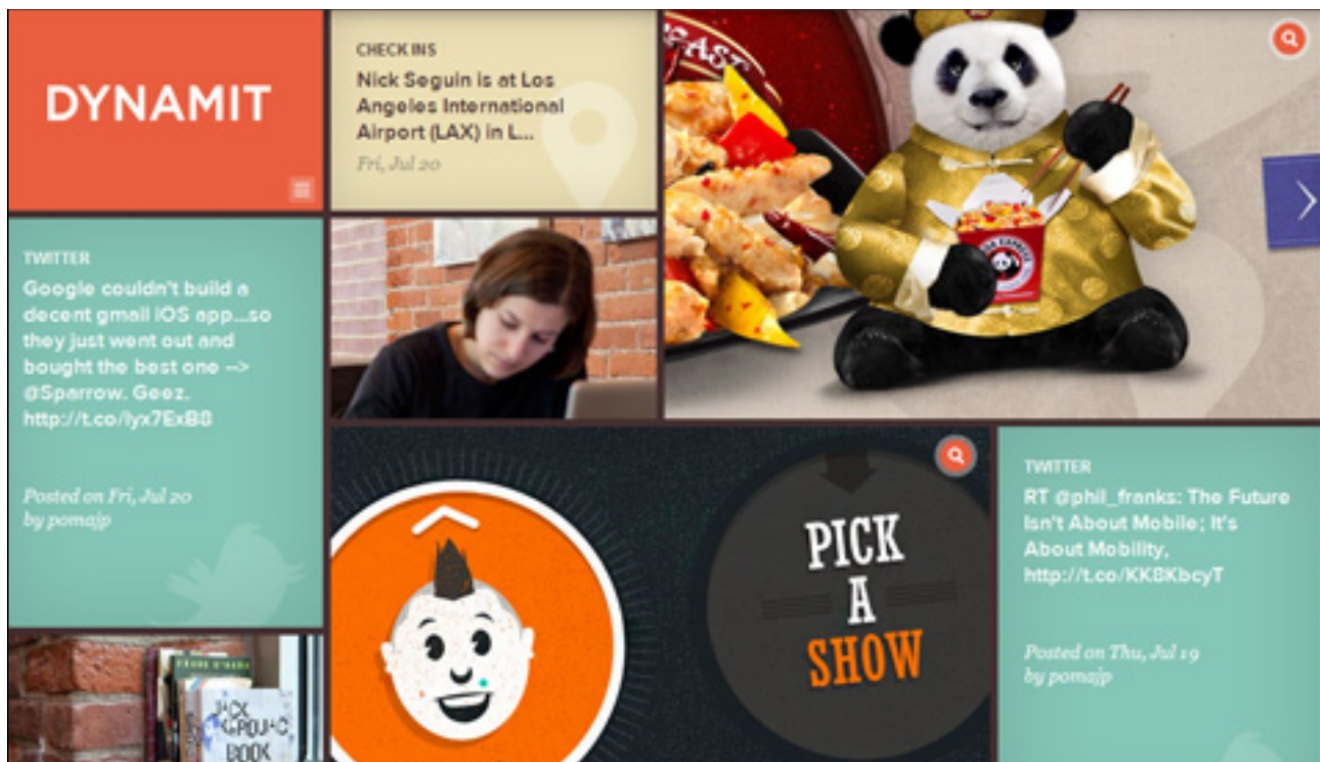


GRID布局

1. 它将网页划分成一个个网格，可以任意组合不同的网格，做出各种各样的布局
- 2.
3. **Grid** 布局与 **Flex** 布局有一定的相似性，都可以指定容器内部多个项目的位置。但是，它们也存在重大区别。
4. **Flex** 布局是轴线布局，只能指定“项目”针对轴线的位置，可以看作是一维布局。
5. **Grid** 布局则是将容器划分成“行”和“列”，产生单元格，然后指定“项目所在”的单元格，可以看作是二维布局。



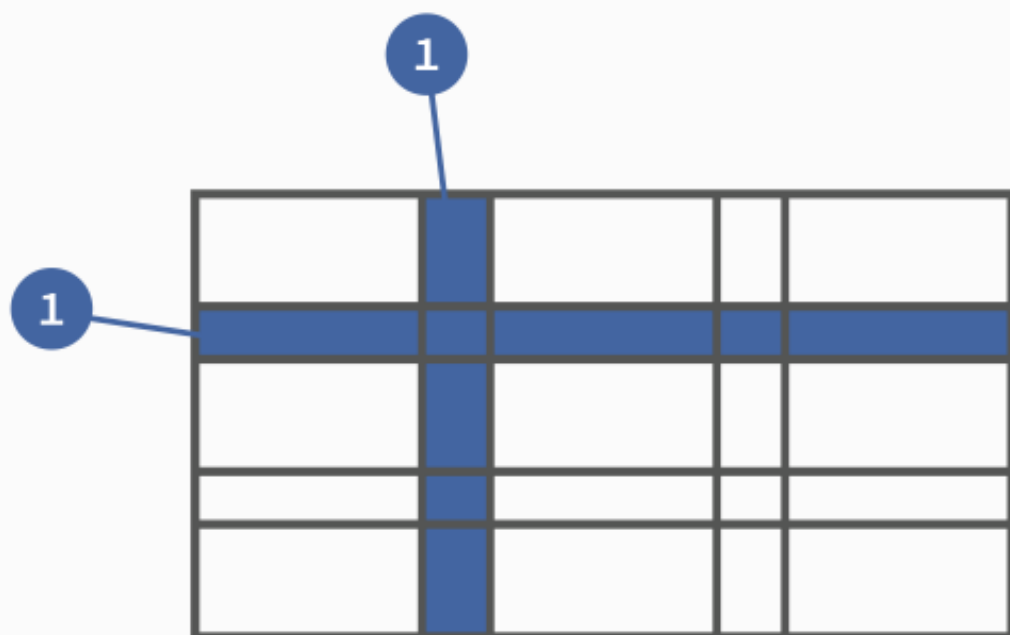
grid布局预热

- 容器和项目

1. `<section>`
2. `<div>`
3. `<p></p>`
4. `</div>`
5. `<div>`
6. `<p></p>`
7. `</div>`

```
8. <div>
9. <p></p>
10. </div>
11. </section>
12.
13. <!-- 以上代码section为容器、div为项目 （项目不包括p标签） -->
```

- 行和列



1. 容器里面的水平区域称为“行”，垂直区域称为“列”。

- 单元格





1. 行和列的交叉区域，称为“单元格”。

- 网格线

1. 划分网格的线，称为“网格线”。水平网格线划分出行，垂直网格线划分出列。

GRID属性

容器属性:添加在容器元素

1. 启动网格布局

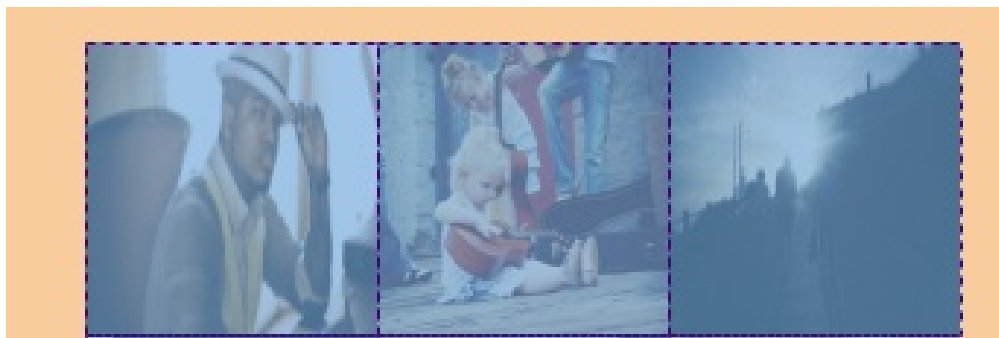
1. `display:grid`
2. `/*`
3. 属性值:
4. `grid` 为块状网格容器（容器自上而下排列）
5. `line-grid` 为内联网格容器（容器横向排列）
6. `*/`

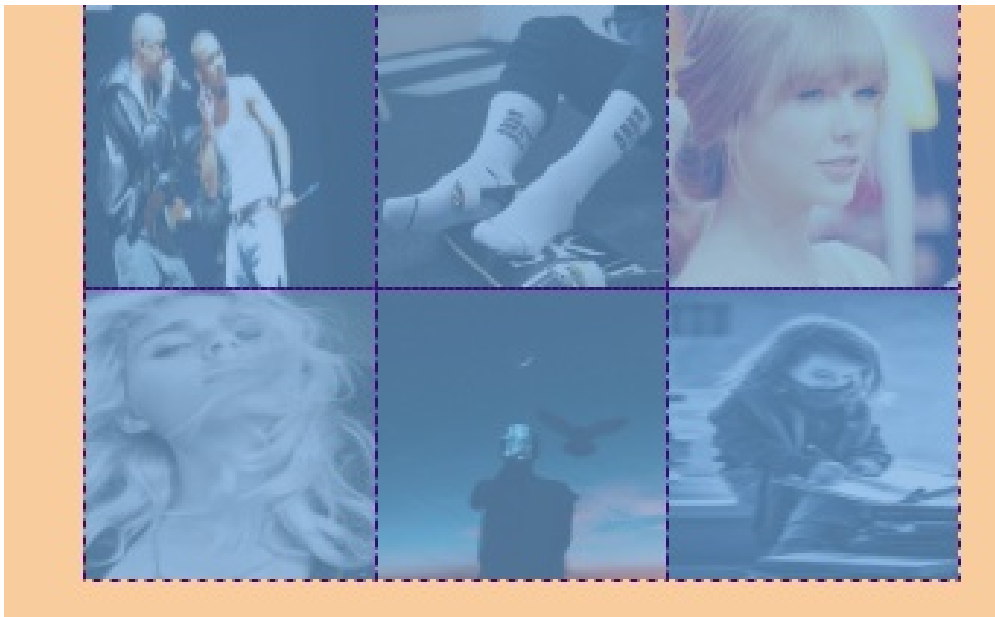
2. 划分行和列

1. `grid-template-columns:`
2. `grid-template-rows:`

- 属性值：绝对大小(根据列数或者行数确定值的个数) 例：200px 200px 200px

1. `grid-template-columns:200px 200px 200px`
2. `grid-template-rows:200px 200px 200px`





- 属性值：百分比（根据列数或者行数确定值的个数） 例：33.33% 33.33% 33.33%

1. `grid-template-columns:33.33% 33.33% 33.33%`
2. `grid-template-rows:33.33% 33.33% 33.33%`



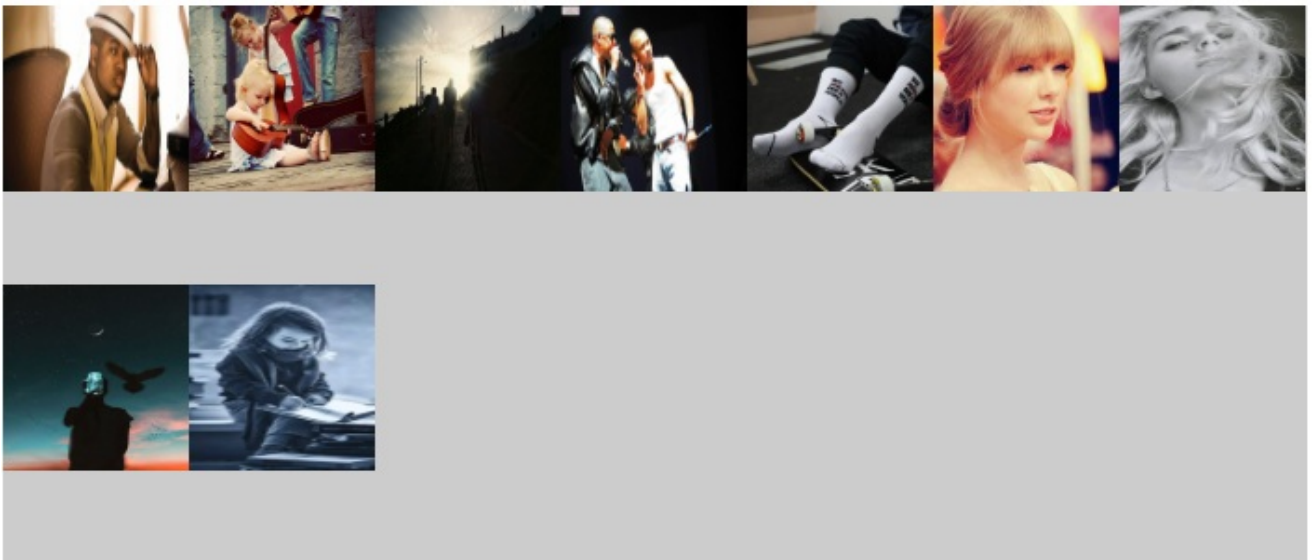
- 功能函数：repeat()

1. `repeat(参数1, 参数2)`

2. `/*`
3. 参数1 : 重复的次数
4. 参数2 : 重复的数值或者重复的模式
- 5.
6. eg:
7. `grid-template-columns: repeat(3, 33.33%);`
8. 等同
9. `grid-template-columns: 33.33% 33.33% 33.33%`
10. `*/`

- auto-fill关键字(自动填充)

1. `grid-template-columns: repeat(auto-fill, 33.33%);`
2. `/*`
3. 当项目宽高固定, 容器不固定的情况下, 自动填充网格列数
4. `*/`



- fr关键字(列宽片段)

1. 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。
1. `grid-template-columns: 1fr 3fr 1fr;`
2. `grid-template-rows: repeat(3, 100px);`





- minmax() 功能函数

1. `grid-template-columns: 1fr 1fr minmax(100px, 1fr);`
2. `/*`
3. `minmax(100px, 1fr)`表示列宽不小于100px，不大于1fr
4. `*/`

- auto 关键字 自动填充

1. `grid-template-columns: 30px auto 20px;`
2. `grid-template-rows: repeat(3, 100px);`



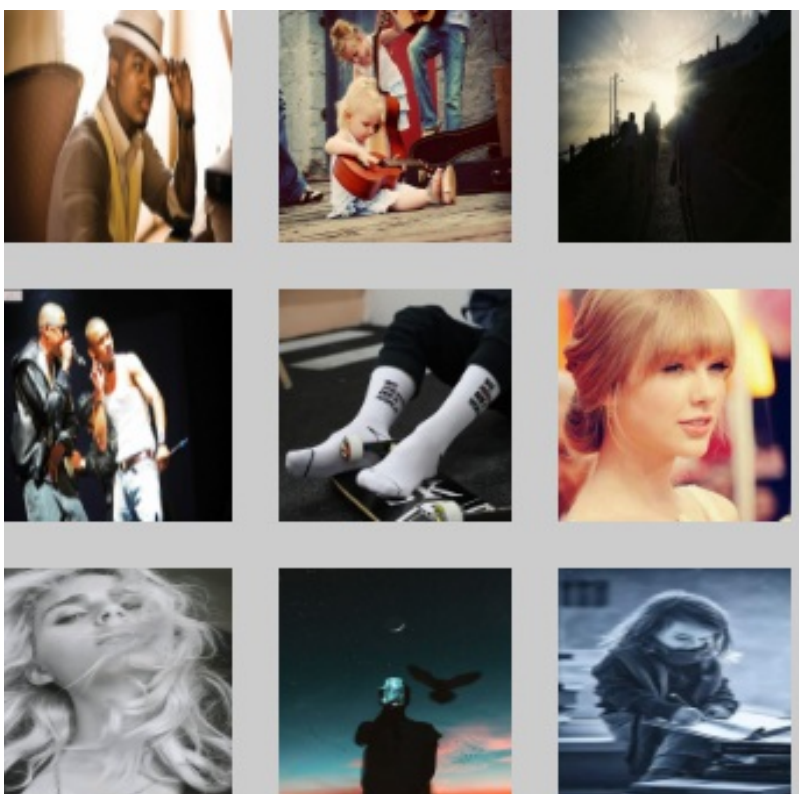


- 网格线名称

1. `grid-template-columns: [c1] 100px [c2] 100px [c3] auto [c4];`
2. `grid-template-rows: [r1] 100px [r2] 100px [r3] auto [r4];`
3. `/*`
4. 指定网格布局为3行x3列，因此有4根垂直网格线和4根水平网格线。方括号里面依次是这八根线的名字。
5. `*/`

1. 设置行间距 || 列间距

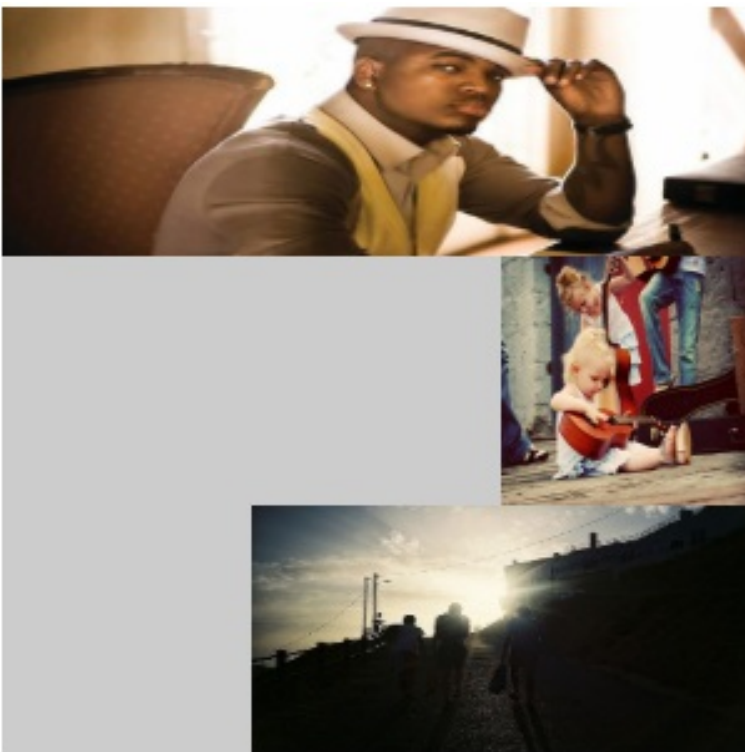
1. `grid-row-gap:20px /* 行间距 */`
2. `grid-column-gap:20px /* 列间距 */`
3. `grid-gap:30px 30px /* 复合式写法 */`
- 4.
5. `/*注：新版本已经省略 grid- 前缀 row-gap \ column-gap \ gap*/`



1. 指定”区域”（area），一个区域由单个或多个单元格组成。

1. `display: grid;`

```
1. display: grid;
2. grid-template-columns: 100px 100px 100px;
3. grid-template-rows: 100px 100px 100px;
4. grid-template-areas: 'a b c'
5.   'd e f'
6.   'g h i';
7.
8. /*
9.   将整个网格容器分为9个区域，每个区域对应一个单元格
10.  通过grid-area 指定项目名称。
11. */
1. display: grid;
2. grid-template-columns: repeat(3, 100px);
3. grid-template-rows: repeat(3, 100px);
4. grid-template-areas: 'a a a'
5.   '. . b'
6.   '. c c';
```



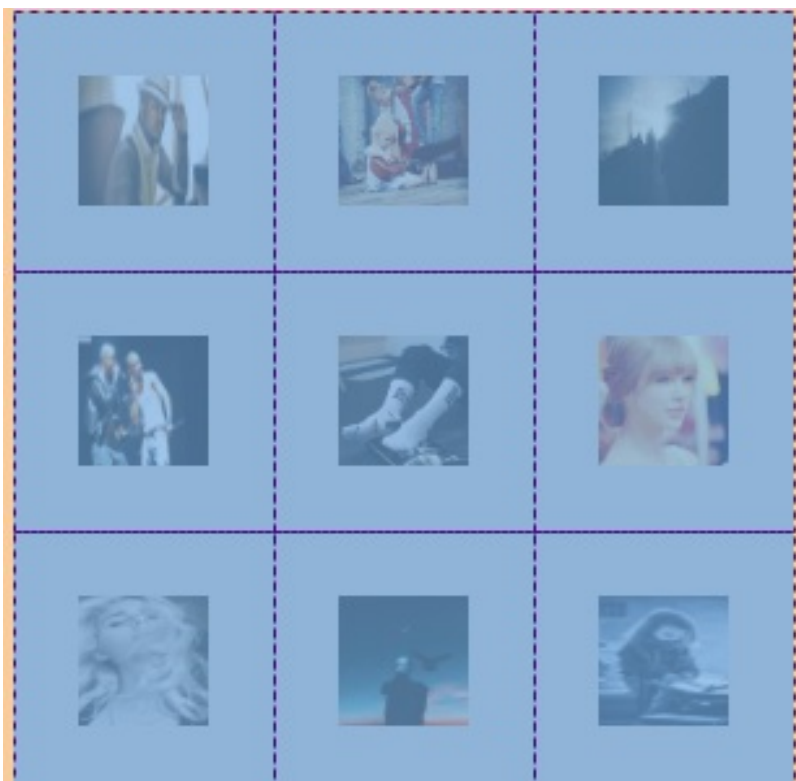
1. 规划子元素放置的顺序（默认为先排行后排列）

```
1. grid-auto-flow: column | row;
2. /*
3.   row dense和column dense。
```


4. 这两个值主要用于，某些项目指定位置以后，剩下的项目怎么自动放置。
5. */

2. 设置单元格内容在单元格内 水平 | 垂直 的对齐方式

1. justify-items: start | end | center | stretch;
2. align-items: start | end | center | stretch;
3. place-items: <justify-items> <align-items> /*复合式写法*/
4. /*
5. start: 对齐单元格的起始边缘。
6. end: 对齐单元格的结束边缘。
7. center: 单元格内部居中。
8. stretch: 拉伸，占满单元格的整个宽度（默认值）。
9. */



1. 设置整个内容区域在容器里面的水平 | 垂直 对齐方式

1. justify-content: start | end | center | stretch | space-around | space-between | space-evenly;
2. align-content: start | end | center | stretch | space-around | space-between | space-evenly;
3. place-content: <justify-content> <align-content> /* 复合式写法 */
- 4.
5. /*

5. `/*`
6. `start` - 对齐容器的起始边框。
7. `end` - 对齐容器的结束边框。
8. `center` - 容器内部居中。
9. `stretch` - 项目大小没有指定时，拉伸占据整个网格容器。
10. `space-around` - 每个项目两侧的间隔相等。所以，项目之间的间隔比项目与容器边框的间隔大一倍。
11. `space-between` - 项目与项目的间隔相等，项目与容器边框之间没有间隔。
12. `pace-evenly` - 项目与项目的间隔相等，项目与容器边框之间也是同样长度的间隔。
13. `*/`



项目属性：添加在子元素上面

1. 指定项目的四个边框，分别定位在哪根网格线。

```
1.  grid-column-start
2.  grid-column-end
3.  grid-row-start
4.  grid-row-end
5.
6.  /*
7.    grid-column-start属性：左边框所在的垂直网格线
8.    grid-column-end属性：右边框所在的垂直网格线
9.    grid-row-start属性：上边框所在的水平网格线
10.   grid-row-end属性：下边框所在的水平网格线
11.  */
1.  grid-column-start:1;
2.  grid-column-end:3;
3.  grid-row-start: 2;
4.  grid-row-end:4;
```



1. grid-column , grid-row

1. `grid-column`属性是`grid-column-start`和`grid-column-end`的合并简写形式
2. `grid-row`属性是`grid-row-start`属性和`grid-row-end`的合并简写形式。
1. `grid-column: 1 / 3;`
2. `grid-row: 1 / 2;`