# 移动web

* 获取屏幕的物理像素尺寸

**widown.screen.width;** // 获取到屏幕的宽度尺寸

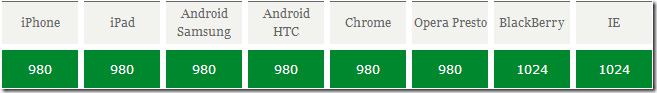
**window.screen.height;** // 获取到屏幕的高度尺寸

* 适口（viewport）是用来约束网站中最顶级的块元素（html）的 它决定的html的大小
* 获取网站适口viewport的大小

document.documentElement.clientWidth;

document.documentElement.clientHeight;

* 移动厂商分别设置了一个默认的viewport的值 如图：



* 适口标签

<meta name="viewport"  
 content="width=device-width, user-scalable=no, initial-scale=1.0, maximum-scale=1.0, minimum-scale=1.0"/>

1. name=”viewport” // 名字适口
2. width=devic-width, // 设别的宽度
3. user-scalable=”no”, // 用户可否对窗口进行缩放 yes 是可以
4. initial-scale=1.0, // 这只页面的初始缩放值
5. maximum-scale=1.0, // 允许用户的最大缩放值
6. minimum-scale=1.0 // 允许用户的最小缩放值

* 媒体查询语句

1. 关键字：
   * 1. **@media** // 媒体
     2. **only** // 指定某个特定的媒体类型，可以省略
     3. **screen**  // 媒体类型 这里指电脑屏幕 平板电脑 智能手机

还有all//指所有类型 print//指打印机和打印预览

* + 1. **and** // 可以将多个媒体特性连接到一起，相当于“且”的意思
    2. **not** // 排除某个媒体类型，相当于“非”的意思，可以省略
    3. **每种媒体类型都具体各自不同的特性，根据不同媒体类型的媒体特性设置不同的展示风格**
       1. **(width:100px)**  // 设备的页面宽**完全等于**多少时的样式
       2. **(height:100px)**  // 设备的页面高**完全等于**多少时的样式
       3. **(min-width:100px)** // 设备的页面宽**大于等于**多少时的样式
       4. **(min-height:100px)** // 设备的页面高**大于等于**多少时的样式
       5. **(max-width:100px)** // 设备的页面宽**小于等于**多少时的样式
       6. **(max-height:100px)** // 设备的页面高**小于等于**多少时的样式
       7. **(device-width:320px**) // 设备的屏幕的可见宽度是多少时的样式
       8. **(device-height:320px)**  // 设备的屏幕的可见高度是多少时的样式
       9. **orientation:portrait** // 竖屏时
       10. **orientation:landscape** // 横屏时

1. 引入方式 两种
2. @media only screen and(max-width:320px){书写样式} // 直接写在样式里
3. <link rel=”stylesheet” href=”min.css” media=”only screen and (max-width:320px)”> // 利用link标签引入

* rem

1. rem //HTML的font-size的大小
2. 适配手机 一般设计图就是两种尺寸
   * 1. 苹果5 //宽320px 设计图尺寸宽为640px

设计图是640px时做适配

为了除尽 除以20 320/20=16 HTML的字体大小就是16px

@media only screen and (width:320px) {

html{

font-size:16px;

}

}

@media only screen and (width:375px) {

html{

font-size:18.75px;

}

}

语法 设计图 宽为100px的盒子 width:(100rem/32);

32的由来 设计图的宽640/20=32

* + 1. 苹果6//宽375px 设计图尺寸宽为750px

设计图是750px时做适配

为了除尽 除以25 375/25=15 HTML的字体大小就是15px

@media only screen and (width:375px) {

html{

font-size:15px;

}

}

@media only screen and (width:320px) {

html{

font-size:12.8px;

}

}

语法 设计图 宽为100px的盒子 width:(100rem/30);

30的由来 设计图的宽750/25=30

* 响应式常用的尺寸

@media only screen and (min-width:980px){  
 body {  
 background-color: red;  
 }  
}  
@media only screen and (max-width:979px) and (min-width:768px) {  
 body {  
 background-color: blue;b  
 }  
}  
@media only screen and (max-width:767px) {  
 body {  
 background-color: pink;  
 }  
}  
@media only screen and (max-width:480px){  
 body {  
 background-color: purple;  
 }  
}

# Scss

* @charset “UTF-8”;
* $变量名：#000; // 变量
* scss引入scss文件

**@import './yin.scss'**;

* 使用变量

body{background-color: **$c1**}

* 注释

// 这是静默注释 这个注释不会显示到编译好的css中 这样的注释是给开发者看的  
/\* 这是多行注释 这个注释会被编译到css中显示 \*/

* 父选择器的标示 &

// article a { color: blue }  
// article a:hover { color: red }  
article {  
 a{  
 color:blue;  
 &:hover{  
 color:red;  
 }  
 }  
}

* 嵌套

// #content article h1 { color: #333 }  
// #content article p { margin-bottom: 4px }  
// #content aside { background-color: #EEE }  
#content {  
 article {  
 h1 {  
 color:#333;  
 }  
 p {  
 margin-bottom: 4px;  
 }  
 }  
 aside{  
 background-color:

* 群组选择器的嵌套

// .container h1, .container h2, .container h3 { margin-bottom: 8px }  
.container{  
 h1,h2,h3{  
 margin-bottom:8px;  
 }  
}

* 子组合选择器

// article > section { border: 1px solid #ccc }  
article{  
 > section{  
 border:1px solid #ccc;  
 }  
}

* 混合器

// 混合器和类名有什么区别  
// 答:混合器是sass中的语法规定 如果这一段css可以用非常简短的名字归纳总结 就用混合器  
// 类名是css中的语法规定 更重要的是 类名会写入到html中 类名一般是有语义化的

// box{  
// border-bottom-left-radius: 10px;  
// border-bottom-right-radius: 10px;  
// border-top-left-radius: 0px;  
// border-top-right-radius: 0px;  
// }  
// @mixin 关键字就是用来定义混合器的  
// 语法 @mixin 混合器的名字 {}  
**@mixin** bottom-radius {  
 border-bottom-left-radius: 10px;  
 border-bottom-right-radius: 10px;  
 border-top-left-radius: 0px;  
 border-top-right-radius: 0px;  
}  
**@mixin** box-sizing {  
 -webkit-box-sizing: border-box;  
 -moz-box-sizing: border-box;  
 -o-box-sizing: border-box;  
 box-sizing: border-box;  
}  
// 调用  
.box{  
 **@****include** bottom-radius();  
 **@include** box-sizing();  
}

* 混合器传参

/\* a:link {color: #FF0000} /!\* 未访问的链接 \*!/  
 a:visited {color: #00FF00} /!\* 已访问的链接 \*!/  
 a:hover {color: #FF00FF} /!\* 鼠标移动到链接上 \*!/  
 a:active {color: #0000FF} /!\* 选定的链接 \*!/  
 normal正常\*/  
 **@mixin** mouse-over(**$normal**,**$link**,**$visited**,**$hover**,**$active**){  
 color:**$normal**;  
 &:link{color:**$link**}  
 &:visited{color:**$visited**}  
 &:hover{color:**$hover**}  
 &:active{color:**$active**}  
 }  
 a{  
 **@include** mouse-over(#ccc, #FF0000,#00FF00, #FF00FF, #0000FF)  
 }

* 混合器默认参数值

**@mixin** mouse-over(**$normal**:black,**$link**:orange,**$visited**:pink,**$hover**:hotpink,**$active**:deeppink){  
 color:**$normal**;  
 &:link{color:**$link**}  
 &:visited{color:**$visited**}  
 &:hover{color:**$hover**}  
 &:active{color:**$active**}  
}  
  
ul{  
 **@include** mouse-over(#999)  
}

* 综合使用

///嵌套  
//.box{border: 1px solid #000;}  
//.box div{margin-bottom:10px;}  
//.box span{font-size:50px;}  
//.box span:hover{font-size:20px;}  
//.box .one{color:pink;}  
//.box .two{color:red;}  
//.box .three{color:purple;}  
//.box .three{color:hotpink;text-align: center;}  
//.box .three div{background-color: purple;}  
//.box .three div:nth-child(2){background-color: #ccc;}  
.box{  
 border: 1px solid #000;  
 div{margin-bottom:10px;}  
 span{  
 font-size:50px;&:hover{font-size:20px;} //&代表父级 伪类的书写  
 }  
 .one{color:pink;}  
 .two{color:red;}  
 .three{  
 color:hotpink;  
 text-align: center;  
 div{background-color: purple;&:nth-child(2){background-color: #ccc;}}  
 }  
}