DAY15 怪异盒子模型和弹性盒子

2023年4月11日 17:28

盒子模型的分类

box-sizing

content-box

标准盒子模型 默认

元素的总宽度=width+左右的padding+左右的border+左右的margin

border-box

怪异盒子模型 IE盒子模型

padding和border不会撑大盒子

元素的总宽度=width+左右的margin

width=内容的宽度+左右的padding+左右的border

弹性盒子

是一种布局方式

主要解决某一个元素中的子元素的布局方式

子元素的大小 排版 顺序都受到父元素的影响和控制

给一个元素设置display: flex,这个元素的儿子们就变成了弹性布局

弹性盒子的四个专有名词

弹性容器

设置了display: flex;的元素

弹性项目

设置了display: flex;的元素的儿子们

主轴

弹性项目的排列方向

主轴是x轴，主轴的起点在左侧，主轴的终点在右侧

主轴是x轴，主轴的起点在右侧，主轴的终点在左侧

主轴是y轴，主轴的起点在顶部，主轴的终点在底部

主轴是y轴，主轴的起点在底部，主轴的终点在顶部

交叉轴

永远与主轴垂直的一个方向

设置元素为弹性容器

display: flex; //让块级元素变成容器

display: inline-flex; //让行内元素变成容器

-添加给父元素的属性

-弹性容器只对子元素生效，与孙子无关，但是可以嵌套使用

-语法默认值

项目主轴方向是水平从左到右排列

项目的总宽度大于容器的总宽度的时候，会自动缩小以适应

-子元素不受元素类型的影响，按照块元素处理

-弹性布局的目的，就是灵活的控制子元素的水平或者垂直对齐方向

-所以说当元素变成了容器之后，容器和项目所有的改变水平或者垂直对齐方式的属性会失去效果

-只有设置了display:flex,其他的容器属性和项目属性才能生效

容器的属性

控制所有的弹性项目

01.项目在主轴方向上空间不够时是否换行

flex-wrap:

nowrap 默认 不换行

wrap 换行

02.主轴方向

flex-direction: row; 默认 水平从左到右

flex-direction: row-reverse; 水平从右到左

flex-direction: column; 垂直从上到下

flex-direction: column-reverse; 垂直从下到上

03.换行和主轴方向的简写

flex-flow: wrap column;

04.设置项目在主轴上的排列方式

justify-content: flex-start; 默认 主轴起点对齐

justify-content: flex-end; 主轴终点对齐

justify-content: center; 主轴中心对齐

justify-content: space-around; 容器两边有间隙

justify-content: space-between; 容器两边没有间隙

05.设置项目在交叉轴上的排列方式

align-items: stretch; 默认 拉伸以适应交叉轴的空间

align-items: flex-start; 交叉轴起点对齐

align-items: flex-end; 交叉轴终点对齐

align-items: center; 交叉轴中心对齐

align-items: baseline; 交叉轴基线对齐 了解

06.多行项目在交叉轴上的排列方式

align-content: stretch; 默认 拉伸以适应交叉轴的空间

align-content: flex-start; 交叉轴起点对齐

align-content: flex-end; 交叉轴终点对齐

align-content: center; 交叉轴中心对齐

align-content: space-around; 容器两边有间隙

align-content: space-between; 容器两边没有间隙

元素水平垂直居中

不需要知道父子元素的宽高

```
父元素{
  display: flex;
  justify-content: center;
  align-items: center;
  }
```

项目的属性

控制某一个弹性项目

01.设置单个项目在交叉轴上的排列方式

align-self:

auto 默认 继承父元素的align-items的属性值;

如果父元素没有设置align-items，就会找自己的align-self的默认值.

stretch 默认值 拉伸以适应 项目在交叉轴方向上不设置尺寸的时候

flex-start 交叉轴起点对齐

flex-end 交叉轴终点对齐

center 交叉轴中心对齐

baseline 交叉轴基线对齐

02.控制项目如何增长

如果主轴方向上空间盈余，项目们按照取值的占比，自动增长对应的尺寸

flex-grow:

0 默认值 不空配空间 不扩展

数字 1 2 3 4 ...

03.控制项目如何收缩

如果主轴方向上空间不足，项目们按照取值的占比，自动减少对应的尺寸

flex-shrink:

1 默认值 分配空间 收缩

0 不分配空间 不收缩

数字 1 2 3 4...

04.设置项目在主轴方向上的尺寸

flex-basis:

auto 默认

项目在主轴上如果没有设置flex-basis，会参考width;

如果width也没有设置，靠内容撑开.

当width和flex-basis冲突，以flex-basis为准

数值

05.项目属性的简写

flex:flex-grow flex-shrink flex-basis

-顺序不可以换

-平分空间:flex:1

项目的排列顺序

order:

auto 默认值

数值 + - 数值越大，排列顺序越靠后

弹性盒子布局

两栏布局

```
<style>
  * {
    margin: 0;
    padding: 0;
  }
  html,
  body {
    height: 100%;
  }
  body {
    display: flex;
  }
  .left {
    width: 200px;
    height: 100%;
    background-color: slategray;
  }
  .right {
    flex: 1;
    height: 100%;
    background-color: plum;
```

```
    }
  </style>

<body>
  <div class="left"></div>
  <div class="right"></div>
</body>
```

上下两栏布局

```
  <style>
    * {
      margin: 0;
      padding: 0;
    }
    html,
    body {
      height: 100%;
    }
    body {
      display: flex;
      flex-direction: column;
    }
    header {
      width: 100%;
      height: 100px;
      background-color: blue;
    }
    main {
      width: 100%;
      flex: 1;
      background-color: slateblue;
      display: flex;
    }
    .left{
      width: 200px;
      height: 100%;
      background-color: seagreen;
    }
    .right{
      height: 100%;
      flex: 1;
      background-color: palevioletred;
    }
  </style>

<body>
  <header></header>
  <main>
    <div class="left"></div>
    <div class="right"></div>
  </main>
```

```
</body>
```

上中下三栏布局

```
  <style>
    *{
      padding: 0;
      margin: 0;
    }
    html,body{
      height: 100%;
    }
    body{
      display: flex;
      flex-direction: column;
    }
    header{
      width: 100%;
      height: 100px;
      background-color: slateblue;
    }
    main{
      width: 100%;
      flex: 1;
      background-color: sienna;
      display: flex;
    }
    footer{
      width: 100%;
      height: 100px;
      background-color: blueviolet;
    }
    .left{
      width: 200px;
      height: 100%;
      background-color: peru;
    }
    .center{
      height: 100%;
      flex: 1;
      background-color: seagreen;
    }
    .right{
      width: 200px;
      height: 100%;
      background-color: peru;
    }
  </style>

<body>
  <header></header>
  <main>
```

```
    <div class="left"></div>
    <div class="center"></div>
    <div class="right"></div>
  </main>
  <footer></footer>
</body>
```