sm min-width: 576px 手机

md min-width: 768px 平板

lg min-width: 992px 电脑

xl min-width: 1200px 大屏

**一、FlexGrid**

FlexGrid是一个基于flexbox的CSS实用程序。有关Flex的更多信息，请访问Flexbox的完整指南。基本网格是通过给出一个容器p-grid类和其子p-col类来定义的。网格的子元素将具有相同的宽度，并根据父元素的宽度进行缩放。

**列固定宽度**一个列可以有固定的宽度，而兄弟列可以有自动宽度。应用p-col-fixed类固定列宽度。

<div class="p-grid">

<div class="p-col-fixed" style="width:100px">Fixed</div>

<div class="p-col">Auto</div>

</div>

**响应**响应式布局是通过将特定于断点的类应用于列来实现的，而p-col-\*则为带有小屏幕的移动设备定义默认行为。不同的断点支持四种屏幕尺寸。

在下面的例子中，大屏幕显示4列，中等屏幕显示2列在2行，最后在小设备上，列堆叠。

<div class="p-grid">

<div class="p-col-12 p-md-6 p-lg-3">A</div>

<div class="p-col-12 p-md-6 p-lg-3">B</div>

<div class="p-col-12 p-md-6 p-lg-3">C</div>

<div class="p-col-12 p-md-6 p-lg-3">D</div>

</div>

**偏移**

偏移量类允许在列上定义左边距，以避免为空格添加空列。

<div class="p-grid">

<div class="p-col-6 p-offset-3">6</div>

</div>

<div class="p-grid">

<div class="p-col-4">4 </div>

<div class="p-col-4 p-offset-4">4</div>

</div>

**嵌套的**可以嵌套列来创建更复杂的布局。

**二、FlexBox**

使用响应式FlexBox实用程序轻松管理组件的布局。

可以使用p-d-flex或p-d-inline-flex类将元素配置为flexbox容器。

<div class="p-d-flex">Flex Container</div>

<div class="p-d-inline-flex">Inline Flex Container</div>

**响应**

行方向为更大的屏幕，列为更小的。

<div class="p-d-flex p-flex-column p-flex-md-row">

<div class="p-mb-2 p-mr-2">Item 1</div>

<div class="p-mb-2 p-mr-2">Item 2</div>

<div class="p-mb-2 p-mr-2">Item 3</div>

</div>

**Justify Content**

对齐内容是沿着主轴的对齐方式，使用p-jc-{value}格式定义，并使用响应替代方案。Value字段接受下面的选项。

start end center between around evenly

**Align Items**

Align Items配置是沿十字轴的对齐，使用p-ai-{value}格式定义，并具有响应性备选方案。Value字段接受下面的选项。

start end center baseline stretch (default)

**Align Self**

对齐自配置是针对特定元素沿着十字轴的对齐，使用p-as-{value}格式和响应备选方案进行定义。Value字段接受下面的选项。

start end center baseline stretch (default)

**Align Content**

Align内容是沿着十字轴的对齐方式，使用p-ac-{value}格式定义，并具有响应性备选方案。Value字段接受下面的选项。

start end center around stretch between

**Margin with FlexBox**

当与间隔实用程序结合时，flexbox提供了无限的可能性。

水平间隔

<div class="p-d-flex p-p-3 card">

<button pButton pRipple type="button" icon="pi pi-check" class="p-mr-2">

</button>

<button pButton pRipple type="button" icon="pi pi-trash" class="p-button-danger">

</button>

<button pButton pRipple type="button" icon="pi pi-search" class="p-ml-auto p-button-help">

</button>

</div>

**垂直线间距离**

<div class="p-d-flex p-flex-column" style="height:150px">

<div>Item 1</div>

<div class="p-mt-auto">Item 2</div>

</div>

**三、表单布局**表单布局是一种CSS实用工具，用于轻松创建表单。FormLayout不包含在PrimeNG中，因为它是由PrimeFlex提供的，一个在PrimeFaces, PrimeNG, PrimeReact和PrimeVue项目之间共享的网格库。

**开始**

FormLayout的核心成员是.p-field类，它包装了输入字段和相关的标签。

<div class="p-field">

<label for="fieldId">Label</label>

<input id="fieldId" type="text" pInputText>

</div>

**垂直布局**

以最简单的形式，在.p-fluid中使用时将创建垂直布局，使组件使用所有可用的宽度。

<div class="p-fluid">

<div class="p-field">

<label for="firstname1">Firstname</label>

<input id="firstname1" type="text" pInputText>

</div>

<div class="p-field">

<label for="lastname1">Lastname</label>

<input id="lastname1" type="text" pInputText>

</div>

</div>

**网格垂直布局**

这就是FormLayout在。p-formgrid类的帮助下连接到PrimeFlex的地方，以优化表单设计的内容。下面的示例将两个字段排列在相邻两个字段的后面。

<div class="p-fluid p-formgrid p-grid">

<div class="p-field p-col">

<label for="firstname1">Firstname</label>

<input id="firstname1" type="text" pInputText>

</div>

<div class="p-field p-col">

<label for="lastname1">Lastname</label>

<input id="lastname1" type="text" pInputText>

</div>

</div>

**水平和固定宽度**

在水平形式中，字段的标签显示在输入的同一行上，与垂直形式相反。在这个例子中，标签有一个固定的宽度，其中输入的容器获得剩余的空间。

<div class="p-field p-grid">

<label for="firstname3" class="p-col-fixed" style="width:100px">Firstname</label>

<div class="p-col">

<input id="firstname3" type="text" pInputText>

</div>

</div>

<div class="p-field p-grid">

<label for="lastname3" class="p-col-fixed" style="width:100px">Lastname</label>

<div class="p-col">

<input id="lastname3" type="text" pInputText>

</div>

</div>

**水平和流体**

使用.p-fluid包装上一个示例中的表单并删除固定宽度会产生流体布局。

<div class="p-fluid">

<div class="p-field p-grid">

<label for="firstname" class="p-col-12 p-mb-2 p-md-2 p-mb-md-0">Firstname</label>

<div class="p-col-12 p-md-10">

<input id="firstname" type="text" pInputText>

</div>

</div>

<div class="p-field p-grid">

<label for="lastname" class="p-col-12 p-mb-2 p-md-2 p-mb-md-0">Lastname</label>

<div class="p-col-12 p-md-10">

<input id="lastname" type="text" pInputText>

</div>

</div>

</div>

**内联**

内联表单用于将内容显示在同一行上，只需将.p-formgroup- Inline添加到表单容器中即可实现。请注意，根据设计要求，如果标签没有在视觉上隐藏，建议使用.p-sr-only来仍然支持屏幕阅读器

<div class="p-formgroup-inline">

<div class="p-field">

<label for="firstname" class="p-sr-only">Firstname</label>

<input id="firstname" type="text" pInputText placeholder="Firstname">

</div>

<div class="p-field">

<label for="lastname" class="p-sr-only">Lastname</label>

<input id="lastname" type="text" pInputText placeholder="Lastname">

</div>

<button pButton type="button" label="Submit"></button>

</div>

**复选框和RadioButton**

Checkbox和RadioButton分别通过.p-field- Checkbox和.p-field- RadioButton类独占布局支持。这里的示例演示了垂直和水平布局的替代方案。

**高级形式**

可以使用.p-field、.p-formgrid和.p-fluid组合来轻松实现具有各种输入字段的响应式表单。

<div class="p-fluid p-formgrid p-grid">

<div class="p-field p-col-12 p-md-6">

<label for="firstname">Firstname</label>

<input id="firstname" type="text" pInputText>

</div>

<div class="p-field p-col-12 p-md-6">

<label for="lastname">Lastname</label>

<input id="lastname" type="text" pInputText>

</div>

<div class="p-field p-col-12">

<label for="address">Address</label>

<textarea id="address" type="text" rows="4" pInputTextarea></textarea>

</div>

<div class="p-field p-col-12 p-md-6">

<label for="city">City</label>

<input id="city" type="text" pInputText>

</div>

<div class="p-field p-col-12 p-md-3">

<label for="state">State</label>

<p-dropdown inputId="state" [options]="states" [(ngModel)]="selectedState" placeholder="Select" optionLabel="name"></p-dropdown>

</div>

<div class="p-field p-col-12 p-md-3">

<label for="zip">Zip</label>

<input id="zip" type="text" pInputText>

</div>

</div>

**四、间距**

PrimeFlex提供了各种间距工具来修改元素的布局。

开始

空格类使用p-{property}{position}-{value}语法，而响应式值使用p-{property}{position}-{breakpoint}-{value}格式。

**Property**

m: margin 外边距

p: padding 内边距

**Position**

各边的位置可选，x-y轴的位置可选或毛坯的位置可选。

t: top

b: bottom

l: left

r: right

x: left and right

y: top and bottom

blank: all sides

**Value**

值字段从0到6不等，$间隔符的默认值是1rem。特殊的auto值只对页边距可用，并用于使元素居中。

0: $spacer \* 0

1: $spacer \* .25

2: $spacer \* .5

3: $spacer \* 1

4: $spacer \* 1.5

5: $spacer \* 2

6: $spacer \* 3

auto: auto margin

**Breakpoint**

断点根据屏幕大小定义了如何定义间距。设计首先是移动的，所以较小的值也可以应用于较大的值，如md也适用于lg或xl(如果它们没有明确定义的话)。

sm: small screens e.g. phones

md: medium screens e.g. tablets

lg: large screens e.g. notebooks

xl: larger screens .e.g monitors

**Examples**

<div class="p-mb-2">Margin bottom with level 2</div>

<div class="p-mt-4">Margin top with level 4</div>

<div class="p-m-2">Margin for all sides with level 2</div>

<div class="p-mx-auto">Auto margins for left and right side</div>

<div class="p-pb-4">Padding bottom with level 4</div>

<div class="p-p-1">Padding for all sides with level 1</div>

<div class="p-m-1 p-p-1 p-m-lg-3 p-b-lg-3">Level 3 spacing for lg screens and level 1 for smaller screens (xs).</div>

**五、文本**

使用PrimeFlex文本实用程序管理文本的对齐和样式。

**开始**

文本类使用p-text-{value}语法。

**对齐**

四个选项可用于对齐： left center right justify

<div class="p-text-left">Left</div>

<div class="p-text-center">Center</div>

<div class="p-text-right">Right</div>

<div class="p-text-justify">Justify</div>

**Wrap**

文本换行定义了当文本超过其容器的大小时如何显示文本。

nowrap wrap truncate

<div style="width: 10rem">长文本自动换行且不过低.</div>

<div class="p-text-nowrap" style="width: 10rem">长文本不会换行并溢出父文本.</div>

<div class="p-text-nowrap p-text-truncate" style="width: 10rem">长文本不会换行不溢出父文本。</div>

**Transform**

属性改变文本的大小写。 lowercase uppercase capitalize

<div class="p-text-lowercase">文本中所有字都小写</div>

<div class="p-text-uppercase">文本中所有字都大写</div>

<div class="p-text-capitalize">首字母大写</div>

**Style**

文本样式适用于字体粗细和样式： bold normal light italic

<div class="p-text-bold">粗体</div>

<div class="p-text-normal">正常的</div>

<div class="p-text-light">灯光</div>

<div class="p-text-italic">斜体</div>

**六、Display**

使用PrimeFlex显示工具有效地配置组件的显示。

**开始**

显示类使用p-d-{value}格式和p-d-{breakpoint}-{value}格式进行响应式设计。

**Values**

Value字段可以是以下选项之一。

none inline inline-block block flex inline-flex

**Breakpoint**

断点定义了如何根据屏幕大小定义display属性。设计首先是移动的，所以较小的值也可以应用于较大的值，如md也适用于lg或xl(如果它们没有明确定义的话)。

sm: 小屏幕，例如手机

md: 中等屏幕，例如平板电脑

lg: 大屏幕，例如笔记本

xl: 大屏幕生产。g监测

**Examples**

<div class="p-d-inline">

显示为内联.</div>

<div class="p-d-flex">

显示为一个flexbox容器.</div>

<div class="p-d-block p-d-lg-inline">

内联用于更大的屏幕，阻塞用于其他屏幕.</div>

<div class="p-d-md-none">

在小屏幕上可见</div>

<div class="p-d-none p-d-md-inline-flex">

隐藏在一个小屏幕上</div>

<div class="p-d-none p-d-print-block">

打印时才可见。</div>

<div class="p-d-block p-d-print-none">

无法打印.</div>