

# Pontos de quebra

API que permite o uso de pontos de quebra em uma ampla variedade de contextos.

Para uma experiência de usuário ideal, as interfaces do material design precisam adaptar seu leiaute em vários pontos de quebra. Material-UI usa uma implementação **simplificada** da [especificação](#) original.

The breakpoints are used internally in various components to make them responsive, but you can also take advantage of them for controlling the layout of your application through the [Grid](#) component.

## Pontos de quebra padrão

Cada ponto de quebra (uma chave) corresponde a uma largura de tela *fixa* (um valor):

- **xs**, extra-pequeno: 0px
- **sm**, pequeno: 600px
- **md**, medium: 900px
- **lg**, large: 1200px
- **xl**, extra-large: 1536px

Esses valores podem ser [customizados](#).

## Consultas de Mídia CSS

Consultas de mídia CSS são a abordagem idiomática para tornar sua interface de usuário responsiva. The theme provides five styles helpers to do so:

- [theme.breakpoints.up\(key\)](#)
- [theme.breakpoints.down\(key\)](#)
- [theme.breakpoints.only\(key\)](#)
- [theme.breakpoints.not\(key\)](#)
- [theme.breakpoints.between\(start, end\)](#)

Na demonstração a seguir, alteramos a cor do plano de fundo (vermelho, azul & verde) com base na largura da tela.

```
const styles = (theme) => ({
  root: {
    padding: theme.spacing(1),
    [theme.breakpoints.down('md')]: {
      backgroundColor: theme.palette.secondary.main,
    },
    [theme.breakpoints.up('md')]: {
      backgroundColor: theme.palette.primary.main,
    },
    [theme.breakpoints.up('lg')]: {
      backgroundColor: green[500],
    },
  },
});
```

```
{{"demo": "MediaQuery.js"}}
```

## Consultas de mídia JavaScript

Às vezes, usar CSS não é suficiente. Você pode querer alterar a árvore de renderização React com base no valor do ponto de quebra, em JavaScript.

### useMediaQuery hook

Você pode aprender mais na página [useMediaQuery](#).

## Pontos de quebra customizados

Você define os pontos de quebra do seu projeto na seção `theme.breakpoints` do seu tema.

- `theme.breakpoints.values` : Padrão são [os valores acima](#). As chaves são seus nomes de tela e os valores são a largura mínima onde esse ponto de quebra deve iniciar.
- `theme.breakpoints.unit` : Padrão é `px`. A unidade usada para os valores do ponto de quebra.
- `theme.breakpoints.step` : Padrão é 5 ( `0.05px` ). O incremento usado para implementar os pontos de quebra exclusivos. For example, `{ step: 5 }` means that `down(500)` will result in `'(max-width: 499.95px)'`.

Se você alterar os valores dos pontos de quebra padrão, você precisará fornecer novos conforme descreveremos:

```
const theme = createTheme({
  breakpoints: {
    values: {
      xs: 0,
      sm: 600,
      md: 900,
      lg: 1200,
      xl: 1536,
    },
  },
});
```

Sinta-se à vontade para ter quantos pontos de quebra você quiser, nomeando-os da maneira que preferir para o seu projeto.

```
const theme = createTheme({
  breakpoints: {
    values: {
      mobile: 0,
      tablet: 640,
      laptop: 1024,
      desktop: 1200,
    },
  },
});
```

Se você estiver usando TypeScript, você também deverá usar a [extensão de módulos](#) para que o tema aceite os valores acima.

```
declare module '@mui/material/styles' {
  interface BreakpointOverrides {
    xs: false; // removes the `xs` breakpoint
    sm: false;
    md: false;
    lg: false;
    xl: false;
    mobile: true; // adds the `mobile` breakpoint
    tablet: true;
    laptop: true;
    desktop: true;
  }
}
```

## API

**theme.breakpoints.up(key) => media query**

### Argumentos

1. **key** (*string | number*): A breakpoint key ( `xs` , `sm` , etc.) or a screen width number in px.

### Retornos

**media query** : Uma string de consulta de mídia pronta para ser usada com a maioria das soluções de estilo, na qual corresponde à largura da tela maior que, e incluindo o tamanho de tela fornecido pela chave do ponto de quebra.

### Exemplos

```
const styles = (theme) => ({
  root: {
    backgroundColor: 'blue',
    // Match [md, ∞)
    // [900px, ∞)
    [theme.breakpoints.up('md')]: {
      backgroundColor: 'red',
    },
  },
});
```

**theme.breakpoints.down(key) => media query**

### Argumentos

1. **key** (*string | number*): A breakpoint key ( `xs` , `sm` , etc.) or a screen width number in px.

### Retornos

**media query** : Uma string de consulta de mídia pronta para ser usada com a maioria das soluções de estilo, na qual corresponde à largura da tela menor que, e incluindo o tamanho de tela fornecido pela chave do ponto de quebra.

## Exemplos

```
const styles = (theme) => ({
  root: {
    backgroundColor: 'blue',
    // Match [0, md)
    //       [0, 900px)
    [theme.breakpoints.down('md')]: {
      backgroundColor: 'red',
    },
  },
});
```

**theme.breakpoints.only(key) => media query**

### Arguments

1. **key** (*string*): A breakpoint key ( `xs` , `sm` , etc.).

### Returns

**media query** : Uma string de consulta de mídia pronta para ser usada com a maioria das soluções de estilo, na qual corresponde à largura da tela incluindo o tamanho de tela fornecido pela chave do ponto de quebra.

## Examples

```
const styles = (theme) => ({
  root: {
    backgroundColor: 'blue',
    // Match [md, md + 1)
    //       [md, lg)
    //       [900px, 1200px)
    [theme.breakpoints.only('md')]: {
      backgroundColor: 'red',
    },
  },
});
```

**theme.breakpoints.not(key) => media query**

### Arguments

1. **key** (*string*): A breakpoint key ( `xs` , `sm` , etc.).

### Returns

**media query** : A media query string ready to be used with most styling solutions, which matches screen widths stopping at the screen size given by the breakpoint key (exclusive) and starting at the screen size given by the next breakpoint key (inclusive).

## Examples

```
const styles = (theme) => ({
  root: {
    backgroundColor: 'blue',
    // Match [xs, md) and [md + 1, ∞)
    //      [xs, md) and [lg, ∞)
    //      [0px, 900px) and [1200px, ∞)
    [theme.breakpoints.not('md')]: {
      backgroundColor: 'red',
    },
  },
});
```

**theme.breakpoints.between(start, end) => media query**

### Arguments

1. **start** (*string*): A breakpoint key ( `xs` , `sm` , etc.) or a screen width number in px.
2. **end** (*string*): A breakpoint key ( `xs` , `sm` , etc.) or a screen width number in px.

### Returns

**media query** : A media query string ready to be used with most styling solutions, which matches screen widths greater than the screen size given by the breakpoint key in the first argument (inclusive) and less than the screen size given by the breakpoint key in the second argument (exclusive).

### Examples

```
const styles = (theme) => ({
  root: {
    backgroundColor: 'blue',
    // Match [sm, md)
    //      [600px, 900px)
    [theme.breakpoints.between('sm', 'md')]: {
      backgroundColor: 'red',
    },
  },
});
```

## Valores padrão

You can explore the default values of the breakpoints using [the theme explorer](#) or by opening the dev tools console on this page ( `window.theme.breakpoints` ).