

DAY11-宽度自适应、伪元素和布局

2023年4月4日 16:32

宽度自适应

1. 一个元素不写宽度或者宽度写100%的时候，宽度自适应父元素
2. 一个脱离文档流的元素(浮动，绝对定位，固定定位)，宽度由内容物决定

高度自适应

1. 一个元素不设置高度或者高度为auto的时候，元素高度由内容物决定
2. 一个元素的高度为300px,当内容物的高度超过300px,元素的高度由内容物决定?
---min-height 最小高度

设置范围

min-height 最小高度

max-height 最大高度

min-width 最小宽度

max-width 最大宽度

伪元素

::before 在元素的最前面添加内容 css2

::after 在元素的最后面添加内容 css2

content 必须写,伪元素要添加的内容

添加了内容之后默认为一个行内元素

::first-letter 第一个字符的样式 css2

::first-line 第一行的样式 css2

::selection 用户选中的内容的样式 css3

只支持背景色和文本颜色

必须是两个::

```
/* 向元素的最前面插入新的内容 */
```

```
div::before {
```

```
/* 1.content必须写 */
/* 2.默认插入的内容是行内元素 */
content: "前面的内容";
width: 200px;
height: 200px;
background-color: blue;
display: block;
font-size: 30px;
color: red;
}

/* 向元素的最后面插入新的内容 */
div:after {
    content: "前面的内容";
    width: 200px;
    height: 200px;
    background-color: blue;
    display: block;
    font-size: 30px;
    color: red;
}

/* 第一个字符的样式 */
p:first-letter {
    color: red;
    background-color: blue;
    font-size: 30px;
}

/* 第一行的样式 */
p:first-line {
    color: red;
    background-color: blue;
    font-size: 30px;
}

/* 用户选中的内容的样式 */
p::selection {
    color: red;
    background-color: blue;
    font-size: 30px;
}
```

visibility: 决定元素是否显示

visible 默认 显示

hidden 隐藏

visibility: hidden;

1. 元素以及里面的内容物全部隐藏
2. 占据页面空间
3. 结果可以逆

visibility: hidden; 和 display:none; 和 opacity:0; 的区别

1. display:none; 不占据空间
2. opacity:0; 和 visibility: hidden; 占据空间
3. display:none; 和 visibility: hidden; 可以逆
4. opacity:0; 不可以逆
5. opacity:0; 有中间值

清除浮动

清除浮动所带来的负面影响----高度塌陷

- 给高度塌陷的盒子加高度

缺点： 违背了高度自适应

- 给高度塌陷的盒子加overflow:hidden

缺点： 溢出部分隐藏

- 给高度塌陷的盒子里面的最后面加一个空的div, 设置clear: both;

缺点： 有很多空的标签

- **万能清除法** -----重置样式表

```
.clearfix::after {  
    content: "";  
    display: block;  
    clear: both;  
    /* 保险 */  
    height: 0;  
    overflow: hidden;  
    visibility: hidden;  
}
```

clear

clear:清除浮动所带来的负面影响

none 默认

left 清除左浮动所带来的负面影响

right 清除右浮动所带来的负面影响

both 清除所有浮动所带来的负面影响

布局

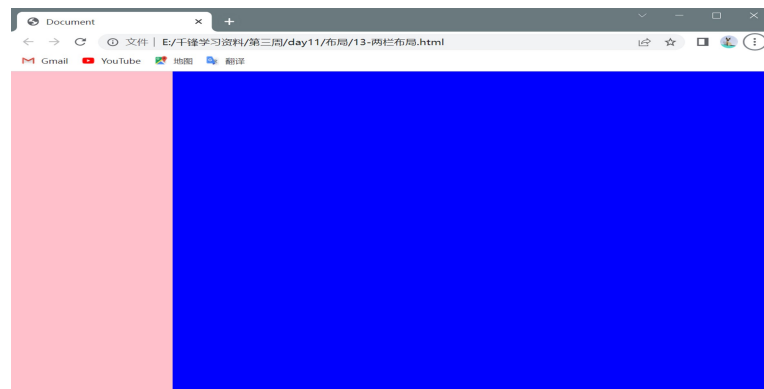
自适应布局

```
html,body{  
  height: 100%;  
}
```

两栏布局

```
/* 1.重置样式 */  
* {  
  padding: 0;  
  margin: 0;  
}  
/* 2.高度自适应 */  
html,  
body {  
  height: 100%;  
}  
/* 3.左边导航栏 */  
.left {  
  width: 200px;  
  height: 100%;  
  background-color: pink;  
  float: left;  
}  
/* 4.右边主要内容区域 */  
.right {  
  width: calc(100% - 200px);  
  height: 100%;  
  background-color: blue;  
  float: right;  
}  
</style>
```

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



</body>

三栏布局

<style>

/* 1.重置样式 */

```
* {  
  padding: 0;  
  margin: 0;  
}
```

/* 2.高度自适应 */

```
html,  
body {  
  height: 100%;  
}
```

/* 3.左边导航栏 */

```
.left {  
  width: 200px;  
  height: 100%;  
  background-color: yellowgreen;  
  float: left;  
}
```

/* 4.中间主要内容区域 */

```
.center {  
  width: calc(100% - 200px - 200px);  
  height: 100%;  
  background-color: pink;  
  float: left;  
}
```

/* 5.右边导航栏 */

```
.right {  
  width: 200px;  
  height: 100%;  
  background-color: blue;  
  float: right;  
}
```

</style>

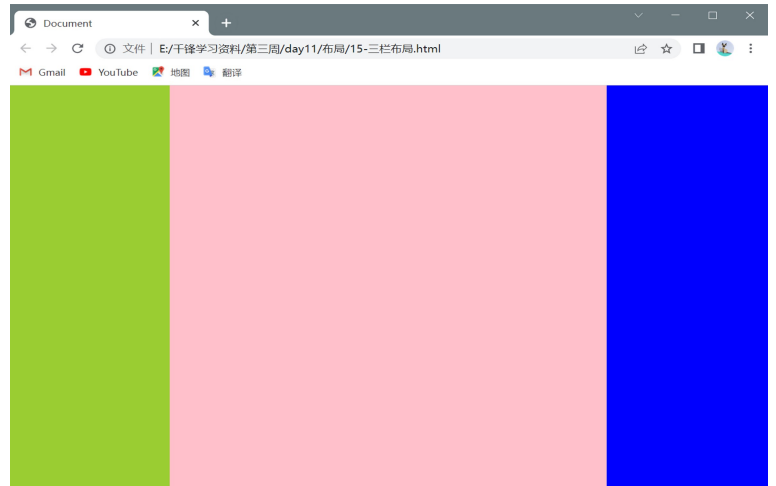
<body>

<div class="left"></div>

<div class="center"></div>

<div class="right"></div>

</body>



上下两栏布局

```
<style>
  /* 1.重置样式 */
  * {
    padding: 0;
    margin: 0;
  }
  /* 2.高度自适应 */
  html,
  body {
    height: 100%;
  }
  /* 3.上下布局 */
  .top {
    height: 100px;
    background-color: pink;
  }
  .bottom {
    height: calc(100% - 100px);
    background-color: cyan;
  }
  /* 4.左右布局 */
  .left {
    width: 200px;
    height: 100%;
    background-color: green;
    float: left;
  }
  .right {
    width: calc(100% - 200px);
    height: 100%;
    background-color: yellow;
    float: right;
  }
</style>
```

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



上中下三栏布局

```
<style>
  /* 1.重置样式 */
  * {
    padding: 0;
    margin: 0;
  }
  /* 2.高度自适应 */
  html,
  body {
    height: 100%;
  }
  /* 3.上中下布局 */
  .top {
    height: 100px;
    background-color: pink;
  }
  .middle {
    height: calc(100% - 100px - 100px);
    background-color: aqua;
  }
  .bottom {
    height: 100px;
    background-color: red;
  }
  /* 4.左中右布局 */
  .left {
    width: 100px;
    height: 100%;
    background-color: yellow;
    float: left;
  }
  .center {
    width: calc(100% - 100px - 100px);
    height: 100%;
    background-color: orange;
    float: left;
  }
  .right {
    width: 100px;
    height: 100%;
    background-color: green;
    float: right;
  }
</style>
```



```
<body>
  <div class="top"></div>
  <div class="middle">
    <div class="left"></div>
    <div class="center"></div>
    <div class="right"></div>
  </div>
  <div class="bottom"></div>
</body>
```

一个元素高度设置了百分比，父元素必须设置高度

计算属性

用于动态计算数值

`calc(100% - 200px)` 运算符号两边必须有空格

```
<style>
  /* 1.重置样式 */
  * {
    padding: 0;
    margin: 0;
  }
  /* 2.高度自适应 */
  html,
  body {
    height: 100%;
  }
  /* 3.左边导航栏 */
  .left {
    width: 200px;
    height: 100%;
    background-color: pink;
    float: left;
  }
  /* 4.右边主要内容区域 */
  .right {
    width: calc(100% - 200px);
    height: 100%;
    background-color: blue;
    float: right;
  }
</style>
```

```
<body>
```



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