

自适应和响应式

为了适配移动端终端设备

实现途径：

1.媒体查询

将设备按照不同的分辨率条件筛选，使符合条件的设备显示对应的样式，从而实现不同分辨率样式不同的效果

语法：

```
@media (max-width: 屏幕最大宽度){  
  
    ... 符合条件的样式 跟style样式表一样 含多个元素样式  
  
}
```

示例：

```
<!DOCTYPE html>  
<html lang="en">  
<head>  
  <meta charset="UTF-8">  
  <meta name="viewport" content="width=device-width, initial-scale=1.0">  
  <title>Document</title>  
  <style>  
    @media (max-width: 960px){ /*屏幕宽度为0~960的样式*/  
      body{  
        background: gold;  
      }  
    }  
    @media (max-width: 760px){ /*屏幕宽度为0~760的样式*/  
      body{  
        background: pink;  
      }  
    }  
    @media (max-width: 640px){ /*屏幕宽度为0~640的样式*/  
      body{  
        background: pink;  
      }  
    }  
    @media (max-width: 520px){ /*屏幕宽度为0~520的样式*/  
      body{  
        background: blue;  
      }  
    }  
    @media (max-width: 480px){ /*屏幕宽度为0~480的样式*/  
      body{
```

```
        background:gray
    }
}
</style>
</head>
<body>

</body>
</html>
```

效果参考b.html

使用媒体查询时，需要对应不同分辨率终端写多套所对应的样式，比较繁琐

2.百分比

通过‘%’单位来实现响应式效果。浏览器高度和宽度发生变化时，通过百分比单位可以使元素的高和宽随着浏览器的变化而变化，从而实现响应式效果

可以取百分比的属性：

- 定位：top, right, bottom, left
- 盒模型：width,height, margin,padding
- 背景：backgroug-position, backgroud-size
- 文本：text-indent
- 字体：font-size

css 百分比都相对于谁？

结论：css的布局块都是盒子，然后一个盒子的百分比其实是相对于其包含块的，也就是他的父级元素。

- 相对于父级宽度的：
 - width
 - max-width
 - min-width
 - padding
 - margin
 - text-indent
 - left
 - right
 - ...
- 相对于父级高度的：
 - height
 - max-height
 - min-height
 - top
 - bottom
- 相对于主轴长度(在flex弹性盒子布局中)
 - flex-basis
- 相对于继承字号的
 - font-size
- 相对于自身字号的
 - line-height
- 相对于自身宽高的

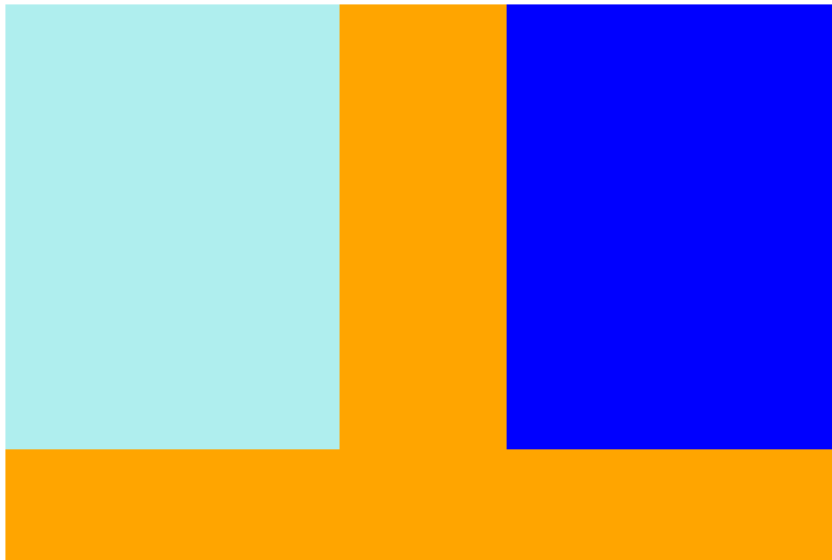
- border-radius 等border相关
- background-size
- ...
- 相对于行高的
 - verticle-align (设置元素垂直对齐方式)

示例：

```
<!DOCTYPE html>
<!-- -->
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
  <style>
    .box{
      width: 900px;
      height: 600px;
      background:orange;
    }
    .item1{
      width: 40%;
      height:80%;
      background: paleturquoise;
      float: left;
    }
    .item2{
      width:40%;
      height: 80%;
      background: blue;
      float: right;
    }
  </style>
</head>

<body>
  <div class="box">
    <div class="item1"></div>
    <div class="item2"></div>
  </div>
</body>
</html>
```



使用百分比做单位时，如果要定义一个元素的宽高，需要根据设计稿手动计算 换算成% 单位，容易出现误差

rem

rem是一个相对单位，1rem等于html元素上字体设置的大小。我们只要设置html上font-size的大小，就可以改变rem所代表的大小。**其实rem布局的本质是等比缩放，一般是基于宽度。**

这样我们就有了一个可控的统一参考系。我们现在有两个目标：

- rem单位所代表的尺寸大小和屏幕宽度成正比，也就是设置html元素的font-size和□屏幕宽度成正比
- rem单位和px单位很容易进行换算，方便我们按照标注稿写css

使用：

借助js

```
//designwidth: 设计稿的实际宽度值，需要根据实际设置
//maxwidth: 制作稿的最大宽度值，需要根据实际设置
//这段js的最后面有两个参数记得要设置，一个为设计稿实际宽度，一个为制作稿最大宽度，例如设计稿为750，最大宽度为750，则为(750,750)
;(function(designwidth, maxwidth) {
    var doc = document,
        win = window,
        docEl = doc.documentElement,
        remStyle = document.createElement("style"),
        tid;

    function refreshRem() {
        var width = docEl.getBoundingClientRect().width;
        maxwidth = maxwidth || 540;
        width > maxwidth && (width = maxwidth);
        var rem = width * 100 / designwidth;
        remStyle.innerHTML = 'html{font-size:' + rem + 'px;}';
    }

    if (docEl.firstChild) {
        docEl.firstChild.appendChild(remStyle);
    }
})
```

```

} else {
    var wrap = doc.createElement("div");
    wrap.appendChild(remStyle);
    doc.write(wrap.innerHTML);
    wrap = null;
}
//要等 viewport 设置好后才能执行 refreshRem, 不然 refreshRem 会执行2次:
refreshRem();

win.addEventListener("resize", function() {
    clearTimeout(tid); //防止执行两次
    tid = setTimeout(refreshRem, 300);
}, false);

win.addEventListener("pageshow", function(e) {
    if (e.persisted) { // 浏览器后退的时候重新计算
        clearTimeout(tid);
        tid = setTimeout(refreshRem, 300);
    }
}, false);

if (doc.readyState === "complete") {
    doc.body.style.fontSize = "16px";
} else {
    doc.addEventListener("DOMContentLoaded", function(e) {
        doc.body.style.fontSize = "16px";
    }, false);
}
})(750, 750);
/*
    第一个参数是设计稿的宽度, 一般设计稿有640, 或者是750, 你可以根据实际调整
    第二个参数则是设置制作稿的最大宽度, 超过750, 则以750为最大限制。
*/

```

1.复制上面这段代码到你的页面的头部的script标签的最前面。

2.根据设计稿大小, 调整里面的最后两个参数值。

3.使用 $1\text{rem}=100\text{px}$ 转换你的设计稿的像素, 例如设计稿上某个块是 $100\text{px} * 300\text{px}$, 换算成rem则为 $1\text{rem} * 3\text{rem}$ 。

示例参考demo/rem.html