

移动开发之CSS样式

前端开发部-卢晶晶



1. viewport
2. reset
3. flex布局的应用
4. 开发中的常见问题

viewport

```
<meta content="width=device-width, initial-scale=1.0, maximum-scale=1.0, user-scalable=0" name="viewport" />
```

viewport, 即视口, 浏览器可视区域。

`document.documentElement.clientWidth` (<html>元素宽度) 来源于viewport的宽度。一般来

说:

- PC: `document.documentElement.clientWidth` = viewport宽

- M: `window.innerWidth`

viewport宽度的获取:

- PC: viewport 大小即浏览器可视区域大小

常见设备默认layout viewport: 移动设备上的默认 viewport 宽度一般都 **大于** 浏览器可视区域 (由于css)

iPhone	iPad	Android Samsung	Android HTC	Chrome	Opera Presto	BlackBerry	IE
980	980	980	980	980	980	1024	1024



viewport

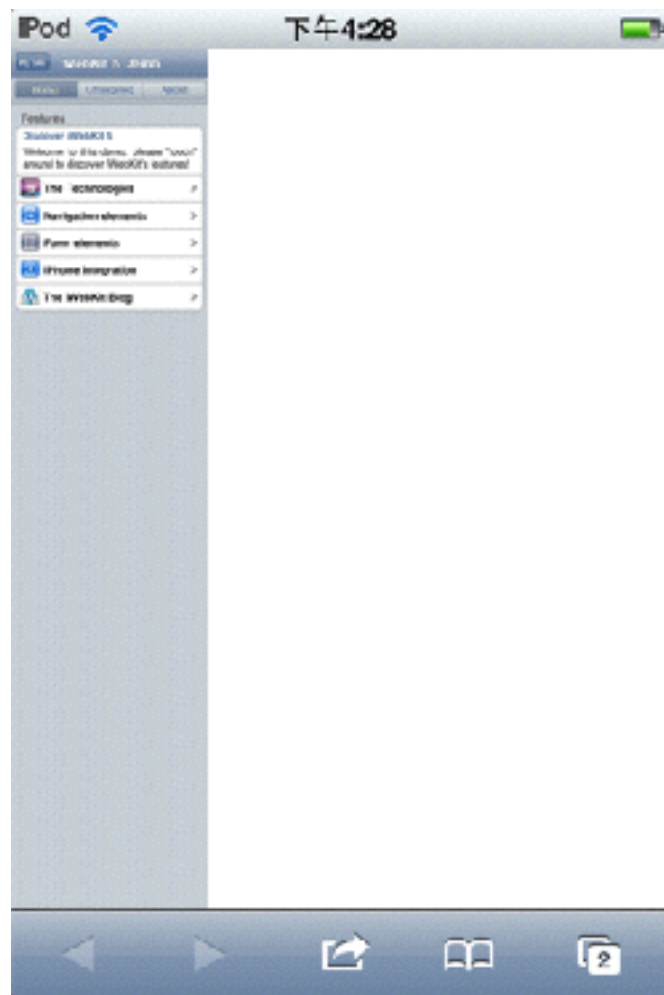
完整显示PC网页



为手机设计的网页



实际显示



viewport

解决办法：device-width

- 默认viewport大于浏览器可视区域，device-width代表移动端浏览器可视区域的大小
- 不同的设备拥有不同的[device-width](#)，将layout viewport的宽度设置为device-width
- 自适应

获取方法：

```
<meta content="width=device-width, initial-scale=1.0, maximum-scale=1.0, user-scalable=0" name="viewport" />
```

`window.innerWidth` 设置该meta标签后，`window.innerWidth`（viewport宽度）就等于`device-width`，由此得到`document.documentElement.clientWidth`（=`window.innerWidth`），从而设置<html>（根元素）的字体大小，进而使用rem



京东

rem

```
var deviceWidth = document.documentElement.clientWidth;
if(deviceWidth > 750) {
    deviceWidth = 750;
}
var rootFontSize = deviceWidth / 7.5;
document.documentElement.style.fontSize = rootFontSize +
'px';
@media only screen and (min-width:320px) and (max-
width:350px) {
    html {
        font-size: 43px !important;
    }
}
@media only screen and (min-width:351px) and (max-
width:375px) {
    html {
        font-size: 50px !important;
    }
}
.....
```

$$\frac{\text{设计稿宽 (750)}}{\text{设计稿元素宽}} = \frac{deviceWidth}{\text{实际元素宽}}$$

$$\text{实际元素宽度: } \frac{deviceWidth}{\text{设计稿宽}} \times \text{设计稿元素宽}$$

$$\frac{deviceWidth}{\text{设计稿宽}} \approx \frac{414}{750} < 12(px)$$

$$\text{rootFontSize} = \frac{deviceWidth}{\text{设计稿宽}/100}$$

$$\text{实际元素宽} = (\text{设计稿元素宽} / 100) \text{ rem}$$

vw,vh

rem	相对于根元素字体大小
vw	相对于视窗的宽度：视窗宽度是100vw
vh	相对于视窗的高度：视窗高度是100vh
vm	相对于视窗的宽度或高度，取决于哪个更小

rem

@media不能完全等比
通过脚本计算需要引入额外js

vw,vh

逻辑清晰，直观

viewport的1% = 1vw

vw+rem

Current aligned	Usage relative	Date relative	Show all										
				Edge *	Firefox	Chrome	Safari	Opera					
IL									iOS Safari *	Opera Mini *	Android *	Chrome for Android	
						49							
						59							
				14	54	60	10.1		10.2		4.4		
				15	55	61	11	47	10.3		4.4.4		
11	16	56	62	TP	48				11	all	56	61	
					57	63		49					
					58	64							

viewport-fit

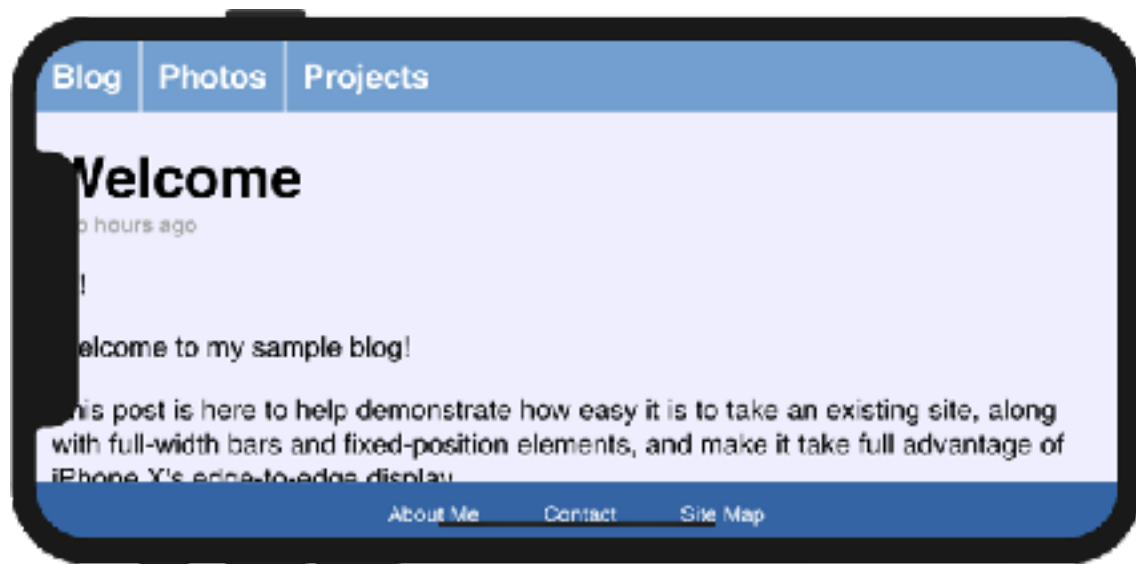
available in iOS 11



viewport-fit

```
<meta name='viewport' content='initial-scale=1, viewport-fit=auto'>
```

```
<meta name='viewport' content='initial-scale=1, viewport-fit=cover'>
```



viewport-fit

```
.post {  
  padding: 12px;  
  padding-left: constant(safe-area-inset-left);  
  padding-right: constant(safe-area-inset-right);  
}
```

Respect Safe Area



1. viewport
2. **reset**
3. flex布局的应用
4. 开发中的常见问题

base.css

```
margin: 0;
padding: 0;
border: 0;
font-size: 100%;
font: inherit;
vertical-align: baseline;
-webkit-tap-highlight-color:
rgba(0, 0, 0, 0);
-webkit-tap-highlight-color:
transparent;
-webkit-box-sizing: border-
box;
-moz-box-sizing: border-box;
-ms-box-sizing: border-box;
box-sizing: border-box;
```

```
ol, ul, li, menu {
    list-style: none
}
/* 阻止旋转屏幕时自动调整字体大小*/
html {
    -webkit-text-size-adjust: none;

    -ms-text-size-adjust: 100%
}
```

```
/* ios默认文本框阴影,禁用系统原生按钮样式*/
input[type=text], textarea {
    -webkit-appearance: none;
}
```

```
a, button, input {
    outline: 0 none;
    text-decoration: none
}
```

禁用UIWebView默认交互行为

WebView加载显示HTML页面时，组件本身提供了一些系统默认的交互行为：

长按文本区域出现放大镜和内容选择



长按链接呼出菜单弹窗



禁止放大镜和文本选择

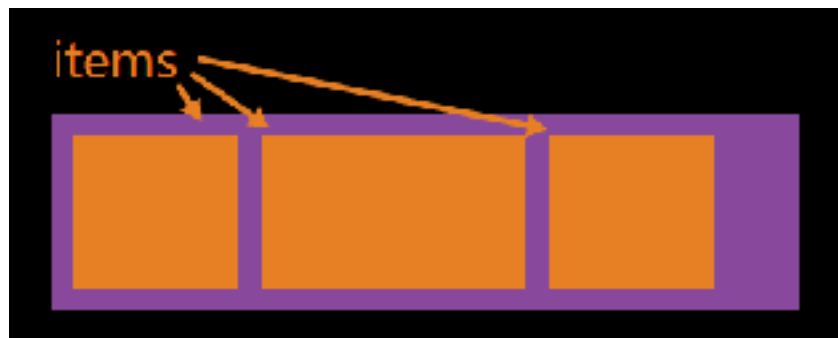
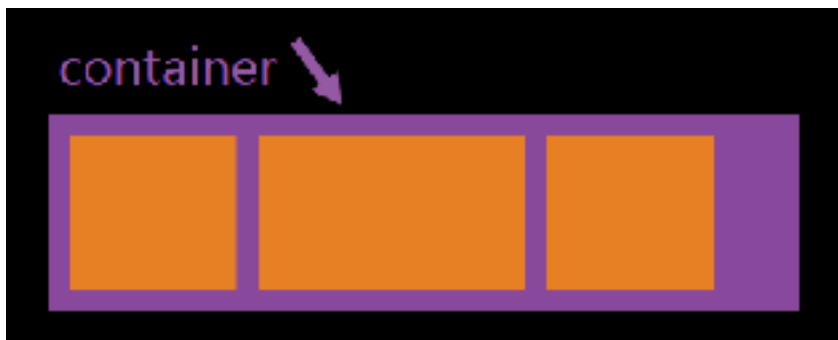
```
*:not(input,textarea) {  
    -webkit-user-select: none;  
  
    -ms-user-select: none;  
    -o-user-select: none;  
    -moz-user-select: none;  
    user-select: none;  
}
```

禁止长按链接呼出弹窗

```
a {  
    -webkit-touch-callout:  
none;  
    -ms-touch-callout: none;  
    -o-touch-callout: none;  
    -moz-touch-callout: none;  
    touch-callout: none;  
}
```

1. viewport
2. reset
- 3. flex布局的应用**
4. 开发中的常见问题

flex布局的应用



```
@mixin box{
  display: -moz-box;
  display: -ms-flexbox;

  display: -webkit-box;

  display: box;
  display: -webkit-
flex;
  display: flex;
}
```

项目属性: order
flex-grow
flex-shrink
flex-basis
flex
align-self

容器属性:
flex-direction
flex-wrap
flex-flow
justify-content
align-items
align-content



等分——无间距



```
ul {  
  @include box;  
  li {  
    @include  
flex(1);  
  }  
}
```

```
@mixin box{  
  display: -moz-box;  
  display: -ms-flexbox;  
  display: -webkit-box;  
  display: box;  
  display: -webkit-flex;  
  display: flex;  
}
```

```
/* 平均分配 */  
@mixin flex($order){  
  -ms-flex: $order;  
  -webkit-box-flex: $order;  
  -webkit-flex: $order;  
  flex: $order;  
}
```



等分——justify-content: space-around



```
ul {  
  display: flex;  
  justify-content: space-around;  
}
```

in JD App



每个子元素两侧的间隔相等

所以，元素之间的间隔是元素与边框间隔的2倍



等分——justify-content: space-between



两端对齐，子元素之间的间隔都相等

```
ul {  
  display: flex;  
  justify-content: space-between;  
}
```

兼容写法:

/* 两边对齐 */

```
@mixin both-side{  
  box-pack: start;  
  -webkit-box-pack: start;  
  justify-content: space-between;  
  -webkit-justify-content: space-between;  
}
```

等分——justify-content: space-between



温馨提示：订单中含有不支持7天无理由退货的商品，
请确认相关商品信息后再交订单

商品 180*11*8251 默认

北京市台江区三环路四环路之间海园广里20号楼10单元003



维达(Minda) 抽纸 超韧3层100抽软抽...

¥21.99

X8

支付方式

全款支付

配送

每周2件 x 3期

2017-09-27至2017-10-11，每周送1次【周三】

京东配送

11.00前下单，【第1/3期】预计2017-09-27【周三】送达

发票信息

纸质(详细 个人)

商品金额

¥131.93

运费(近期赠免运费)

+ ¥0.00

实付款: ¥131.93

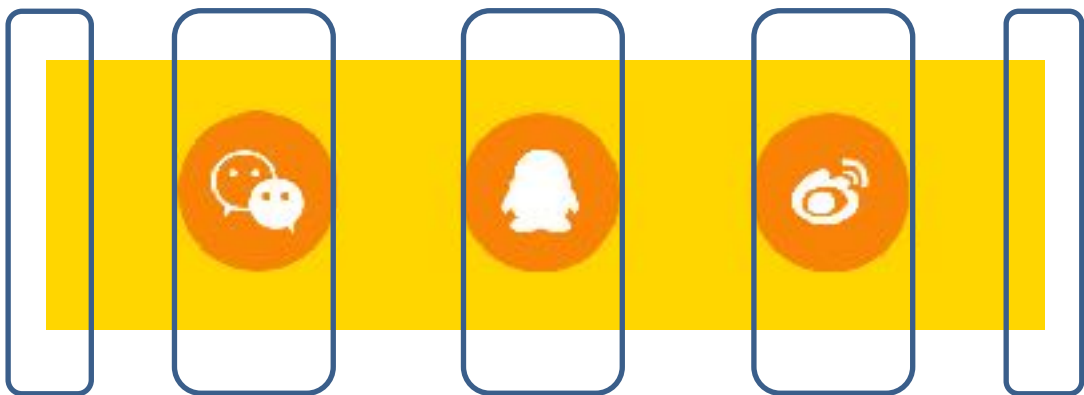
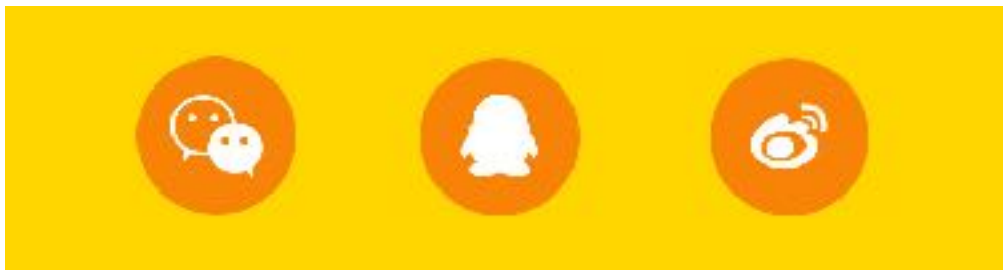
提交订单



等分——space-between

space-between的一个应用：

li（三个圆），父元素剩余间距平分



```
<ul>
  <li></li>
  <li></li>
  <li></li>
</ul>
```

```
ul {
  display: flex;
  justify-content: space-between;
}
li:before, li:after {
  content: ' ';
}
li {
  width: .8rem;
  height: .8rem;
}
```

多列布局



父元素

```
display: flex;
```

子元素

①自适应:

```
flex: 1;
```

②定宽:

```
width: xxx;
```



居中

```
/* 垂直居中 */
@mixin vertical-center {
  box-align: center;
  -webkit-box-align: center;
  -ms-flex-align: center;
  align-items:center;
  -webkit-align-items:center;
}
```

```
/* 水平居中 */
@mixin horizontal-center {
  box-pack: center;
  -webkit-box-pack: center;
  justify-content:center;
  -webkit-justify-
content:center;
}
```



flex-wrap

日	一	二	三	四	五	六
2017年09月						
				1	2	
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28 开始	29	30
2017年10月						
1	2	3	4	<u>5</u>	6	7
8	9	10	11	<u>12</u>	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				
2017年11月						

```
.calendar .flex {  
    display: -ms-flexbox;  
  
    display: -webkit-box;  
  
    display: box;  
    display: flex;  
    -ms-flex-wrap: wrap;  
    flex-wrap: wrap;  
}
```



flex in 小程序

微信小程序实现了flex布局，并且对flex的支持很好，对view组件采用flex布局，无需考虑浏览器兼容性问题

```
<view class="flex-row" style="display: flex;">  
  <view class="flex-view-item">1</view>  
  <view class="flex-view-item">2</view>  
  <view class="flex-view-item">3</view>  
</view>
```



flex in RN

React Native通过一个基于FlexBox的布局引擎，在所有移动平台上实现了一致的跨平台样式和布局方案

RN的flexbox主要有以下几个属性：

1. **flex**

2. **flexDirection**: row, column (没有row-reverse, column-reverse)

3. **alignItems**: flex-start, flex-end, center, stretch (没有baseline)

4. **alignSelf**

5. **justifyContent**: flex-start, flex-end, center, space-between, space-around

6. **flexWrap**: nowrap, wrap (没有wrap-reverse)

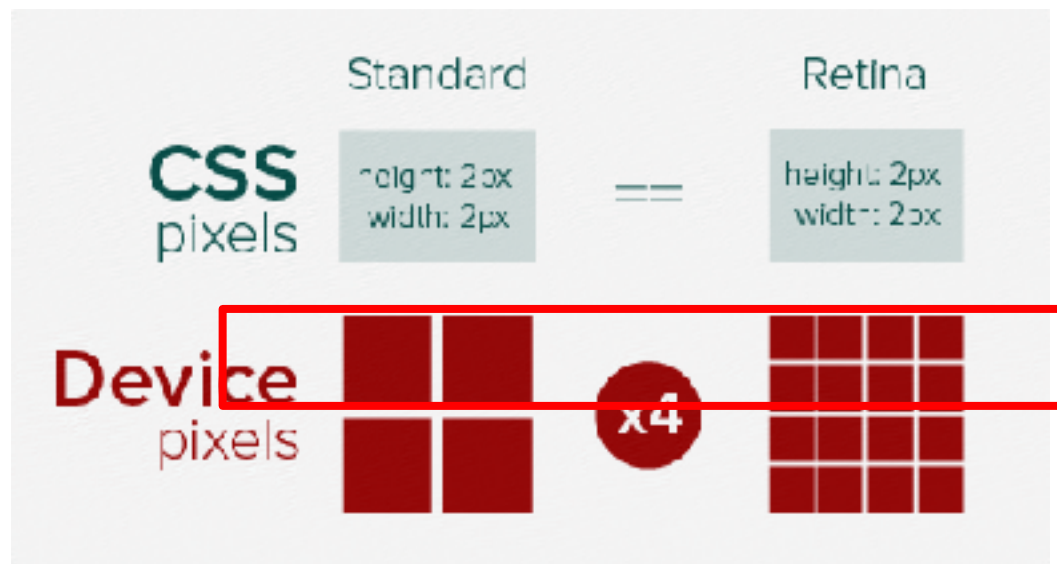


1. viewport
2. reset
3. flex布局的应用
4. 开发中的常见问题

1px on retina

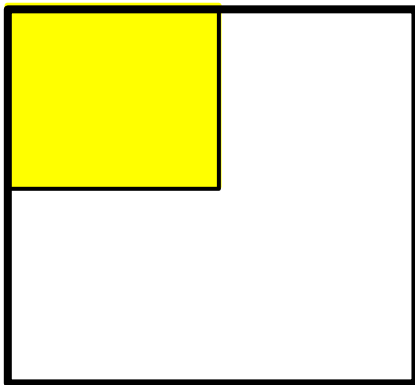


```
border: 1px solid #000;
```



1px on retina

一边



```
@mixin border-bottom-1px($color) {  
  position: relative;  
  &:before {  
    content: '';  
    display: block;  
    position: absolute;  
    width: 100%;  
    left: 0;  
    bottom: 0;  
    height: 1px;  
    background-color: $color;  
    -webkit-transform: scaleY(0.5);  
    transform: scaleY(0.5);  
    pointer-events: none;  
  }  
}
```

四边

```
@mixin bd-all-1px($bStyle,$color,$radius){  
  position: relative;  
  &:after{  
    content: '';  
    position: absolute;  
    top: 0; left: 0;  
    width: 200%;  
    height: 200%;  
    -webkit-box-sizing: border-box;  
    box-sizing: border-box;  
    -webkit-transform: scale(0.5);  
    transform: scale(0.5);  
    -webkit-transform-origin: left top;  
    transform-origin: left top;  
    border: 1px $bStyle $color;  
    border-radius: $radius;  
  }  
}
```

rem小数像素问题

浏览器对小数单位精度解析的差异导致无法精确处理小数

css sprite背景图片边角缺失



解决方法



1. 使用单个图片
2. 小图标base64
3. 使用 iconfont
4. 制作雪碧图时每个icon之间留出多一些空白间隙
5. js判断缩放比例



IOS Fixed + Input



IOS Fixed + Input



第三方: iScroll.js, 不推荐

<header> fixed

```
main {  
  /* 绝对定位 */  
  position: absolute;  
  top: 50px;  
  bottom: 34px;  
  /* 内部滚动 */  
  overflow-y: scroll;  
  /* 增加该属性, 可以增加弹性 */  
  -webkit-overflow-scrolling: touch;  
}
```

<footer> fixed

IOS不能实时触发scroll事件

粘性布局

解决方法: `position: sticky`

IE	Edge *	Firefox	Chrome	Safari	Opera	IOS Safari *	Opera Mini *	Android *	Chrome for Android
			49						
			59						
	14	54	60	10.1		10.2		4.4	
	15	55	61	11	47	10.3		4.4.4	
11	16	56	62	TP	48	11	all	56	61
		57	63		49				
		58	64						

IOS不能实时触发scroll事件

```
.sticky {  
  position: -webkit-sticky;  
  position: sticky;  
  top: 0;  
}
```

position: sticky的生效条件

1. 设置了position: sticky;的元素要生效必须要至少设置top, bottom, left, right中的一个
2. sticky 只在父元素里生效, 父元素必须可以滚动 (overflow: visible;)
3. 达到设定的阈值



flex item被内容撑开

因为等分是基于剩余空间，被内容撑开后无法实现等分。

解决方法：

```
width: 0;
```



安卓手机line-height垂直居中偏上



原因:

1. font-size为奇数或小数, 即不可被2整除 (rem)
2. font-size小于12px

解决方法: 把字号放大, 再使用transform进行缩放



谢谢！

