链滴

1 行代码，10 种现代布局

作者：Rabbitzzc
原文链接：https://ld246.com/article/1595421590454
来源网站：链滴

1行代码，10种现代布局

~!!! 一般B端系统可以使用，C端系统有时候需要考虑到兼容性

作为前端开发，在开发需求的时候，选择一个布局并实现一个布局，有时候还是蛮头疼的，有时候写布局以后，上线被产品发现布局或者样式有问题。所以这篇文章总结了几行功能强大的CSS，这些代可以完成一些繁琐头疼的工作，构建可靠的现代布局。

- code： https://codesandbox.io/s/one-line-css-ten-modern-layouts-3k5r7?file=/index.html
- demo： https://3k5r7.csb.app/

01. Super Centered: place-items: center

place-items: center是一个非常强大的居中布局属性，一行代码能够同时设置水平垂直居中。

首先指定 display:grid，然后编写 place-items：center。place-items是同时设置 align-items和 justify-items的简写。 通过将其设置为居中，等同于同时设置 align-items和 justify-items为居中。

```
.parent {
  display: grid;
  place-items: center;
}
```

02. Flex: grow shrink baseWidth

例如，营销站点的常见布局，可能有3行，通常带有图像，标题和一些文本，描述了产品的某些功能。在移动设备上，我们希望它们能很好地堆叠在一起，并随着屏幕尺寸的增加而扩展。

而通过使用 Flexbox布局来实现此效果，不需要媒体查询@media等即可在调整网页屏幕大小时调整些元素的位置。展示效果会非常的自然。

flex属性包括：flex: <flex-grow> <flex-shrink> <flex-basis>，分别是规定项目将相对于其他灵活项目进行扩展的量，规定项目将相对于其他灵活的项目进行收缩的量，以及项目的长度。

! 如果需要详细了解flex的三个属性，可以参考：https://www.zhangxinxu.com/wordpress/2019/1/css-flex-deep/

flex: 0 1 width

```
.parent {
  display: flex;
  flex-wrap: wrap;
  justify-content: center;

}

.child {
  flex: 0 1 150px;
}
```

flex: 1 1 width

```
.parent {
  display: flex;
}

.child {
  flex: 1 1 150px;
}
```

上面两种不同的设置，实际的布局效果也是不同的，具体可以根据产品需求来设置。

03. Sidebar: grid-template-columns: minmax(<min>, <max>)

在demo中，是利用了 minmax函数的网格布局。 在这里所做的是将最小边栏大小设置为150px，但较大的屏幕上，将其扩展到25%。 边栏将始终占据其父级水平空间的25%，直到25%小于150px。

首先给父类设置 grid-template-columns为 minmax(150px, 25%) 1fr。侧边栏设置最小值为150px 25%，右边内容区域对剩余空间进行等分。

CSS Grid带来新的单位：fr，代表对剩余空间进行等分

```
.parent {
  display: grid;
  /** 横向设置*/
  grid-auto-flow: column;
  grid-template-columns: minmax(150px, 25%) 1fr;
}
```

04. Hamburger grid-template-rows: auto 1fr auto

在上面通过 grid-template-columns设置了横向的侧边栏布局，实际上，通过 grid-template-rows性，也是可以固定一个div，另个一个div自适应的，只需要将上面属性改为 *-rows即可。

grid-template-row还有一个作用，可以利用其 auto 1fr auto属性值，来实现汉堡amburger夹层布局。

```
.parent {
  display: grid;
  /* 上下auto，中间布局大*/
  grid-template-rows: auto 1fr auto;
}
```

05. (Classic Holy Grail Layout)圣杯布局： grid-template: auto 1fr auto / auto 1fr auto

圣杯布局，有页眉、页脚、左边栏、右边栏（侧边栏）和中间的核心内容。

要使用一行代码编写整个网格，请使用 grid-template属性，可以同时设置行和列。

```
.parent {
   display: grid;
   grid-template: auto 1fr auto / 150px 1fr 150px;
}
```

06. (12-Span Grid)12网格布局：grid-template-columns: repeat(12, 1fr)

12网格布局，可以使用 repeat函数在CSS中快速编写网格。使用 repeat(12,1fr) ，网格模板列会提供2列，每列为1fr的距离。

```
.parent {
  display: grid;
  grid-template-columns: repeat(12, 1fr);
}
```

07. RAM (Repeat, Auto, MinMax)子元素自适应宽度: grid-template-columns(auto-fit, minmax(<base>, 1fr))

创建具有自动放置和灵活的子节点的响应式布局。很整洁。这里需要记住的关键术语是 repeat、auto-fit|fill)和minmax()，它的缩写是RAM。

```
.parent {
  display: grid;
  grid-gap: 1rem;
  /* 重复次数自适应，根据子类元素数量确定， 每个子元素最小宽度最150 */
  grid-template-columns: repeat(auto-fit, minmax(150px, 1fr));
}
```

08. Line Up(对齐)：justify-content: space-between

justify-content 属性定义了浏览器之间，如何分配顺着弹性容器主轴(或者网格行轴) 的元素之间及其围的空间。

justify-content: space-between　均匀排列每个元素首个元素放置于起点，末尾元素放置于终点。

例子中的 card元素中的子元素的排列即为展示效果。

```
.card {
  display: flex;
  flex-direction: column;
  padding: 1rem;
  justify-content: space-between;
}
```

09. Clamp 函数：居中布局的有一个方案

clamp()函数作用是返回一个区间范围的。width:clamp将设置绝对最小和最大大小以及实际大小。

clamp(MIN, VAL, MAX)

其中MIN表示最小值，VAL表示首选值，MAX表示最大值。意思是，如果VAL在MIN和MAX范围之，则使用VAL作为函数返回值；如果VAL大于MAX，则使用MAX作为返回值；如果VAL小于MIN，使用MIN作为返回值。

注意兼容性：https://caniuse.com/#search=clamp。兼容是比较差的

```
.parent {
```

```
  width: clamp(23ch, 50%, 46ch);
}
```

10. Respect for Aspect: aspect-ratio: <width> / <height>

终于到最后一个了。

aspect-ratio CSS 媒体属性 可以用来测试 viewport 的宽高比。因此可以利用检测容器的宽高比实现适应的布局。

```
.video {
  aspect-ratio: 16 / 9;
}
```

aspect-ratio的兼容性更差，只支持chrome84+。https://caniuse.com/#search=aspect-ratio

总结

上面列举了10行强大的CSS，来实现现代常用的布局。希望对大家有帮助！

本文使用 mdnice 排版