# css学习第一部分，查看：[css学习笔记1](学习笔记05-css学习1.docx)

# css学习第二部分，查看：[css学习笔记2](学习笔记05-css学习2.docx)

# css学习第三部分：参考：[css学习笔记3](学习笔记05-css学习3.docx)

# css学习第四部分

# css网格系统：display：grid；

## css显示网格可以创建一些比较复制的界面，这些界面用flex-box很难做到，如

|  |
| --- |
|  |

## Grid应用实例

|  |
| --- |
|  |

## Grid vs Flex

## 如果你要把你的子元素排除一行或者一列，就可以使用flexbox。如果是2维的，就使用grid

|  |
| --- |
|  |

## 其实，你也可以把他们组合起来用

|  |
| --- |
|  |
|  |

## 这里有一个老师制作的网站，可以练习练习

### [CSS Grid vs Flexbox (appbrewery.github.io)](https://appbrewery.github.io/grid-vs-flexbox/)

## 效果对比：

### 你可以用grid和flexbox分别创建一行元素，当窗口足够大时是这样子的

|  |
| --- |
|  |

### 然后我们慢慢缩小窗口宽度，很难发现，当窗口宽度不能够容纳所有元素在一行时，grid会自动换行，并且元素不收缩，flexbox的元素会收缩来适应窗口宽度

|  |
| --- |
|  |

### Grid的对齐是非常严格，非常整齐，flexbox很难做到grid的对齐效果

|  |
| --- |
|  |

## CSS 网格属性

|  |  |
| --- | --- |
| **属性** | **描述** |
| [column-gap](https://www.runoob.com/cssref/css3-pr-column-gap.html) | 指定列之间的间隙 |
| [gap](https://www.runoob.com/cssref/css3-pr-gap.html) | row-gap 和 column-gap 的简写属性 |
| [grid](https://www.runoob.com/cssref/css-pr-grid.html) | grid-template-rows, grid-template-columns, grid-template-areas, grid-auto-rows, grid-auto-columns, 以及 grid-auto-flow 的简写属性 |
| [grid-area](https://www.runoob.com/cssref/pr-grid-area.html) | 指定网格元素的名称，或者也可以是 grid-row-start, grid-column-start, grid-row-end, 和 grid-column-end 的简写属性 |
| [grid-auto-columns](https://www.runoob.com/cssref/pr-grid-auto-columns.html) | 指的默认的列尺寸 |
| [grid-auto-flow](https://www.runoob.com/cssref/pr-grid-auto-flow.html) | 指定自动布局算法怎样运作，精确指定在网格中被自动布局的元素怎样排列。 |
| [grid-auto-rows](https://www.runoob.com/cssref/pr-grid-auto-rows.html) | 指的默认的行尺寸 |
| [grid-column](https://www.runoob.com/cssref/pr-grid-column.html) | grid-column-start 和 grid-column-end 的简写属性 |
| [grid-column-end](https://www.runoob.com/cssref/pr-grid-column-end.html) | 指定网格元素列的结束位置 |
| [grid-column-gap](https://www.runoob.com/cssref/pr-grid-column-gap.html) | 指定网格元素的间距大小 |
| [grid-column-start](https://www.runoob.com/cssref/pr-grid-column-start.html) | 指定网格元素列的开始位置 |
| [grid-gap](https://www.runoob.com/cssref/pr-grid-gap.html) | grid-row-gap 和 grid-column-gap 的简写属性 |
| [grid-row](https://www.runoob.com/cssref/pr-grid-row.html) | grid-row-start 和 grid-row-end 的简写属性 |
| [grid-row-end](https://www.runoob.com/cssref/pr-grid-row-end.html) | 指定网格元素行的结束位置 |
| [grid-row-gap](https://www.runoob.com/cssref/pr-grid-row-gap.html) | 指定网格元素的行间距 |
| [grid-row-start](https://www.runoob.com/cssref/pr-grid-row-start.html) | 指定网格元素行的开始位置 |
| [grid-template](https://www.runoob.com/cssref/pr-grid-template.html) | grid-template-rows, grid-template-columns 和 grid-areas 的简写属性 |
| [grid-template-areas](https://www.runoob.com/cssref/pr-grid-template-areas.html) | 指定如何显示行和列，使用命名的网格元素 |
| [grid-template-columns](https://www.runoob.com/cssref/pr-grid-template-columns.html) | 指定列的大小，以及网格布局中设置列的数量 |
| [grid-template-rows](https://www.runoob.com/cssref/pr-grid-template-rows.html) | 指定网格布局中行的大小 |
| [row-gap](https://www.runoob.com/cssref/css3-pr-row-gap.html) | 指定两个行之间的间距 |

## 网格实例

|  |
| --- |
|  |

## 实例，制作棋盘

### Index.html内容如下

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>    <title>Chessboard</title>    <style>      /\* Write your code here \*/      /\* Each square should be 100px by 100px \*/      /\* My colours for white: #f0d9b5, black: #b58863 \*/      .container{        width: 800px;        height: 800px;        margin: 0 auto; /\*这个设置是前提是块级元素，并且设置宽高为px百分比是不行的 \*/        display:grid;        grid-template-columns: 1fr 1fr 1fr 1fr 1fr 1fr 1fr 1fr;        grid-template-rows: 1fr 1fr 1fr 1fr 1fr 1fr 1fr 1fr;      }        .white{        width: 100px;        height: 100px;        background-color: #f0d9b5;      }      .black{        width: 100px;        height: 100px;        background-color: #b58863;      }    </style>  </head>  <body>    <div class="container">      <div class="white"></div>      <div class="black"></div>      <div class="white"></div>      <div class="black"></div>      <div class="white"></div>      <div class="black"></div>      <div class="white"></div>      <div class="black"></div>      <div class="black"></div>      <div class="white"></div>      <div class="black"></div>      <div class="white"></div>      <div class="black"></div>      <div class="white"></div>      <div class="black"></div>      <div class="white"></div>      <div class="white"></div>      <div class="black"></div>      <div class="white"></div>      <div class="black"></div>      <div class="white"></div>      <div class="black"></div>      <div class="white"></div>      <div class="black"></div>      <div class="black"></div>      <div class="white"></div>      <div class="black"></div>      <div class="white"></div>      <div class="black"></div>      <div class="white"></div>      <div class="black"></div>      <div class="white"></div>      <div class="white"></div>      <div class="black"></div>      <div class="white"></div>      <div class="black"></div>      <div class="white"></div>      <div class="black"></div>      <div class="white"></div>      <div class="black"></div>      <div class="black"></div>      <div class="white"></div>      <div class="black"></div>      <div class="white"></div>      <div class="black"></div>      <div class="white"></div>      <div class="black"></div>      <div class="white"></div>      <div class="white"></div>      <div class="black"></div>      <div class="white"></div>      <div class="black"></div>      <div class="white"></div>      <div class="black"></div>      <div class="white"></div>      <div class="black"></div>      <div class="black"></div>      <div class="white"></div>      <div class="black"></div>      <div class="white"></div>      <div class="black"></div>      <div class="white"></div>      <div class="black"></div>      <div class="white"></div>    </div>  </body>  </html> |

### 效果:

|  |
| --- |
|  |

### 注意，必须给网格容器设置固定宽度，否则当窗口变宽，会有空白把元素隔开，那就破坏了棋盘布局

## 网格元素宽高设置

### 网格容器设置元素宽度有下面的方法，最后一个是小测验

|  |
| --- |
|  |

#### 可以去这个网址练习：<https://appbrewery.github.io/grid-sizing/>

### 1.用固定数字设置，如图

|  |
| --- |
|  |

#### 注意：这种设置是写死的，不是响应式的，如果你改为rem，他还是写死的

### 上面的设置可以简写为如下，/左边的是行宽

|  |
| --- |
|  |

#### 注意：不推荐这种写法

### 2.固定+auto

|  |
| --- |
|  |

### 注意：这种写法，auto部分有响应式功能，列会占用浏览器窗口减去固定宽度的全部，但是row是内容高度度，这个坑需要注意

### 3.fractional size，

#### 这个设置会根据fr的大小来设置元素宽高占比，然后按照这个比例来瓜分容器宽高

|  |
| --- |
|  |

#### 这个是响应式的

### 4.minmax方式

|  |
| --- |
|  |

#### 这个其实很有用，相当于第一列固定，第二列可以有响应式效果，侧栏不会变，内容可以变化

### 5，repeat方式，可以把指定数量的元素设置为同一宽度，格式如下：

|  |
| --- |
|  |

#### 这是特殊案例

|  |
| --- |
|  |

#### 当元素总数量不够，他会先满足能够满足的行，把剩余的放到另外一行

#### 还有一种特殊情况就是元素总个数比你设置的要多，有点丑

|  |
| --- |
|  |

#### 然后你可以这么做

|  |
| --- |
|  |

## 网格容器子元素放置：grid placement

### justify-items：水平方向放置，如左中右

### align-items: 垂直方向放置

### place-items：垂直方向，水平方向，相当于上面两个的组合

可以取下面这些值。

* start：对齐单元格的起始边缘。
* end：对齐单元格的结束边缘。
* center：单元格内部居中。
* stretch：拉伸，占满单元格的整个宽度（默认值）。

### justify-content 属性，是整个内容区域在容器里面的水平位置（左中右） align-content 属性，是整个内容区域的垂直位置（上中下） place-content 属性：上面两个的组合

#### 可以取这些值：

**start | end | center | stretch | space-around | space-between | space-evenly;**

### grid子元素样式设置，需要设置到每一个组元素上面

### 实例：实现下面的目标

|  |
| --- |
|  |

### 设置子元素占2列，注意如果总列数不够用，这个设置没有作用，变换换行

|  |
| --- |
|  |

### 注意，此时如果需要第二个元素也占用2列，你必须这么写

|  |
| --- |
|  |

### 还有一种写法不需要设置order

|  |
| --- |
|  |

### 这种写法有一种简写：

|  |
| --- |
|  |

#### 注意格式：row-start/column-start/row-end/column-end,如图

|  |
| --- |
|  |

### 设置子元素站2行，有2种写法

|  |
| --- |
|  |

### 网格布局和flex布局有一个很大的不同就是它允许元素重叠

|  |
| --- |
|  |

## 可以到这个网站练习网格布局：

## [Grid Garden - A game for learning CSS grid (cssgridgarden.com)](https://cssgridgarden.com/)