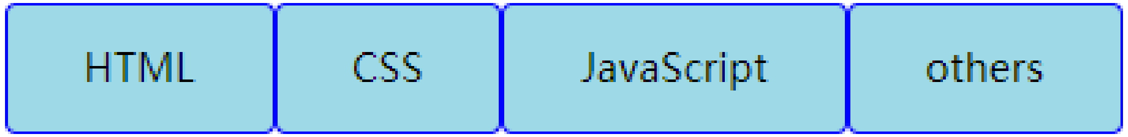
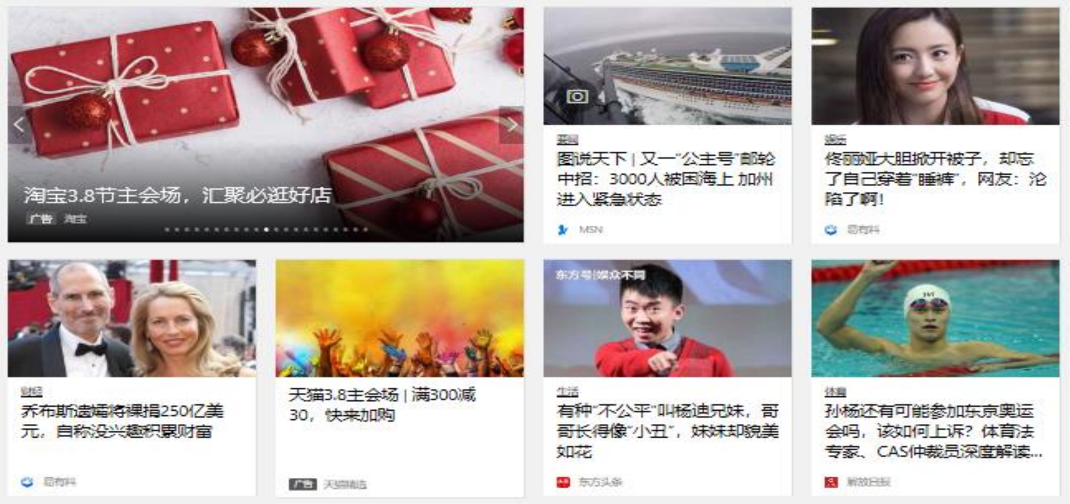
概述

弹性盒子flexbox
一维布局



HTML | CSS | JavaScript | others

网格布局grid
二维布局



淘宝3.8节主会场，汇聚必逛好店
广告 淘宝

要闻
图说天下 | 又一"公主号"邮轮中招：3000人被困海上 加州进入紧急状态
MSN

娱乐
佟丽娅大胆掀开被子，却忘了自己穿着"睡裤"，网友：沦陷了啊！
看有料

财经
乔布斯遗孀将裸捐250亿美元，自称没兴趣积累财富
看有料

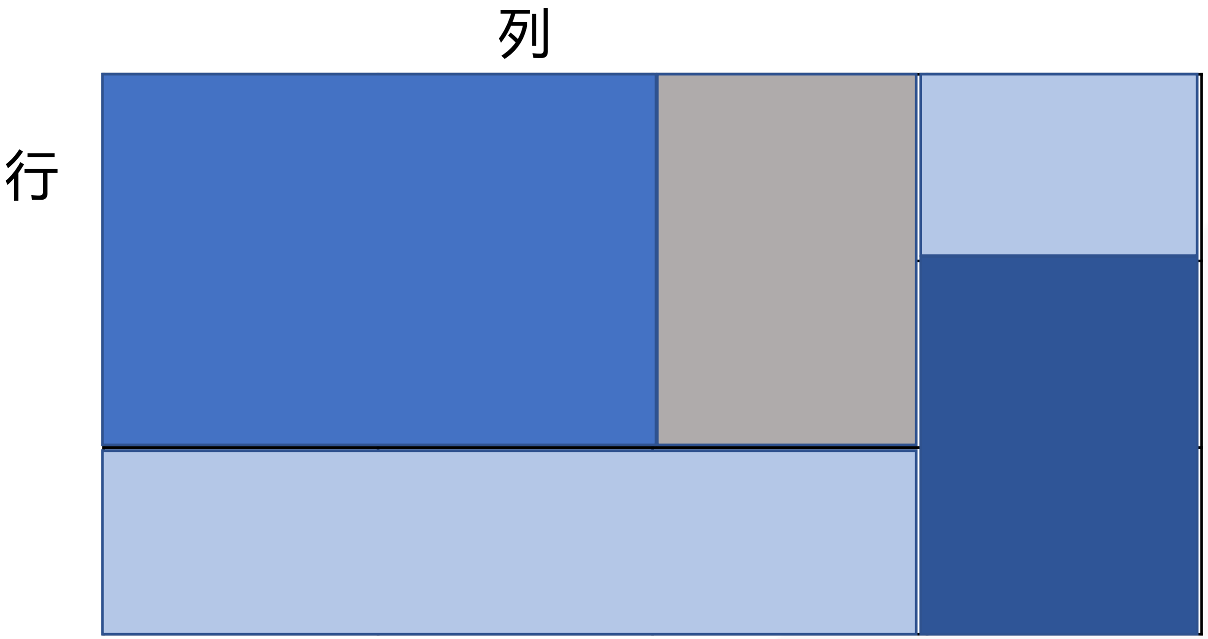
天猫3.8主会场 | 满300减30，快来加购
广告 天猫精选

东方号|娱众不同
生活
有种"不公平"叫杨迪兄妹，哥哥长得像"小丑"，妹妹却貌美如花
东方头条

体育
孙杨还有可能参加东京奥运会吗，该如何上诉？体育法专家、CAS仲裁员深度解读...
解放日报

网格布局

网格布局（Grid Layout）



网格容器（Grid Container）
网格元素（Grid Item）

网格容器样式

1. display属性：定义网格容器

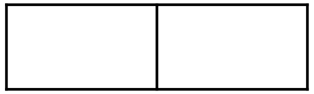
```
.grid-container{
        display: grid;
}
```

网格容器样式

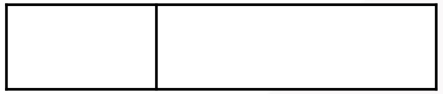
2. grid-template-columns和grid-template-rows

划分网格的行和列，取值：px、%、auto、fr、repeat()函数

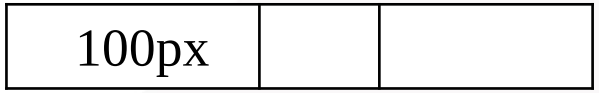
grid-template-columns: 1fr 1fr;



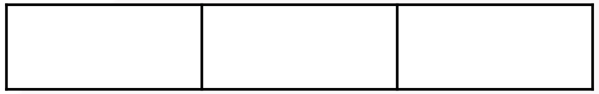
grid-template-columns: 1fr 2fr;



grid-template-columns: 100px 1fr 2fr;

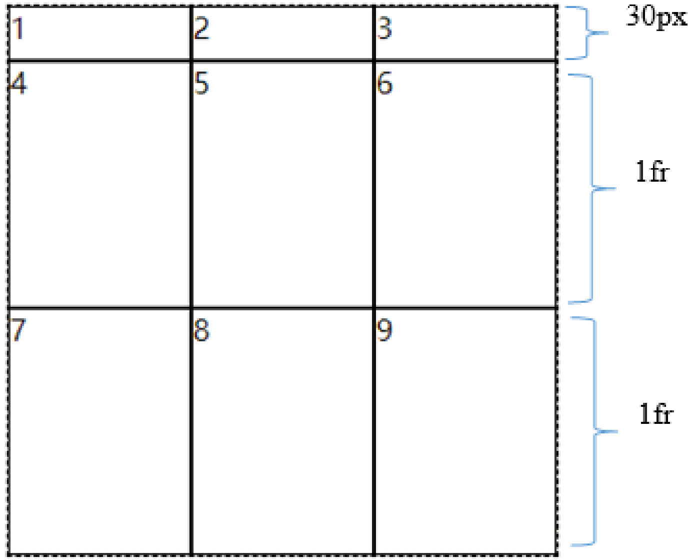


grid-template-columns: repeat(3,1fr);



网格布局练习

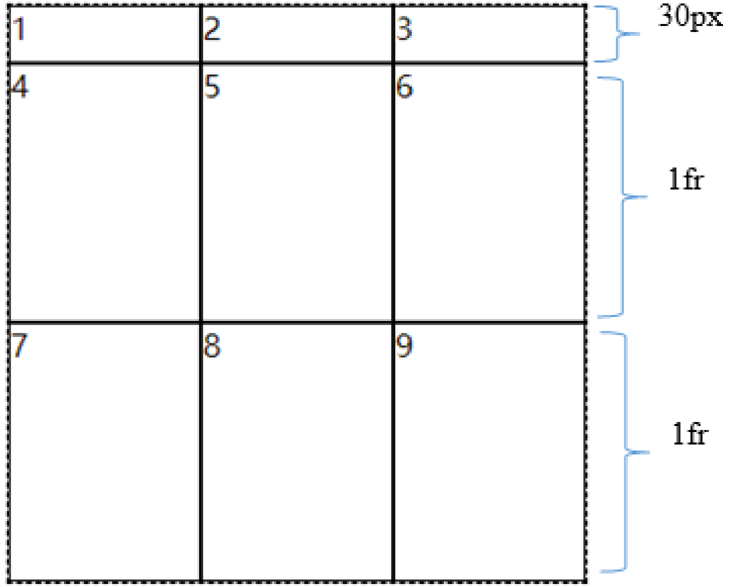
完成如下网格



```
<div class="grid-container">
        <div class="grid-item">1</div>
        ……
        <div   class="grid-item">9</div>
</div>
 .grid-item{
              border: 1px solid;
 }
```

网格布局练习

完成如下网格



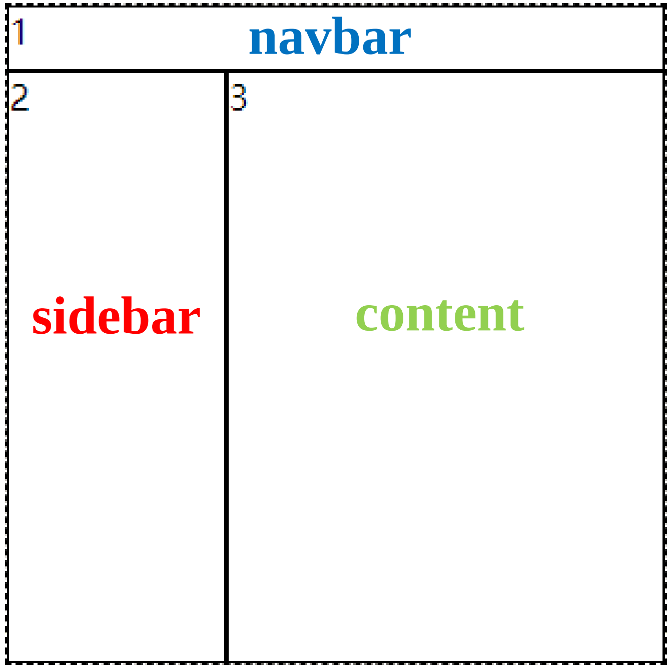
```
.grid-container{
        width: 300px;
        height: 300px;
        border:1px dashed;

        display: grid;
        grid-template-rows:30px 1fr 1fr;
        grid-template-columns:1fr 1fr 1fr;
}
```

网格容器样式

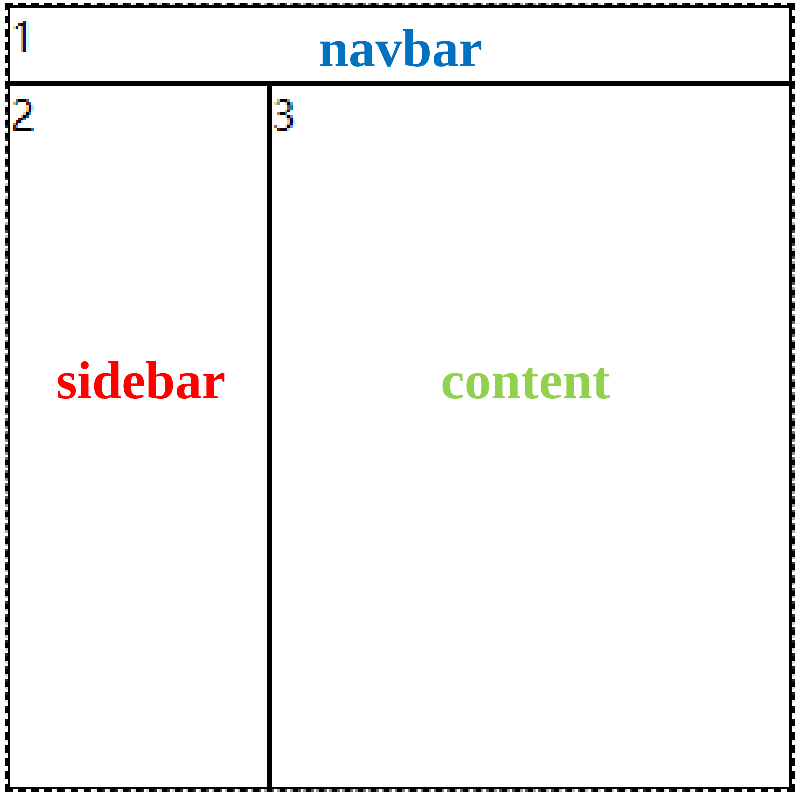
3. grid-template-areas属性

命名单元格，相同名称的单元格被划分为一个区域



```
<div class="grid-container">
	<div class="grid-item">1</div>
	<div class="grid-item">2</div>
	<div class="grid-item">3</div>
</div>
```

网格容器样式



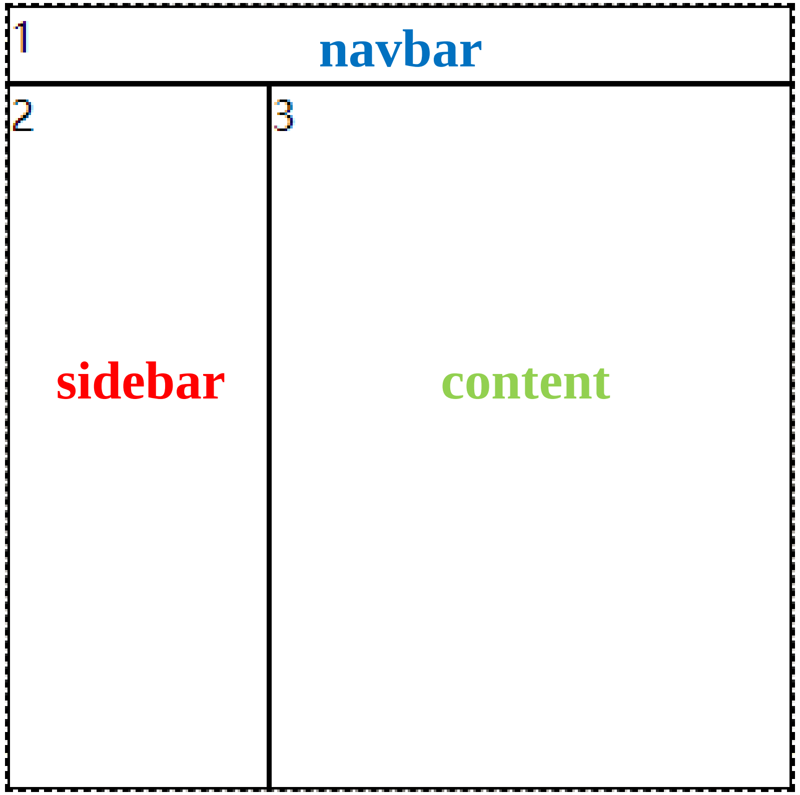
1 navbar

2 sidebar

3 content

```
.grid-container{
        display: grid;
        grid-template-rows:30px 1fr 1fr;
        grid-template-columns:1fr 1fr 1fr;
        grid-template-areas:
        "navbar navbar navbar"
        "sidebar content content"
        "sidebar content content ";
        width: 300px;
        height: 300px;
        border:1px dashed;
}
```

网格容器样式



1 navbar

2 sidebar

3 content

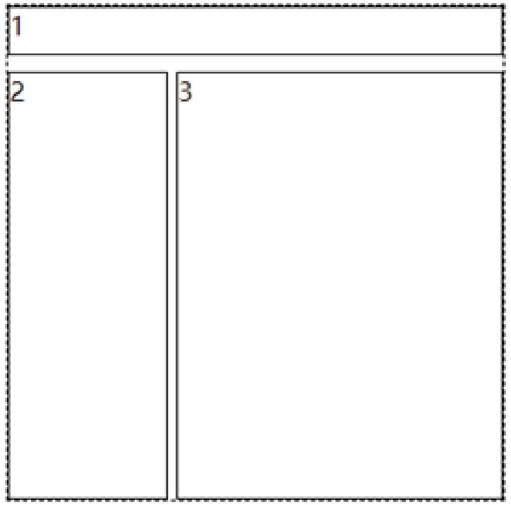
```
.grid-item{
        border: 1px solid;
}
.grid-item:nth-child(1){
        grid-area : navbar;
}
.grid-item:nth-child(2){
        grid-area : sidebar;
}
.grid-item:nth-child(3){
        grid-area : content;
}
```

网格容器样式

4. grid-gap 属性

grid-gap： grid-row-gap grid-column-gap

grid-row-gap:行间距，grid-column-gap：列间距



```
.grid-container{
        grid-row-gap:10px;
        grid-column-gap:5px;
}
```

网格容器样式

5. justify-items、align-items、place-items 属性

（1）justify-items:单元格内容的水平位置

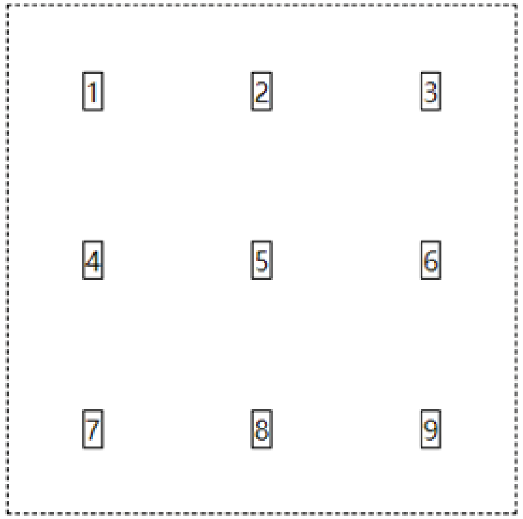
justify-items : start | end | center | 默认stretch;

（2）align-items:单元格内容的垂直位置

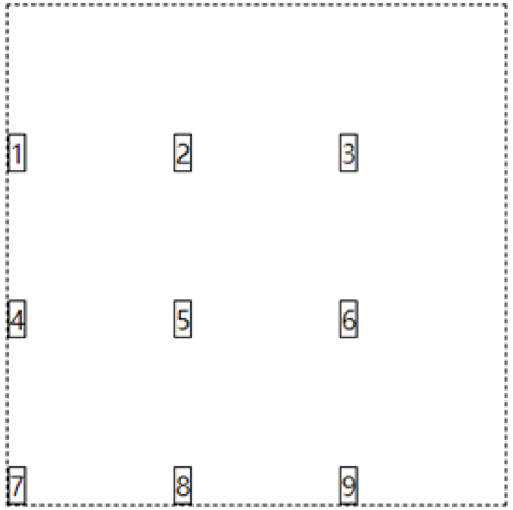
align-items : start | end | center | 默认stretch;

place-items: align-items属性 justify-items属性;

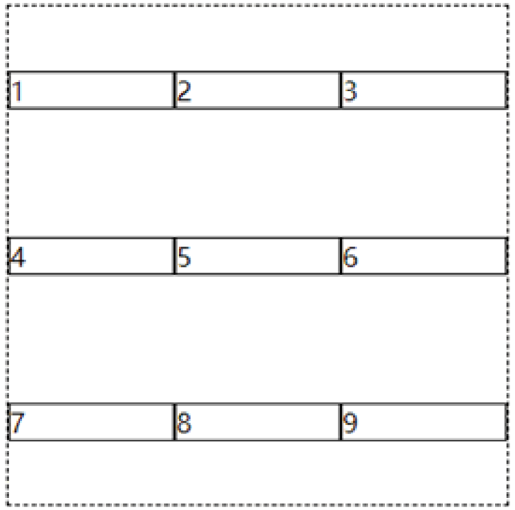
网格容器样式



justify-items:center;
align-items:center;



justify-items:start;
align-items:end;



justify-items:stretch;
align-items:center;

💻 网格容器样式

6. justify-content、align-content、place-content 属性

（1）justify-content:整个内容区域在容器里面的水平位置

justify-content: start | end | center | stretch |
space-around | space-between | space-evenly;

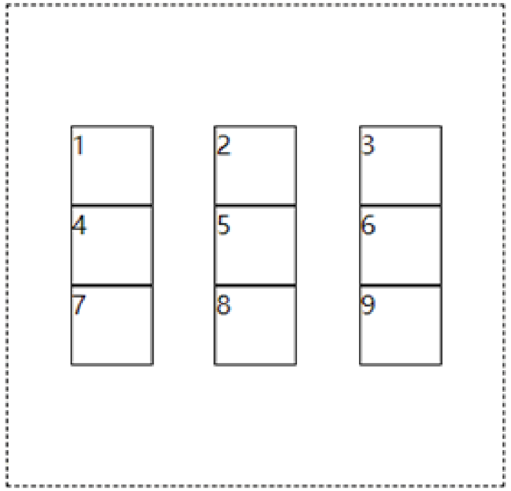
（2）align-content:整个内容区域的垂直位置

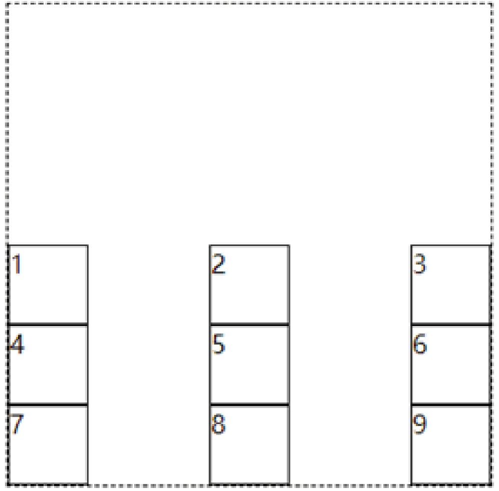
align-content: start | end | center | stretch |
space-around | space-between | space-evenly;

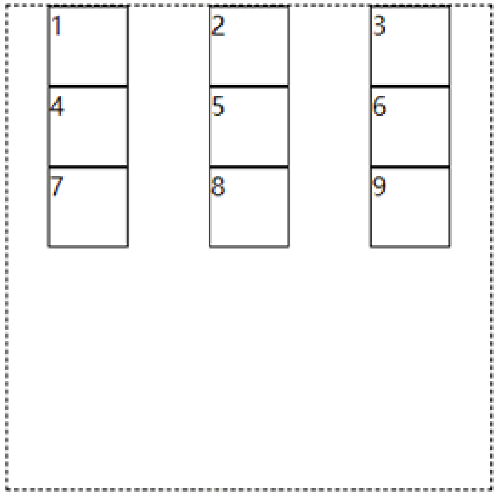
（3）place-content :justify-content属性 align-content属性

网格容器样式

justify-content:space-between;
align-content:end;





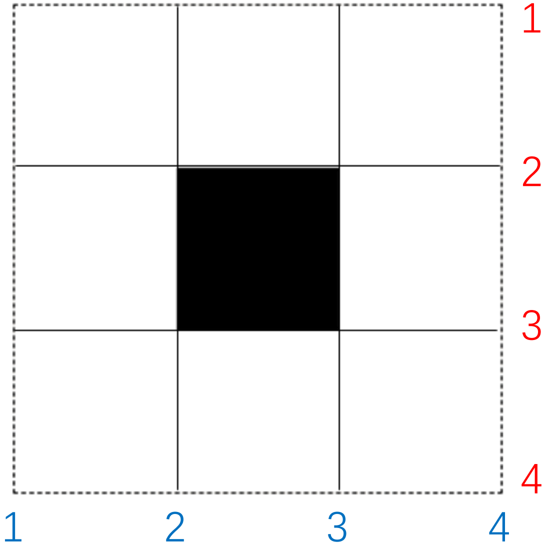


justify-content:space-evenly;
align-content:center;

justify-content:space-around;
align-content:start;

网格元素样式

1.　grid-column-*、grid-row-*属性

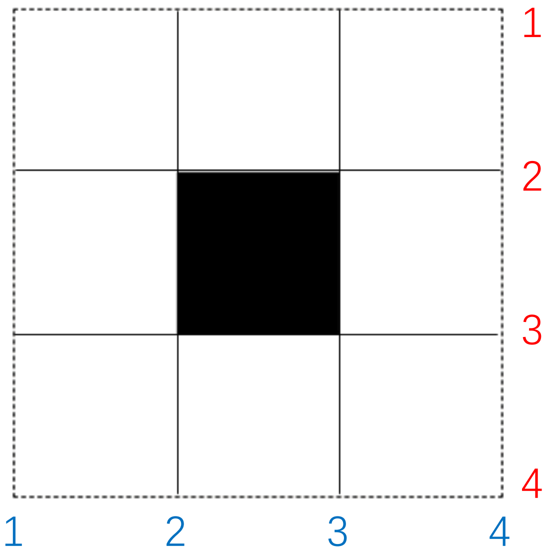


grid-row-start:2;
grid-row-end:3;

grid-column-start:2;
grid-column-end:3;

网格元素样式

1. grid-column-*、grid-row-*属性



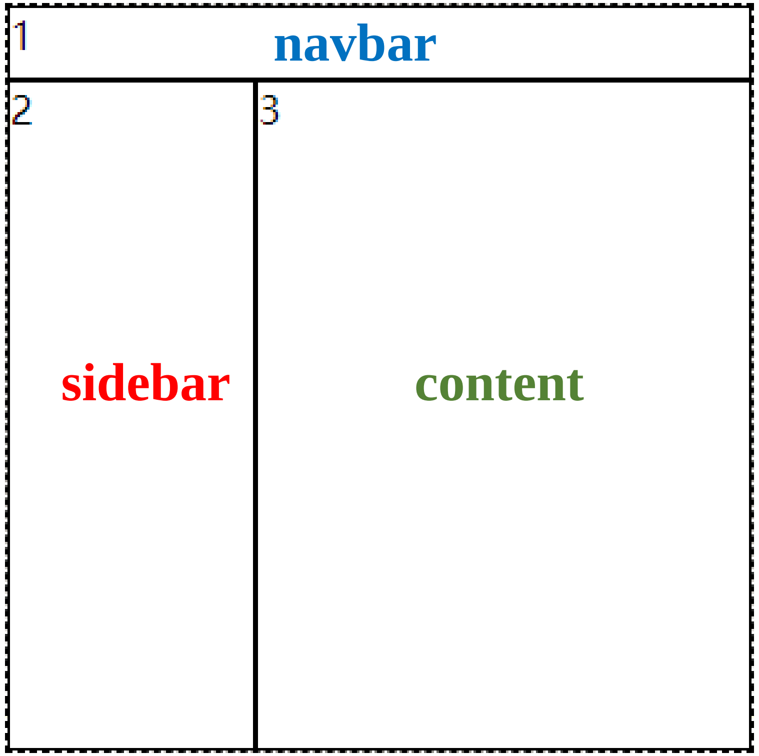
```
.grid-item{
        grid-row-start:2;
        grid-row-end:3;
        grid-column-start:2;
        grid-column-end:3;
 }
```

```
.grid-item{
       grid-row:2/3;
       grid-column:2/3;
  }
```

```
.grid-item{
       grid-area:2/2/3/3;
}
```

网格元素样式

2. grid-area属性



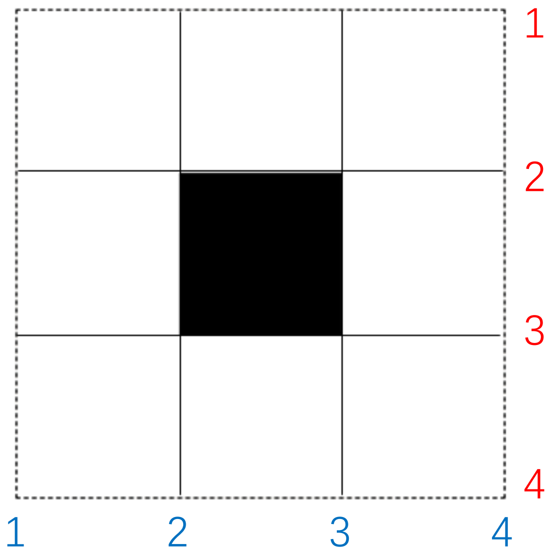
```
.grid-item:nth-child(1){
        grid-area : navbar;
}

.grid-item:nth-child(2){
        grid-area : sidebar;
}

.grid-item:nth-child(3){
        grid-area : content;
}
```

网格元素样式

2. grid-area属性



```
.grid-item{
        grid-area:2/2/3/3;
}
```

网格元素样式

2. justify-self、align-self、place-self 属性

单个网格元素的对齐方式

（1）justify-self属性：单元格内容的水平位置，用法同justify-items属性。

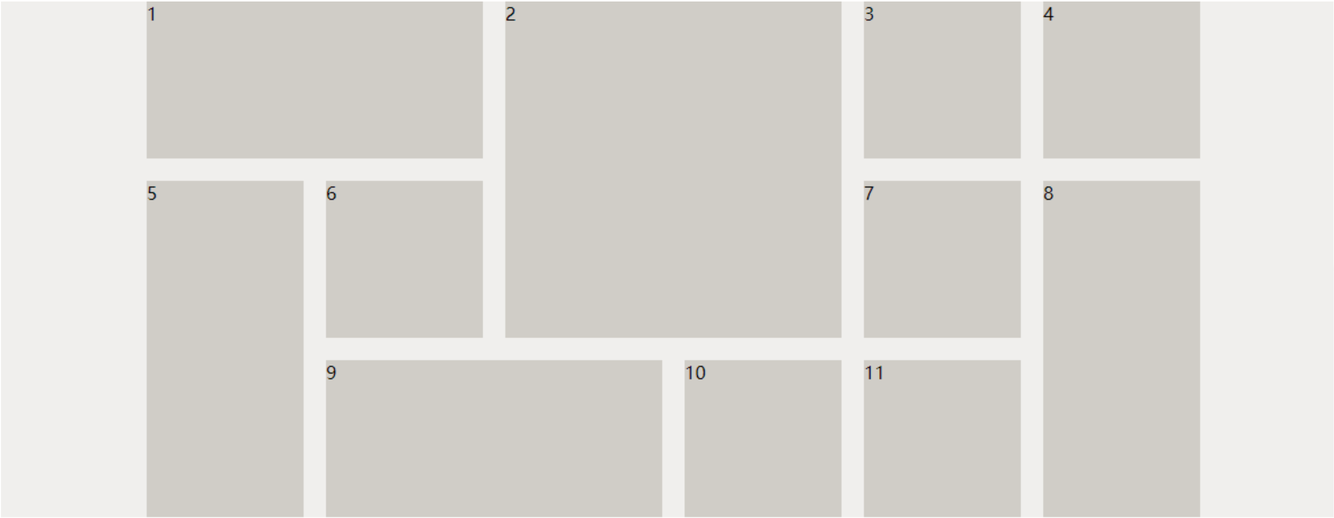
justify-self: start | end | center | stretch;

（2）align-self属性:单元格内容的垂直位置，用法同align-items属性。

align-self: start | end | center | stretch;

place-self: align-self属性 justify-self属性;

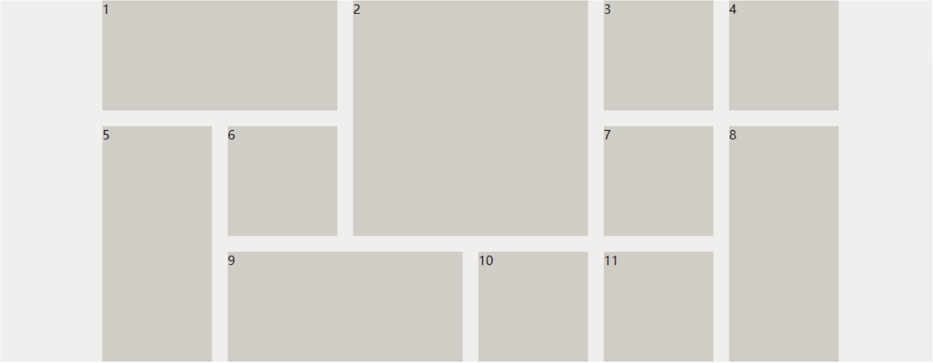
案例



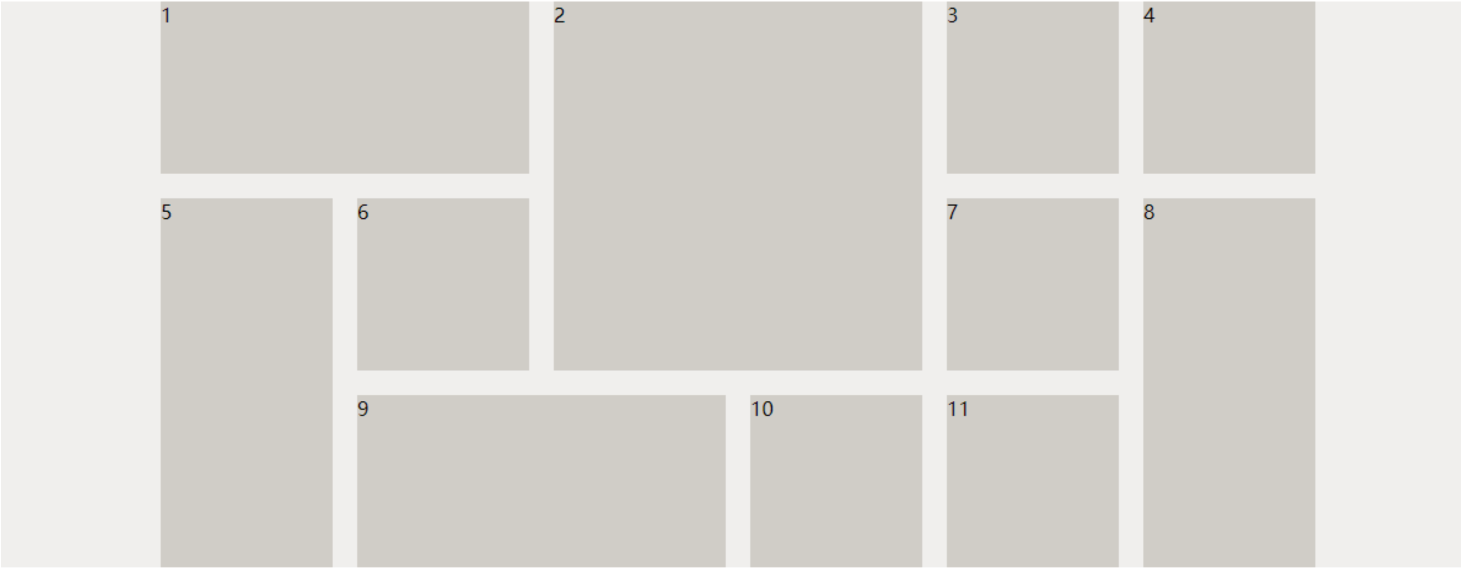
```
<div class="grid-container">
          <div class="grid-item">1</div>
          ……
          <div class="grid-item">11</div>
</div>
```

案例

```
.grid-container{
        display: grid;
        justify-content: center;
        align-content: center;
        grid-template-columns: repeat(6,140px);
        grid-template-rows: repeat(3,140px);
        grid-gap: 20px;
}
```



案例



```
.grid-item:nth-child(1){
	grid-column: 1 / span 2;

	grid-row: 1 / 2;

}
```

```
.grid-item:nth-child(2){
	grid-column: 3 / span 2;

	grid-row: 1 / span 2;

}
```