**逻辑像素：—— 物理像素**

**IOS 750 X 1334**

**安卓 最低分辨率：360 640**

**实际像素：——**

<!-- 1、移动端视口问题

375 x 667 浏览器测试窗口

750 x 1334 正常iphone屏幕

a) 布局视口 css中设置

b) 视觉视口 375 x 667

逻辑像素/css像素（软件能达到的尺寸）

软件怎么办？将两个实际物理尺寸当做一个逻辑像素来处理

设计师拿到的设计稿大小：750x1334

物理像素 （硬件所能达到的尺寸）

一般将逻辑像素与物理像素的比叫做dpr

iPhone6的dpr 2

c) 理想视口

一般情况下将屏幕大小作为布局大小最合适

我们需要提前设置 width = device-width

initial-scale=1 不能随意缩放 -->

<!-- 单位为 rem -->

移动端布局方式

1.百分比

2.rem

3.flex

传统的布局方式，基于盒子模型，依赖display+float+position

**flex布局**以后：子元素身上的float、clear、vertical-align 将会失效

必须有容器承载

父元素上设置

display：flex；（新版本兼容）

display：-webkit-flex；（旧版本兼容）

flex-warp：nowarp;(默认值 不换行)

flex-warp：warp;（换行）

flex-warp:warp-reverse;(第一行在最下方)

主轴对齐的方向

（水平方向）

水平

flex-direction：row; (从左到右)

flex-direction：row-reverse; (从右到左)

垂直

flex-direction：column; (从上到下)

flex-direction：column-reverse;(从下到上)

主轴的对齐方式

justify-content：flex-start；(从左到右)

justify-content：flex-end； (从右到左)

justify-content：flex-center；（居中）

justify-content：space-between；（文本两端对齐）

justify-content：space-around； （文本间隔相等）

单轴 交叉对齐方式

（Y轴单行）

align-iteams:flex-start;(交叉轴的起点)

align-iteams:flex-end;(交叉轴的终点)

align-iteams:flex-center;(交叉轴的中心点)

align-iteams:flex-baseline;(第一行文字的基点)

align-iteams:flex-stretch;(如果没有设高度或者高度为auto的时候，自动占满整个容器)

多轴 交叉轴的对齐方式

（Y轴多行）

align-content:flex-start;{交叉轴的起点}

align-content:flex-end;

align-content:center;

align-content:space-around;

align-content:space-between;

align-content:stretch;

子类的设置

order:1;{设置项目的排列顺序}

flex-grow：1;(设置项目扩张 里面数字是扩大倍数 默认是0)

flex-shrink：1；（设置项目缩小 空间不足时会对该项目进行缩小）

单个项目对齐方式

align-self:flex-start;

align-self:flex-end;

align-self:center;

align-self:baseline;

align-self:stretch;

font-size:10px;

375 750

html 100

375\*100/750=html