

6 de dezembro de 2020