

TaskBuilder 组件定位详解

如何设计出界面美观、布局合理、易于操作的前端页面，是每个 TaskBuilder 开发者都需要考虑的问题，很多刚开始使用 TaskBuilder 的同学，在进行组件定位和页面布局时遇到很多问题不知道如何解决，下面就给大家分享一下 TaskBuilder 的组件定位和页面布局有关的一些基础概念、实现方法和典型应用场景。

注：以下内容和截图都是基于 TaskBuilder v1.3.24 版编写的，如果与您正在使用的版本功能不一致，请升级到最新版本。

网页元素常见定位方式

本章所说的“**定位方式**”是指确定 HTML 元素（或 TFP 组件）在页面里的显示位置的相关规则，也有人把这些规则称为“**布局**”，为了与介绍页面如何合理划分内容显示区域的“**页面布局**”的概念进行区分，我们将这里介绍的内容统一称为“**定位方式**”。

相对于专业编程书籍，下述的介绍会比较简洁，不会把这些概念的技术细节展开进行非常详细的介绍，因为目的只是希望大家大致了解一下一些网页设计的基本知识，以便大家将来在使用 TaskBuilder 设计前端页面时，知道问题出在哪里，应该怎么解决。既然让大家使用 TaskBuilder 进行低代码开发，就不想让大家像专业程序员一样，再学习太多的专业知识，我们尽量把大家需要的常用功能都封装成 TaskBuilder 里可以可视化设置的组件、参数或选项，使用鼠标拖拽点选和输入必要的参数就能实现想要的效果。在介绍完基本概念之后，会详细讲解如何用 TaskBuilder 结合这些概念进行页面设计。

网页元素常见的定位方式有以下几种：

1. 流式定位

这是网页元素默认的定位方式，网页元素按照其 HTML 标签的先后顺序，在网页内依次显示，就像液体一样“流动”，所以称为“流式定位”，这种方式将所有网页元素的默认显示方式分为以下几种类型：

块级元素 (display: block)：可以设置宽度，默认宽度为 100%，但不管多宽都会独占一行，多个块级元素按从上到下的顺序纵向显示，默认显示为块级元素的包括：段落(p)、层(div)、表单(form)等。

内联元素 (display: inline)：不会独占一行，可以多个元素共享一行，按从左至右的顺序横向显示，默认显示为内联元素的包括：文本标签(label)、链接(a)等。

行内块元素 (display: inline-block)：这种定位方式兼具块级元素和内联元素的优势，既可以像块级元素一样设置宽度和高度，又可以一行放置多个，默认显示为内联元素的包括：图片(img)、输入项(input)、多行文本(textarea)等。

可以通过设置 HTML 元素的 CSS 样式的 display 属性来改变元素默认的流式定位方式，例如将层(div)元素设置为行内块元素后，就可以跟文本标签(label)、图片(img)等元素放在同一行显示，不再独占一行。

流式定位存在一些问题，主要包括：

- 块级元素很难被有效利用起来，因为其独占一行的特性，不能一行显示多个。
- 内联元素无法设置宽度、高度和垂直方向边距，不好精确控制元素大小和显示位置。
- 内联元素之间水平方向会因为代码中的空格、换行等产生间隙，垂直方向存在默认的底部间隙，这给水平布局带来额外的麻烦。

2. 浮动定位

浮动定位会让元素脱离原来的标准文档流浮动起来，直至它的外边缘遇到父元素的

内边缘或者另一个兄弟浮动元素的外边缘。

任何 HTML 元素都可以设置为浮动,可以通过设置 HTML 元素的 CSS 样式的 float 属性来设置具体的浮动方式,可以选择向左浮动 (float: left) 或向右浮动 (float: right)。

相对于网页默认的流式定位,浮动定位有以下优势:

- 多个块级元素可以共享一行。
- 内联元素可以设置宽度和高度。
- 元素之间的空格、换行不影响元素的显示位置。

浮动定位最初被设计出来是为了实现文字环绕图片显示的效果,但由于其上述优势,后来大家也用浮动定位来设置更多网页元素的位置,代替默认的流式定位,也是 TaskBuilder 默认的组件定位方式。

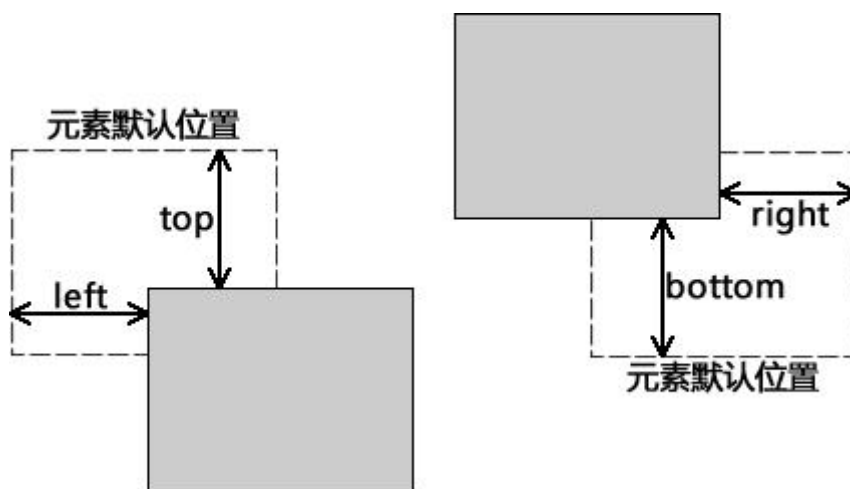
不过,浮动定位也有一些问题,不是所有的场合都适合,大家根据需要选择,这些问题包括:

- 居中对齐问题:浮动定位不太好实现垂直和水平居中对齐。
- 父元素的高度塌陷问题:如果元素不浮动,会自动撑开父元素的高度,浮动后,由于浮动元素已经脱离原来的标准流,如果父元素没有设置高度,且父元素中没有其他非浮动元素时,会造成父元素的高度塌陷(高度变成了 0)。

3. 坐标定位

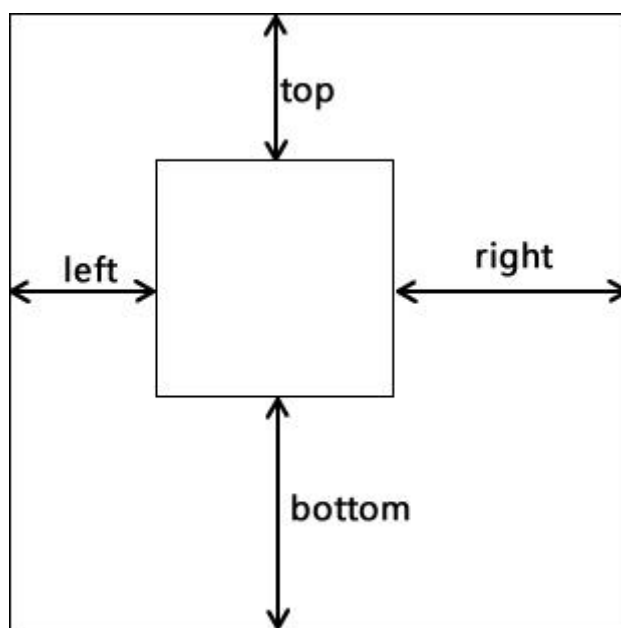
坐标定位是指为 HTML 元素设置上下左右的偏移量来确定其具体的显示位置或大小,通过设置元素 CSS 的 position 属性,可以选择坐标定位的具体方式,包括:

相对定位 (position: relative) : 即相对于元素在流式定位时的正常位置进行定位,您可以通过 top、right、bottom、left 这 4 个 CSS 属性来设置元素相对于正常位置的偏移量,在此过程中不会对其它元素造成影响。



相对定位示意图

绝对定位 (position: absolute)：即相对于第一个非静态定位的父级元素进行定位，可以通过 top、right、bottom、left 这 4 个属性来设置元素相对于父级元素位置的偏移量。如果没有满足条件的父级元素，则会相对于浏览器窗口来进行定位。元素使用绝对定位后，其他标准流元素会填补它的位置。绝对定位可以通过设置元素 CSS 样式的 left 和 top 属性来精确控制元素的显示位置，再设置 right 或 bottom 这两个 CSS 样式，实现元素的宽度或高度根据父元素的宽度和高度动态设置和变化。



非静态定位的父级元素

绝对定位示意图

固定定位 (position: fixed) : 即相对于浏览器的窗口进行定位, 可以使用 top、right、bottom、left 这 4 个 CSS 属性来定义元素相对于浏览器窗口的位置。使用固定定位的元素无论如何滚动浏览器窗口, 元素的位置都是固定不变的。

粘性定位 (position: sticky) : 它是相对定位和固定定位的结合体, 能够实现类似吸附的效果, 当滚动页面时它的效果与相对定位相同, 当元素要滚动到屏幕之外时则会自动变成固定定位的效果。用粘性定位很容易实现元素置顶显示的效果。

4. 表格定位

顾名思义, 表格定位就是用表格来实现网页内容的定位, 先绘制一个多行多列的表格, 并设定表格各列的宽度和各行的高度, 然后在表格的单元格里插入相关网页元素, 最终实现这些元素的位置定位。

这种定位方式比较适合可以将页面内容清晰明确地划分为多行多列的场景, 例如各种业务单据和数据报表等。

有两种方式可以实现表格布局: HTML Table (<table>标签) 和 CSS Table (display:table 等相关属性)。HTML Table 是早期网页设计采用的表格定位方式, 由于其渲染性能较低, 且有些界面效果较难实现, 用的人已经比较少了, 现在推荐使用 CSS Table, 能实现一些 HTML Table 做不到的效果。

目前, TaskBuilder 只是在报表设计器里集成了一个第三方的表格插件, 在前端 tfp 页面设计器里, 还没有提供表格定位相应的容器组件, 未来会提供。(现有的可编辑表格和数据表格不是用来为其他组件提供定位的, 所以不能算作是表格定位的容器组件)

5. 弹性定位

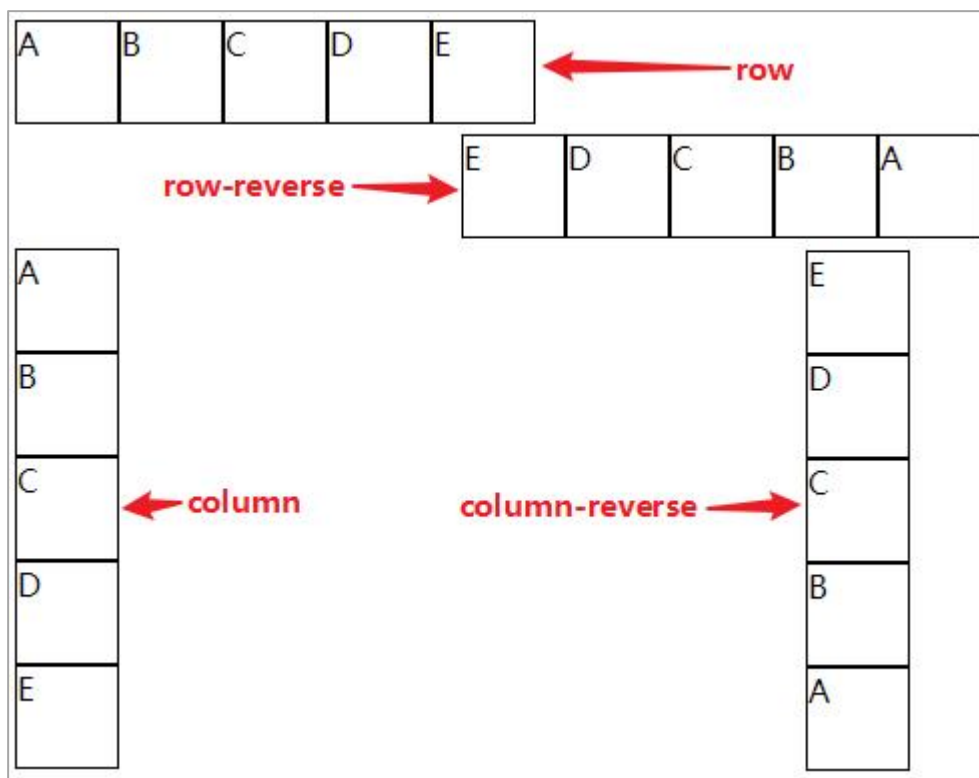
弹性定位又叫弹性布局, 这种定位方式最大的特点就是可以让子元素的大小可以根据父元素的大小自动扩张或收缩, 从而可以做到自动适应不同终端的屏幕尺寸, 或者在

用户改变浏览器窗口大小时自动伸缩。

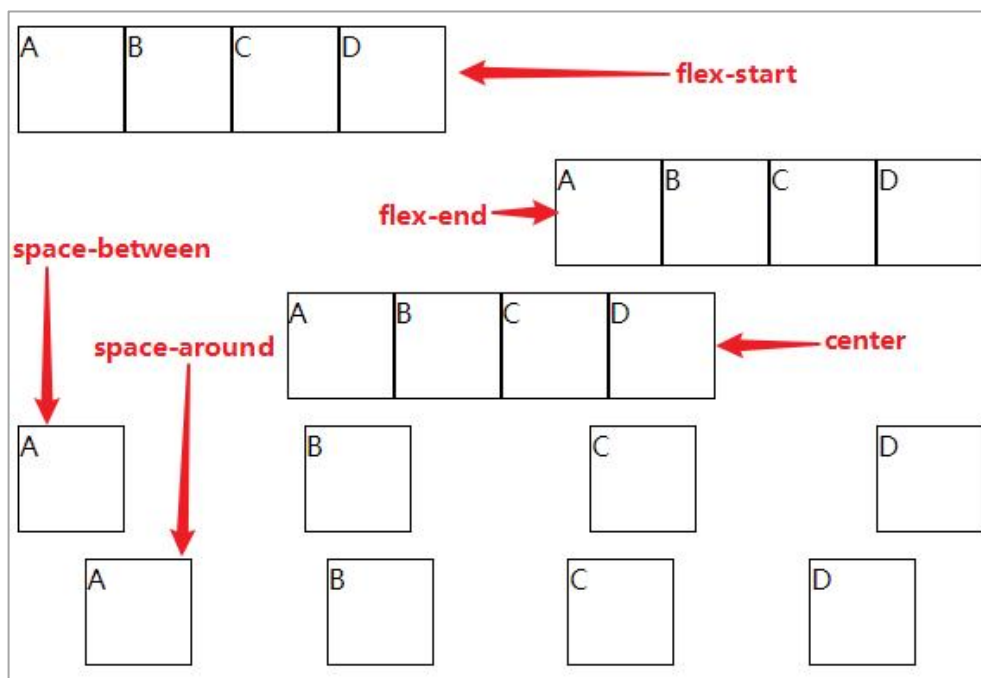
这种定位方式需要先将一个父元素的 CSS 属性 display 设置为 flex, 然后该元素里的子元素即可实现弹性定位。

采用弹性定位的容器元素可以设置以下 CSS 样式：

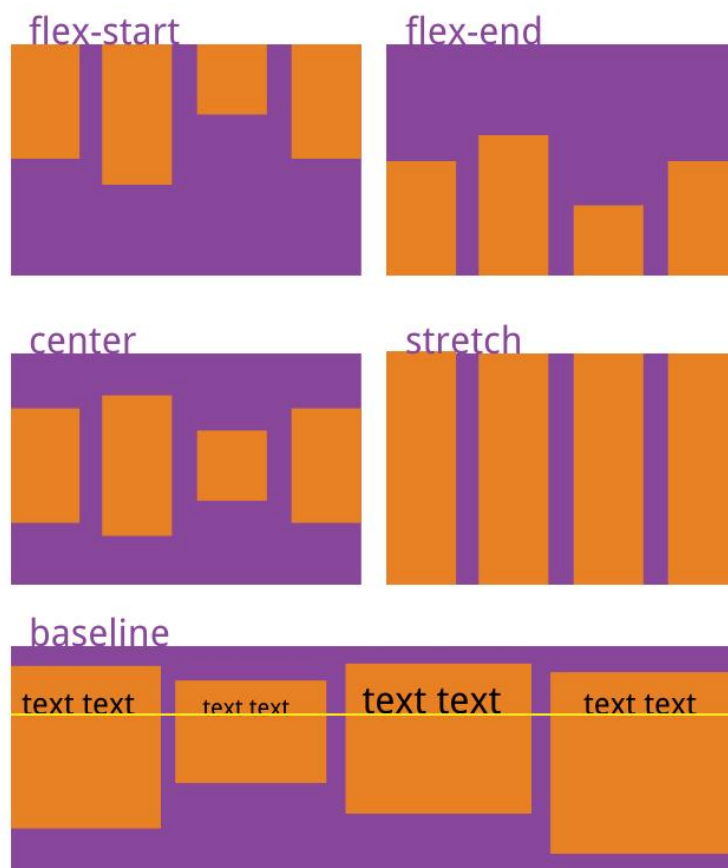
元素排列方向 (flex-direction)： 用来设置子元素的排列方向，有四种方式可以选择：从左到右显示 (row)、从右到左显示 (row-reverse)、从上到下显示 (column)、从下到上显示 (column-reverse)。



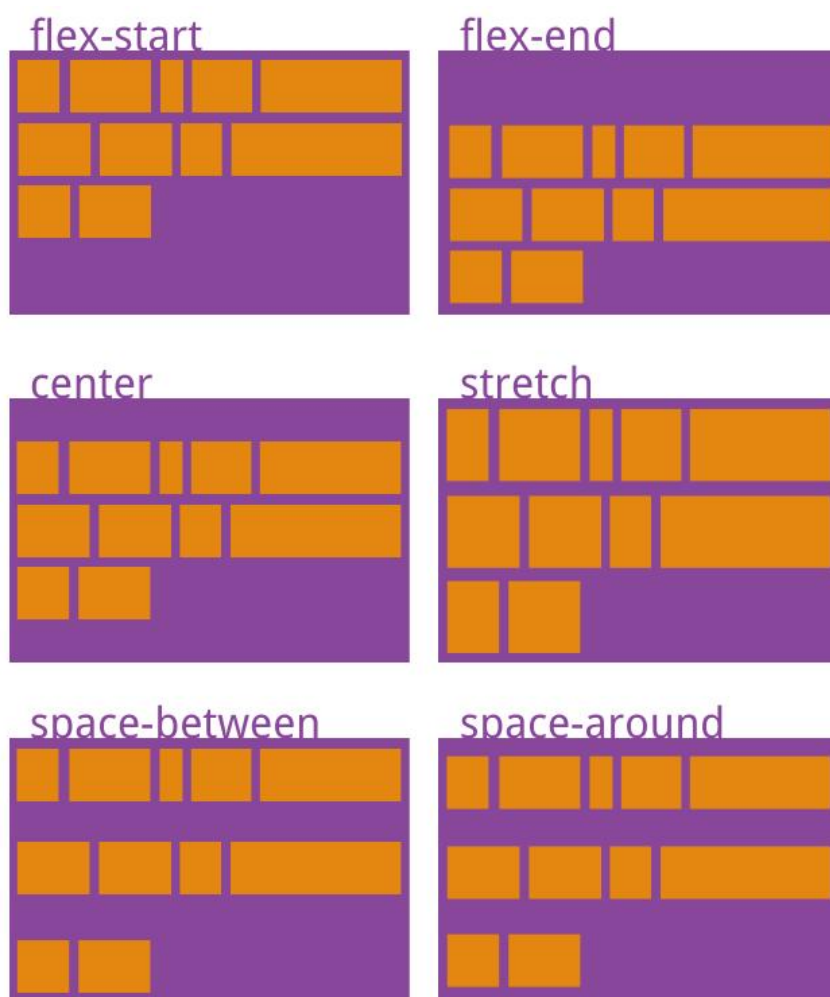
元素水平对齐方式 (justify-content)： 用来设置子元素在水平方向的对齐方式，可以选左对齐 (flex-start)、右对齐 (flex-end)、居中对齐 (center)、两端对齐 (space-between) 和等距对齐 (space-around)。



元素垂直对齐方式 (align-items)：用来设置子元素在垂直方向的对齐方式，可以选择顶部对齐 (flex-start)、底部对齐 (flex-end)、居中对齐 (center)、基线对齐 (baseline)、自动伸展 (stretch)。



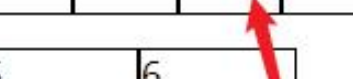
内容整体对齐方式 (align-content)： 当有多行子元素时，可以用该属性设置全部子元素整体的垂直对齐方式，包括自动拉伸 (stretch)、在容器的顶部排列 (flex-start)、在容器的底部排列 (flex-end)、在容器内居中排列 (center)、均匀分布，上下靠边 (space-between)、均匀分布，每行等距 (space-around)。



元素是否自动换行 (flex-wrap)： 用来设置子元素是否自动换行。

1	2	3	4	5	6	7	8	9	10
---	---	---	---	---	---	---	---	---	----

1	2	3	4	5	6
7	8	9	10		



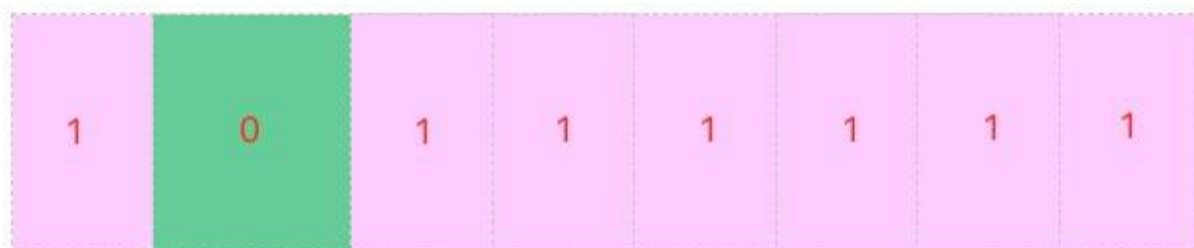
弹性定位容器内的子元素可以设置以下 CSS 样式：

扩展量 (flex-grow)： 必须参数，用来设置当前元素相对于其他元素的增长量，默认值为 0，即如果存在剩余空间，也不自动放大。

如果所有元素的扩展量都设置为 1，将等分剩余空间。如果一个元素的扩展量设置为 2，其他元素都为 1，则前者占据的剩余空间将是其他项的 2 倍，如下图所示：



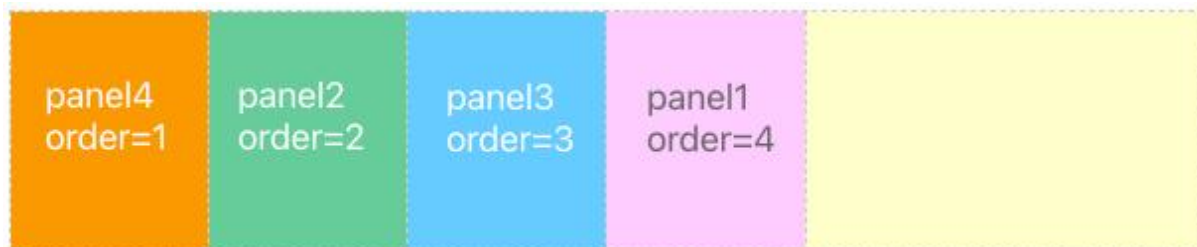
收缩量 (flex-shrink)： 必须参数，用来设置当弹性容器空间不足时，该元素相对于其他元素的收缩量，默认值为 1。如果所有元素的 flex-shrink 属性都为 1，当空间不足时，都将等比例缩小。如果一个元素的 flex-shrink 属性为 0，其他元素都为 1，则空间不足时，前者不缩小。



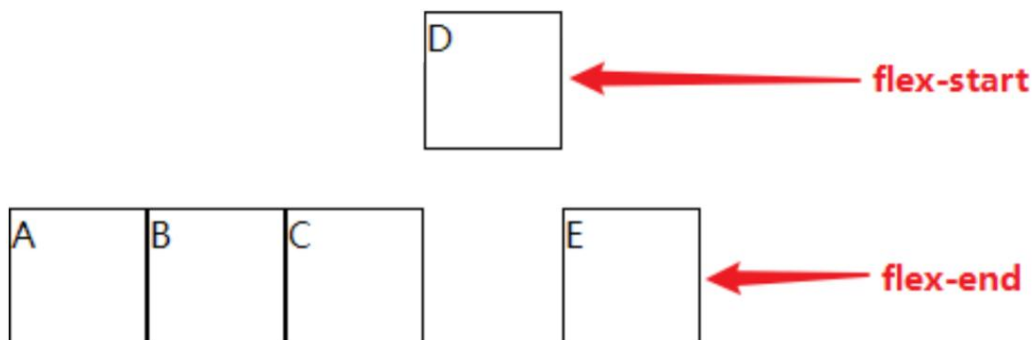
基准长度 (flex-basis)： 必须参数，元素的基准长度定义了父容器在分配多余空间之前，当前元素占据的主轴空间 (main size，如果是横向排列，则指父容器在水平方向的空间，如果是纵向排列，则指父容器在垂直方向的空间)。浏览器根据这个属性，计算主轴是否有多余空间。合法值为 auto (默认值，表示根据其他情况自动伸缩)，或者以具体的值加 "%"、"px"等单位的形式，表示该元素将占据固定空间。

显示顺序 (order)： 属性用来设置元素在容器中出现的顺序，您可以通过具体的

数值来定义元素在容器中的位置，默认值为 0。下图所示的四个普通的面板组件，在弹性面板里从左至右的正常插入顺序是 panel1、panel2、panel3、panel4，但通过设置这四个面板的显示顺序，将 panel1 和 panel4 的显示位置进行了调换。



对齐方式 (align-self)：该属性允许您为某个特定的元素设置不同于其它元素的对齐方式，该属性可以覆盖父容器组件（弹性面板）里的垂直对齐属性的值。该属性的可选顶部对齐 (flex-start)、底部对齐 (flex-end)、居中对齐 (center)、基线对齐 (baseline)、自动伸展 (stretch)。



由于弹性定位的自动伸缩特性，且易于实现子元素的水平和垂直对齐，目前很多主流的前端 UI 框架都大量地采用了该定位方式。

在 TaskBuilder 中如何设置组件的定位方式

以上介绍的这些定位方式，除了表格定位以外，其他方式 TaskBuilder 都可以支持。下面给大家介绍一下在 TaskBuilder 中如何设置组件的定位方式。

1. 设置新建组件默认的定位方式

在使用 TaskBuilder 创建前端 TFP 页面时，有一个“定位方式”的选项，可以用来设置在该页面内新建组件时默认的定位方式，为了简化操作，目前仅支持两种默认定位方式：**浮动定位** 和 **绝对定位**，默认是浮动定位，如下图所示：



上图是创建空白 TFP 页面的示意图，创建数据查询、添加、修改和详情 TFP 页面时，也都有这个定位方式的选项。

如果在创建 TFP 页面时，定位方式选择的是浮动定位，则后续从组件库里拖拽组件到该页面上时，新建的组件会按下面的规则进行显示：

- 容器组件按照流式定位里的块级元素进行显示，多个容器组件默认从上到下显示，每个独占一行，宽度默认为 100%
- 非容器组件按照浮动定位从左至右浮动显示，一行可以显示多个

如果在创建 TFP 页面时，定位方式选择的是绝对定位，则后续从组件库里拖拽组件到该页面上时，不管是容器组件还是非容器组件，新建的组件默认都会采用绝对定位，即鼠标在哪里松开，新建的组件就会插入到该位置的容器组件中，并显示在鼠标松开的位置，如果还需调整位置，可以继续用鼠标拖拽调整。

绝对定位和传统的桌面 GUI 应用开发采取的定位方式非常相似，例如：VB、Delphi、

PowerBuilder、VS WinForm 等，如果原来用过这些开发工具，对绝对定位就会很熟悉，很容易上手。

注意：新建组件时，有一种特殊情况不会按照上述规则处理，即从组件库中拖拽组件到 **弹性面板** 组件中时，新建的组件不会采用当前页面里设置的默认定位方式，而是会按弹性定位进行处理，后面会详细讲解弹性面板组件的使用。

2. 修改已有组件的定位方式

上面介绍的默认定位方式只是为了确定在新建组件时采用哪种定位方式，并不代表创建完的组件只能是这种定位方式。如果想修改页面上已有组件的定位方式，可以在 TaskBuilder 前端页面设计器中先选中组件，然后在顶部的外观样式设置栏的“显示和定位”中，点击定位方式选项，在弹出的列表中选择相应的定位方式即可，如下图所示：



(1) 流式定位设置

如果将组件设置为流式定位，则可以在右侧选择具体的显示方式，可以选系统默认（即该组件对应的 HTML 元素在浏览器里默认的显示方式）、显示为内联元素、显示为块级元素和显示为行内块元素，如下图所示：



(2) 浮动定位设置

如果选择的是浮动定位，则可以在右侧的浮动方式中选择是左浮动还是右浮动，如果选的是左浮动，则组件靠左对齐，这也是浮动定位时，非容器组件默认的对齐方式，如下图所示：



如果选的是右浮动，则组件会靠右浮动，如下图所示：

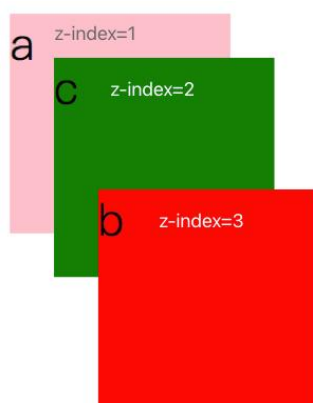


(3) 坐标定位设置

如果选择的是相对定位、绝对定位、固定定位和粘性定位这几种定位方式，则可以设置组件的堆叠顺序和上下左右的具体偏移量，如下图所示：



其中，堆叠顺序（CSS 样式名：z-index）用来确定组件在当前层叠上下文中的层叠级别，参数类型为整数，数值大的在上面，数值小的在下面，例如下图所示的 a、b、c 三个面板组件，就是按照堆叠顺序从小到大、从底到上的顺序显示的。



如果上下偏移量只设置其中一个，另外一个为空，则组件的高度由正常的 CSS 样式的 height 确定。如果左右偏移量只设置其中一个，另外一个为空，则组件的宽度由正常的 CSS 样式的 width 确定。组件的宽度和高度可以在 TaskBuilder 前端页面设计器顶部样式设置栏的“显示和定位”里查看和设置，如下图所示：



如果上下偏移量都设置，并且未设置高度，则组件高度会根据父组件的高度和自身上下偏移量动态确定，计算公式如下：

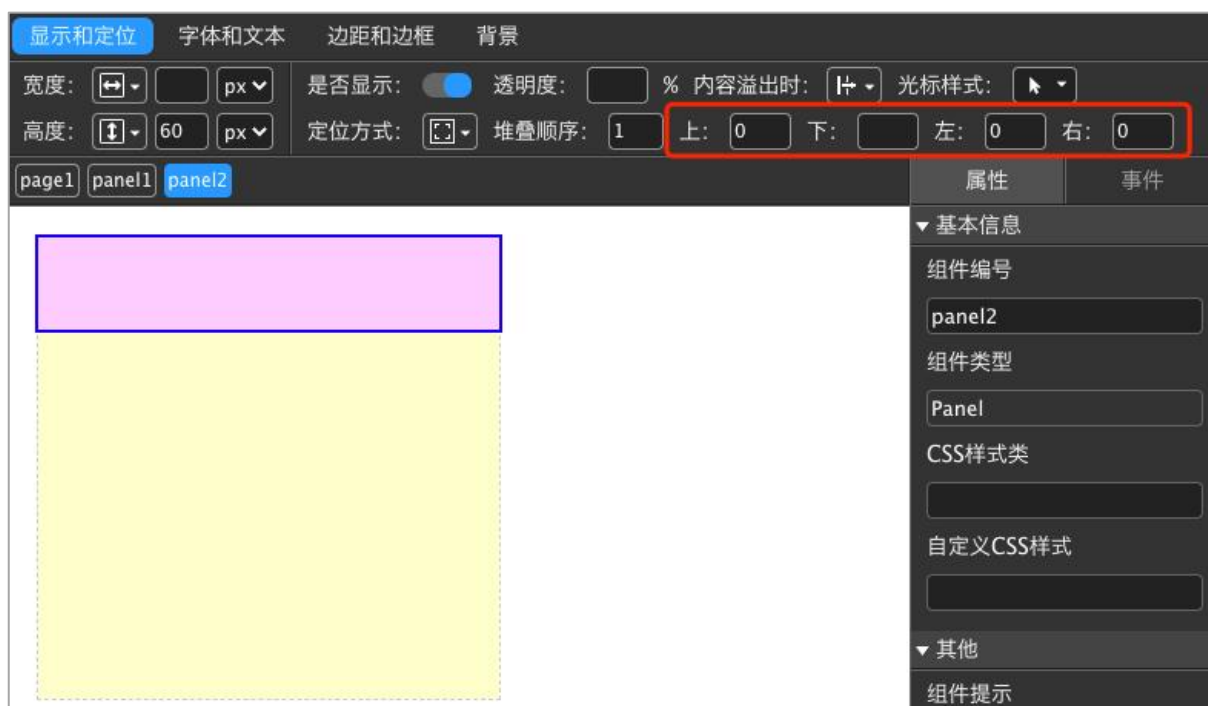
$$\text{组件高度} = \text{父组件内容区高度} - \text{上偏移量} - \text{下偏移量}$$

如果左右偏移量都设置，并且未设置宽度，则组件宽度会根据父组件的宽度和自身左右偏移量动态确定，计算公式如下：

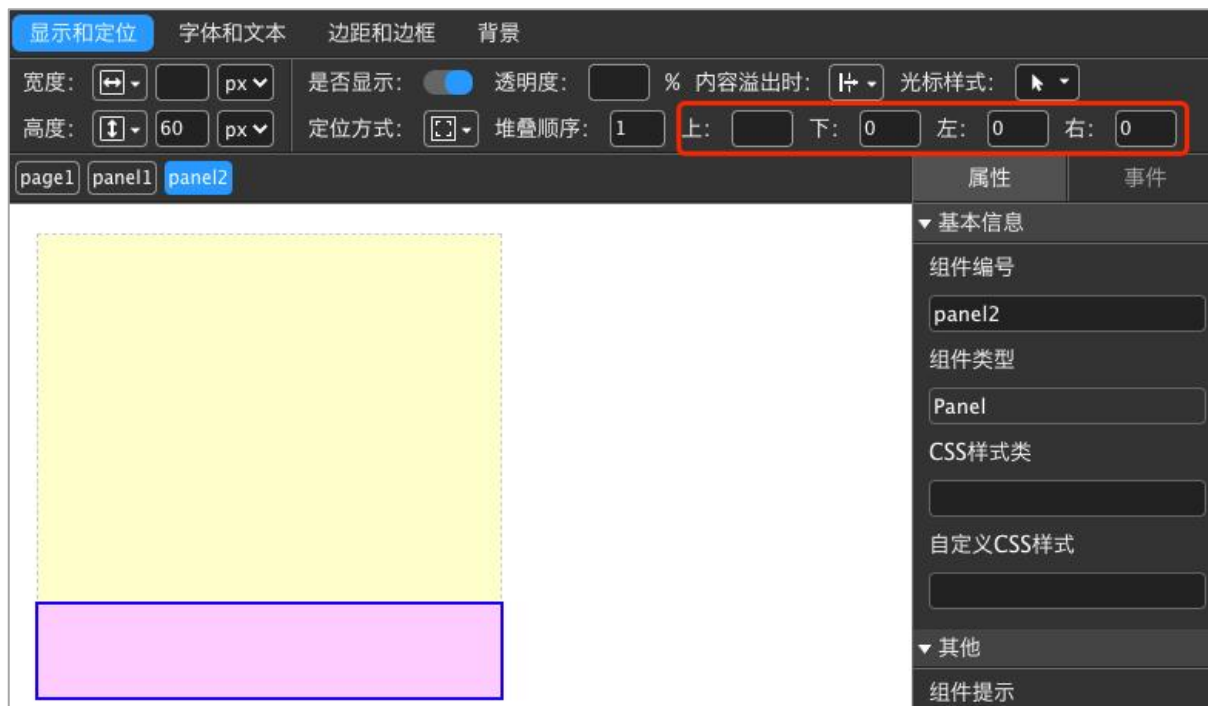
$$\text{组件宽度} = \text{父组件内容区宽度} - \text{左偏移量} - \text{右偏移量}$$

采用 **绝对定位** 时，如果组件设置了上下左右中的三个偏移量，另外一个为空，则可以实现固定停靠在父组件内容区某一侧的效果，即使父组件的大小发生了变化，停靠位置不会发生变化。

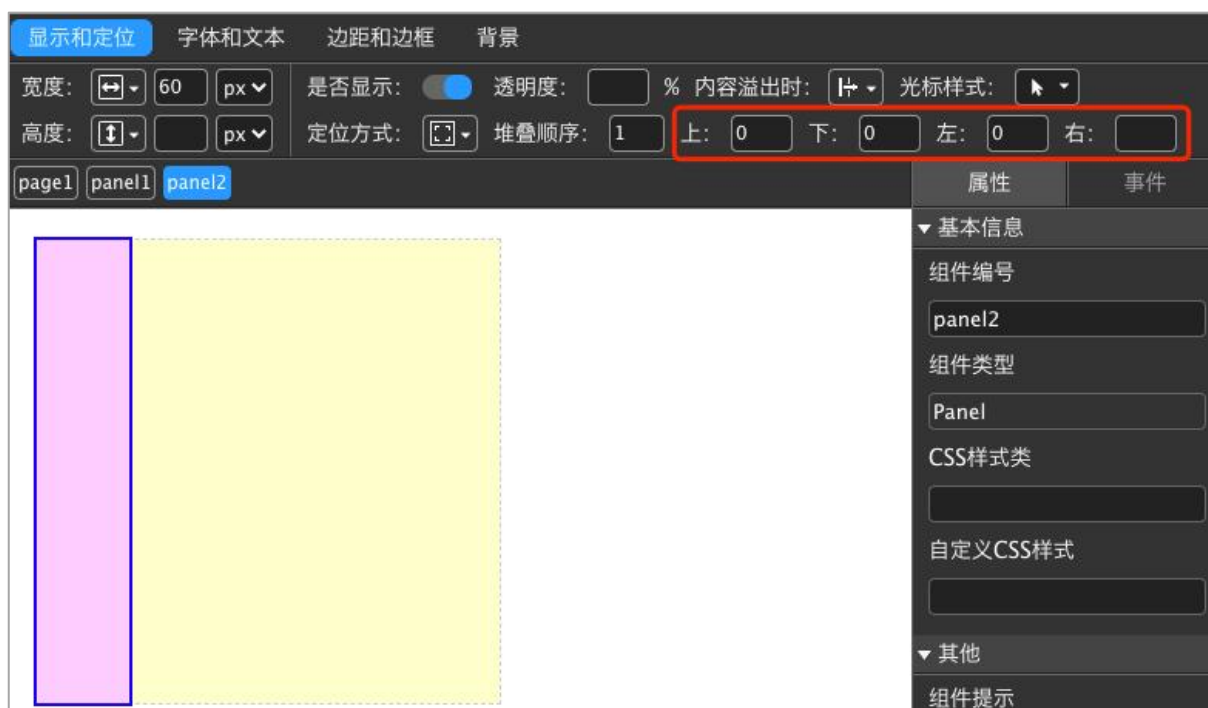
例如，设置了左右和上侧偏移量，下侧未设置，则组件向上停靠，如下图：



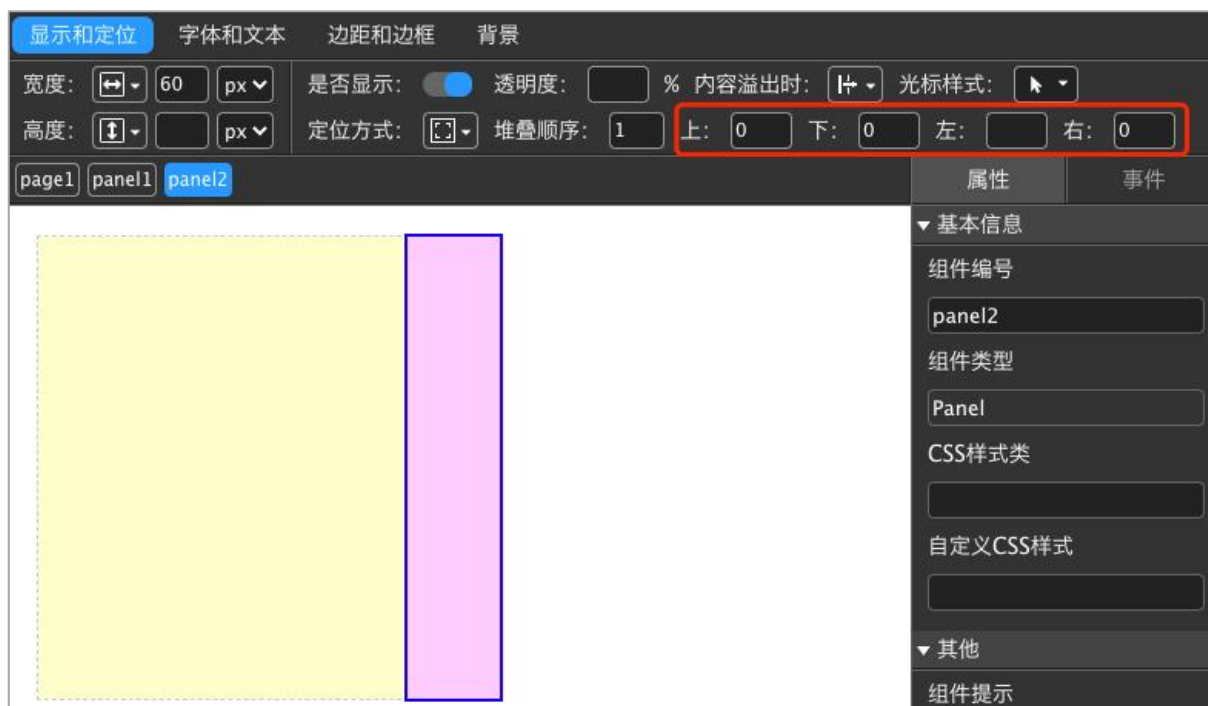
如果设置了左右和下侧偏移量，上侧未设置，则组件向下停靠，如下图：



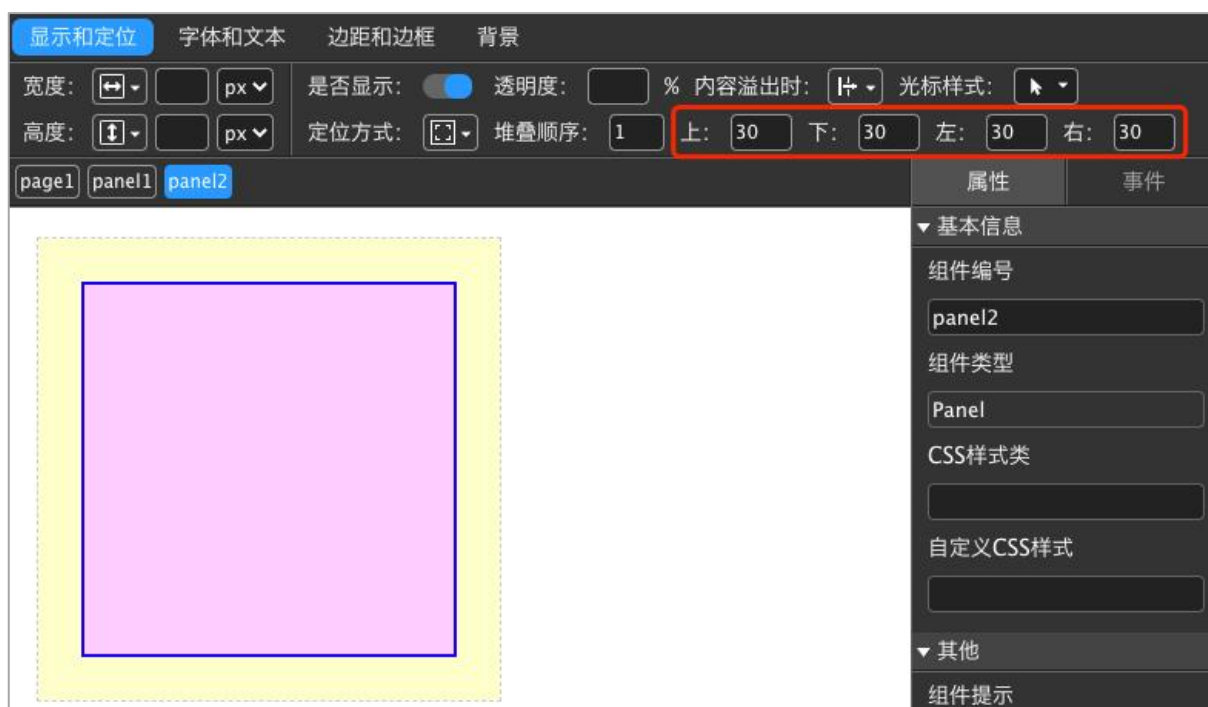
如果设置了上下和左侧偏移量，右侧未设置，则组件向左停靠，如下图：



如果设置了上下和右侧偏移量，左侧未设置，则组件向右停靠，如下图：



如果上下左右四个偏移量都设置，且不设置组件的宽度和高度，则该组件会铺满整个父组件的内容区，并且与父组件上下左右边框的边距就是这四个偏移量的值，当父组件的大小发生变化时，该组件的大小随之变化，保持与父组件的边距不变，如下图所示：



(4)弹性定位设置

弹性定位需要先将一个父元素设置为弹性盒子，子元素（项目）才能进行弹性定位，不是说将某个独立的网页元素设置一下就能实现弹性定位，所以在 TaskBuilder 里，没有将弹性定位设置为默认的定位方式，而是可以通过 **弹性面板** 和 **弹性栅格** 这两个组件来实现弹性定位，下面分别介绍一下这两个组件。

1) 弹性面板组件

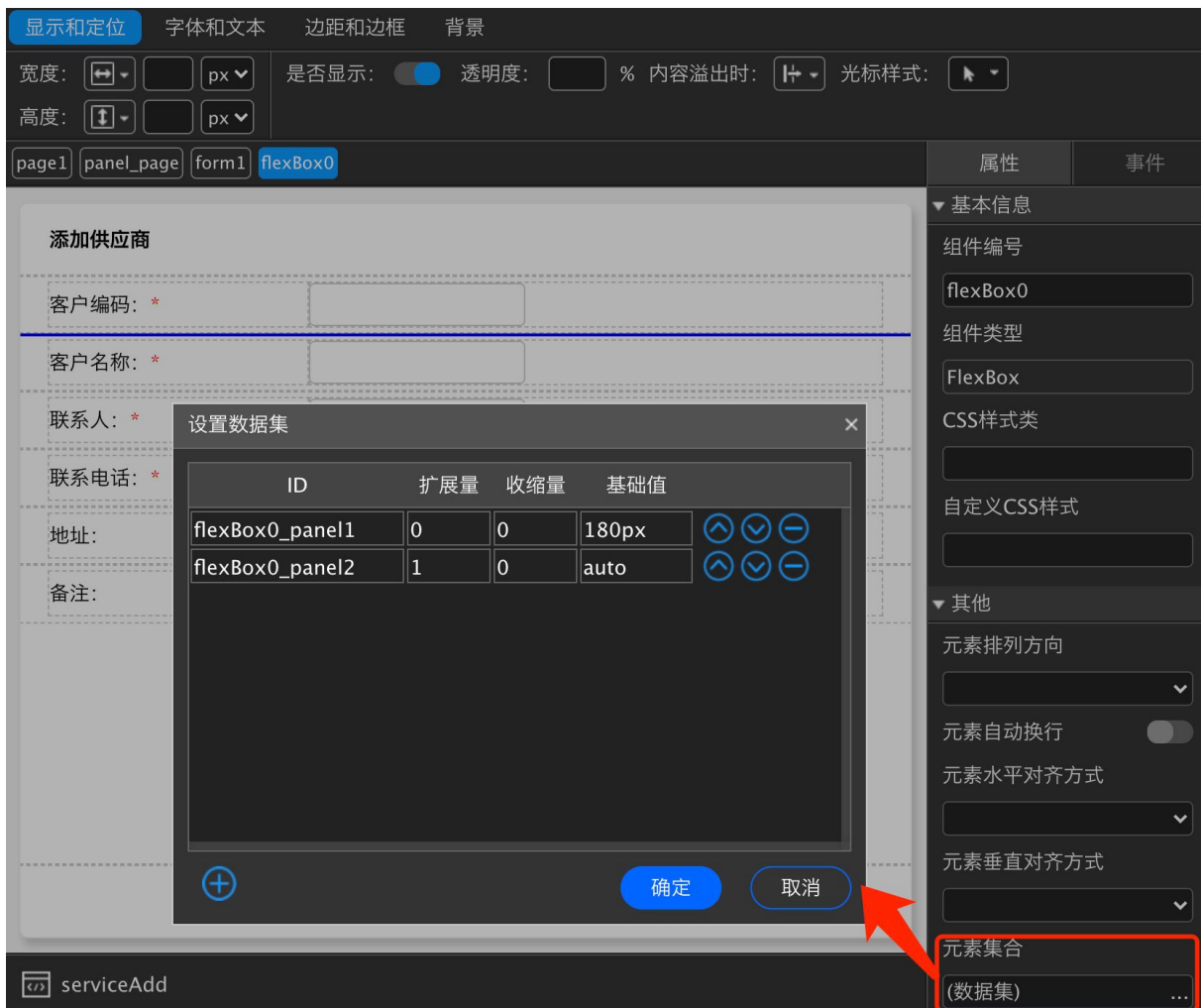
弹性面板组件可以像普通面板组件一样，作为其他组件的容器，区别就是，在面板组件内创建新的子组件时，默认的定位方式按照当前页面选择的定位方式进行定位，而在弹性面板组件里创建新组件时都会按照弹性定位进行定位，默认子组件（元素）从左至右排列、水平左对齐、垂直居中对齐、不自动换行，可以在属性设置中修改这些设置。

新建的弹性面板组件，默认里面是没有任何子元素的，需要开发者自己从组件库拖拽创建所需的子组件，子组件的弹性定位有关的样式可以在顶部样式设置栏的“显示和定位”里设置，如下图所示：



2) 弹性栅格组件

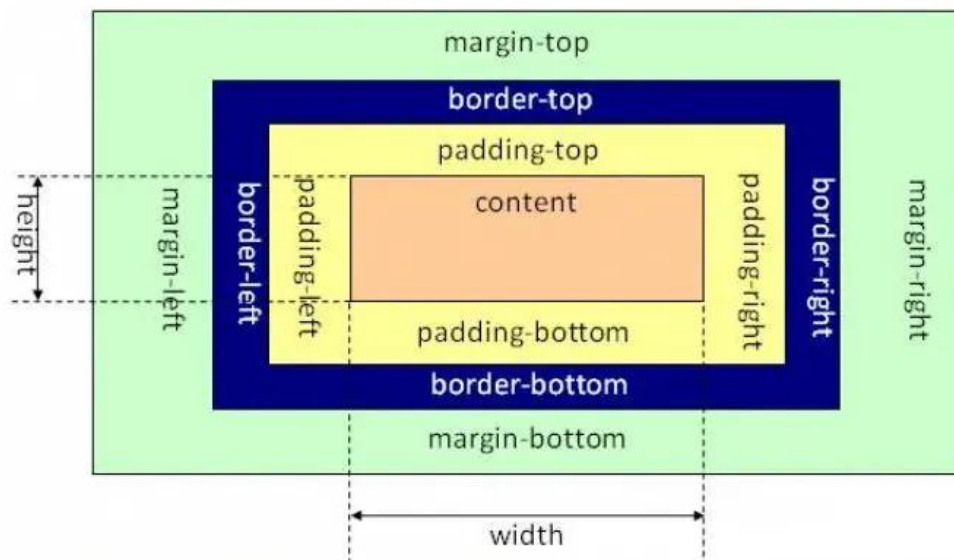
弹性栅格组件最外层跟弹性面板组件一样，都是一个采用弹性布局的容器元素，新建弹性栅格组件时，默认会创建两个子元素，并从左至右横向显示，点击属性设置里的“元素集合”，可以打开元素设置对话框，在此批量设置具体要显示多少个子元素，及各个子元素的扩展量、收缩量和基准长度等。弹性栅格里的每个子元素都是一个面板组件，里面又可以拖拽放入其他组件，这样就就可以快速创建多列或多行的栅格容器。



相对弹性面板组件，弹性栅格组件多了一层，它不是直接把其他组件拖拽到其内部，而是先设置有多少个栅格，然后再把其他组件拖入到这些栅格里，拖入到这些栅格里的组件具体采用什么定位方式，取决于当前页面设置的默认定位方式。

网页元素的边距和边框

由于网页元素的边距和边框会影响到元素的显示位置，所以在此也统一介绍一下。网页元素在浏览器中实际占用的空间大小由外边距（margin）、边框（border）、内边距（padding）和内容区这四个部分构成，如下图所示：



在 TaskBuilder 前端页面设计器顶部的样式设置栏里的“边距和边框”中可以设置选中组件的外边距、内边距和边框，如下图所示：



下面，分别介绍一下这几个参数的具体含义：

外边距 (margin)： 外边距用来设置该元素与父元素边框及相邻兄弟元素之间的间距，外边距没有背景颜色，是完全透明的，可以使用负值。内联元素的外边距在水平方向上设置有效，垂直方向上无效。

内边距 (padding)： 内边距用来设置该元素的子元素与其边框之间的间距，与外边距不同，内边距会受到元素背景属性的影响，不允许指定负内边距值。内联元素水平方向上的内边距(`padding-left`, `padding-right`)可以正常设置，垂直方向上的内边距不会影响它的内容区高度，但是会影响他的背景高度。

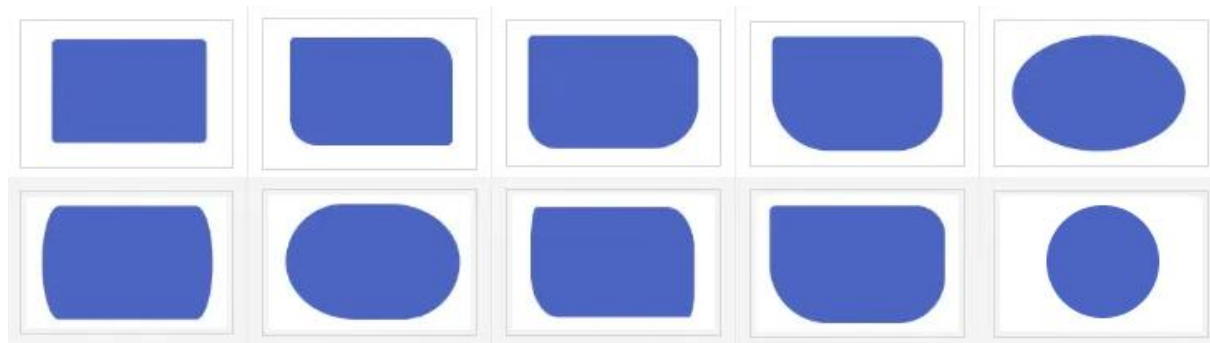
边框 (border)： 边框是围绕元素内容和内边距的一条或多条线，对于这些线条，您可以自定义它们的宽度、样式以及颜色。

在 TaskBuilder 前端页面设计器顶部的样式设置栏里，点击“边距和边框”，再点击“边框”右侧的设置按钮，即可打开如下图所示的边框设置窗口：



在该窗口内，可以设置全部边框的宽度、样式和颜色，也可以分别设置上下左右四个边里的一个或某几个的边的宽度、样式和颜色。

另外，在该窗口底部，还可以设置边框的四个角是否使用圆角，依次为左上角、右上角、右下角和左下角。如果要想让某个角显示为圆角，则输入具体的圆角半径即可。通过四个圆角值的不同组合，可以设置出各种各样的形状，包括圆角矩形、椭圆形、圆形等等，如下图所示：



组件定位在 TaskBuilder 中的典型应用

上面介绍了很多种定位方式，但比较常用的主要就是绝对定位和弹性定位，所以下面重点介绍一下这两种定位方式在 TaskBuilder 里的典型应用场景。

1. 绝对定位的典型应用

在创建前端 TFP 页面时，如果定位方式选择的 **绝对定位**，则可以用鼠标随意拖拽摆放组件的位置，这种方式虽然看起来很灵活，但页面上组件多了之后，组件之间的对齐、间距等就不太好调整，例如多行组件之间的行间距如果保持一致，就得开发者自己计算好每行的 y 坐标，然后挨个设置这行的各个组件，非常麻烦。

其实绝对定位不仅仅只是能通过设置左侧和上侧偏移量确定组件的精确显示位置，而且，通过设置上下左右四个偏移量的不同组合，可以实现组件停靠在父容器的某一侧或者填充组件剩余空间等效果。另外，还可以实现组件宽度或高度可以随着页面大小自动变化，达到自适应屏幕大小的效果。

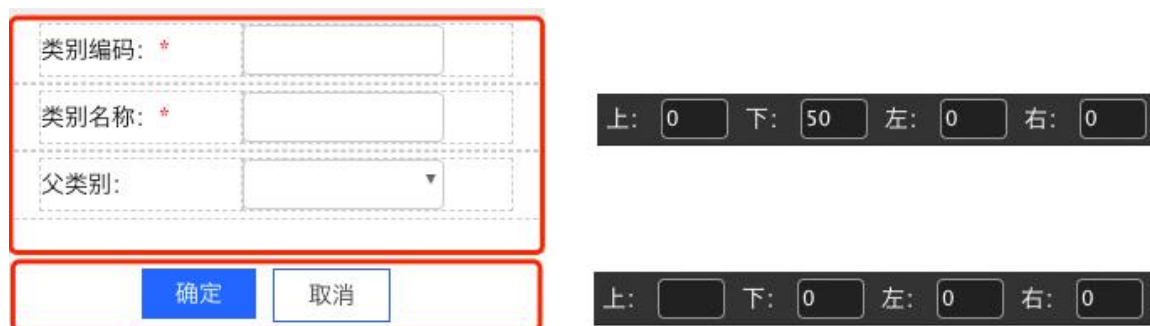
那么哪些应用场景更加适合绝对定位呢，下面就给大家介绍一下。

(1) 页面整体布局

绝对定位比较适合进行页面整体布局，例如下图所示的进销存演示项目里的商品管理页面，就是用两个采用了绝对定位的面板组件，将页面划分为左右两个区域，左侧的面板组件设置了上下和左侧的偏移量，并且设置了固定的宽度，高度随页面变化而变化，这样就实现了向左停靠的效果。右侧的面板组件上下左右都设置了偏移量，且左侧的偏移量是左侧面板的宽度加上两边的间距，这样就实现了填充右侧剩余空间的效果，宽度和高度多会随着页面的大小变化而变化。



另外，采用弹出对话框形式显示的页面一般都是分上下两栏布局，底部显示一个固定高度的工具栏，用来摆放确定和取消等按钮，上部剩余区域显示页面的具体内容。这种情况就可以采用绝对定位，上下分别放两个面板组件。下面的面板组件将左右和底部的偏移量都设置为 0，高度设置为 50，实现向下停靠。上面的面板组件将左右和上部的偏移量设置为 0，底部的偏移量设置为 50，不设置宽度和高度，实现对页面剩余空间的填充，界面如下图所示：



(2) 容器组件定位

很多容器组件的位置需要停靠在父容器的指定位置，宽度或高度需要随着页面大小变化而变化，这时候就可以采用绝对定位。用 TaskBuilder 的向导创建的数据查询 TFP 页面，顶部默认会放一个面板组件，用来添加各种查询输入项，以及添加、查询、导入和导出等按钮，该面板组件会设置固定的高度，左右和顶部的偏移量会设置为 0，实现

顶部停靠；页面剩余的空间会放一个数据列表组件，该组件左右和底部的偏移量会设置为 0，顶部的偏移量设置为上部面板组件的高度，实现填充页面剩余空间的效果。

计量单位编码	计量单位名称	修改	删除
{dwbm}	{dwmc}	修改	删除

(3) 多层堆叠显示

有些情况需要在页面里弹出一个单独的层显示相关内容，这个层位于页面里其他组件的上面，并遮挡住这些组件的内容，这时候就可以采用绝对定位。将组件设置为绝对定位后，可以设置其堆叠顺序，值越大显示会靠上层。例如在任擎后台管理系统里弹出的对话框和提示框，都是在最外层页面里动态创建的基于绝对定位的层（div，对应面板组件），并将堆叠顺序设置得足够大（>10000），保证其能显示在页面里所有其他组件上面，然后再在里面显示其他内容：



对话框示例



提示框示例

(4) 组件精确定位

如果需要对组件的显示位置精确到一个像素都不能差，那这时，就建议采用绝对定位，通过设置组件的左侧偏移量和顶部的偏移量，相当于设置了组件显示位置的 x 坐标和 y 坐标，来实现精确定位。

2. 弹性定位的典型应用

在 TaskBuilder 内，弹性定位不是页面可选的默认定位方式，但提供了弹性面板和弹性栅格两个组件可以实现弹性定位，具体介绍如下：

(1) 弹性面板组件的应用

弹性面板组件适合用来摆放不同类型的子组件、而且需要采用弹性布局的场合，例如想让多个组件水平或垂直居中，通过设置弹性面板组件属性里的元素排列方向、水平

或垂直对齐方式就可以很容易实现。

拖入到弹性面板里的组件都会被当做弹性元素进行弹性定位，可以在顶部样式设置栏里的“显示和定位”中设置扩展量、收缩量、基准长度和对齐方式等参数。

在没有弹性面板组件之前，在数据查询页面里，放置查询条件输入项和添加、查询等按钮的面板组件默认采用的浮动定位，在添加新的输入项或按钮时，新建的组件默认左侧和最后一个组件的右边框紧挨着，顶部跟面板的上边框紧挨着的，如下图所示：



计量单位编码	计量单位名称	修改	删除
{dwbm}	{dwmc}	修改	删除

这样肯定是不美观，原来需要再设置一下组件的左外边距和上外边距，才能实现新增的组件和前面的组件之间保留一定的间距，并垂直居中。

现在使用弹性面板组件，就不用这么麻烦，因为弹性面板组件默认设置的就是让各个子组件垂直居中显示，并且默认加了一个左外边距，这样新增加的组件不需要再进行其他设置就能达到理想效果，如下图所示：



计量单位编码	计量单位名称	修改	删除
{dwbm}	{dwmc}	修改	删除

另外，弹性面板组件和 **数据卡片** 组件结合，可以用卡片的形式动态显示多条数据记录的信息。将弹性面板组件的元素排列方向设置为从左至右横向显示，并自动换行；数据卡片组件通过绑定后台数据查询服务，就可以为每一条查询结果动态生成一个数据卡片，为卡片内的输入项设置好数据绑定格式即可将数据记录中指定字段的内容显示在

卡片中，大致效果如下图所示：



(2) 弹性栅格组件的应用

相对弹性面板，弹性栅格组件可以批量设置指定数量的子元素，然后再往这些子元素里拖拽其他组件，这种方式可以快速构建多行或多列布局，比较适合以下几种场景：

1) 表单输入项定位

弹性栅格组件默认会创建两个子元素，左窄右宽，比较适合在添加、修改或详情页面里快速实现表单输入项所在行的左右两栏布局，每个输入项拖一个弹性栅格组件，然后在其左侧放置显示输入项名称的文本标签组件，右侧放置具体的输入项，如下图所示：

添加供应商	
客户编码：*	
客户名称：*	
联系人：*	
联系电话：*	

2) 多行或多列内容布局

前面介绍的用绝对定位进行页面多栏布局的场合，其实也可以采用弹性栅格组件实现，具体采用哪种方式，看个人喜好，但如果栏数（行或列）比较多，采用弹性栅格会

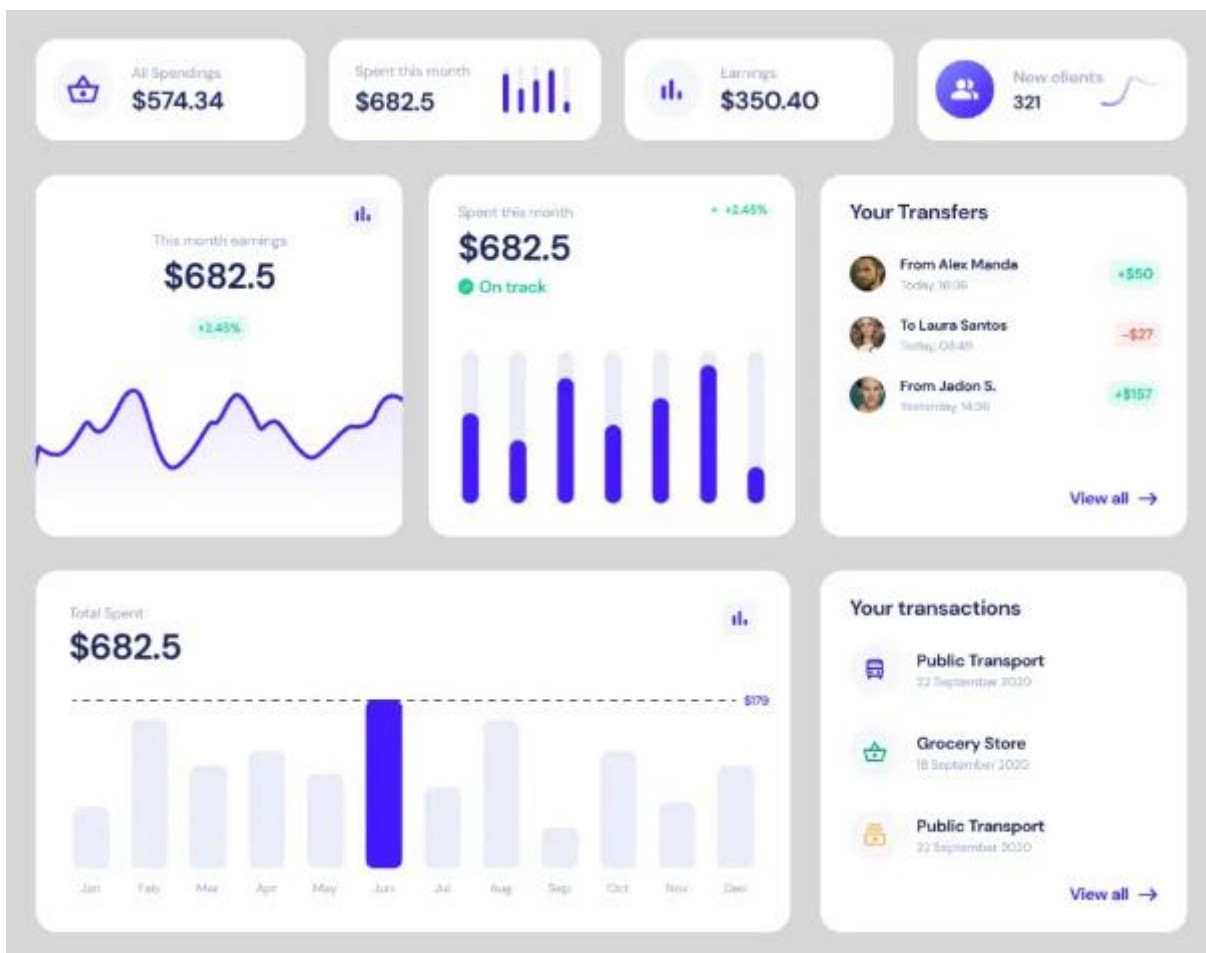
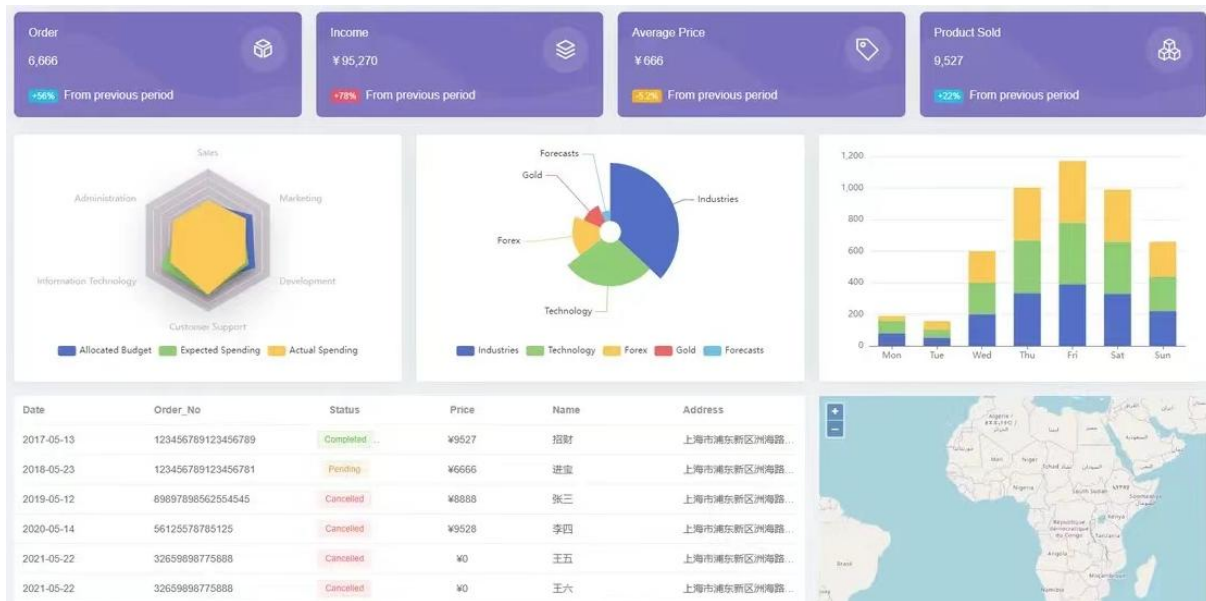
更加方便。例如下图所示的任擎后台管理系统默认的首页里，就大量地采用了弹性栅格，整个页面最外层用一个弹性栅格把页面分为左右两列，右侧的宽度固定，不能伸缩，左侧宽度根据屏幕自适应，两侧再放其他弹性栅格组件层层嵌套实现复杂的多行多列的页面布局，显示丰富的内容。



弹性栅格组件如果用好了，可以实现非常美观的界面效果，下面给大家分享几张从网上找的 UI 界面图，理论上用弹性栅格组件都可以实现，大家可以自己尝试一下。



《TaskBuilder 组件定位详解》





另外，在手机端 H5 页面中，使用弹性栅格组件，也可以很容易实现经典的九宫格布局或多行多列的应用列表界面，如下图所示：

