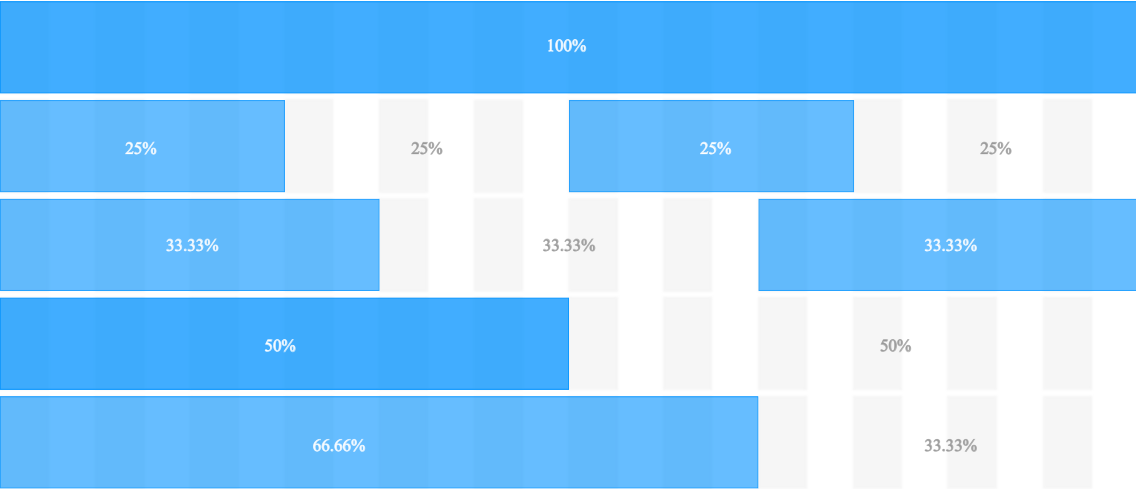


24 栅格系统。

设计理念



在多数业务情况下，Ant Design 需要在设计区域内解决大量信息收纳的问题，因此在 12 栅格系统的基础上，我们将整个设计建议区域按照 24 等分的原则进行划分。

划分之后的信息区块我们称之为『盒子』。建议横向排列的盒子数量最多四个，最少一个。『盒子』在整个屏幕上占比见上图。设计部分基于盒子的单位定制盒子内部的排版规则，以保证视觉层面的舒适感。

概述

布局的栅格化系统，我们是基于行（row）和列（col）来定义信息区块的外部框架，以保证页面的每个区域能够稳健地排布起来。下面简单介绍一下它的工作原理：

- 通过 `row` 在水平方向建立一组 `column`（简写 `col`）。
- 你的内容应当放置于 `col` 内，并且，只有 `col` 可以作为 `row` 的直接元素。
- 栅格系统中的列是指 1 到 24 的值来表示其跨越的范围。例如，三个等宽的列可以使用 `<Col span={8} />` 来创建。
- 如果一个 `row` 中的 `col` 总和超过 24，那么多余的 `col` 会作为一个整体另起一行排列。

我们的栅格化系统基于 Flex 布局，允许子元素在父节点内的水平对齐方式 - 居左、居中、居右、等宽排列、分散排列。子元素与子元素之间，支持顶部对齐、垂直居中对齐、底部对齐的方式。同时，支持使用 `order` 来定义元素的排列顺序。

布局是基于 24 栅格来定义每一个『盒子』的宽度，但不拘泥于栅格。

API

Ant Design 的布局组件若不能满足你的需求，你也可以直接使用社区的优秀布局组件：

- [react-flexbox-grid](#)
- [react-blocks](#)

Row

成员	说明	类型	默认	版本
----	----	----	----	----

			值	
align	垂直对齐方式	top middle bottom	top	
gutter	栅格间隔，可以写成像素值或支持响应式的对象写法来设置水平间隔 { xs: 8, sm: 16, md: 24}。或者使用数组形式同时设置 [水平间距, 垂直间距]	number object array	0	
justify	水平排列方式	start end center space-around space-between	start	
wrap	是否自动换行	boolean	true	4.8.0

Col

成员	说明	类型	默认值	版本
flex	flex 布局属性	string number	-	
offset	栅格左侧的间隔格数，间隔内不可以有栅格	number	0	
order	栅格顺序	number	0	
pull	栅格向左移动格数	number	0	
push	栅格向右移动格数	number	0	
span	栅格占位格数，为 0 时相当于 <code>display: none</code>	number	-	
xs	屏幕 < 576px 响应式栅格，可为栅格数或一个包含其他属性的对象	number object	-	
sm	屏幕 ≥ 576px 响应式栅格，可为栅格数或一个包含其他属性的对象	number object	-	
md	屏幕 ≥ 768px 响应式栅格，可为栅格数或一个包含其他属性的对象	number object	-	
lg	屏幕 ≥ 992px 响应式栅格，可为栅格数或一个包含其他属性的对象	number object	-	
xl	屏幕 ≥ 1200px 响应式栅格，可为栅格数或一个包含其他属性的对象	number object	-	
xxl	屏幕 ≥ 1600px 响应式栅格，可为栅格数或一个包含其他属性的对象	number object	-	

响应式栅格的断点扩展自 [Bootstrap 4 的规则](#) (不包含链接里 `occasionally` 的部分)。