Espaçamento

Bootstrap tem um grande leque de atalhos responsivos de margem e padding para modificar a aparência de um elemento.

Como funciona

Use valores de `margin` e `padding` responsivos, em um elemento ou em alguns de seus lados, usando nossas classes de utilitárias. Elas dão suporte para propriedades individuais, todas as propriedades, proprieades verticais ou horizontais. Essas classes são construídas a partir de um mapa Sass, que possui valores variando de `.25rem` até `3rem`.

Notação

Utilitários de espaçamento que se aplicam a todos breakpoints (de `xs` até `xl`), não possuem abreviações. Isto é porque essas classes são usadas de `min-width: 0` pra cima, sem media queries limitantes. No entanto, os demais utilitários possuem abreviações de breakpoint.

As classes são definidas usando o formato `{propriedade}{lados}-{tamanho}` para o breakpoint `xs` e `{prorieade}{lados}-{breakpoint}-{tamanho}` para `sm`, `md`, `lg` e `xl`.

O campo *propriedade* pode ser:

- `m` - para usar `margin`;
- `p` - para usar `padding`.

O campo *lados* pode ser:

- `t` - para usar `margin-top` ou `padding-top`;
- `b` - para usar `margin-bottom` ou `padding-bottom`;
- `l` - para usar `margin-left` ou `padding-left`;
- `r` - para usar `margin-right` ou `padding-right`;
- `x` - para usar margem ou padding, tanto na esquerda quanto direita;
- `y` - para usar margem ou padding, tanto na parte superior quanto na inferior;
- Deixe o campo em branco, para usar `margin` ou `padding` em todos os quatro lados do elemento.

O campo *tamanho* pode ser:

- `0` - para remover o `padding` ou `margin`;
- `1` - para usar a `margin` ou `padding` com valor de `$spacer * .25`;
- `2` - para usar a `margin` ou `padding` com valor de `$spacer * .5`;
- `3` - para usar a `margin` ou `padding` com valor de `$spacer`;
- `4` - para usar a `margin` ou `padding` com valor de `$spacer * 1.5`;
- `5` - para usar a `margin` ou `padding` com valor de `$spacer * 3`;
- `auto` - para usar a `margin` com valor de `auto`.

Ps: você pode adicionar mais tamanhos, colocando mais valores no mapa Sass `$spacers`.

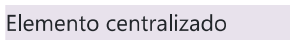
Exemplos

Aqui estão, alguns exemplos de como funcionam as classes:

```scss
.mt-0 {
  margin-top: 0 !important;
}

.ml-1 {
  margin-left: ($spacer * .25) !important;
}

.px-2 {
  padding-left: ($spacer * .5) !important;
  padding-right: ($spacer * .5) !important;
}

.p-3 {
  padding: $spacer !important;
}
```

Centralização horizontal

Adicionalmente, o Bootstrap também possui uma classe `.mx-auto` para centralização horizontal de elementos de largura fixa e `display: block;`.




```html
<div class="mx-auto" style="width: 200px;">
  Elemento centralizado
</div>
```

Negative margin

In CSS, `margin` properties can utilize negative values (`padding` cannot). As of 4.2, we've added negative margin utilities for every non-zero integer size listed above (e.g., `1`, `2`, `3`, `4`, `5`). These utilities are ideal for customizing grid column gutters across breakpoints.

The syntax is nearly the same as the default, positive margin utilities, but with the addition of `n` before the requested size. Here's an example class that's the opposite of `.mt-1`:

```scss
.mt-n1 {
  margin-top: -0.25rem !important;
}
```

Here's an example of customizing the Bootstrap grid at the medium (`md`) breakpoint and above. We've increased the `.col` padding with `.px-md-5` and then counteracted that with `.mx-md-n5` on the parent `.row`.


```html
<div class="row mx-md-n5">
  <div class="col py-3 px-md-5 border bg-light">Custom column padding</div>
  <div class="col py-3 px-md-5 border bg-light">Custom column padding</div>
</div>
```