

## 第六篇 移动网页开发

# Contents

## 01.移动端知识

viewport

## 02.vw/vmin适配方案

case

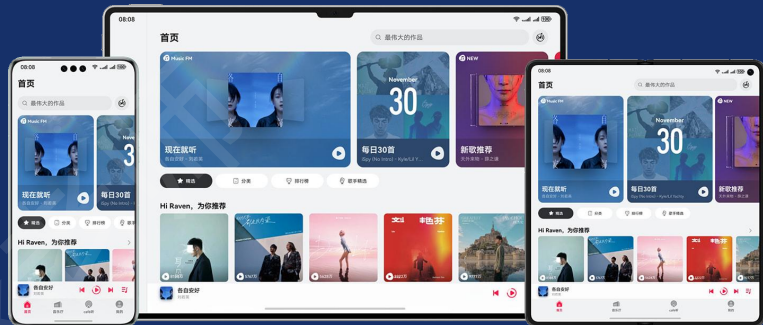
## 03.rem适配方案

case

## 移动端知识



电脑 (PC)



移动端 (手机、折叠屏、平板等)

**全球整体：**移动端流量占比约 63% ，PC端37%，其中电商与社交娱乐等行业移动端主导流量（占比70-90%）

# 移动端知识

常见的移动端开发：



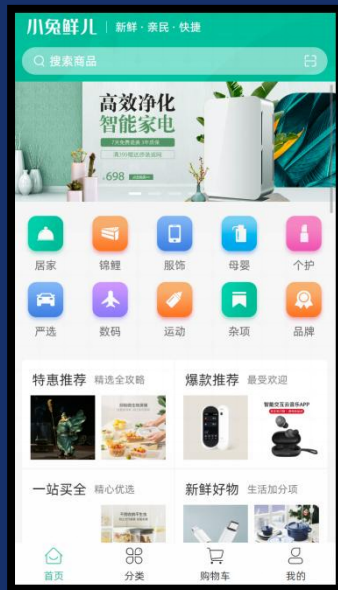
原生APP开发

针对安卓、iOS、鸿蒙移动操作系统开发的 APP



小程序端

微信小程序、支付宝小程序等。uniapp、taro等



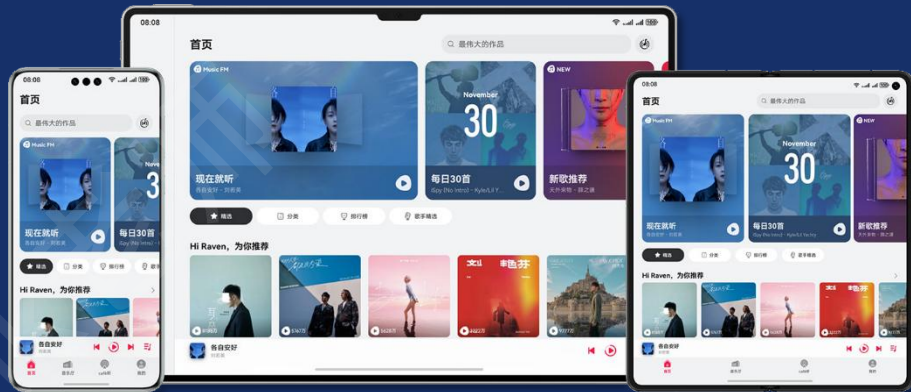
web端 (H5)

H5、CSS3、JavaScript 等技术

## 移动端知识



手机尺寸大小不一

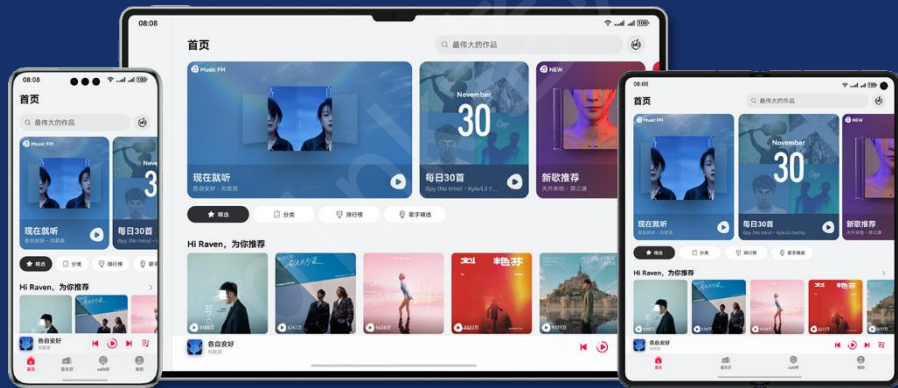


折叠屏、平板、车机等

页面要在移动端设备完成适配，正常显示

## 移动端知识

PC页面会设置版心，移动端如何设置呢？



移动端页面宽度根据**设备宽度**来定

# 移动端知识

视口标签：

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

属性名	作用	取值	默认值	示例	注意事项
width	设置视口宽度	device-width 或具体数值（如 375）	980px（移动端默认）	<code>&lt;meta name="viewport" content="width=device-width"&gt;</code>	移动端默认的980px称为 <b>布局视口</b> device-width 以匹配设备宽度称为 <b>理想视口</b>
initial-scale	页面初始缩放比例	数值（如 1.0）	1.0	<code>&lt;meta name="viewport" content="initial-scale=1.0"&gt;</code>	与 width 和 device-width 配合使用，避免缩放混乱
minimum-scale	允许用户缩放的最小比例	数值（如 0.5）	0.1	<code>&lt;meta name="viewport" content="minimum-scale=0.5"&gt;</code>	若设为 1.0，则用户无法放大页面
maximum-scale	允许用户缩放的最大比例	数值（如 2.0）	10.0	<code>&lt;meta name="viewport" content="maximum-scale=3.0"&gt;</code>	限制缩放范围，但可能影响无障碍访问
user-scalable	是否允许用户手动缩放	yes 或 no	yes	<code>&lt;meta name="viewport" content="user-scalable=no"&gt;</code>	禁用缩放可提升体验，但需谨慎使用（无障碍场景需权衡）

# 移动端知识

调试工具移动端：





# 总结

## 1. 移动端开发需要设置版心吗？

不需要。 让我们页面宽度和移动端屏幕一样宽就行。

## 2. 如何保证我们页面宽度和移动端屏幕一样宽？

视口标签。 最核心的是宽度设置为移动端**设备宽度**。

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

# 总结

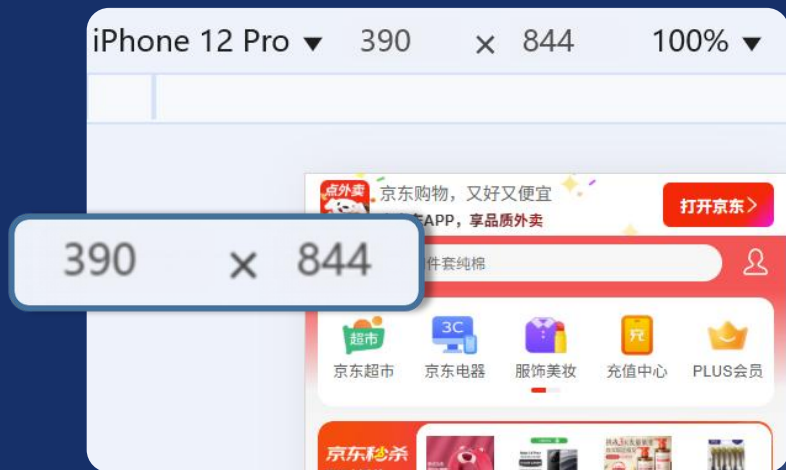
## 3. 移动端开发怎么预览?

调试工具。有移动端模拟器。



## 移动端知识

物理像素和CSS像素：



iphone 12 pro

## 移动端知识

物理像素和CSS像素：



物理像素（也称设备像素）：

是设备屏幕的实际物理像素数量，即硬件层面的像素点总数，出厂默认。

例如：

一台设备的屏幕分辨率为 2532×1170，表示横向有2532个物理像素，纵向有1170个物理像素。

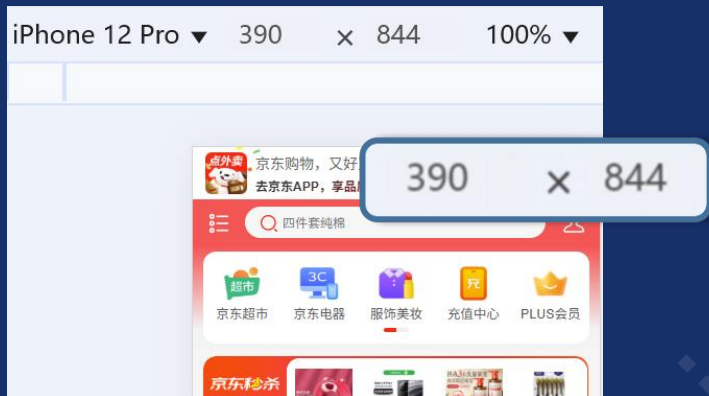
硬件相关：

由屏幕硬件决定，不同设备的物理分辨率可能不同。

将更多的物理像素点压缩至一块屏幕里，从而达到更多的分辨率，提高了屏幕显示的细腻程度

## 移动端知识

物理像素和CSS像素：



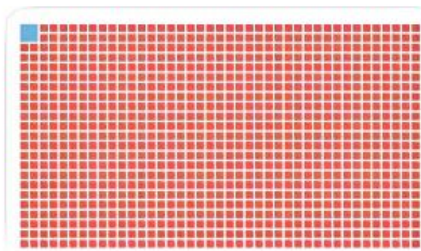
**CSS像素**（也称设备独立像素或者逻辑像素）：

是CSS中使用的虚拟像素单位，用于**布局**和**样式计算**。

它是一个抽象概念，不直接对应物理像素，而是通过**设备像素比**（DPR）与物理像素关联。

比如：PC端，1个CSS像素就是1个物理像素，但是手机端要看DPR

普通标准屏



设备独立像素 (CSS pixels)



物理像素 (physical pixels)



# 移动端知识

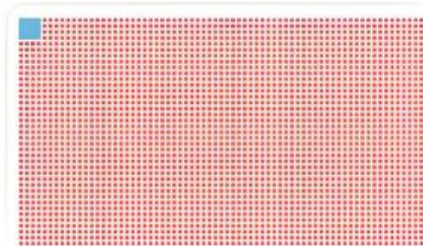
物理像素和CSS像素：

iphone

型号	物理尺寸 (inch)	逻辑分辨率(css像素)	物理分辨率 (物理像素)	倍率 (dpr)
15, 15.pro, 14.pro	6.1	393×852	1179×2556	3
15.proMax, 15.plus, 14.proMax	6.7	430×932	1290×2796	3
14.plus, 13.proMax, 12.proMax	6.7	428×926	1284×2778	3
14, 13, 13pro, 12, 12pro	6.1	390×844	1170×2532	3
11.proMax, xs.max	6.5	414×896	1242×2688	3
11.pro, xs, x	5.8	375×812	1125×2436	3
11, xr	6.1	414×896	828×1792	2
8, 7, 6s, 6	4.7	375×667	750×1334	2

<https://remeins.com/index/app/sizegf>

## 2 倍屏



```
width: 2px;
height: 2px;
```



设备独立像素 ( CSS pixels )



物理像素 ( physical pixels )

# 移动端知识

## 设计稿：

移动端设计稿通常以 750px 作为标准宽度（尤其是微信小程序、H5页面等）。

1. 早期移动端设计以 iPhone 为基准，750px (@2x) 成为行业默认规范。
2. iPhone 6/7/8 的物理分辨率是 750×1334px，设计稿按 750px 宽度制作时，能直接对应这些设备的 1:1 视觉还原，方便设计师精准把控细节（如字体大小、间距等）。

设计稿按 750px（2倍大小）制作后：

1. 开发时直接缩放到 50%（如 CSS 中写 375px），即可适配逻辑分辨率。
2. 图片和图标按 2倍尺寸输出，在高清屏（Retina）上显示更清晰。（2倍图）



# 总结

## 1. 物理像素和CSS像素分别是什么？

物理像素。是设备屏幕的实际物理像素数量，出厂默认，平时我们说的设备分辨率，提高屏幕显示细腻程度。

CSS像素。用于布局和样式计算。跟设备 DPR有关系。



# 总结

## 1. 开发设计稿如何设置？

设计师设计稿是按照物理像素设计。

我们前端开发工程师要按照CSS 像素开发代码。

所以我们设计稿拿到后，设置为@2X 或者 @3X 就可以了



# Contents

## 01.移动端知识

viewport

## 02.vw/vmin适配方案

case

## 03.rem适配方案

case

# 移动端知识

实现移动端适配方案:

## 1. 响应式:

一套代码通过响应式方案完成PC、平板、手机端的适配。



代表: vivo官网、华为官网等

## 2. 单位适配:

PC和平板通过响应式完成适配。

单独制作手机端页面, 通过rem、vw单位完成页面适配。



代表: 淘宝手机端、B站手机端、京东手机端等

不管哪种方案, 页面布局还是使用grid、flex、百分比等, 元素大小需要进行适配

# 移动端适配

## vw/vh单位适配:

vw (Viewport Width) 是一种基于视窗宽度的相对单位, 非常适合用于移动端适配。

### vw 单位定义:

- ❑ 1vw = 视窗宽度的 1%
- ❑ 如果视窗宽度是 375px, 则 1vw = 3.75px
- ❑ 同理 100vw = 视窗完整宽度。
- ❑ vh: 视窗高度的 1%
- ❑ 注意vw和%不一样, %相对于父容器

### 设计稿转换公式:

$$\text{vw值} = (\text{元素在设计稿中的px值} / \text{设计稿宽度}) \times 100\text{vw}$$

$$(106 / 375) * 100\text{vw} = 28.26\text{vw}$$

### 注意:

网页是主要对屏幕宽度适配, 所以通常用vw即可, vh很少用

强、过多的阳光中紫外线的...



12 收藏 120 评论

```
.box {  
  width: 28.26vw;  
  height: 28.26vw;  
}
```

# 移动端适配

vw/vh单位适配:

vw适配方案:

## 方式1: 使用 calc() 函数

```
.nav {  
  width: calc(106 / 375 * 100vw);  
  height: calc(106 / 375 * 100vw);  
}
```

书写繁琐

## 方式2: 使用 CSS 预处理器 (推荐)

- less或者sass实现, 后续工程化讲

但是实际开发里面直接写px不需要计算

## 方式3: 结合 PostCSS 插件 (最佳实践)

- less/sass代码直接写px
- 使用 postcss-px-to-viewport 插件, 打包发布的时候会统一自动转换为vw。
- 但是需要脚手架知识, 后续Vue阶段讲。

## 方式4: 使用trae插件转换 (当前阶段)

### px to rem & rpx & vw (cssrem) V3.1.1

存储库 | 2,834 | ★★★★★ (0) | cipchk

Converts between px and rem & rpx & vw units in VSCode

禁用

卸载



自动更新



## 移动端适配

vw/vh单位适配:

插件配置:

**px to rem & rpx & vw (cssrem)** V3.1.1

存储库 | 2,834 | ★★★★★ (0) | cipchk

Converts between px and rem & rpx & vw units in VSCode

禁用 卸载 ☒ 自动更新 ⚙️

**Cssrem: Vw**

☒ 是否启用vw支持

**Cssrem: Vw Design**

规定设计稿宽度（一般等同于浏览器视口宽度），默认 `750`

375

# 综合案例

优医问诊项目：



登录页



首页



我的

## 综合案例

优医问诊项目：

技术栈：flex + vw 实现适配。 px 利用插件完成vw转换

小图标：svg 格式为主



## 综合案例

优医问诊项目：

CSS 提供了四个独立的圆角属性，分别控制四个角：

`border-top-left-radius`（左上角）

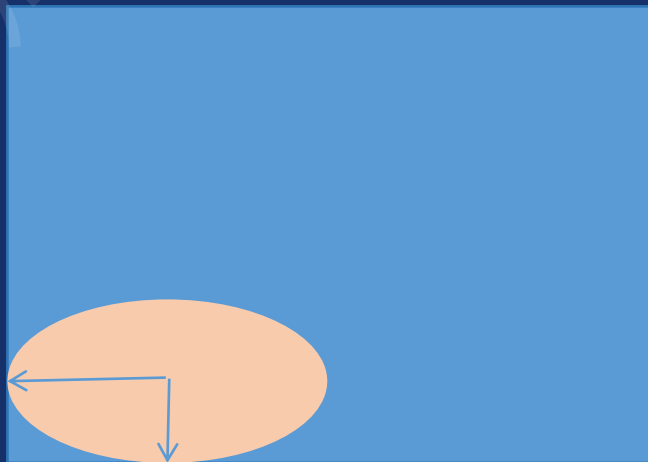
`border-top-right-radius`（右上角）

`border-bottom-right-radius`（右下角）

`border-bottom-left-radius`（左下角）

如果后面跟两个值分别表示该角的水平半径和垂直半径

```
border-bottom-left-radius: 150px 20px;
```



## 综合案例

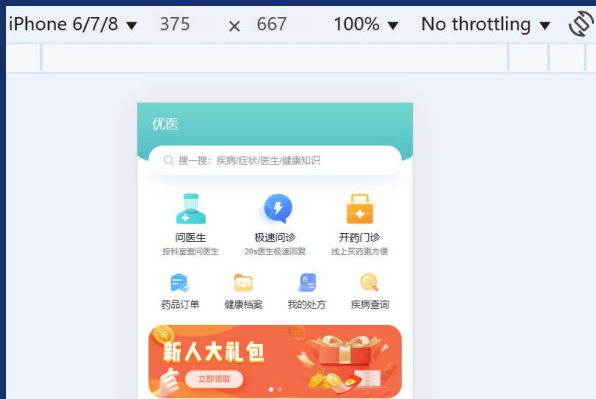
优医问诊项目：



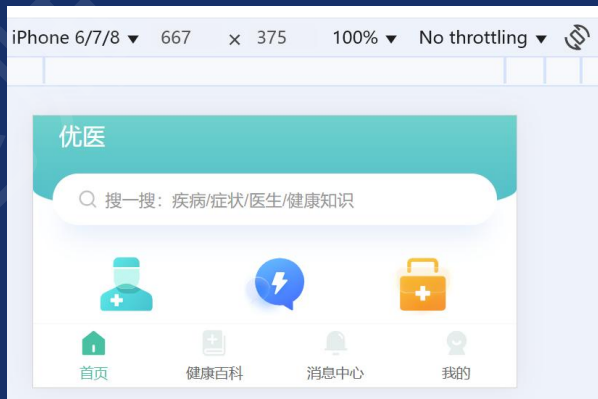
```
a {  
  -webkit-tap-highlight-color: rgba(0, 0, 0, 0);  
  /* 移除移动端点击高亮 */  
}
```

## 移动端适配

vmín/vmax单位:



竖屏状态下



横屏状态下

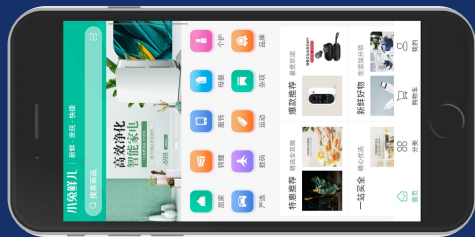
当设备旋转（如手机从竖屏转横屏）时，视口的宽高会变化，但 `vmín` 始终取当前宽高的较小值，`vmax`取较大值。

## 移动端适配

vw/vmax单位：

特性	vw (Viewport Width)	vmin (Viewport Minimum)
定义	视口宽度的 1%	视口较小尺寸（宽度或高度）的 1%
计算依据	仅基于视口宽度（width）	取视口宽度与高度的较小值
方向敏感性	始终基于宽度，无视口方向	根据视口方向动态调整（宽度或高度取较小值）
竖屏（如手机）	基于宽度（如 375px 宽度的 50vw = 187.5px）	基于高度（如 812px 高度的 50vmin = 406px）
代码示例	width: 50vw; （视口宽度的 50%）	height: 50vmin; （视口较小边的 50%）
优势	精准控制横向尺寸	自动适应不同方向，适合全屏或动态比例布局
劣势	无法响应视口高度变化	在宽屏设备上可能不如 vw 直观

场景：如何需要照顾手机端横屏状态，可以使用vmin，比如B站全部vmin



# Contents

## 01.移动端知识

viewport

## 02.vw/vmin适配方案

case

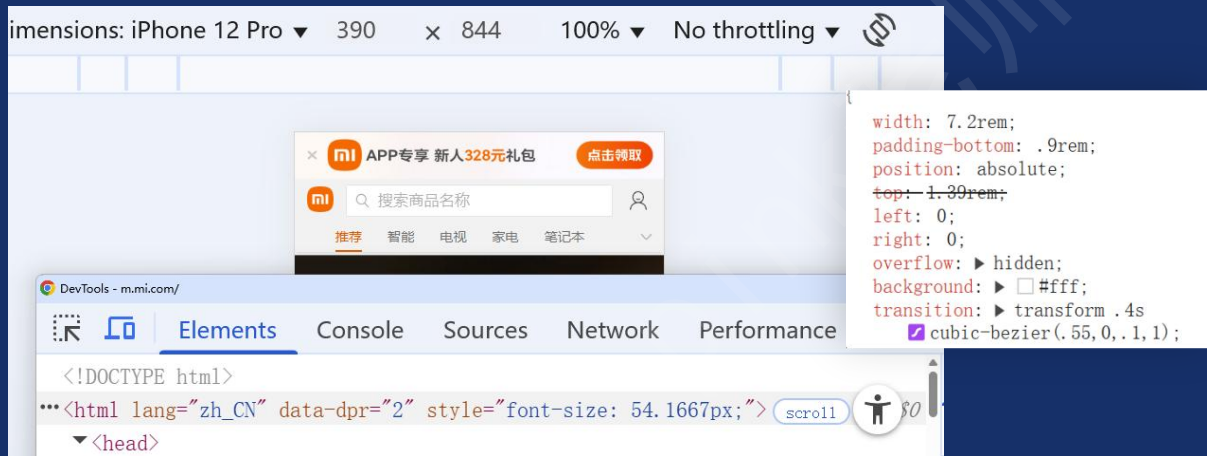
## 03.rem适配方案

case

## 移动端适配

rem单位适配:

通过动态修改根元素 (html) 的 `font-size`, 使得页面元素基于 rem 的布局能够自适应不同屏幕尺寸。  
比如京东、淘宝、小米移动端使用rem适配。



# 移动端适配

rem单位适配:

动态修改html文字大小（基准值）方案:

## 方式1：媒体查询方案

```
html {  
  font-size: 20px;  
  font-size: 5.3333vw  
}  
  
@media screen and (max-width: 320px) {  
  html {  
    font-size: 17.06667px  
  }  
}  
  
@media screen and (min-width: 540px) {  
  html {  
    font-size: 28.8px  
  }  
}
```

京东写法

## 方式2：js方式引入（推荐）

阿里巴巴手淘团队开发的 移动端适配方案 [Flexible.js](#)  
通过动态调整根元素（<html>）的 font-size，结合 rem 单位实现页面自适应。

```
<script src="./js/flexible.js"></script>
```

```
// set 1rem = viewWidth / 10  
function setRemUnit() {  
  var rem = docEl.clientWidth / 10  
  docEl.style.fontSize = rem + 'px'  
}
```

拼多多、淘宝写法

# 总结

## 1. rem单位跟谁的文字大小有关系？

html 根元素 文字大小有关。跟父元素等其他元素无关

## 2. 如何动态修改html 文字大小？

媒体查询或者 利用 Flexible.js（提倡）动态修改文字大小



## 移动端适配

rem单位适配：

动态修改html文字大小（基准值）方案：



设计稿转换公式：

rem值 = 设计稿元素尺寸 / html的font-size（基准值）

$$(106 / 37.5) = 2.826 \text{ rem}$$

# 移动端适配

rem单位适配:

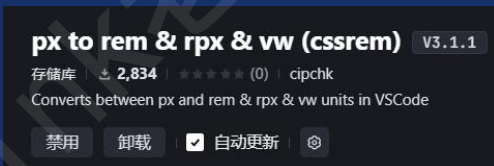
动态修改html文字大小（基准值）方案：

## 方式3：使用PostCSS插件（最佳实践）

- ❑ less/sass代码直接写px
- ❑ 使用 postcss-pxtorem 插件，打包发布的时候会统一自动转换为rem。
- ❑ 但是需要脚手架知识，后续Vue阶段讲。

## 方式4：转换插件（目前阶段）

1. Flexible.js 实现动态修改 html 根字号大小
2. 利用插件实现 px 转换为 rem单位



### Cssrem: Root Font Size

基准font-size（单位：`px`），default: 16

flexible.js 基准值算法：

$$375（设计稿） / 10 = 37.5$$

## 综合案例

优医问诊项目：

< 注册

密码登录 短信验证码登录 >

请输入手机号

请输入密码

☐ 我已同意 用户协议 及 隐私条款

登录

忘记密码?

第三方登录

登录页

# 移动端适配总结

## 1. 视口标签

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

移动端页面宽度根据**设备宽度**来走



# 移动端适配总结

## 2. 物理像素和CSS像素

设计师设计稿是按照物理像素设计。

我们前端开发工程师要按照CSS 像素开发代码。

所以我们设计稿拿到后，设置为@2X 或者 @3X 就可以了



# 移动端适配总结

## 3. 移动端适配方案

技术栈：

1. flex + vw 实现适配。 px 利用插件完成vw转换
2. flex + rem 实现适配。 px 利用插件完成rem转换



对比维度	VW 适配方案	REM 适配方案
基础单位	1vw = 视窗宽度的 1% (如视窗宽度 375px, 则 1vw = 3.75px)	1rem = 根元素 ( ) 的 font-size 值 (需动态计算)
实现方式	直接使用 vw 单位, 无需 JavaScript	通过 JavaScript 或媒体查询动态调整根元素的 font-size
设计稿换算	元素尺寸 = 设计稿尺寸 / 设计稿宽度 * 100vw (例: 设计稿 375px, 元素 100px → 26.667vw)	元素尺寸 = 设计稿尺寸 / 基准字体大小 (例: 基准 20px, 元素 100px → 5rem)
动态适配	视窗宽度变化时, vw 单位自动按比例缩放	根元素字体动态调整, 所有 rem 单位同步缩放
优先级	优先使用	根据需求使用

## 移动端适配总结

作业：



我的