

- > Diferença entre margin e padding
- > Configurando a propriedade margin
- > Configurando a propriedade padding

IOS – Instituto de
Oportunidade Social

Padding



A propriedade **padding** configura o **espaço vazio entre o conteúdo do elemento HTML (geralmente texto) e a borda**. Por **padrão**, o **preenchimento** é definido como 0. Se você configurar uma cor de fundo para um elemento, a cor é aplicada tanto ao preenchimento quanto ao conteúdo das áreas.

> Vamos Praticar

IOS – Instituto de
Oportunidade Social

Margin



A propriedade **margin** simplesmente **adiciona uma margem ao seu elemento**. Você pode utilizar qualquer **medida Css (px, pt, em, %...)** como tamanho da propriedade margin, além disso você pode atribuir **valores negativos**, mas tenha cuidado com eles.

margin

Notação abreviada para configurar a margem em torno de um elemento. Com essa propriedade você pode configurar as quatro margens:
`margin: top right bottom left`

Um valor numérico (px ou em), por exemplo:
`body { margin: 25px 50px 75px 100px; }`
Definirá as margens da página
Margem superior (top margin) será 25px
Margem direita (right margin) será 50px
Margem inferior (bottom margin) será 75px
Margem esquerda (left margin) será 100px.
Ao eliminar a margem, não use a unidade px ou em, por exemplo: `body {margin: 0}`

> Vamos Praticar

IOS – Instituto de
Oportunidade Social

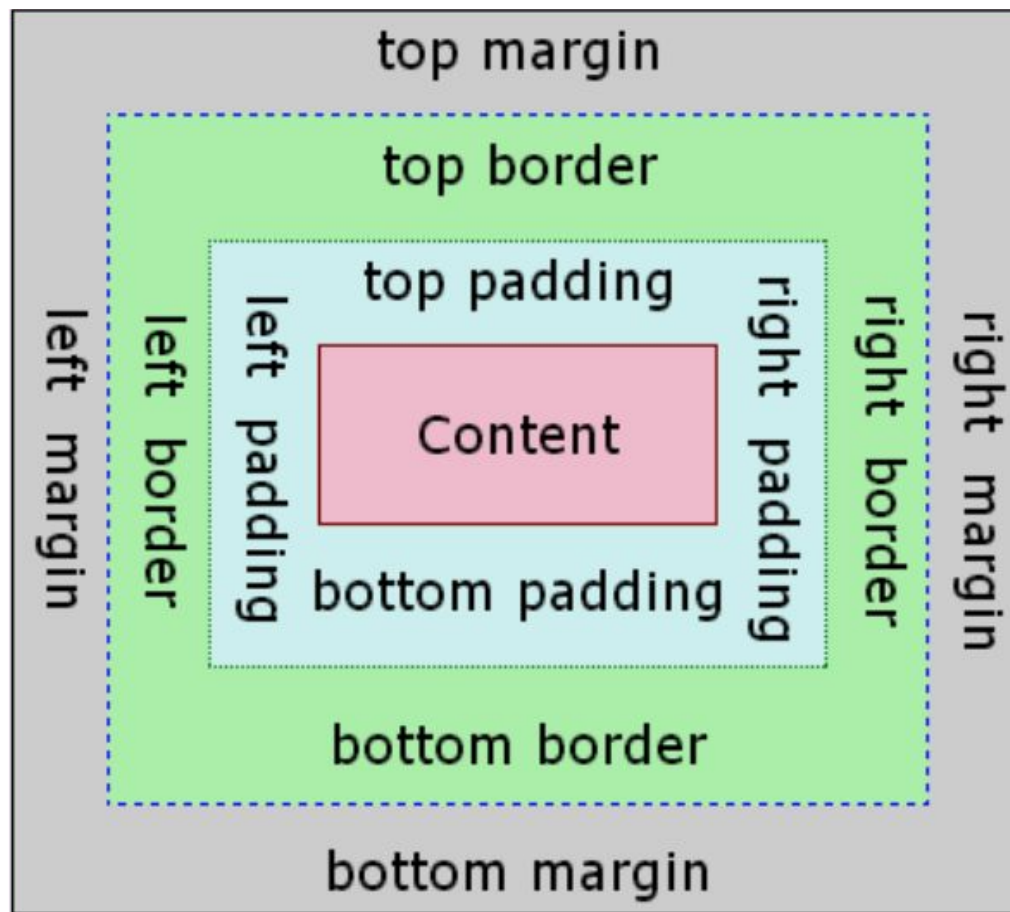
Diferença entre margin e padding



Quando estamos começando em CSS geralmente podemos confundir os conceitos de **margin** e **padding**. Isso acontece, porque os dois são essenciais, nos permitindo dar espaçamento entre elementos do HTML, mas existem importantes diferenças fundamentais no uso desses dois estilos.

Antes de ver a diferença entre essas duas propriedades, vamos entender o conceito de **Box Model**. No **CSS**, o termo **Box Model** é usado quando estamos trabalhando com o design ou o layout em uma página web. Cada elemento HTML em um documento é visto como uma caixa (**box**) retangular e o **CSS box model** é essencialmente essa caixa com diversas camadas em volta do conteúdo, que é formada por **padding**, **border** e **margin**.

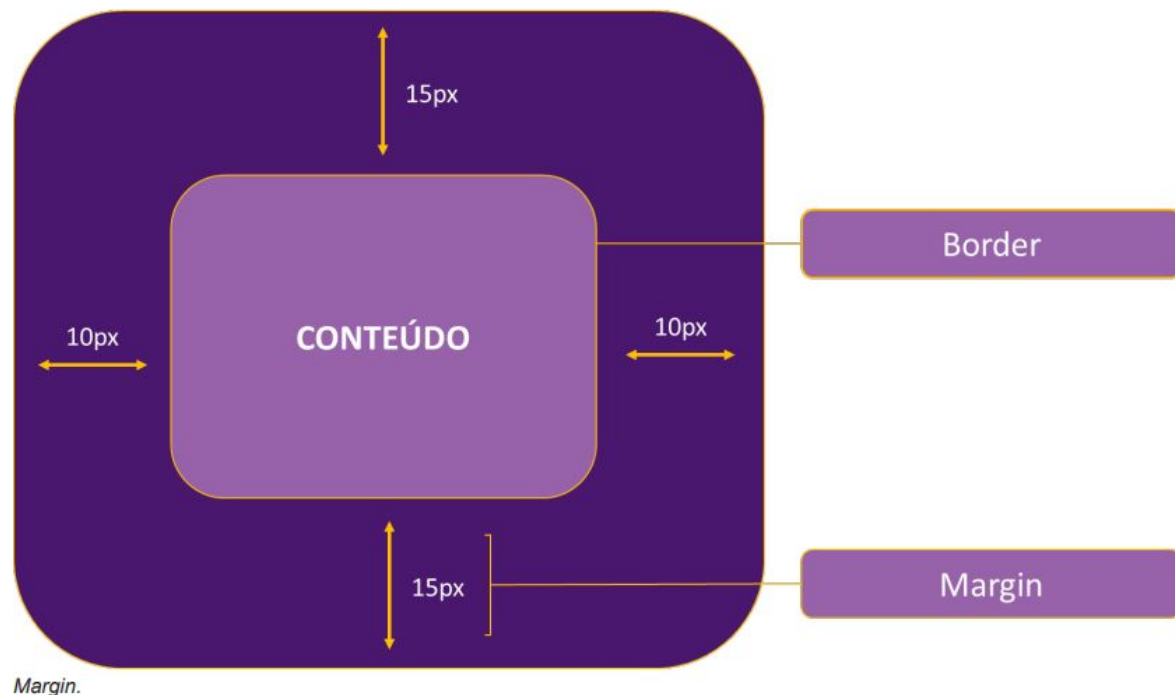
Diferença entre margin e padding



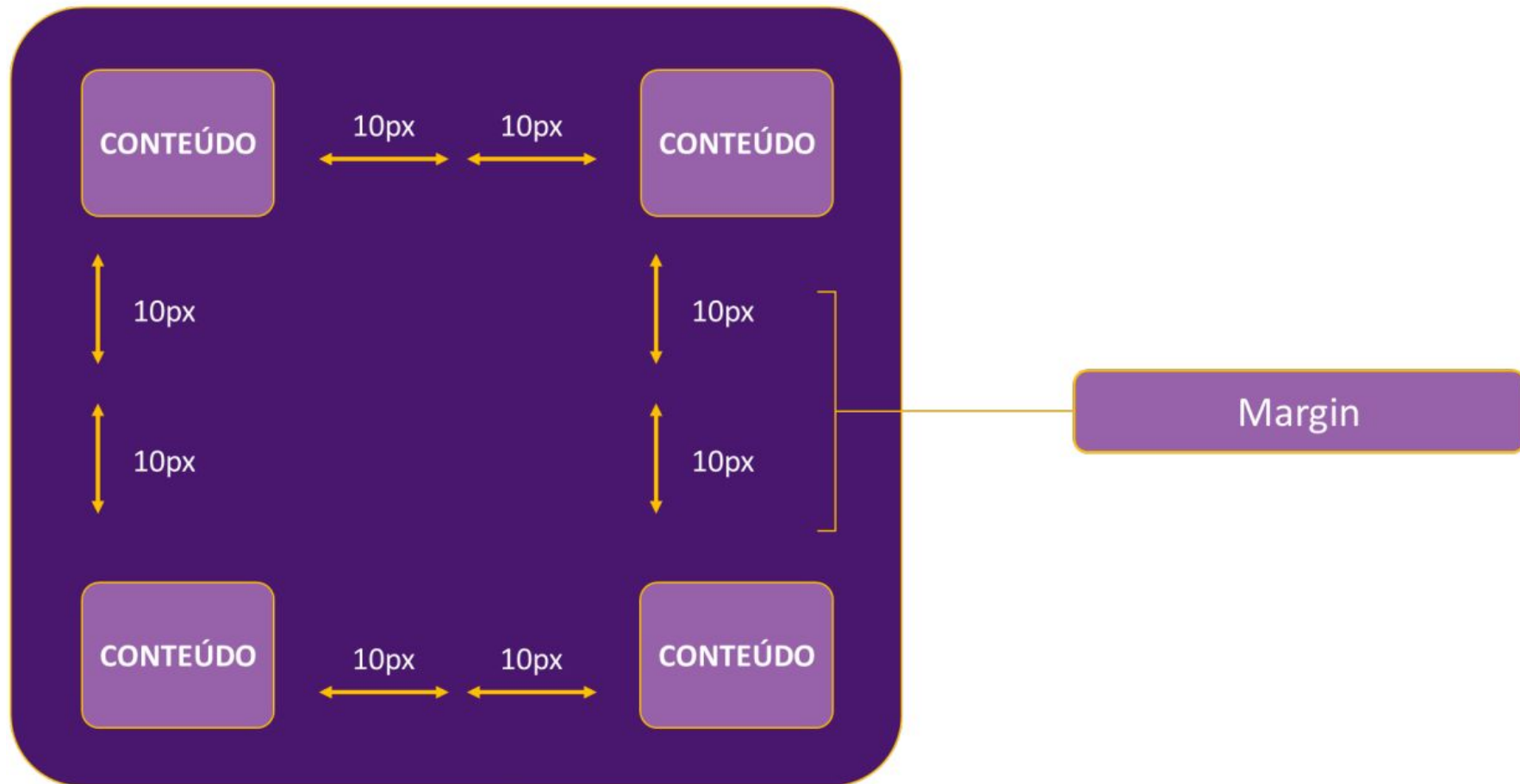
Diferença entre margin e padding.

Diferença entre margin e padding

A propriedade **margin** é o espaço entre a borda do elemento HTML e o próximo elemento da sua página web. Ou seja, está do lado de fora da borda.



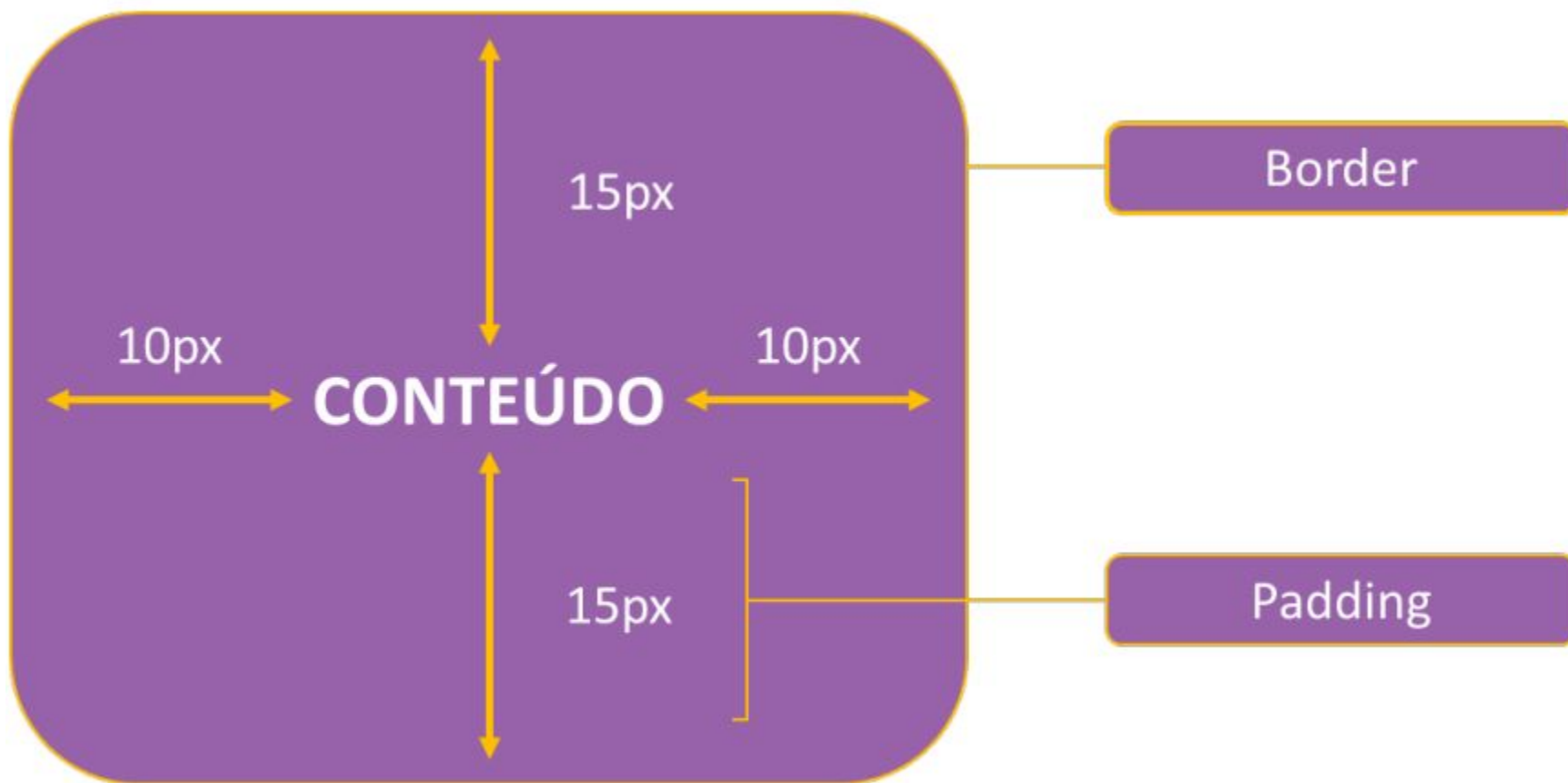
Diferença entre margin e padding



Margem entre outros elementos HTML da página web.

Por sua vez, a propriedade padding é o espaço entre o conteúdo e a borda. Ou seja, está do lado de dentro da borda. Assim, a propriedade **padding** é usada para definir o espaço em torno do conteúdo e, também, contorna os quatro lados e pode ser configurado separadamente cada um dos lados.

Diferença entre margin e padding



Padding.

IOS – Instituto de
Oportunidade Social

Configurando a propriedade margin



No **CSS**, a propriedade **margin** permite definir a margem com valores numéricos (**px**, **cm**, **pt**, **em**, **rem**, etc.) ou valores percentuais (**%**). Ambos podem ser **negativos** ou **positivos**. Você pode configurar cada uma das margens de um elemento HTML separadamente ou todas de uma vez.

Configurando a propriedade margin



Para configurar cada uma das margens você pode utilizar as instruções: **margin-top**, **margin-right**, **margin-bottom** e **margin-left**. Por exemplo, o trecho a seguir configura a margem superior com **100 pixels**, a inferior com **100 pixels**, a margem da esquerda com **80 pixels** e a da direita com **150 pixels** de um parágrafo.

```
p {  
    margin-top: 100px;  
    margin-bottom: 100px;  
    margin-right: 150px;  
    margin-left: 80px;  
}
```

Configurando a propriedade margin



Para fazer a mesma configuração anterior, mas com uma única instrução você deve usar o seguinte trecho:

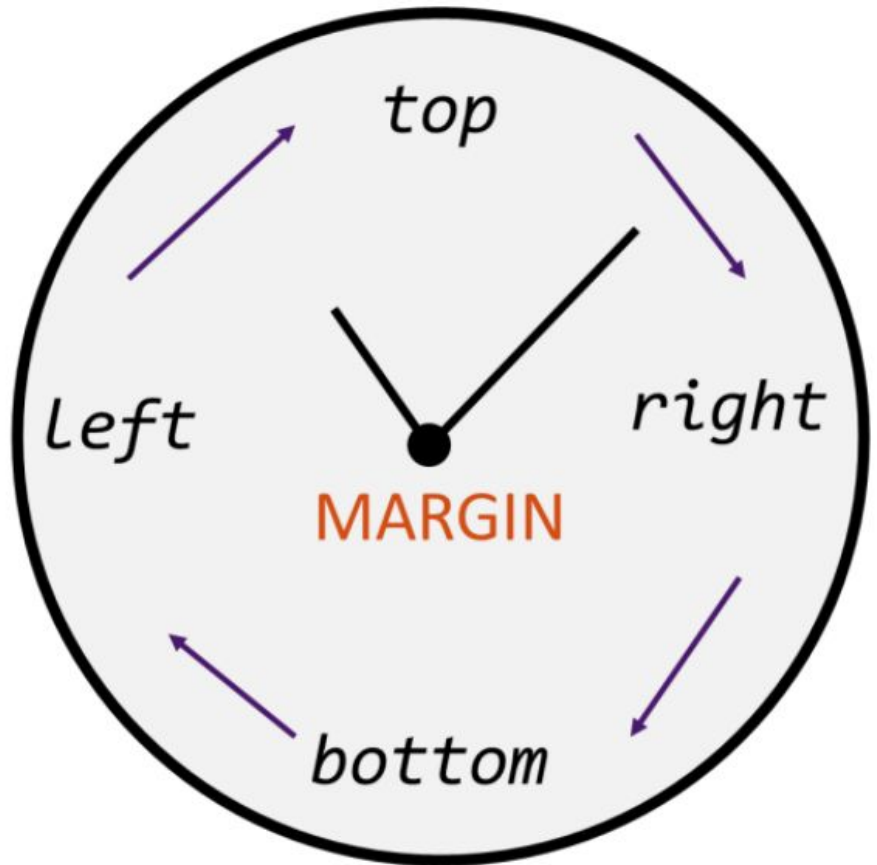
```
p {  
    /* margin: top right bottom left */  
    margin: 100px 150px 100px 80px;  
}
```

Configurando a propriedade margin



Dica!

Para memorizar essa notação da CSS, basta lembrar dos ponteiros de um relógio! Ela também pode ser usada em outros elementos, como o **padding** por exemplo.



Existem outros formatos de usar a instrução única. Por exemplo, caso você queira configurar as margens **top** e **bottom** com valores de **25 pixels** e **right** e **left** com **50 pixels**, basta usar o trecho:

```
p {  
    /* margin: (top e bottom) (right left)*/  
    margin: 25px 50px;  
}
```

Ou configurar as margens **top** com **25 pixels** e **right** e **left** com **50 pixels** e bottom com **75 pixels**, basta usar o trecho:

```
p {  
    /* margin: (top) (right bottom) (left) */  
    margin: 25px 50px 75px;  
}
```

> Vamos Praticar

Configurando a propriedade margin



Ou configurar as quatro **margens** com **25 pixels**, basta usar o trecho:

```
p {  
    /* margin: top right bottom left */  
    margin: 25px;  
}
```

> Vamos Praticar

Você também pode usar o valor **auto**, que deixa para o navegador calcular a margem do elemento automaticamente ou usar o valor **inherit**, que especifica que a margem deverá ser herdada do **elemento pai** (**primeiro elemento acima na hierarquia do design da página web**).

> Vamos Praticar

IOS – Instituto de
Oportunidade Social

Configurando a propriedade padding



No **CSS**, a propriedade padding permite definir o espaçamento com valores numéricos (**px**, **cm**, **pt**, **em**, **rem**, etc.), que podem ser apenas **positivos**. Você pode configurar separadamente cada espaçamento de um elemento **HTML** ou todos de uma vez. Para configurar cada uma das margens você pode utilizar as instruções: **padding-top**, **padding-right**, **padding-bottom** e **padding-left**.

> Vamos Praticar

Configurando a propriedade padding



Por exemplo, o trecho a seguir configura o espaçamento superior com 100 pixels, o inferior com 100 pixels, o da esquerda com 80 pixels e o da direita com 150 pixels de um parágrafo.

```
p {  
    padding-top: 100px;  
    padding-bottom: 100px;  
    padding-right: 150px;  
    padding-left: 80px;  
}
```