移动端开发

1 移动端适配

<meta name="viewport" content="width=device-width, initial-scale=1.0, user-scalable=no, maximum-scale=1, minimum-scale=1">

获取页面宽度

window.innerWidth

document.documentElement.clientWidth

document.documentElement.getBoundingClientRect().width

获取dpr

window.devicePixelRatio

box-sizing

宽高为内容尺寸，不包括间距和边框 box-sizing: content-box; padding: 10px; border: 10px solid #ccc;

宽高 包括内容、间距和边框（默认） box-sizing: border-box; padding: 10px; border: 10px solid #ccc;

移动端初始样式

\* {

box-sizing: border-box;

padding: 0;

margin: 0;

}

图标字体

阿里巴巴图标字体库iconfont https://www.iconfont.cn

需要修改url路径

习惯使用<i></i>标签引用 <i class="iconfont icon-name"></i>

flex弹性布局

display: flex | inline-flex;

flex：父容器有固定宽度

inline-flex：父容器靠子元素撑起宽度

弹性容器属性

布局方向 flex-direction: row | row-reverse | column | column-reverse;

是否换行 flex-wrap: nowrap | wrap | wrap-reverse;

简写 flex-flow: direction wrap; 默认row nowrap

主轴对齐 justify-content: flex-start | flex-end | center | space-between（两侧） | space-around（平分）;

交叉轴对齐 align-items: flex-start | flex-end | center | baseline | stretch（默认，使用设置高度或占满）;

交叉轴多行对齐 align-content: flex-start | flex-end | center | space-between | space-around | stretch

弹性内容属性

对齐排序 order: number;

是否按比例扩张 flex-grow: number; 默认0，不扩张

是否按比例收缩 flex-shrink: number; 默认1，收缩

宽度 flex-basis: auto | width; 主轴宽度

简写 flex: grow shrink basis;

单独项目对齐 align-self: auto | flex-start | flex-end | center | baseline | stretch;

媒体查询

匹配满足条件的屏幕

type: all、screen、print、speech（阅读器）

逻辑与或非: and、,（逗号）、not

媒体特征表达式：width、max-width、min-width、-webkit-device-pixel-ratio、-webkit-max-device-pixel-ratio、-webkit-min-pixel-ratio、orientation: landscape（横屏） | portrait（竖屏）

@media screen and (min-width: 900px), screen and (max-width: 1024px) {

body {

background-color: red;

}

}

缩放窗口自动换行适配

适配区间

xs < 576px

sm 576px ~ 768px

md 768px ~ 992px

lg 992px ~ 1200px

xl > 1200px

.row {

width: 100%;

display: flex;

flex-wrap: wrap;

}

.col { width: 100%; }

@media (min-width: 576) { .col { width: 50%; }}

@media (min-width: 768) { .col { width: 25%; }}

@media (min-width: 992) { .col { width: 16%; }}

@media (min-width: 1200) { .col { width: 8%; }}

移动端常用单位

px、%、em（参照自身或继承的字体大小，推荐缩进行首时使用）、rem（参照html根元素字体大小，推荐布局时使用）、vw（可视窗口宽度）、vh（可视窗口高度）

计算rem适配尺寸（1rem = 20px）

<script>

setRemUnit();

function setRemUnit() {

var el = document.documentElement;

var vw = el.clientWidth;

el.style.fontSize = vw / 375 \* 20 + 'px';

}

</script>

取消图片间距

img { vertical-align: top; width: 100%; border: none; }

2 响应式布局

实现方式：float和flex布局

实现原理：媒体查询（@media）、列样式（col-1~col-12、col-、col-sm-、col-md-、col-lg-、col-xl-）

优点：一个网站能够兼容多种终端，不用为每种终端开发一个网站。

缺点：需要兼容各种终端，工作量大，效率低下。代码累赘，加载时间加长。

适用：结构比较简单的展示性页面。

不适用：结构复杂，交互多的功能页面。

3 移动端屏幕适配

尺寸单位多使用rem、%，字体单位使用px，宽高等比例缩放

计算屏幕适配缩放单位（rem）

// 通过自动执行的匿名函数封装，避免污染全局变量

(function() {

// 使用严格编码模式

'use strict';

// 通用方式实现：实时计算设备缩放比例，解决1px边框问题

var viewportEl = document.querySelector('meta[name="viewport"]');

var dpr = window.devicePixelRatio || 1;

var maxWidth = 540;

var minWidth = 320;

dpr = dpr >= 3 ? 3 : (dpr >= 2 ? 2 : 1);

document.documentElement.setAttribute('data-dpr', dpr);

document.documentElement.setAttribute('max-width', maxWidth);

document.documentElement.setAttribute('min-width', minWidth);

var scale = 1 / dpr;

var content = 'width=device-width, initial-scale=' + scale + ', maximum-scale=' + scale + ', minimum-scale='+ scale + ', user-scalale=no';

if (viewportEl) {

viewportEl.setAttribute('content', content);

} else {

viewprotEl = document.createElement('meta');

viewportEl.setAttribute('name', 'viewport');

viewportEl.setAttribute('content', content);

document.head.appendChild(viewportEl);

}

// 简单实现方式（不考虑1px边框）

// 初始化计算

setRemUnit();

// 绑定浏览器窗口改变事件，随时计算

window.addEventListener('resize', setRemUnit());

function setRemUnit() {

// 取得视口宽度

var viewWidth = document.documentElement.getBoundingClientRect().width || window.innerWidth;

// 设置缩放系数

var ratio = 18.75;

// 设置视口宽度最大和最小值

if (maxWidth && (viewWidth / dpr > maxWidth)) {

viewWidth = maxWidth \* dpr;

} else {

viewWidth = minWidth \* dpr;

}

// 计算缩放尺寸

document.documentElement.style.fontSize = viewWidth / ratio + 'px';

}

})();

适配布局

固定顶部和底部菜单，中间滚动

.header-container { height: 2.5rem; }

.main-container { padding: 2.5rem 0; }

.tabbar-container { height: 2.5rem; }

html, body { width: 100%; height: 100%; }

.header-container, .tabber-container { position: fixed; left: 0; z-index: 1000; width: 100%; }

.header-container { top: 0; }

.tabber-container { bottom: 0; }

4 移动端事件

5 移动调试和终端检测

6 常见问题

html5兼容性查询Can I use https://caniuse.com

第三方库html5shiv

js兼容性检测

// 使用特性检测，并添加浏览器前缀

var requestAnimationFrame = window. requestAnimationFrame ||

window.webkitRequestAnimationFrame ||

window.mozRequestAnimationFrame ||

window.oRequestAnimationFrame ||

window.msRequestAnimationFrame ||

// 自定义实现

function (fn) {

setTimeout(fn, 16);

}

requestAnimationFrame(function() { ... });

css兼容性

使用第三方检测库Modernize生成兼容代码

支持时使用 .flexbox header { display: flex; justify-content: center; align-items: center; }

不支持时使用 .no-flexbox header { text-align: center; line-height: 50px; }

动画兼容性

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | dom | canvas |
| js | setInterval | √ | √ |
| setTimeout | √ | √ |
| requestAnimationFrame | √√ | √ |
| css3 | transition | √√√√ | x |
| animation | √√√ | x |

动画代码生成 Animate.css

单击300ms延迟（区分双击放大）

部分浏览器关闭特性

<meta name="viewport" content="width=device-width" ...>

第三方库fastclick屏蔽特性

if ('addEventListener' in document) {

document.addEventListener('DOMContentLoaded', function() {

FastClick.attach(document.body);

}, false);

}

文字溢出省略

单行（需外层容器设置布局）

.text-ellipsis {

overflow: hidden;

text-overflow: ellipsis;

white-space: nowrap;

}

多行（高度自适应，webkit兼容性更好）

.multiline-ellipsis {

overflow: hidden;

text-overflow: ellipsis;

display: -webkit-box;

-webkit-line-clamp: 2;

-webkit-box-orient: vertical;

white-space: normal !important;

word-wrap: break-word;

}

居中

方法1（浏览器兼容时推荐使用）

display: flex;

justify-content: center;

align-items: center;

方法2（容器）

position: absolute;

left: 50%;

top: 50%;

transform: translate(-50%, -50%);

方法3（块元素文字）

display: block;

text-align: center;

padding: 30px 0;

方法4

单行文字

display: inline-block;

width: 300px;

height: 100px;

text-align: center;

line-height: 100px;

多行文字

每一行都居中 text-align: center;

多行文字整体居中 display: flex; justify-content: center;

方法5（容器：反向设置宽高一半坐标）

position: absolute;

left: 50%;

top: 50%;

margin-left: -150px;

margin-top: -50px;

7 移动性能优化

8 其他

监听视口变化 window.onresize();

获取视口的宽度 window.innerWidth;

获取视口的高度 window.innerHeight;