MÓDULO

PROGRAMAÇÃO WEB – HTML/CSS

UNIDADE

ESTILOS DE POSIÇÃO

Instituto Tecnológico

ÍNDICE

OBJETIVOS

Com esta unidade didática, pretende-se que desenvolva os seguintes objetivos de aprendizagem:

- Criação de estruturas lógicas de elementos.
- Posicionamento de elementos em páginas web.

INTRODUÇÃO

O posicionamento de elementos numa página web é uma parte fundamental na criação de layouts. É necessário conhecer, por isso, as várias propriedades de posicionamento que o CSS tem para oferecer. Entender os estilos de posição e as opções a eles associadas torna-se essencial na hora de decidir onde colocar os elementos na página.

Uma estrutura lógica e bem definida permite modificar e expandir a página web de uma forma rápida e fácil, bem como ajuda na hora de realizar o seu posicionamento na página.

Também a forma como os elementos flutuam na página e como os espaços criados à sua volta funcionam é essencial para criar layouts atrativos.

1. O ATRIBUTO POSITION

Para determinar o posicionamento de um elemento recorre-se ao atributo position (posição), que pode assumir cinco valores distintos: static, relative, absolute, fixed e sticky (estático, relativo, absoluto, fixo e "pegajoso", respetivamente). Consoante o valor escolhido, o elemento será posicionado de forma diferente.

Uma vez atribuído o tipo de posicionamento do elemento, deve estabelecer-se a sua posição exata, utilizando as propriedades de posição top, right, bottom e left (superior, direita, inferior e esquerda, respetivamente). Estas propriedades definem a distância à qual o elemento é posicionado, e devem receber valores em qualquer unidade de medida suportada por CSS, embora os mais comuns sejam ponto (pt), pixel (px) e percentagem (%). Também estas propriedades funcionam de forma diferente dependendo do valor escolhido.

Aviso

As propriedades de posição top, right, bottom e left apenas funcionam se o atributo position (static, relative, fixed, absolute ou sticky) já estiver definido.

1.1. STATIC

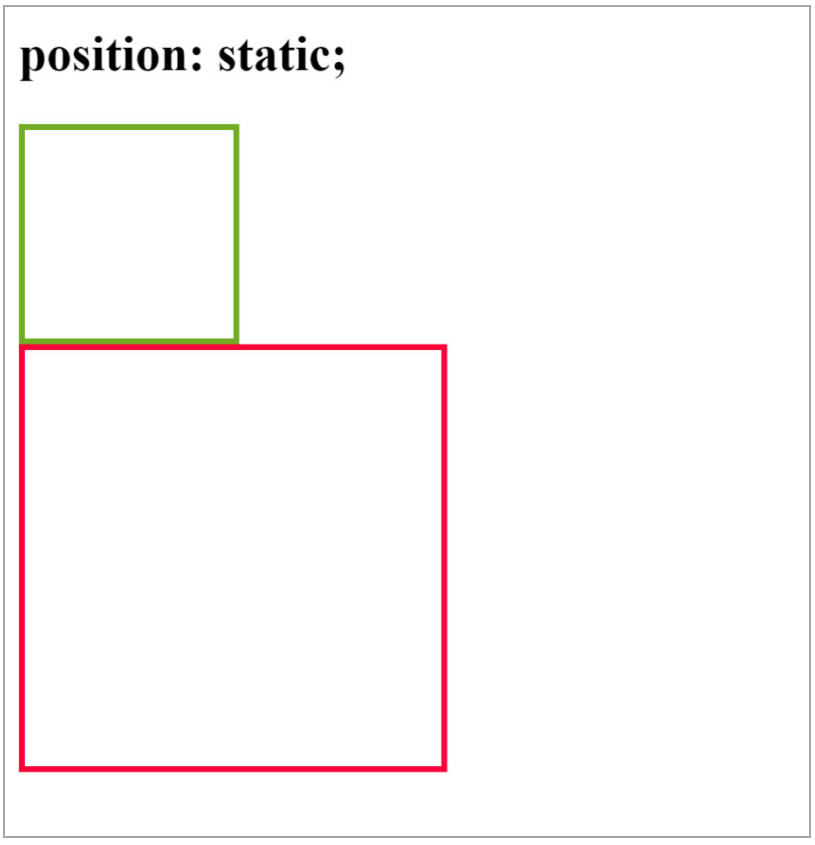
Por defeito, e caso não seja especificado o atributo position, os elementos HTML assumem a posição estática (position: static). Os elementos static não são afetados pelas propriedades top, right, left e bottom, ou seja, aparecem como o definido em HTML e posicionado de acordo com o fluxo normal da página, ou seja, no próximo espaço livre do documento.

Exemplo

```
<!DOCTYPE html>
<html lang="pt">
<head>
<title> Position CSS</title>
    <meta charset="UTF-8">
    <meta name="description" content="position static">
           <meta name="author" content="Sara Granja">
<style>
#square1 {
  position: static;
  width: 100px;
  height: 100px;
  border: 3px solid #73AD21;
}
#square2 {
  position: static;
  width: 200px;
  height: 200px;
  border: 3px solid #fc0339;
}
</style>
</head>
<body>
```

```
<h2>position: static;</h2>
<div id="square1"> </div>
<div id="square2"> </div>
</body>
</html>
```

Resultado:



1.2. RELATIVE

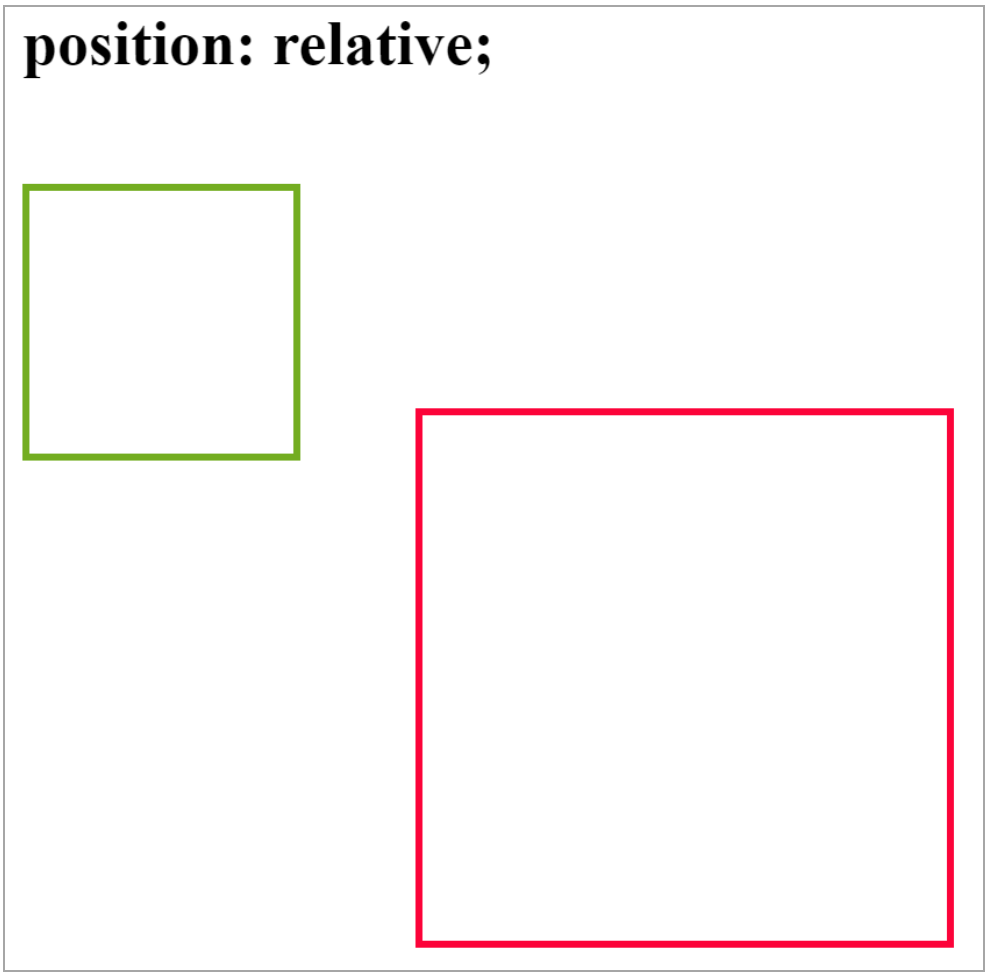
O valor relative (relativo) faz com que os elementos se comportem da mesma forma que quando possuem o valor static, ou seja, seguem o fluxo normal da página, e vão ser colocados no espaço livre seguinte do documento. No entanto, podem receber valores para as propriedades top, right, bottom e left, posicionando-se à distância definida do elemento anterior ou da borda do elemento que o contém.

Exemplo

```
<!DOCTYPE html>
<html lang="pt">
<head>
<title> Position CSS</title>
      <meta charset="UTF-8">
   <meta name="description" content="position relative">
   <meta name="author" content="Sara Granja">
<style>
#square1 {
  position: relative;
  top: 20px;
  width: 100px;
  height: 100px;
  border: 3px solid #73AD21;
}
#square2 {
  position: relative;
  left: 150px;
  width: 200px;
  height: 200px;
  border: 3px solid #fc0339;
```

```
}
</style>
</head>
<body>
<h2>position: static;</h2>
<div id="square1"> </div>
<div id="square2"> </div>
</body>
</html>
```

Resultado:



1.3. FIXED

O valor de posicionamento fixed (fixo) faz com que o elemento seja posicionado em relação à viewport (janela do navegador). Esta propriedade permite que o elemento permaneça sempre no mesmo ponto da viewport, mesmo ao fazer o scroll da página. Os valores das propriedades top, right, bottom e left são os que ditam o posicionamento do elemento.

Aviso

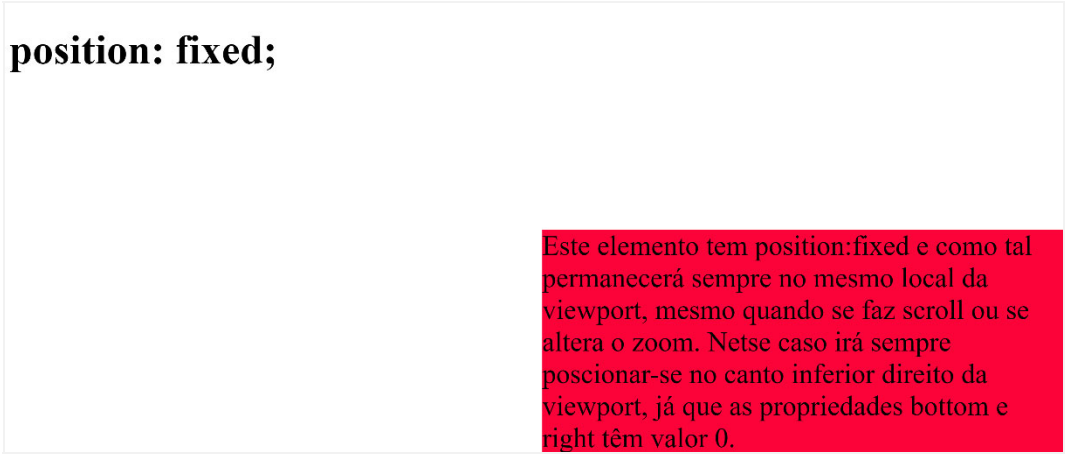
No Internet Explorer, o valor fixed não funciona. Para poder testá-lo terá de se utilizar o Mozilla Firefox ou o Google Chrome.

Exemplo

```
<!DOCTYPE html>
<html lang="pt">
<head>
<title> Position CSS</title>
    <meta charset="UTF-8">
    <meta name="description" content="position fixed">
    <meta name="author" content="Sara Granja">
<style>
div.fixed {
  position: fixed;
  bottom: 0;
  right: 0;
  width: 300px;
  height:auto;
background-color:#fc0339;}
```

```
</style>
</head>
<body>
<h2>position: fixed;</h2>
<div class="fixed">
Este elemento tem position:fixed e como tal permanecerá sempre no mesmo local da viewport, mesmo quando se faz scroll ou se altera o zoom. Neste caso irá sempre posicionar-se no canto inferior direito da viewport, já que as propriedades bottom e right têm valor 0.
</div>
</body>
</html>
```

Resultado:



1.4. ABSOLUTE

Quando se atribui o valor absolute (absoluto) a um elemento, o mesmo vai ser posicionado de acordo com os valores definidos para top, right, left e bottom, em relação à posição do seu elemento ancestral mais próximo, ao contrário do que acontece com o valor fixed, que é relativo à viewport. Assim, consegue-se posicionar um elemento onde se pretender, sem que este mude de posição.

Contudo, se um elemento posicionado com valor absolute não tiver ancestrais posicionados, o mesmo vai ser relativo ao documento (body) e mover-se-á acompanhando o scrolling da página.

Exemplo

```
<!DOCTYPE html>
<html lang="pt">
<head>
<title> Position CSS</title>
    <meta charset="UTF-8">
    <meta name="description" content="position absolute">
    <meta name="author" content="Sara Granja">
<style>
div.relative {
  position: relative;
  width: 600px;
  height: 300px;
  background-color: #73AD21;
}
div.absolute {
  position: absolute;
  top: 100px;
  right: 20px;
```

```
    width: 200px;
    height: 100px;
    background-color: #ffffff;
}
</style>
</head>
<body>
<h2>position: absolute;</h2>
<div class="relative">Este elemento tem atributo <i>relative</i> e
está posicionado em relação à sua posição normal
<div class="absolute">Este elemento tem atributo <i>absolute</i> e
está posicionado em relação ao seu ancestral (a div.relative)</div>
</div>
</body>
</html>
```

Resultado:

1.5. STICKY

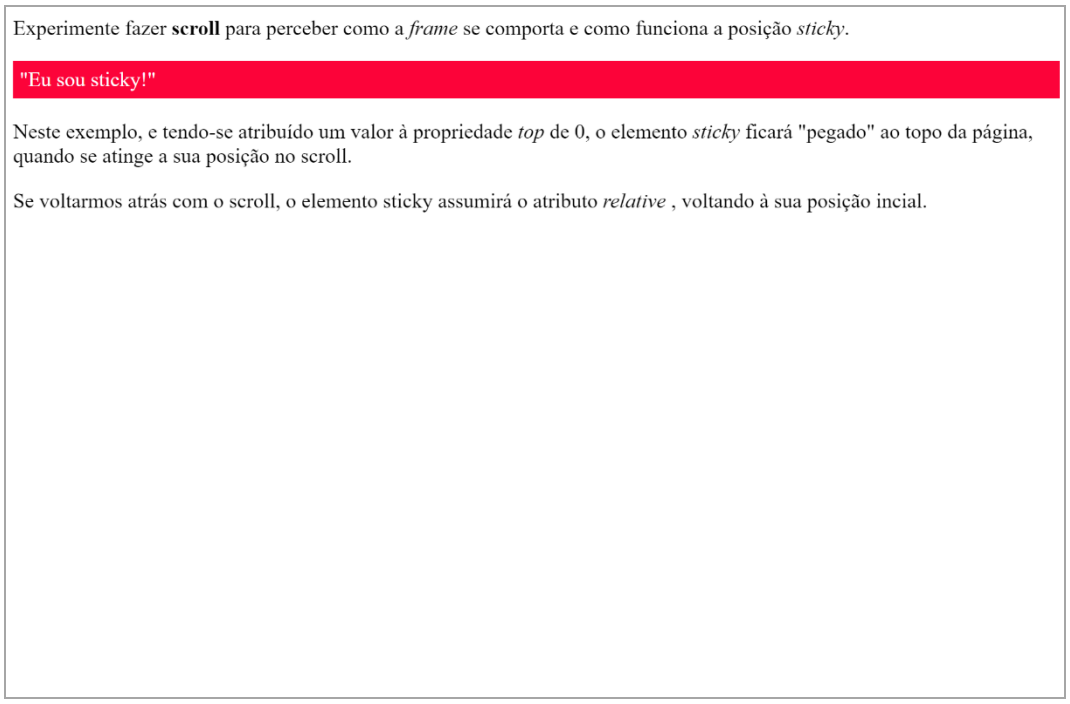
Um elemento com valor sticky posiciona-se tendo em conta a localização do scroll do utilizador.

O Internet Explorer, o Edge 15 e versões anteriores não suportam o valor position:sticky. O Safari requer que se adicione um prefixo -webkit- (como se mostra abaixo) e que se especifique pelo menos um dos valores de top, right, bottom ou left.position: -webkit-sticky.

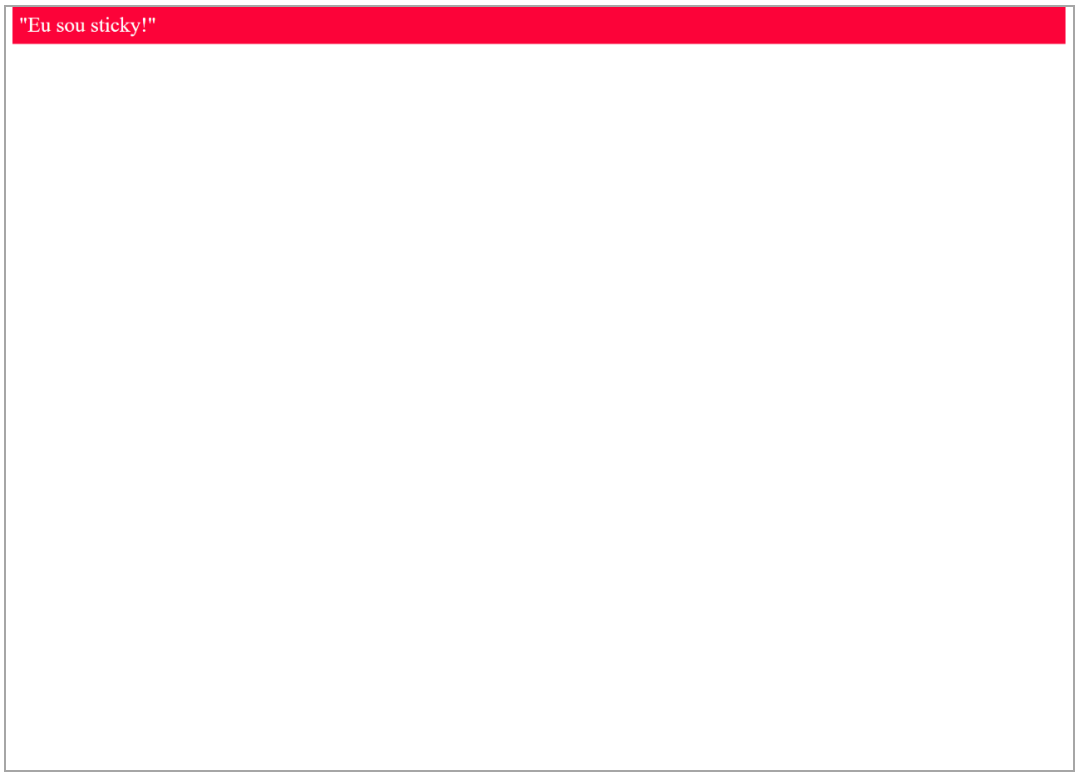
Exemplo

```
<!DOCTYPE html>
<html lang="pt">
<head>
<title> Position CSS</title>
      <meta charset="UTF-8">
      <meta name="description" content="position sticky">
      <meta name="author" content="Sara Granja">
<style>
div.sticky {
  position: -webkit-sticky;
  position: sticky;
  top: 0;
  padding: 5px;
  background-color: #fc0339;
  color: white;
}
</style>
```

```
</head>

<body>


<p>Experimente fazer <b>scroll</b> para perceber como a
<i>frame</i> se comporta e como funciona a posição
<i>sticky</i>.</p>


<div class="sticky">"Eu sou sticky!"</div>


<div style="padding-bottom:2000px">


  <p>Neste exemplo, e tendo-se atribuído um valor à propriedade
<i>top</i> de 0, o elemento <i>sticky</i> ficará "pegado" ao topo
da página, quando se atinge a sua posição no scroll.</p>


  <p>Se voltarmos atrás com o scroll, o elemento sticky assumirá o
atributo <i> relative</i>, voltando à sua posição inicial.</p>




</div>


</body>

</html>
```



Resultado do exemplo position sticky – vista geral antes do scroll.



Resultado do exemplo position:sticky – vista geral após o scroll.

2. O ATRIBUTO FLOAT

O atributo float (flutuar) especifica como o elemento ao qual se atribui esta propriedade vai flutuar na página. É usado em elementos aos quais se atribui um posicionamento relative, e pode assumir quatro valores distintos:

- **left**: o elemento flutua à esquerda do ecrã.
- **right**: o elemento flutua à direita do ecrã.
- **none**: o elemento não flutua e será exibido onde ocorre no texto. Este é o valor padrão.
- **inherit**: o elemento herda o valor flutuante do seu ancestral.

No ponto anterior, foi explicado que a posição relative vai colocar o elemento no espaço disponível seguinte do fluxo do documento, e dependendo do seu container. Ao adicionar a propriedade float, pode escolher-se se este elemento fica alinhado à esquerda ou à direita (ficando sempre situado no espaço livre seguinte do fluxo do documento).

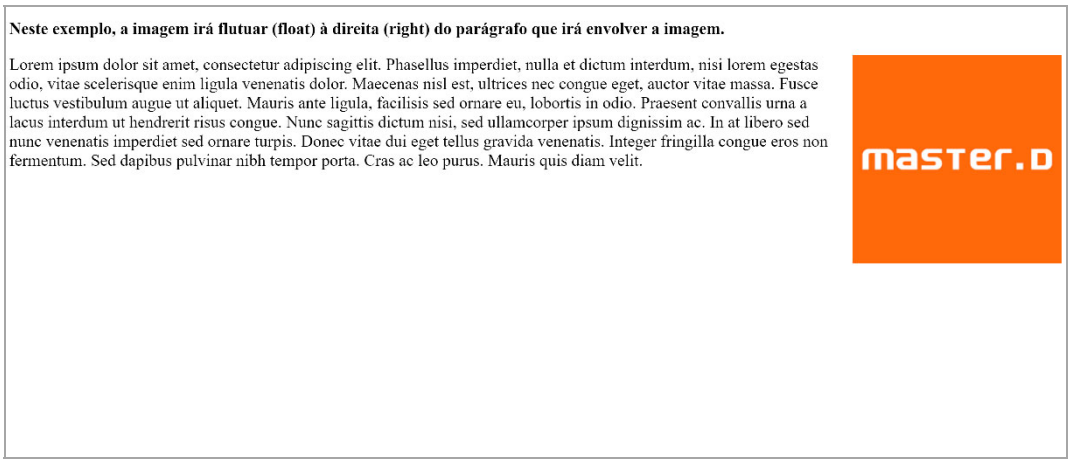
Atenção

O atributo float não funciona nas posições absolute e fixed, já que estas ocupam as posições top, right, bottom e left, e nunca seguem o fluxo do documento.

Exemplo

```
<!DOCTYPE html>
<html lang="pt">
<head>
<title> Float CSS</title>
    <meta charset="UTF-8">
    <meta name="description" content="float CSS">
    <meta name="author" content="Sara Granja">
<style>
img {
  float: right;
}
</style>
</head>
<body>


<p><strong>Neste exemplo, a imagem irá flutuar (float) à direita
(right) do parágrafo que irá envolver a imagem.</strong></p>


<p><img src="logo_masterd.png" alt="Logo MasterD"
style="width:200px;height:auto;margin-left:15px;">
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus
imperdiet, nulla et dictum interdum, nisi lorem egestas odio, vitae
scelerisque enim ligula venenatis dolor. Maecenas nisl est,
ultrices nec congue eget, auctor vitae massa. Fusce luctus
vestibulum augue ut aliquet. Mauris ante ligula, facilisis sed
ornare eu, lobortis in odio. Praesent convallis urna a lacus
interdum ut hendrerit risus congue. Nunc sagittis dictum nisi, sed
ullamcorper ipsum dignissim ac. In at libero sed nunc venenatis
imperdiet sed ornare turpis. Donec vitae dui eget tellus gravida
venenatis. Integer fringilla congue eros non fermentum. Sed dapibus
pulvinar nibh tempor porta. Cras ac leo purus. Mauris quis diam
velit.</p>


</body>
</html>
```

Resultado:

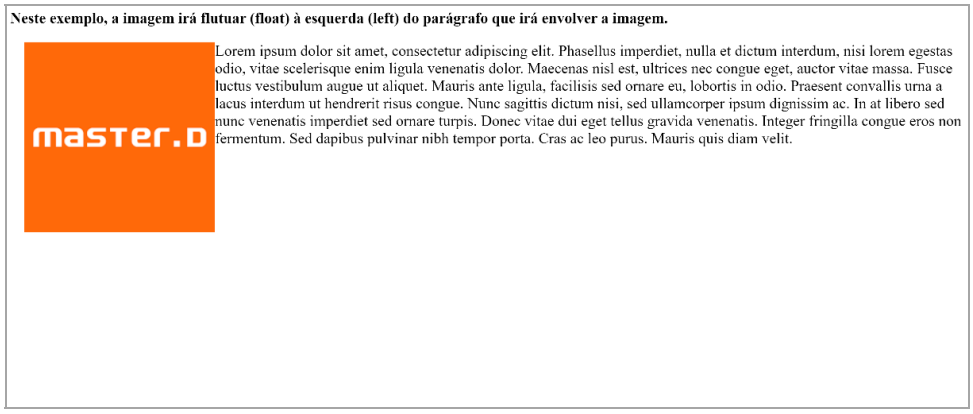


Resultado do exemplo float:right.

Exemplo

```
<!DOCTYPE html>
<html lang="pt">
<head>
<title> Float CSS</title>
    <meta charset="UTF-8">
    <meta name="description" content="float CSS">
    <meta name="author" content="Sara Granja">
<style>
img {
  float: left;
}
</style>
</head>
<body>
<p><strong>Neste exemplo, a imagem irá flutuar (float) à esquerda
(left) do parágrafo que irá envolver a imagem.</strong></p>
```

```
<p><img src="logo_masterd.png" alt="Logo MasterD"
style="width:200px;height:auto;margin-left:15px;">

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus
imperdiet, nulla et dictum interdum, nisi lorem egestas odio, vitae
scelerisque enim ligula venenatis dolor. Maecenas nisl est,
ultrices nec congue eget, auctor vitae massa. Fusce luctus
vestibulum augue ut aliquet. Mauris ante ligula, facilisis sed
ornare eu, lobortis in odio. Praesent convallis urna a lacus
interdum ut hendrerit risus congue. Nunc sagittis dictum nisi, sed
ullamcorper ipsum dignissim ac. In at libero sed nunc venenatis
imperdiet sed ornare turpis. Donec vitae dui eget tellus gravida
venenatis. Integer fringilla congue eros non fermentum. Sed dapibus
pulvinar nibh tempor porta. Cras ac leo purus. Mauris quis diam
velit.</p>


</body>

</html>
```

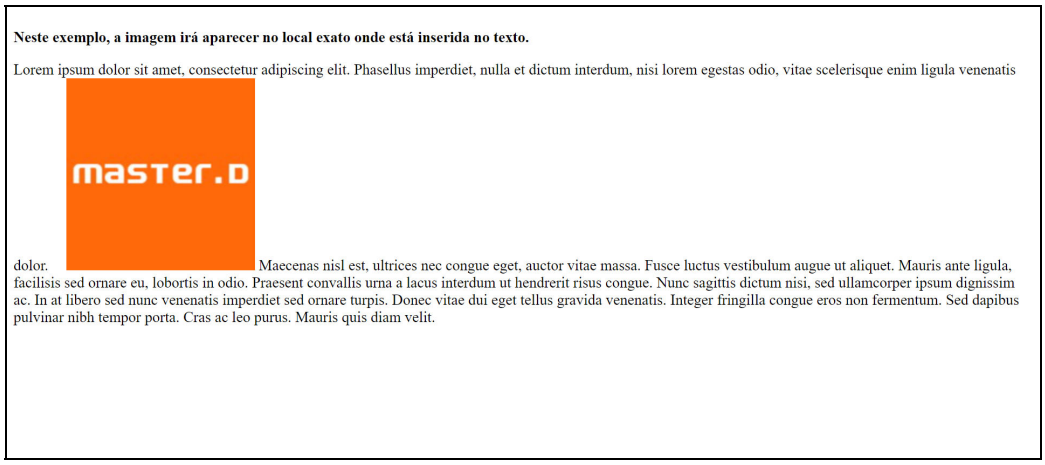


Resultado do exemplo float:left.

Exemplo

```
<!DOCTYPE html>
<html lang="pt">
<head>
<title> Float CSS</title>
     <meta charset="UTF-8">
     <meta name="description" content="float CSS">
     <meta name="author" content="Sara Granja">
<style>
img {
  float: none;
}
</style>
</head>
<body>


<p><strong>Neste exemplo, a imagem irá aparecer no local exato onde está inserida no texto. </strong></p>


<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus imperdiet, nulla et dictum interdum, nisi lorem egestas odio, vitae scelerisque enim ligula venenatis dolor.

<img src="logo_masterd.png" alt="Logo MasterD" style="width:200px;height:auto;margin-left:15px;">

Maecenas nisl est, ultrices nec congue eget, auctor vitae massa. Fusce luctus vestibulum augue ut aliquet. Mauris ante ligula, facilisis sed ornare eu, lobortis in odio. Praesent convallis urna a lacus interdum ut hendrerit risus congue. Nunc sagittis dictum nisi, sed ullamcorper ipsum dignissim ac. In at libero sed nunc venenatis imperdiet sed ornare turpis. Donec vitae dui eget tellus gravida venenatis. Integer fringilla congue eros non fermentum. Sed dapibus pulvinar nibh tempor porta. Cras ac leo purus. Mauris quis diam velit.</p>


</body>
```

```
</html>
```



Resultado do exemplo float:none.

3. O ATRIBUTO CLEAR

A propriedade clear especifica de que lado de um elemento flutuante outro elemento não pode flutuar.

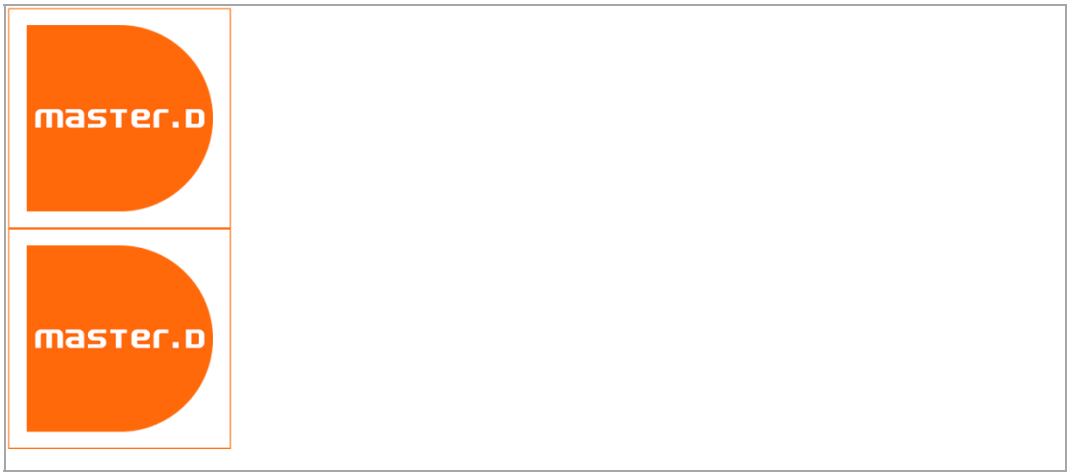
O atributo clear pode ter um dos seguintes valores:

- **none** (nenhum): valor padrão. Permite elementos flutuantes em ambos os lados.
- **left** (esquerda): não são permitidos elementos flutuantes do lado esquerdo.
- **right** (direita): não são permitidos elementos flutuantes do lado direito.
- **both** (ambos): não são permitidos elementos flutuantes em ambos os lados.
- **inherit** (herdar): o elemento herda o valor clear do seu elemento ancestral.

A forma mais comum de usar o atributo clear é depois de atribuir a propriedade float a um elemento.

Exemplo

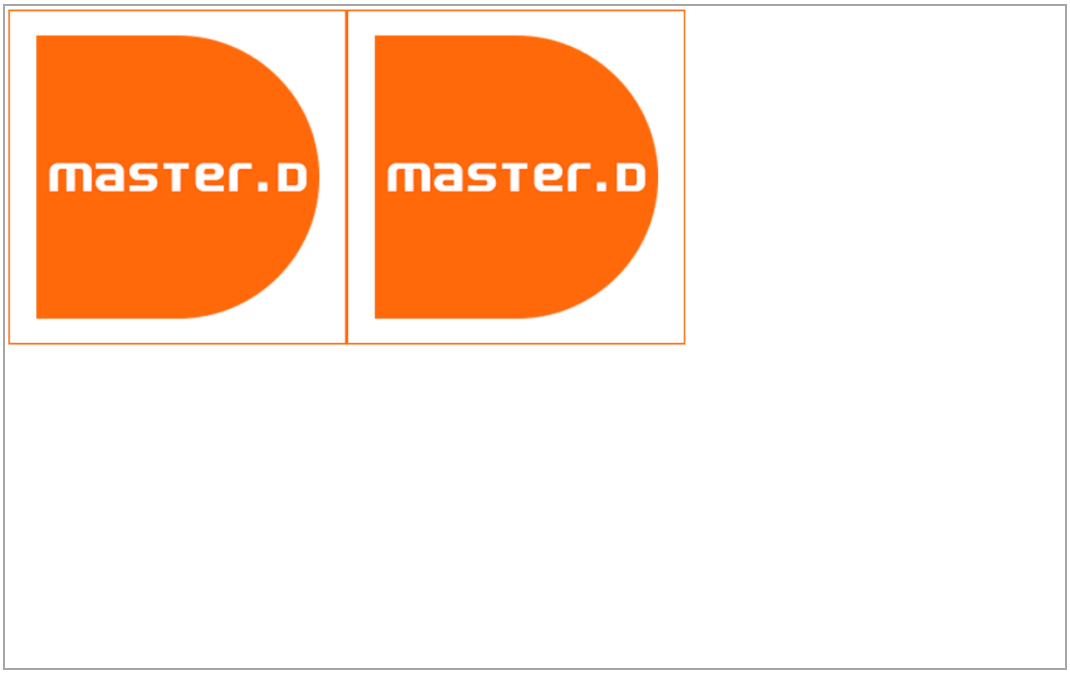
```
<!DOCTYPE html>
<html lang="pt">
<head>
<title> Clear CSS</title>
    <meta charset="UTF-8">
    <meta name="description" content="clear CSS">
    <meta name="author" content="Sara Granja">
<style>
img
{
float: left;
clear: both;
}
</style>
</head>


<body>
<img src="logo_masterd_roundside.png">
<img src="logo_masterd_roundside.png">
</body>
</html>
```



Resultado do exemplo float:left, com valor clear:both.

```
<!DOCTYPE html>
<html lang="pt">
<head>
<title> Clear CSS</title>
    <meta charset="UTF-8">
    <meta name="description" content="clear CSS">
    <meta name="author" content="Sara Granja">
<style>
img
{
float: left;
}
</style>
</head>
<body>


<body>
<img src="logo_masterd_roundside.png">
<img src="logo_masterd_roundside.png">
</body>
```

```
</html>
```



Resultado do exemplo float:left, sem valor clear.

4. O ATRIBUTO Z-INDEX

O atributo z-index permite a sobreposição de camadas: especifica a ordem em que os elementos são sobrepostos, ou seja, que elemento deve ser posicionado por cima ou por baixo de outro.

Um maior valor de z-index permite que o elemento esteja em cima de outro com um valor menor, independentemente da ordem em que os mesmos aparecem no código HTML. O valor deve ser um número inteiro, e pode assumir valores positivos ou negativos.

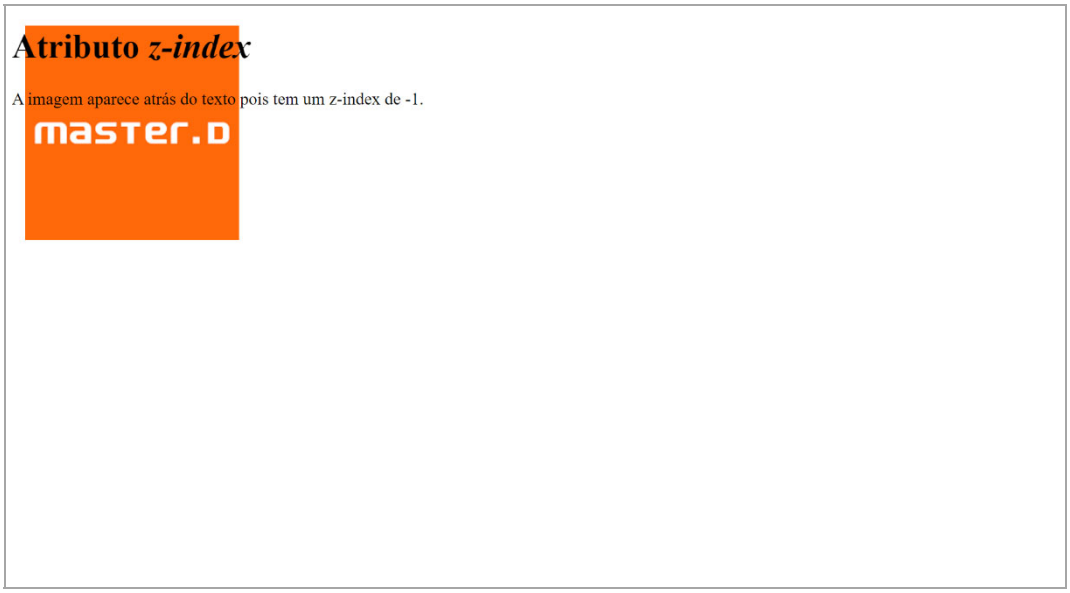
Nota

Se dois elementos são posicionados sem um valor *z-index* atribuído, os mesmos serão posicionados de acordo com a ordem no código HTML. O elemento escrito em ultimo aparecerá na frente.

Exemplo

```
<!DOCTYPE html>
<html lang="pt">
<head>
```

```
<title> Z-Index CSS</title>
    <meta charset="UTF-8">
    <meta name="description" content="z-index">
    <meta name="author" content="Sara Granja">
<style>
img {
  position: absolute;
  left: 20px;
  top: 20px;
  z-index: -1;
}
</style>
</head>
<body>

<h1>Atributo <i>z-index</i></h1>
<img src="logo_masterd.png" width="200px" height="auto">
<p>A imagem aparece atrás do texto pois tem um z-index de -1.</p>

</body>
</html>
```



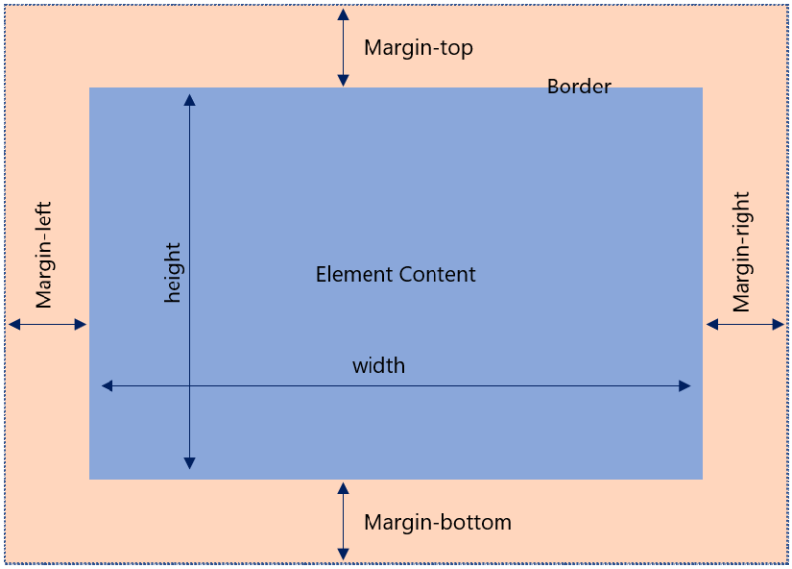
Resultado do exemplo de aplicação do atributo z-index.

5. O ATRIBUTO MARGIN

Uma margem (margin) é um espaço invisível que rodeia um elemento e que permite, ao mesmo tempo, manter distância de outros elementos. Em CSS, a propriedade margin permite definir esta distância e não é afetada por outros atributos do elemento (por exemplo, background-color), ficando sempre fora do mesmo.

Existem cinco variáveis para este atributo:

- **margin**: gera margem em todos os lados do elemento.
- **margin-top**: gera margem acima do elemento.
- **margin-bottom**: gera margem abaixo do elemento.
- **margin-left**: gera margem à esquerda do elemento.
- **margin-right**: gera margem à direita do elemento.



Esquema ilustrativo dos vários tipos de margem.

Os valores das margens podem ser:

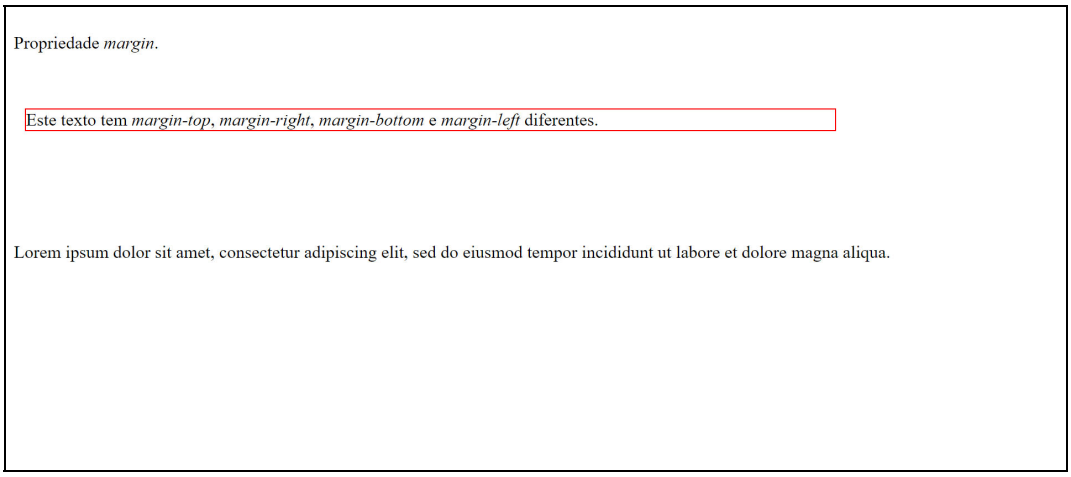
- **auto**: o navegador calcula a margem automaticamente.
- **lenght** (comprimento): especifica uma margem em px, pt, cm, ou qualquer outra unidade reconhecida por CSS.
- **%:** especifica uma margem em % da largura (width) do elemento que o contém.
- **inherit**: especifica que a margem deve ser a mesma do elemento ancestral.

Dica

As margens podem assumir valores negativos.

Exemplo

```
<!DOCTYPE html>
<html lang="pt">
<head>
<title>Margin CSS</title>
     <meta charset="UTF-8">
     <meta name="description" content="margin">
     <meta name="author" content="Sara Granja">
<style>
#texto1{
margin-top:50px;
margin-right:200px;
margin-bottom:100px;
margin-left:10px;
border:1px solid red;}
</style>
</head>
<body>


<p>Propriedade <i>margin</i>. </p>
<div id="texto1">Este texto tem <i>margin-top</i>, <i>margin-
right</i>, <i>margin-bottom</i> e <i>margin-left</i>
diferentes.</div>
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do
eiusmod tempor incididunt ut labore et dolore magna aliqua.</p>


</body>
</html>
```



Resultado do exemplo de aplicação da propriedade margin.

5.1. MARGIN – SHORTCODE

Para simplificar o código HTML/CSS pode usar-se o shortcode (código curto), com o qual é possível especificar todos os valores da margem numa só linha.

Imagine-se um elemento com diferentes valores para as quatro margens:

- margin-top: 25 px.
- margin-right: 50 px.
- margin-bottom: 75 px
- margin-left: 100 px.

O mesmo pode ser escrito em shortcode, da seguinte forma:

```
margin: 25px 50px 75px 100px;
```

Veja-se agora o caso de um elemento que tem margin-left e margin-right iguais:

- margin-top: 25 px.
- margin-right: 50 px.
- margin-bottom: 75 px.
- margin-left: 50 px.

```
margin: 25px 50px 75px;
```

Poderá ainda acontecer que valores de margin-top e margin-bottom sejam iguais e as margin-right e margin-left também (ainda que diferentes das anteriores). Assim:

- margin-top: 25 px.
- margin-right: 50 px.
- margin-bottom: 25 px.
- margin-left: 50 px.

O shortcode será:

```
padding: 25px 50px;
```

Por último, veja-se o caso em que o elemento assume valores iguais para as quatro margens diferentes:

- margin-top: 25 px.
- margin-right: 25 px.
- margin-bottom: 25 px.
- margin-left: 25 px.

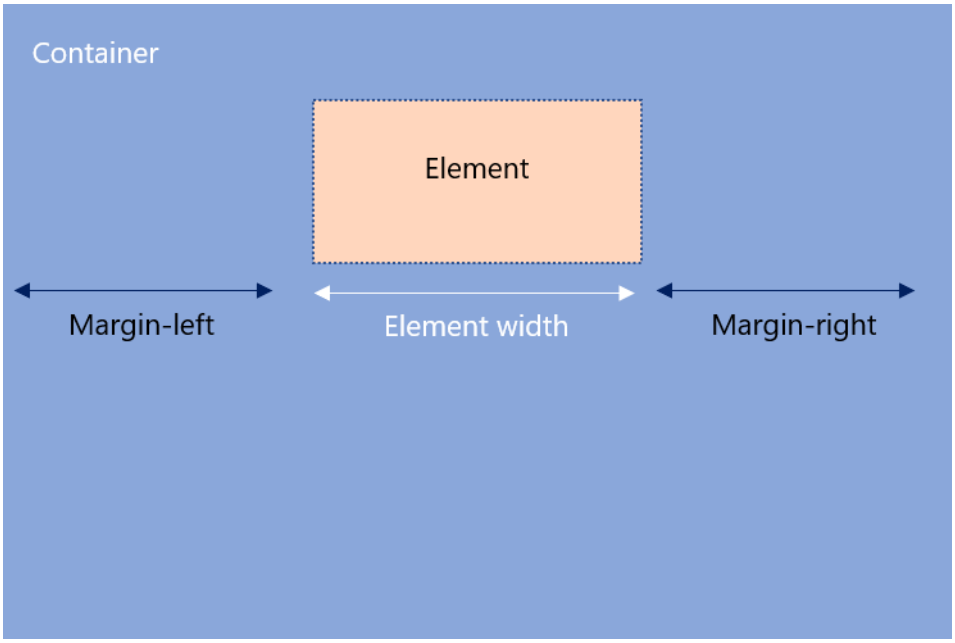
```
margin: 25px;
```

Os valores das margens podem ser:

- **auto**: o navegador calcula a margem automaticamente.
- **lenght** (comprimento): especifica uma margem em px, pt, cm, ou qualquer outra unidade reconhecida por CSS.
- **%**: especifica uma margem em % da largura (width) do elemento que o contém.
- **inherit**: especifica que a margem deve ser a mesma do elemento ancestral.

5.2. O VALOR AUTO

Para centrar um elemento horizontalmente no seu container pode usar-se o valor margin:auto. Com este valor o elemento será colocado, automaticamente, no centro do container, tendo em conta o espaço ocupado pelo elemento e dividindo, igualmente, o espaço restante pelas margens direita e esquerda.



Esquema ilustrativo da localização e distância quando aplicado o valor auto à margin.

6. O ATRIBUTO PADDING

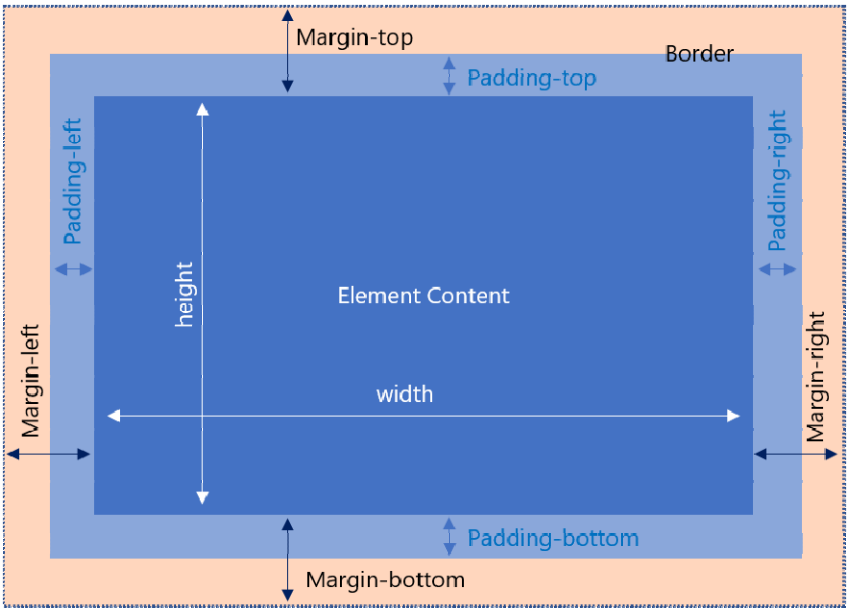
O atributo padding comporta-se exatamente da mesma forma que o atributo margin, com a diferença de que gera espaço em torno do elemento, dentro de quaisquer fronteiras (borders) definidas (no caso de não ter fronteiras definidas, o espaço será com a margem externa do elemento).

Nota

Ao contrário da margin, o espaço do elemento ao qual se atribui um valor para o padding é afetado por outras propriedades aplicadas ao elemento (por exemplo: background-color).

Existem cinco valores para o padding:

- **padding**: gera um espaço interno em todos os lados do elemento.
- **padding-top**: gera um espaço interno acima do elemento.
- **padding-bottom**: gera um espaço interno abaixo do elemento.
- **padding-left**: gera um espaço interno à esquerda do elemento.
- **padding-right**: gera um espaço interno à direita do elemento.



Esquema dos valores de margin e padding.

Os tipos de valores assumidos pelo padding são:

- **length** (comprimento): especifica um padding em px, pt, cm, etc.
- **%**:especifica um padding em % da largura do container do elemento.
- **inherit**: especifica que o padding assume o valor do seu ancestral.

Nota

Na propriedade padding não se podem atribuir valores negativos.

6.1. PADDING – SHORTCODE

Tal como acontece com a propriedade margin, também o padding pode ser escrito em shortcode, seguindo a mesma metodologia.

Para um elemento com valores de padding todos diferentes:

- padding-top: 5 px.
- margin-right: 10 px.
- margin-bottom: 15 px.
- margin-left: 20 px.

O shortcode da propriedade poderá ser escrito da seguinte forma:

```
padding: 5px 10px 15px 20px;
```

No caso de um elemento que tem padding-left e padding-right iguais:

- padding-top: 5 px.
- padding-right: 10 px.
- padding-bottom: 15 px.
- padding-left: 10 px.

O shortcode será:

```
padding: 5px 10px 15px;
```

Se os valores de padding-top e padding-bottom forem iguais e os padding-right e padding-left também (mas diferentes dos anteriores):

- padding-top: 5 px.
- padding-right: 10 px.

- padding-bottom: 5 px.
- padding-left: 10 px.

O shortcode será:

```
margin: 5px 10px;
```

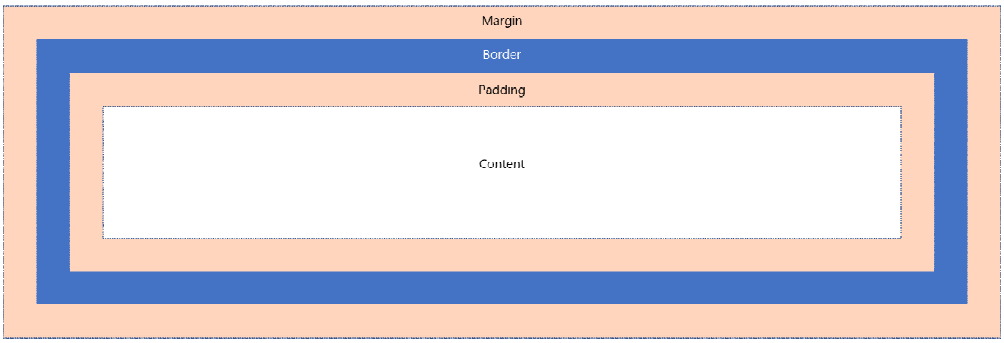
Por último, no caso em que o elemento tem valores iguais nos diferentes quatro padding:

- padding-top: 5 px.
- padding-right: 5 px.
- padding-bottom: 5 px.
- padding-left: 5 px.

```
padding: 5px;
```

6.2. PADDING E O ELEMENTO WIDTH

Ao atribuir um valor de padding a um elemento, e tendo em conta a CSS Box Model, este valor será adicionado à largura (width) do container do elemento. Pode-se, por isso, ter resultados indesejáveis. Para resolver o problema, terá de se recorrer à propriedade box-sizing, que fará com que a largura do container do elemento se mantenha com o valor inicial, independentemente do valor de padding atribuído.



CSS Box Model.

Exemplo

No caso de se ter um elemento div, com 300 px de largura (width) e se atribuir a esse elemento um padding de 25 px, no final, a largura desse elemento será de 350 px (width + padding-left + padding-right = 300 + 25 + 25 = 350 px).

```
div{
  width: 300px;
  padding: 25px;
}
```

Para que a largura do container se mantenha em 300 px, independentemente do valor do padding, terá de se acrescentar a propriedade border-box.

```
div{
  width: 300px;
  padding: 25px;
  box-sizing: border-box;
  }
```

CONCLUSÃO

Nesta unidade didática foram vistos os vários estilos de posição que a programação HTML/CSS permite: como e de que forma posicionar um elemento no seu container, através dos valores static, relative, fixed, absolute e sticky.

Foram também abordados outros atributos que os elementos podem tomar, tais como o float, que permite controlar o alinhamento dos elementos relativos, o clear, que evita que elementos sejam localizados nas laterais, e o z-index, para que se possa definir a sobreposição de elementos.

Expuseram-se ainda os conceitos de margin e padding como forma de criar espaço (externo e interno, respetivamente) em torno dos elementos.

AUTOAVALIAÇÃO

1. **Qual o valor que, por defeito, assumem os elementos HTML caso não seja atribuído nenhum valor ao atributo position?**

   a) Static.

   b) Relative.

   c) Fixed.

   d) Absolute.

2. **Onde é colocado um elemento ao qual se atribui o valor relative do atributo position?**

   a) No espaço livre seguinte do container e seguindo o fluxo normal da página.

   b) Num local aleatório entre o topo e o final da página.

   c) Num local fixo da página.

   d) No espaço livre anterior ao container.

3. Qual o valor a atribuir ao atributo position, se se pretender que um elemento seja visto sempre na mesma posição, mesmo fazendo scroll da página?

a) Static.

b) Relative.

c) Fixed.

d) Sticky.

4. O que acontece se, a um elemento com position:absolute se acrescentar o valor float:right?

a) Ele aparecerá à direita no espaço disponível seguinte no fluxo de documentos.

b) Ele aparecerá à direita do item seguinte que for criado.

c) Ele aparecerá à direita do último item na página.

d) Ele aparecerá à esquerda no espaço disponível.

5. Qual o valor a atribuir à propriedade position, para que um elemento se localize na posição do scroll?

a) Relative.

b) Sticky.

c) Fixed.

d) Static.

6. O que acontece a um elemento ao qual foi atribuído um valor clear:both?

a) Os elementos flutuantes que possam existir aparecem em ambos os lados do elemento ao qual foi atribuído o valor clear:both.

b) Quaisquer elementos que possam existir à esquerda desse elemento vão desaparecer.

c) Quaisquer elementos que possam existir em ambos os lados vão desaparecer.

d) Quaisquer elementos que possam existir à direita desse elemento vão desaparecer.

7. Se um elemento A tiver um valor z-index menor do que um elemento B, qual será o resultado em termos de sobreposição dos mesmos?

a) O elemento A aparecerá sobreposto ao elemento B.

b) O elemento B aparecerá sobreposto ao elemento A.

c) Os dois elementos desaparecerão.

d) Os elementos A e B aparecerão ao lado um do outro.

8. Qual o valor usado para a propriedade margin, para que um elemento fique centralizado horizontalmente no seu container:

a) margin: 50%.

b) margin: inherit.

c) margin: auto.

d) margin: 100px.

9. **Qual é o shortcode de um elemento que tenha os seguintes valores de padding: padding-top de 100 px, padding-right de 50 px, padding - bottom de 100 px e padding-left de 50 px?**

   a) padding: 100px.

   b) padding: 100px 50px.

   c) padding: 100px 100px 50px 50px.

   d) padding: 50px 100px 50px 100px.

10. **Quais as margens de um elemento a, caso lhe sejam atribuídas as seguintes propriedades?**

```
a{
margin: 20px 50px;
}
```

   a) Margens direita e esquerda, 20 px e margens de topo e inferiores, 50 px.

   b) Margens de topo e inferior, 20 px e margens direita e esquerda, 50 px.

   c) Todas as margens têm igual valor que será de 70 px.

   d) Margem de topo, 20 px, e margem inferior 50 px.

SOLUÇÕES

1.	a	**2.**	d	**3.**	c	**4.**	d	**5.**	b
6.	c	**7.**	b	**8.**	c	**9.**	b	**10.**	b

PROPOSTAS DE DESENVOLVIMENTO DO ESTUDO

O site da W3Schools é uma ótima ferramenta para praticar, tendo alguns exemplos e exercícios que poderá realizar:

- https://www.w3schools.com/css/css_positioning.asp.

BIBLIOGRAFIA

- Murphy, C. e Persson N. (2009), *HTML y CSS*. Madrid: Anaya Multimedia-Anaya Interactiva.