

Breakpoints 断点

API 解锁了各种情况下断点的使用。

为了获得最佳的用户体验，在不同的断点下，material design 的接口需要适应它们的布局。Material-UI 使用了原先 [specification](#) 的 **简化** 实现。

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.

默认的断点

每个断点（一个键）匹配了一个**固定**的屏幕宽度（一个值）：

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

这些值可以是 [定制](#) 的。

CSS 媒体查询 (Media queries)

CSS media queries 是一种做出响应式的用户界面的特有方法。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\(start, end\)](#)

在下面的演示中，我们根据屏幕宽度来更改背景颜色 (红色、蓝色和绿色)。

```
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"}}
```

JavaScript 媒体查询

有的时候，仅使用 CSS 是远远不够的。您可能会有基于 JavaScript 中的断点值来更改 React 渲染树的需求。

useMediaQuery hook

您可以在 [userMediaQuery](#) 页面上了解更多信息。

自定义断点

您可以选择在 theme 中的 `theme.breakpoints` 部分定义项目的断点。

- [theme.breakpoints.values](#)：默认为 [以上值](#)。键值 (key) 为屏幕的名字 (screen names)，而值 (values) 是该断点应开始的最小宽度。
- `theme.breakpoints.unit`：默认为 `px`。这个用于断点值的单位。
- `theme.breakpoints.step`：默认为 `5 (0.05px)`。The increment divided by 100 used to implement exclusive breakpoints. For example, `{ step: 5 }` means that `down(500)` will result in `'(max-width: 499.95px)'`.

如果您需要更改断点的默认值，则需要提供所有的断点值：

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

您可以随意设置任意数量的断点，并且也可以在项目中以您喜欢的任何方式为断点命名。

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

如果您使用的是 TypeScript，您还需要使用 [module augmentation](#) 来让主题接受上述值。

```
declare module '@material-ui/core/styles' {
  interface BreakpointOverrides {
    xs: false; // 移除 `xs` 断点
    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

Arguments

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

Returns

`media query` : 一个媒体查询字符串，适用于大多数的样式解决方案，它匹配的屏幕宽度大于（包含）断点键给出的屏幕尺寸。

Examples

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

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

Arguments

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

Returns

`media query` : 一个媒体查询字符串，适用于大多数的样式解决方案，它匹配的屏幕宽度小于（不包含）断点键给出的屏幕尺寸。

Examples

```
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` : 一个媒体查询字符串，适用于大多数的样式解决方案，它会匹配屏幕宽度，并包括断点键给出的屏幕尺寸。

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',  
    },  
  },  
});
```

默认值

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`).