

# Html 笔记

## Grid布局

兼容性是个问题：部分浏览器在部分语法问题上依然存在不兼容现象

c

容器和项目

项目仅能为最顶级子元素

### 初步使用方法

```
div {  
    display: grid;  
}  
/*行内*/  
div {  
    display: inline-grid;  
}
```

定义行列高宽

`grid-template-columns` 属性,  
`grid-template-rows` 属性  
单位 `px` `em` `%` `repeat`

使用repeat函数

`repeat()`接受两个参数，第一个参数是重复的次数（上例是3），第二个参数是所要重复的值。

`repeat()`重复某种模式也是可以的。

**auto-fill** 关键字

有时，单元格的大小是固定的，但是容器的大小不确定。如果希望每一行（或每一列）容纳尽可能多的单元格，这时可以使用**auto-fill**关键字表示自动填充。

```
.container {  
  display: grid;  
  grid-template-columns: repeat(auto-fill, 100px);  
}
```

### (3) fr 关键字

为了方便表示比例关系，网格布局提供了fr关键字（fraction 的缩写，意为"片段"）。如果两列的宽度分别为1fr和2fr，就表示后者是前者的两倍。

```
.container {  
  display: grid;  
  grid-template-columns: 1fr 1fr;  
}
```

fr可以与绝对长度的单位结合使用，这时会非常方便。

```
.container {  
  display: grid;  
  grid-template-columns: 150px 1fr 2fr;  
}
```

### auto 关键字

auto关键字表示由浏览器自己决定长度。

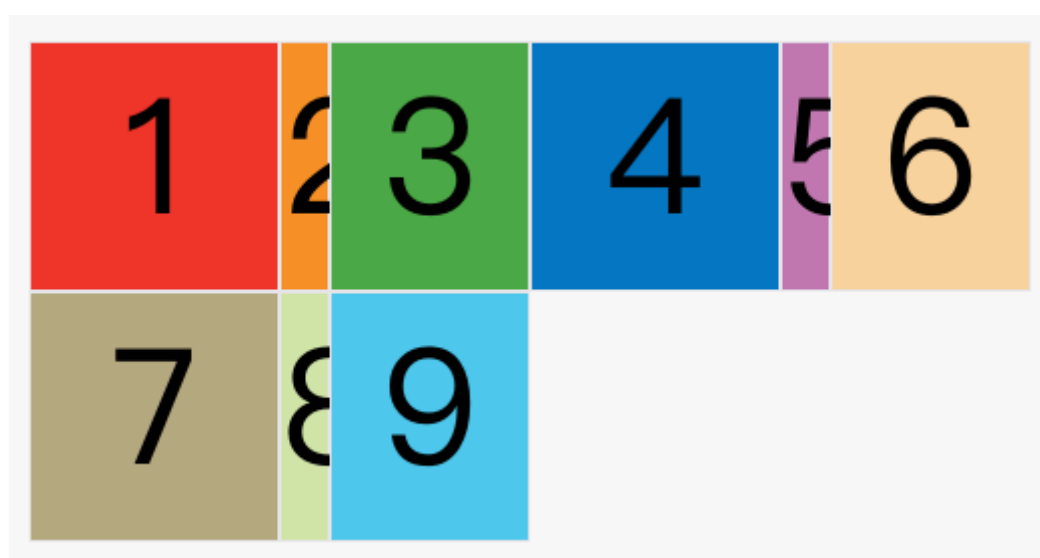
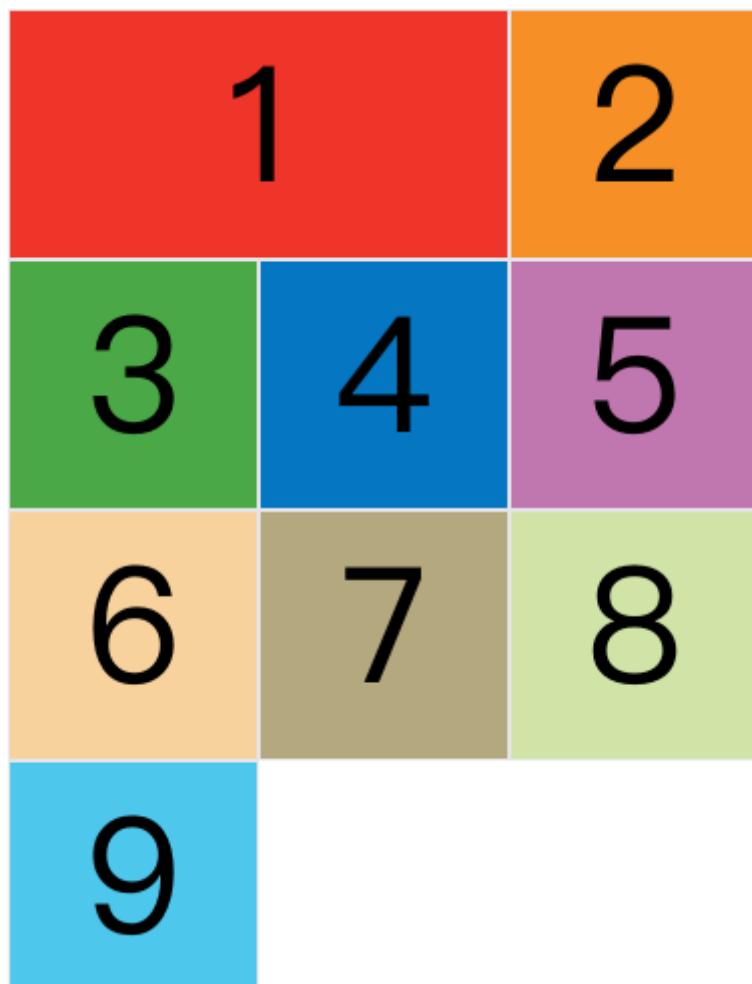
```
grid-template-columns: 100px auto 100px;
```

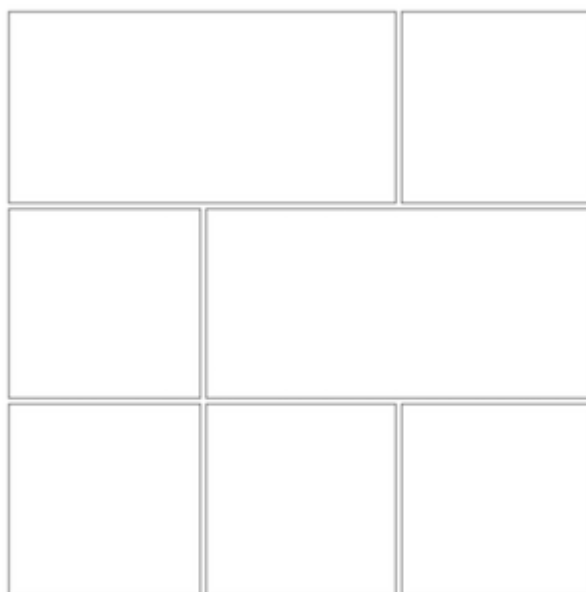
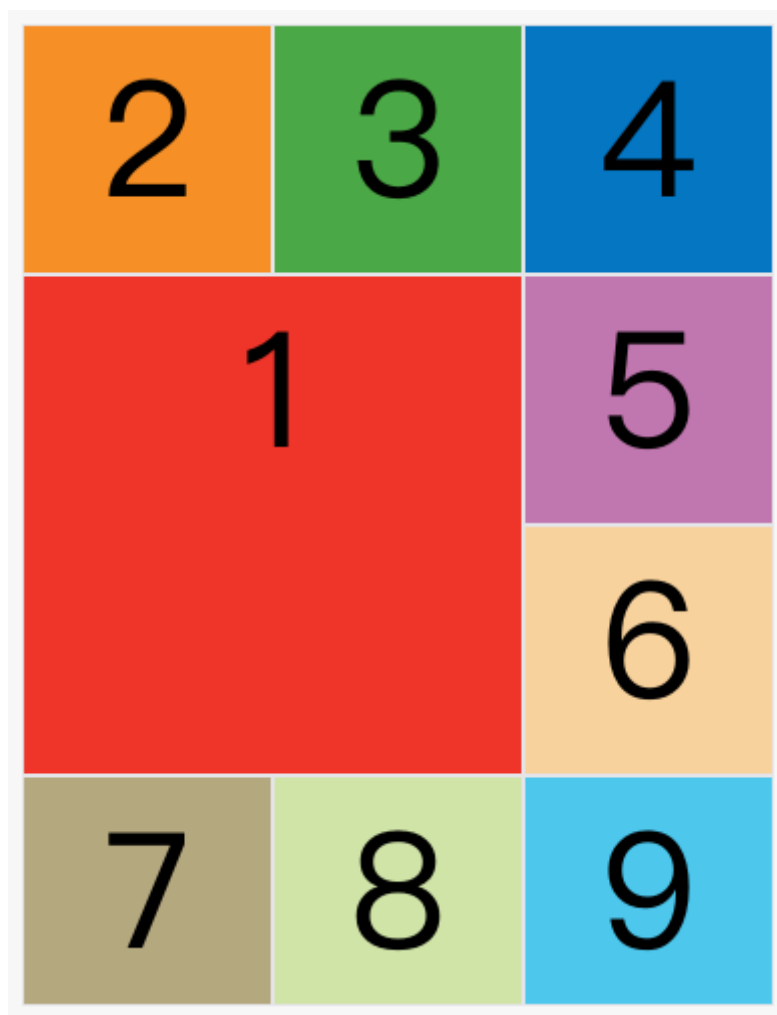
### grid-template-areas 属性

## 实际完成 目标

1	2	3
4	5	6
7	8	9

1	4	7
2	5	8
3	6	9





## 预见的问题

### 未定义

对于目标一定要完全设置，不要存在未定义的区域，避免不受控制的未定义。

### 失效

注意，设为网格布局以后，容器子元素（项目）的`float`、`display: inline-block`、`display: table-cell`、`vertical-align`和`column-*`等设置都将失效。

