

4 个像素

物理像素(分辨率):

也叫设备像素，电脑的分辨率，设备呈像的最小单元

CSS 像素:

浏览器的像素，在一个标准的显示密度下，一个 CSS 像素对应着一个设备像素。，是 web 开发中的最小单元

独立像素

（就是机子的像素）计算机坐标中的一个点也是一个抽象的层，是设备提供出来的接口

位图像素:

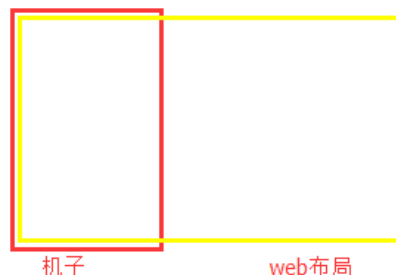
图片的像素是图片的最小单元。

1 个位图像素对应于 1 个物理像素，图片才能得到完美清晰的展示

3 个视口

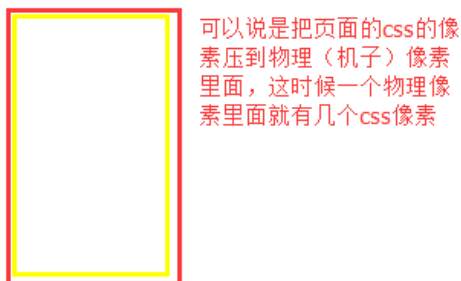
布局视口(layout viewport:)

可以通过 `document.documentElement.clientWidth` 来获取
默认的布局视口宽度远大于屏幕的宽度（css 像素与物理像素一比一的情况下），就是物理像素是固定的，但是 web 页面的像素是大于机子的物理像素的



视觉视口 (visual viewport:)

可以通过 `window.innerWidth` 获取视觉视口的宽度等于屏幕的宽度
不过视觉视口的 css 像素的数量会随着用户的缩放而改变。



理想视口

在页面中加入 viewport 的 meta 标签，理想视口才会生效。

```
<meta name="viewport" content="width=device-width" />
```

这一行代码告诉我们，布局视口的宽度应该与理想视口的宽度一致

2 个操作缩放

pc 端的缩放:会改变元素的布局（布局视口）

移动端（缩放只改变视觉视口内 css 像素的个数）

放大---> 视觉视口包含的 css 像素的个数变少

缩小---> 视觉视口包含的 css 像素的个数变多

1 个比例

- 像素比： 物理像素 /设备独立像素
- 一个方向上占据一块屏幕所需要的物理像素的个数 / 一个方向上占据一块屏幕所需要的设备独立像素的个数
- 设备物理像素和设备独立像素的比例，
也就是 $\text{devicePixelRatio} = \text{物理像素} / \text{设备独立像素}$
- 可以通过 `window.devicePixelRatio` 来获取

物理像素与 css 像素比例的维护是谁在维护？ 维护规则是什么？

视觉视口(1.决定用户能看到什么；2.包住整个布局视口)

物理像素:屏幕的分辨率

css 像素: 布局视口尺寸

加 name 为 viewport 的 meta 标签

###四个像素之间的关系

物理像素和设备独立像素:

像素比: 一个方向上占据一块屏幕所需要的物理像素的个数 / 一个方向上占据一块屏幕所需要的设备独立像素的个数 = 2;

物理像素和位图像素:

1:1 的时候才能完美清晰的展示

物理像素和 css 像素

普通屏:1 比 1

高清屏: 加 name 为 viewport 的 meta 标签

css 像素和设备独立像素

没有加 name 为 viewport 的 meta 标签:没有关系

加 name 为 viewport 的 meta 标签:可以认为 css 像素就是设备独立像素

理想视口与设备之间的关系

理想视口的尺寸：设备独立像素所代表的值

不同浏览器在同一设备上理想视口的尺寸可能会不一样

一款浏览器在不同设备上理想视口的尺寸可能会不一样

等比问题

- 没有加 name 为 viewport 的 meta 标签

一个相同 css 像素大小的区域在不同的设备是等比的, 在不同的设备上占据的实际物理大小（英寸）不一样

- 加 name 为 viewport 的 meta 标签

一个相同 css 像素大小的区域在不同的设备是不等比的, 在不同的设备上占据的实际物理大小（英寸）是一样的

等比是不是一个必须的需要？

百分百还原设计图 ---> 在不同设备上要等比（文字要完美清晰）

----> 必须加 meta 标签（不等比）

---> 适配!!!!（加上 meta 标签后也得等比）

视觉视口包含的 css 像素的个数

默认情况（css 像素和物理像素 1:1）---> 屏幕的分辨率

移动端浏览器初始化的时候（视觉视口必须要包住布局视口） --->

布局视口包含的 css 像素的个数就是视觉视口所包含的

怎么获取三个视口的值

布局视口：document.documentElement.clientWidth(基本没有兼容性问题),要加理想视口才行

视觉视口：window.innerWidth（有一点兼容性问题）

理想视口：screen.width（兼容性问题极大）

完美视口以及 meta 标签

过大的元素 ---> 完美视口 <meta name="viewport" content="width=device-width,initial-scale=1.0" />

哪些操作会影响布局视口

width=320

initial-scale=2.0

pc 端

//获取 pc 端视口的大小 最干净那个视口

```
console.log(document.documentElement.clientWidth);
```

//获取 pc 端浏览器视口（viewport）宽度（单位：像素），如果存在垂直滚动条则包括它。

```
console.log(window.innerWidth);
```

//获取 pc 端浏览器窗口外部的宽度。表示整个浏览器窗口的宽度，包括侧边栏（如果存在）、窗口镶边（window chrome）和调正窗口大小的边框

```
console.log(window.outerWidth);
```

//获取 pc 端屏幕的分辨率

```
console.log(window.screen.width);
```

移动端

var layout = document.documentElement.clientWidth;//layout 的宽度，没有兼容性问题

var visual = window.innerWidth;//visual 的宽度，接近全部支持

var dream = screen.width;//一半代表理想视口的宽度，一半代表设备的分辨率!! 有很大的兼容性问题