

ESPAÇAMENTOS E DIMENSÕES

Temos algumas maneiras de trabalhar com dimensões e espaçamentos. Para espaçamento interno e externo usamos respectivamente `padding` e `margin`, e para redimensionar elementos podemos usar as propriedades de largura e altura ou `width` e `height`. Vamos ver mais a fundo essas propriedades.

9.1 DIMENSÕES

É possível determinar as dimensões de um elemento, por exemplo:

```
p {  
  background-color: red;  
  height: 300px;  
  width: 300px;  
}
```

Todos os parágrafos do nosso HTML ocuparão 300 pixels de altura e de largura, com cor de fundo vermelha.

Se usarmos o inspetor de elementos do navegador veremos que o restante do espaço ocupado pelo elemento vira `margin`

Seus livros de tecnologia parecem do século passado?



Conheça a **Casa do Código**, uma **nova** editora, com autores de destaque no mercado, foco em **ebooks** (PDF, epub, mobi), preços **imbatíveis** e assuntos **atuais**.

Com a curadoria da **Caelum** e excelentes autores, é uma abordagem **diferente** para livros de tecnologia no Brasil.

[Casa do Código, Livros de Tecnologia.](#)

9.2 ESPAÇAMENTOS

Padding

A propriedade **padding** é utilizada para definir um espaçamento interno em elementos (por espaçamento interno queremos dizer a distância entre o limite do elemento, sua borda, e seu respectivo conteúdo) e tem as subpropriedades listadas a seguir:

- padding-top
- padding-right
- padding-bottom
- padding-left

Essas propriedades aplicam uma distância entre o limite do elemento e seu conteúdo acima, à direita, abaixo e à esquerda respectivamente. Essa ordem é importante para entendermos como funciona a *shorthand property* (encurtamento) do padding.

Podemos definir todos os valores para as subpropriedades do padding em uma única propriedade, chamada exatamente de `padding`, e seu comportamento é descrito nos exemplos a seguir:

Se passado somente um valor para a propriedade `padding`, esse mesmo valor é aplicado em todas as direções.

```
p {  
  padding: 10px;  
}
```

Se passados dois valores, o primeiro será aplicado acima e abaixo (equivalente a passar o mesmo valor para `padding-top` e `padding-bottom`) e o segundo será aplicado à direita e à esquerda (equivalente ao mesmo valor para `padding-right` e `padding-left`).

```
p {  
  padding: 10px 15px;  
}
```

Se passados três valores, o primeiro será aplicado acima (equivalente a `padding-top`), o segundo será aplicado à direita e à esquerda (equivalente a passar o mesmo valor para `padding-right` e `padding-left`) e o terceiro valor será aplicado abaixo do elemento (equivalente a `padding-bottom`).

```
p {  
  padding: 10px 20px 15px;  
}
```

Se passados quatro valores, serão aplicados respectivamente a `padding-top`, `padding-right`, `padding-bottom` e `padding-left`. Para facilitar a memorização dessa ordem, basta lembrar que os valores são aplicados em **sentido horário**.

```
p {
```

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

Uma dica para omitir valores do padding:

Quando precisar omitir valores, sempre omita no sentido anti-horário começando a partir da subpropriedade `-left`.

Como os valores tem posicionamento fixo na hora de declarar os espaçamentos, o navegador não sabe quando e qual valor deve ser omitido. Por exemplo:

Se tivermos um padding:

```
h1 {
  padding: 10px 25px 10px 15px;
}
```

O código não pode sofrer o encurtamento porque por mais que os valores de `top` e `bottom` sejam iguais, os valores `right` e `left` não são e eles são os primeiros a serem omitidos. Veja o que acontece quando vamos omitir o valor de `10px` do `bottom`:

```
h1 {
  padding: 10px 25px 15px;
}
```

O navegador vai interpretar da seguinte maneira:

```
h1 {
  padding: top right bottom;
}
```

Que no final vai ficar igual a:

```
h1 {
  padding-top: 10px;
  padding-right: 25px;
  padding-bottom: 15px;
  padding-left: 25px;
}
```

E esses valores não são os que nós colocamos no começo com `padding: 10px 25px 10px 15px;`

Margin

A propriedade `margin` é bem parecida com a propriedade `padding`, exceto que ela adiciona espaço após o limite do elemento, ou seja, é um espaçamento além do elemento em si (espaçamento externo). Além das subpropriedades listadas a seguir, há a *shorthand property* `margin` que se comporta da mesma maneira que a *shorthand property* do `padding` vista no tópico anterior.

- `margin-top`
- `margin-right`
- `margin-bottom`
- `margin-left`

Há ainda uma maneira de permitir que o navegador defina qual será a dimensão da propriedade `padding` ou `margin` conforme o espaço disponível na tela: definimos o valor `auto` para os espaçamentos que quisermos.

No exemplo a seguir, definimos que um elemento não tem nenhuma margem acima ou abaixo de seu conteúdo e que o navegador define uma margem igual para ambos os lados de acordo com o espaço disponível:

```
p {  
  margin: 0 auto;  
}
```