

Foundation 网格实例

下面我们收集了一些网格常用的实例。

三个均等列

该实例演示了如何创建三个均等列 (33.3%/33.3%/33.3%)，在中型和大型设备上显示三个列，在小型设备上自动堆叠：

实例

```
<div class="row">
  <div class="medium-4 columns" style="background-color:yellow;">
    <p>.medium-4</p>
  </div>
  <div class="medium-4 columns" style="background-color:pink;">
    <p>.medium-4</p>
  </div>
  <div class="medium-4 columns" style="background-color:yellow;">
    <p>.medium-4</p>
  </div>
</div>
```

[尝试一下 »](#)

三个不均等列

该实例演示了如何创建三个不均等列 (25%/50%/25%)，在中型和大型设备上显示三个列，在小型设备上自动堆叠：

实例

```
<div class="row">
  <div class="medium-3 columns" style="background-color:yellow;">
    <p>.medium-3</p>
  </div>
  <div class="medium-6 columns" style="background-color:pink;">
    <p>.medium-6</p>
  </div>
  <div class="medium-3 columns" style="background-color:yellow;">
    <p>.medium-3</p>
  </div>
</div>
```

[尝试一下 »](#)

两个均等列

该实例演示了如何创建两个均等列 (50%/50%)，在小型、中型和大型设备上列的比例始终为 50%/50%：

实例

```
<div class="row">
  <div class="small-6 columns" style="background-color:yellow;">
    <p>.small-6</p>
  </div>
  <div class="small-6 columns" style="background-color:pink;">
    <p>.small-6</p>
  </div>
</div>
```

[尝试一下 »](#)

两个不均等列

该实例演示了如何创建两个不均等列 (33.3%/66.6%)，在小型、中型和大型设备上列的比例始终为 33.3%/66.6%：

实例

```
<div class="row">
  <div class="small-8 columns" style="background-color:yellow;">
    <p>.small-8</p>
  </div>
  <div class="small-4 columns" style="background-color:pink;">
    <p>.small-4</p>
  </div>
</div>
```

[尝试一下 »](#)

修改列的顺序

通过使用 `.small|medium|large-push-*` 和 `.small|medium|large-pull-*` 类来修改列的顺序：

实例

```
<div class="row">
  <div class="small-4 small-8-push columns" style="background-color:yellow;">
    <p>.small-4 .small-8-push</p>
  </div>
  <div class="small-8 small-4-pull columns" style="background-color:pink;">
    <p>.small-8 .small-4-pull</p>
  </div>
</div>
```

[尝试一下 »](#)

嵌套列

你可以使用嵌套网格(列中插入列)：

实例

```
<div class="row">
  <div class="small-8 columns">.small-8
    <div class="row">
      <div class="small-8 columns">.small-8 Nested
        <div class="row">
          <div class="small-8 columns">.small-8 Nested Again</div>
          <div class="small-4 columns">.small-4</div>
        </div>
      </div>
      <div class="small-4 columns">.small-4</div>
    </div>
  <div class="small-4 columns">.small-4</div>
</div>
```

[尝试一下 »](#)

混合：手机、桌面设备

Foundation 网格系统有三个列：.small-* (手机), .medium-* (平板), 和 .large-* (桌面设备)。这些类可以动态组合使用，让布局更加灵活：

提示： 每个类都能放大，如果你希望小型和大型屏幕设备的宽度一样可以设置指定 .small-*。

实例

```
<div class="row">
  <div class="small-6 large-8 columns">.small-6 .large-8</div>
  <div class="small-6 large-4 columns">.small-6 .large-4</div>
</div>
<div class="row">
  <div class="small-2 large-4 columns">.small-2 .large-2</div>
  <div class="small-4 large-4 columns">.small-4 .large-2</div>
  <div class="small-6 large-4 columns">.small-6 .large-2</div>
</div>
<div class="row">
  <div class="small-3 large-5 columns">.small-3 .large-5</div>
  <div class="small-9 large-7 columns">.small-9 .large-7</div>
</div>
```

[尝试一下 »](#)

混合：手机、平板和桌面设备

实例

```
<div class="row">
  <div class="medium-6 large-8 columns">.medium-6 .large-8</div>
```

```
<div class="medium-6 large-4 columns">.medium-6 .large-4</div>
</div>
<div class="row">
  <div class="small-4 medium-3 large-7 columns">.small-4 .medium-3 .large-7</div>
  <div class="small-4 medium-6 large-3 columns">.small-4 .medium-6 .large-3</div>
  <div class="small-4 medium-3 large-2 columns">.small-4 .medium-3 .large-2</div>
</div>
```

[尝试一下 »](#)

居中列

列居中可以使用 `.small-centered` 类。中型和大型设备可以继承小型设备的居中，但你需要在大型设备上设置居中类 `.large-centered`。

实例

```
<div class="row">
  <div class="small-4 small-centered columns">small-4 small-centered</div>
</div>
<div class="row">
  <div class="small-6 small-centered columns">small-6 small-centered</div>
</div>
<div class="row">
  <div class="small-6 large-centered columns">small-6 large-centered</div>
</div>
<div class="row">
  <div class="small-8 small-centered large-uncentered columns">small-8 small-centered large-uncentered</div>
</div>
<div class="row">
  <div class="small-10 small-centered columns">small-10 small-centered</div>
</div>
```

[尝试一下 »](#)

列偏移量

可以使用 `.large-offset-*` (或 `.small-offset-*`) 类设置列向右移。左侧外边距的列数量使用 * 号控制:

实例

```
<div class="row">
  <div class="large-1 columns">1</div>
  <div class="large-11 columns">11</div>
</div>
<div class="row">
  <div class="large-1 columns">1</div>
  <div class="large-10 large-offset-1 columns">10, offset 1</div>
</div>
<div class="row">
```

```
<div class="large-1 columns">1</div>
<div class="large-9 large-offset-2 columns">9, offset 2</div>
</div>
<div class="row">
  <div class="large-1 columns">1</div>
  <div class="large-8 large-offset-3 columns">8, offset 3</div>
</div>
```

[尝试一下 »](#)

不完整列

如果一行中的列数量之和不是 12，Foundation 将自动将最后一列向右浮动，并使用空白来填充剩下的列。

可选项 `.end` 类用于设置最后一列的元素向左边浮动：

实例

```
<div class="row">
  <div class="medium-3 columns">.medium-3</div>
  <div class="medium-3 columns">.medium-3</div>
  <div class="medium-3 columns">.medium-3</div>
</div>
<div class="row">
  <div class="medium-3 columns">.medium-3</div>
  <div class="medium-3 columns">.medium-3</div>
  <div class="medium-3 columns end">.medium-3 .end</div>
</div>
```

[尝试一下 »](#)

宽屏

网格 (`.row`) 最大尺寸 (`max-width`) 为 62.5rem。在宽屏设备上尺寸可能大于 62.5rem，这样列就无法完整填充页面，即便宽度设置为 100%。但是我们可以通过 CSS 来设置新的 `max-width`：

实例

```
<style>
.row {
  max-width: 100%;
}
</style>
```

[尝试一下 »](#)

如果你想使用默认的 `max-width`，但是背景颜色需要跨域整个页面，这时你在容器元素上使用 `.row` 类，并指定你需要的背景颜色：

实例

```
<div style="background-color:coral;padding:25px;">
  <div class="row">
    <div class="small-6 columns" style="background-color:yellow;">.small-6</div>
    <div class="small-6 columns" style="background-color:pink;">.small-6</div>
  </div>
</div>
```

[尝试一下 »](#)[← Foundation 块状网格](#)[Foundation 图标参考手册 →](#)[📝 点我分享笔记](#)