

# 21-第二十一章-css3 响应式布局

HTML4

## 一、媒体类型

`all` 所有媒体  
`screen` 彩屏设备  
`print` 用于打印机和打印预览  
`speech` 应用于屏幕阅读器等发声设备  
`braille` 应用于盲文触摸式（已废弃）  
`embossed` 用于打印的盲文（已废弃）  
`projection` 用于投影设备（已废弃）  
`tty` 不适用像素的设备（已废弃）  
`tv` 电视（已废弃）

## 二、关键字

`and` 并且  
`not` 用来排除某种指定的媒体类型  
`only` (限定某种设备)某种特定的媒体类型

## 三、媒体特性

( width:600px ) 宽  
( max-width:600px ) 最大宽度 <=600  
( min-width: 480px ) 最小宽度 >=480  
( orientation:portrait ) 竖屏  
( orientation:landscape ) 横屏

---

## 四、媒体查询

1、方式一 媒体查询：@media [not|only] mediatype [and] (media feature) {  
CSS-Code;  
}

```
1. @media screen and (min-width:400px) and (max-width:500px) {.box {margin: 0 auto;}}
```

//值与值之间需要有空格

---

2、方式二：@import url('index.css') [not|only] mediatype [and] (media feature) [and] (media feature); 需顶行写

```
1. @import url("css/reset.css") screen;
```

---

3、方式三：样式引入

```
1. <link rel="" type="" href="A.css" media="screen and (min-width: 800px)">
2. <link rel="" type="" href="B.css" media="screen and (min-width: 600px) and (max-width: 800px)">
3. <link rel="" type="" href="C.css" media="screen and (max-width: 600px)">
4.
5.
6. <link rel='stylesheet' media='all and (orientation:portrait)' href='portrait.css'>
7. <link rel='stylesheet' media='all and (orientation:landscape)' href='landscape.css'>
8. <link rel='stylesheet' type='text/css' media="print" href='print.css' />
```

## 五、viewport视口

`<meta name="viewport" content="" />` 视口的作用：在移动浏览器中，当页面宽度超出设备，浏览器内部虚拟的一个页面容器，将页面容器缩放到设备这么大，然后展示

```
width [pixel_value | device-width] 例如width = 640
height [pixel_value | device-height]
initial-scale 初始比例
minimum-scale 允许缩放的最小比例
maximum-scale 允许缩放的最大比例
user-scalable 是否允许缩放 ( yes || no 或 1 | 0 )
```

兼容

Edge 模式告诉 IE 以最高级模式渲染文档，也就是任何 IE 版本都以当前版本所支持的最高级标准模式渲染，避免版本升级造成的影响。简单的说，就是什么版本 IE 就用什么版本的标准模式渲染

```
1. <meta http-equiv="X-UA-Compatible" content="IE=edge">
```

使用以下代码强制 IE 使用 Chrome Frame 渲染

```
1. <meta http-equiv="X-UA-Compatible" content="chrome=1">
```

提示 IE 用户安装 Google Frame

最佳的兼容模式方案，结合考虑以上两种：

```
1. <meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">
```

## 六、rem公式

```
1. (function(win,doc){
2.     var docEl = doc.documentElement || document.body;//获取HTML
    L标签
3.
4.     //判断是移动设备还是PC,移动 就采用 'orientationchange',横竖屏事
    件, PC端就采用onresize, 窗口改变时间
5.     var resize = 'orientationchange' in win ? 'orientationcha
    nge':'resize';
6.     function rem(){
7.
8.         var w = docEl.clientWidth/720>1?720:docEl.clientWidt
        h;
9.         docEl.style.fontSize= 100*(w/720)+'px';
10.    }
11.    doc.addEventListener('DOMContentLoaded',rem,false);
12.    //监听 'DOMContentLoaded事件:DOM加载完成执行,如果DOMContentLoaded事件, 那么
    执行rem函数
13.
14.    win.addEventListener(resize,rem,false);
15.    //win下监听resize事件,如果resize事件, 那么执行rem函数
16. })(window,document)
```