|  |  |  |  |
| --- | --- | --- | --- |
| 主容器主轴的排列方向和换行 | flex-direction | column-reverse | 主轴为水平方向，起点在右端 |
| column | 主轴为垂直方向，起点在上沿，并且支持换行 |
| row(default) | 主轴为水平方向，起点在左端, 从左到右排列，并且子元素的宽度会自动伸缩铺满整个容器的宽度 |
| row-reverse | 主轴为水平方向，起点在右端, 从右到左排列，并且子元素的宽度会自动伸缩铺满整个容器的宽度 |
| flex-wrap | nowrap (default) | 不换行 |
| wrap | 换行，第一行在上方 |
| wrap-reverse | 换行，第一行在下方 |
| flex-flow | 是flex-direction属性和flex-wrap属性的简写形式, 默认值为row nowrap | |
| 伸缩项目在主轴上的对齐方式 | justify-content | flex-start (default) | 左对齐 |
| flex-end | 右对齐 |
| center | 居中 |
| space-between | 两端对齐，项目之间的间隔都相等 |
| 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边框的间隔大一倍 |
|  | |
| 伸缩项目在交叉轴(侧轴)上的对齐方式 | align-items | flex-start | 纵轴的起点对齐 |
| flex-end | 纵轴的终点对齐 |
| center | 纵轴的中点对齐 |
| baseline | 项目的第一行文字的基线对齐 |
| stretch (default) | 如果项目未设置高度或设为auto，将占满整个容器的高度 |
|  | |
| 伸缩项目在多根轴线的对齐方式 | align-content | flex-start | 与纵轴的起点对齐 |
| flex-end | 与纵轴轴的终点对齐 |
| center | 与纵轴轴的中点对齐 |
| space-between | 与纵轴轴两端对齐，轴线之间的间隔平均分布 |
| space-around | 每根轴线两侧的间隔都相等。所以，轴线之间的间隔比轴线与边框的间隔大一倍 |
| stretch(default) | 轴线占满整个交叉轴 |
|  | |
| 元素样式的属性 | Order | 元素(项目)的排列顺序。数值越小，排列越靠前，默认为0 | |
| flex-grow | 元素(项目)的放大比例，如果为0表示不放大。如果所有元素的flex-grow都为1，则平分剩余大小（不是大小相同，是平分剩余的大小） | |
| flex-shrink | 元素(项目)的缩小比例，默认为1，即如果空间不足，该项目将缩小 | |
| flex-basic | 在分配多余空间之前，项目占据的大小（纵向排列时是height，横向排列时是width） | |
| flex | flex-grow, flex-shrink和flex-basis的简写，默认值为0 1 auto, 后两个属性可选 | |
| align-self | 允许单个项目有与其他项目不一样的对齐方式，可覆盖align-items属性。默认值为auto，表示继承父元素的align-items属性，如果没有父元素，则等同于stretch | |

<div class="box">

<span class="item"></span>

</div>

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | .box{  display: flex;  } |  | .box{  display: flex;  justify-content: center;  } |  | .box{  display: flex;  justify-content: flex-end;  } |
|  | .box{  display: flex;  align-item: center;  } |  | .box{  display: flex;  justify-content: center;  align-item: center;  } |  | .box{  display: flex;  justify-content: center;  align-item: flex-end;  } |

|  |  |  |  |
| --- | --- | --- | --- |
|  | .box{  display: flex;  justify-content: flex-end;  align-item: flex-end;  } |  | .box{  display: flex;  justify-content: space-between;  } |
|  | .box{  display: flex;  flex-direction: column;  justify-content: space-between;  } |  | .box{  display: flex;  flex-direction: column;  justify-content: space-between;  align-item: center;  } |
|  | .box{  display: flex;  flex-direction: column;  justify-content: space-between;  align-item: flex-end;  } |  | .box{ display: flex; }  .item:nth-child(2) { align-self: center; } |
|  | .box{  display: flex;  justify-content: space-between;  }  .item:nth-child(2) { align-self: flex-end; } |  | .box{ display: flex; }  .item:nth-child(2) { align-self: center; }  .item:nth-child(3) { align-self: flex-end; } |
|  | .box{  display: flex;  flex-wrap: wrap;  justify-content: flex-end;  align-content: space-between;  } |  | .box{  display: flex;  flex-wrap: wrap;  align-content: space-between;  } |
|  | .box{  display: flex;  flex-direction: column;  flex-wrap: wrap;  align-content: space-between;  } |  | .box{  display: flex;  flex-wrap: wrap;  } |

|  |  |  |
| --- | --- | --- |
|  | <div class="box">  <div class="column">  <span class="item"></span>  <span class="item"></span>  </div>  <div class="column">  <span class="item"></span>  <span class="item"></span>  </div>  </div> | .box {  display: flex;  flex-wrap: wrap;  align-content: space-between;  }  .column {  flex-basis: 100%;  display: flex;  justify-content: space-between;  } |
|  | <div class="box">  <div class="column">  <span class="item"></span>  <span class="item"></span>  <span class="item"></span>  </div>  <div class="column">  <span class="item"></span>  </div>  <div class="column">  <span class="item"></span>  <span class="item"></span>  </div>  </div> | .box {  display: flex;  flex-wrap: wrap;  }  .column {  flex-basis: 100%;  display: flex;  }  .item:nth-child(2) { justify-content: center; }  .item:nth-child(3) { justify-content: space-between; } |

## 网格布局

|  |  |  |
| --- | --- | --- |
| **基本网格布局**  平均分布。在容器里面平均分配空间，跟上面的骰子布局很像，但是需要设置项目的自动缩放 |  | <div class="Grid">  <div class="Grid-cell">...</div>  <div class="Grid-cell">...</div>  <div class="Grid-cell">...</div>  </div>  .Grid { display: flex; }  .Grid-cell { flex: 1; } |
| **百分比布局**  某个网格的宽度为固定的百分比，其余网格平均分配剩余的空间 |  | <div class="Grid">  <div class="Grid-cell u-1of4">...</div>  <div class="Grid-cell">...</div>  <div class="Grid-cell u-1of3">...</div>  </div>  .Grid { display: flex; }  .Grid-cell { flex: 1; }  .Grid-cell.u-full { flex: 0 0 100%; }  .Grid-cell.u-1of2 {: 0 0 50%; }  .Grid-cell.u-1of3 { flex: 0 0 33.3333%; } |
| **圣杯布局**  页面从上到下，分成三个部分：头部（header），躯干（body），尾部（footer）。其中躯干又水平分成三栏，从左到右为：导航、主栏、副栏 |  | <body class="HolyGrail">  <header>...</header>  <div class="HolyGrail-body">  <main class="HolyGrail-content">...</main>  <nav class="HolyGrail-nav">...</nav>  <aside class="HolyGrail-ads">...</aside>  </div>  <footer>...</footer>  </body>  .HolyGrail {  display: flex;  min-height: 100vh;  flex-direction: column;  }  header,footer { flex: 1; }  .HolyGrail-body {  display: flex;  flex: 1;  }  .HolyGrail-content { flex: 1; }  /\*小屏幕，躯干的三栏自动变为垂直叠加\*/  @media (max-width: 768px) {  .HolyGrail-body {  flex-direction: column;  flex: 1;  }  .HolyGrail-nav, .HolyGrail-ads,.HolyGrail-content { flex: auto;}  } |
| **输入框的布局**  在输入框的前方添加提示，后方添加按钮 |  | <div class="InputAddOn">  <span class="InputAddOn-item">...</span>  <input class="InputAddOn-field">  <button class="InputAddOn-item">...</button>  </div>  .InputAddOn {display: flex;}  .InputAddOn-field { flex: 1; } |
| **悬挂式布局**  主栏的左侧或右侧，需要添加一个图片栏 |  | <div class="Media">  <img class="Media-figure" src="" alt="">  <p class="Media-body">...</p>  </div>  .Media {  display: flex;  align-items: flex-start;  }  .Media-figure { margin-right: 1em;}  .Media-body { flex: 1; } |
| **固定的底栏** | 有时，页面内容太少，无法占满一屏的高度，底栏就会抬高到页面的中间。这时可以采用Flex布局，让底栏总是出现在页面的底部  <body class="Site">  <header>...</header>  <main class="Site-content">...</main>  <footer>...</footer>  </body> | .Site {  display: flex;  min-height: 100vh;  flex-direction: column;  }  .Site-content { flex: 1;} |

<http://www.techug.com/post/flex-examples.html>