黑马程序员-济南校区-前端与移动开发-技术面试宝典-初版

目录

一. HTML+CSS.....1
二. 移动端开发.....4
三. JS.....6
四. vue.....13
五. 小程序.....18

一. HTML+CSS

1. css3 新特性有哪些

【参考答案】

实现圆角（border-radius）
阴影（box-shadow）
文字加特效（text-shadow）
线性渐变（gradient）
旋转（transform）
媒体查询，多栏布局等

2. 如何清除浮动

【参考答案】

1:给父容器加 overflow:hidden
2: 空标签：在父容器最后添加一个 div 设置样式：display:block;visibility:hidden;clear:both(一定要加);
3: 伪元素:

```
.clearfix::after{
    display:none;
    content:'';
    clear:both;
    width:0;
    height:0;
    zoom:1(兼容 IE)
}
```

4:给父元素加高度

3. H5 的新增特性有哪些

【参考答案】

用于绘画的 canvas 元素
用于媒介回放的 video 和 audio 元素
对本地离线存储的更好的支持
新的特殊内容元素，比如 article、footer、header、nav、section
新的表单控件，比如 calendar、date、time、email、url、search

4. 实现 div 垂直居中有几种方式，请列举

【参考答案】

绝对定位实现垂直居中
定位+transform: translate(-50%, -50%)实现水平+垂直居中
fixed 定位居中
flex 弹性布局居中

5. 行内元素与块级元素有什么区别，分别列举常用的行内元素和块级元素

【参考答案】

①行内元素和其他行内元素都会在一条水平线上排列；块级元素会在新的一行开始排列，各个块级元素独占一行，垂直向下排列

②行内元素无法设置宽高，宽度高度随文本内容的变化而变化，但可以设置行高（line-height），同时在设置外边距 margin 上下无效，左右有效，内填充 padding 上下无效，左右有效；块级元素可以设置宽高，并且宽度高度以及外边距，内填充都可随意控制

③块级元素可以包含行内元素和块级元素，还可以容纳内联元素和其他元素；行内元素不能包含块级元素，只能容纳文本或者其他行内元素

6. 谈谈你对 css 浮动（float）与定位（position）的理解

【参考答案】

关于浮动：默认为 none,对应标准流的情况。当 float : left;时，元素就会向其父元素的左侧靠紧，脱离标准流，同时宽度不再伸展至充满父容器，而是根据自身内容来确定。

关于定位：position 取值有 static absolute relative fixed;

static 是默认的,即标准流排下来,就是 static 定位方式

fixed 相对于浏览器窗口定位的，这个会脱离标准流，一般[回到顶部]这种按钮就是 fixed 做的

relative 照自己原有的文档流位置定位，不会脱离原有的文档流

absolute 就是相对于父元素的非默认定位元素位置定位，这个会脱离标准流

7. 列举 5 种 css 鼠标（hover）效果

【参考答案】

hover 可以实现很多的 css 效果，例如：层叠与放大、跳跃、淡入淡出、旋转等

8. h5 自定义属性格式是什么，简单列举几种在日常项目中的用法

【参考答案】

形式：html 标签中的 data-* 属性
用法：
① 请求后台接口需要携带上一个请求的返回值，可以使用该属性保存
② 保存在自定义属性，方便其他程序获取

9. 在项目中如何调用自定义字体

【参考答案】

可以使用 CSS3 中的模块：@font-face 将自定义字体嵌入到网页中

10.margin 和 padding 对元素会有什么影响?

【参考答案】

行内元素设置外边距 margin 上下无效，左右有效，内填充 padding 上下无效，左右有效；
块级元素可任意设置，但会改变盒子原有大小

11.css 中哪些属性可以继承?

【参考答案】

1. 可继承的样式： font-size font-family color, text-indent;
2. 不可继承的样式：border padding margin width height;

12.响应式页面中需要考虑是主要分辨率有哪些（需要兼容 PC、平板、手机）

【参考答案】

手机：max-width:600px
平板：min-width:600px max-width:960px
PC 端：min-width:960px

13.实现元素隐藏的方式

【参考答案】

```
opacity: 0
display: none
visibility: hidden
position: -2000px
```

14.visibility: hidden 和 display: none 的区别

【参考答案】

visibility: hidden: 将元素隐藏，但是在网页中该占的位置还是占着
display: none 将元素的显示设为无，即在网页中不占任何的位置

15.列表单 form 的属性列举 3 个?

【参考答案】

```
methood=”get/post
action=url
target=”_blank/_parent“(前一个重新打开一个页面，后一个直接本页面跳转)
```

16.怎么理解盒模型

【参考答案】

标准盒模型：盒模型的宽高只是内容（content）的宽高，box-sizing:border-box; 转 ie 盒模型;

ie 怪异盒模型：宽高是内容(content)+填充(padding)+边框(border)的总宽高，box-sizing:content-box; 转标准盒模型’

二．移动端开发

1. 移动端怎么进行适配

【参考答案】

流式布局（百分比布局）、伸缩布局(flex)、rem 布局（均配合媒体查询使用）

2. 实现左边固定，右边自适应

【参考答案】

参考思否网站：七种实现左侧固定，右侧自适应两栏布局的方法

3. rem 和 em 有什么区别

【参考答案】

em:em 的大小决定于当前的元素是否设置了 font-size,如果设置了就根据当前的计算，如果当前没有，看父元素是否有，父元素有，就根据父元素计算，父元素没有，就依次查找，直到找到 body,如果 body 都没有，那么就根据浏览器默认的 font-size 计算

rem: rem 是通过设置 html 根元素的字体大小

4. 简述你对 flex 布局的了解

【参考答案】

① flex 是 Flexible Box 的缩写，意为"弹性布局"，然后去介绍 flex 有主轴和侧轴 2 条轴，主轴是 justify-content,侧轴是:align-items，然后再说怎么将一个容器设置是弹性盒子，给盒子添加 display:flex,之后在和下面的容器的几个属性结合一下，简单介绍一下

②容器的属性

flex-direction: row | row-reverse | column | column-reverse; 属性决定主轴的方向（即项目的排列方向）

flex-wrap: nowrap | wrap | wrap-reverse; 默认情况下，项目都排在一条线（又称"轴线"）上。flex-wrap 属性定义，如果一条轴线排不下，如何换行。

flex-flow: <flex-direction> || <flex-wrap>; flex-flow 属性是 flex-direction 属性和 flex-wrap 属性的简写形式，默认值为 row nowrap

justify-content: flex-start | flex-end | center | space-between | space-around; justify-content 属性定义了项目在主轴上的对齐方式。

align-items: flex-start | flex-end | center | baseline | stretch; align-items 属性定义项目在交叉轴上如何对齐。

align-content: flex-start | flex-end | center | space-between | space-around | stretch; align-content 属性定义了多根轴线的对齐方式。如果项目只有一根轴线，该属性不起作用。

5. 移动端开发常见的兼容性问题

【参考答案】

①安卓浏览器看背景图片，有些设备会模糊。

②图片加载较慢，针对这种情况，手机开发一般用 canvas 方法加载

③假如手机网站不用兼容 IE 浏览器，一般我们会使用 zeptojs。zeptojs 内置 Touch

events 方法

④防止手机中网页放大和缩小,设置 meta 中的 viewport,例如:<meta name="viewport" content="width=device-width, initial-scale=1.0, maximum-scale=1.0, user-scalable=0" />

⑤上下拉动滚动条时卡顿、慢。

解决: body {
-webkit-overflow-scrolling: touch;
overflow-scrolling: touch;
}

Android3+和 iOS5+支持 CSS3 的新属性为 overflow-scrolling

⑥长时间按住页面出现闪退

解决: element {
-webkit-touch-callout: none;
}

三. JS

1. js 中关于数组常用方法

【参考答案】

不要直接上来就是 push, shift, pop 在这里可以说一些比较别人不经常说的, 比方说: splice, join, map, filter 等等, 还有 es6 的 find, findIndex 等

在回答方法的时候顺带着把该方法的作用也说一下

2. js 中关于字符串的常用方法

【参考答案】

split(), substring(), substr(), concat()等方法, 在回答方法的时候顺带着把该方法的作用也说一下

3. 数组方法那些会改变原素组

【参考答案】

shift: 将第一个元素删除并且返回删除元素, 空即为 undefined

unshift: 向数组开头添加元素, 并返回新的长度

pop: 删除最后一个并返回删除的元素

push: 向数组末尾添加元素, 并返回新的长度

reverse：颠倒数组顺序

sort：对数组排序

splice:splice(start,length,item)删，增，替换数组元素，返回被删除数组，无删除则不返回

4. 什么是事件委托？实际运用场景？

【参考答案】

事件委托就是利用事件冒泡，只指定一个父类元素绑定事件，就可以管理所有子类元素

运用场景：

① ajax 局部刷新区域

② 绑定层级比较低的时候，不在 body 上绑定

③ 定次数较少的时候，把多个事件绑定合并到一次事件委托中，由这个事件委托的回调，来进行分发

5. 数组如何去重

【参考答案】

es5：这个实现方法很多，我列举其中的 2 种：

第一种：var arr1 = [1, 23, 1, 1, 1, 3, 23, 5, 6, 7, 9, 9, 8, 5];

```
function rep(arr) {
  var ret = [];
  for (var i = 0; i <arr1.length; i++) {
    if (ret.indexOf(arr1[i]) ==-1) {
        ret.push(arr1[i]);
    }
}
return ret;
}
var result = rep(arr1);
console.log(result)
```

第二种：var arr = [1, 2, 3, 4, 5, 1, 2, 3];

```
var r = arr.filter(function (element, index, self) {
  return self.indexOf(element) === index;
});
console.log(r);
```

es6：var arr =Array.from(new Set(数组))

6. typeof 和 instanceof 的区别

【参考答案】

在 js 中，判断一个基本数据的变量的类型会常用 typeof,但是在判断复杂数据类型的时候，无论是什么样的类型都会输出'object'，这就需要用到 instanceof 来检测某个对象是不是另一个对象的实例

7. 获取页面中元素的 js 方法

【参考答案】

document.getElementById(id 名)，获取的是单个元素
document.querySelector(可以是 id,可以是类，标签，等等)，获取的是单个元素
document.querySelectorAll(选择器都可以），获取的是一类元素，是个伪数组
document.getElementsByTagName(标签的名字） 获取的是一类元素，是个伪数组

8. 谈谈你对原型/原型链的理解

【参考答案】

每个构造函数一旦创建都有 prototype 指针指向它的原型对象（构造函数.prototype）。而原型对象（构造函数.prototype）会默认生成一个 constructor 指针又指向构造函数。

9. 什么是闭包，在哪用过闭包，闭包的使用场景是什么

【参考答案】

概念： 闭包就是能够读取其他函数内部变量的函数。
在哪使用过：需要获取函数内部变量时
应用场景：闭包可以用在许多地方。它的最大用处有两个，一个是可以读取函数内部的变量，另一个就是让这些变量的值始终保持在内存中

10. 谈谈对 this 的理解

【参考答案】

① 普通函数中，this 指向全局对象 window
② 定时器中，this 指向调用对象 window
③ 构造函数中，this 指向当前实例化的对象
④ 事件处理函数中，this 指向事件触发对象

在 js 中一般理解就是谁调用这个 this 就指向谁

11. 怎么改变 this 的指向

【参考答案】

call、apply、bind 三者均为改变 this 指向的方法

12. call/apply/bind 的区别

【参考答案】

call（无数个参数），第一个参数：改变 this 指向；第二个参数：实参；使用之后会自动执行该函数

apply（两个参数），第一个参数：改变 this 指向；第二个参数：数组（里面为实参）；使用时候会自动执行函数

bind（无数个参数），第一个参数：改变 this 指向；第二个参数之后：实参；返回值为一个新的函数；使用的时候需要手动调用下返回 的新函数（不会自动执行）

13. BOM 中常用的 API 有哪些

【参考答案】

navigator：window 中封装浏览器属性和配置信息的对象

cookieEnabled：识别浏览器是否启用 cookie，返回值 true/false

userAgent：保存了浏览器名称和版本的字符串

plugins：保存浏览器中所有插件信息的集合，每个 plugin 对象的 name 属性保存了插件的名称

screen：保存显示屏信息的对象

history：保存窗口的历史记录栈

location：指代当前窗口正在访问的 url 地址对象

location.href:保存了当前窗口正在访问的 url 地址,设置 href 属性为新 url，会在当前窗口打开新 url

location.search()：获取 url 上面？后面的参数

location.reload()：刷新当前页面

location.reload(true/false):true —— 无论是否更改,都获取更新;false —— 被修改的页面，重新获 取，未被修改的页面，从缓冲获取

定时器：让程序按指定时间间隔，自动执行任务，任务是所有定时器的核心

14. prop 和 attr 的区别

【参考答案】

对于 HTML 元素本身就带有的固有属性，在处理时，使用 prop 方法。

对于 HTML 元素我们自己自定义的 DOM 属性，在处理时，使用 attr 方法。

15. null 和 undefined 的区别

【参考答案】

null 是一个表示”无”的对象，转为数值时为 0；undefined 是一个表示”无”的原始值，转为数值时为 NaN。

当声明的变量还未被初始化时，变量的默认值为 undefined。

null 用来表示尚未存在的对象，常用来表示函数企图返回一个不存在的对象。

undefined 表示”缺少值”，就是此处应该有一个值，但是还没有定义。

16. 深拷贝和浅拷贝的区别

【参考答案】

浅拷贝只复制指向某个对象的指针，而不复制对象本身，新旧对象还是共享同一块内存。

深拷贝会另外创造一个一模一样的对象，新对象跟原对象不共享内存，修改新对象不会改到原 对象。

深拷贝的方法：

① 递归拷贝

② 使用 Object.create()方法

③ jquery 有提供一个$.extend 也可以实现

④ 函数库 lodash，也有提供 cloneDeep 用来实现

17. 谈谈对 promise 的理解

【参考答案】

Promise 是异步编程的一种解决方案。所谓 Promise，简单说就是一个容器，里面保存着某个未来才会结束的事件（通常是一个异步操作）的结果。从语法上说，Promise 是一个对象，从它可以获取异步操作的消息。

Promise 提供统一的 API，各种异步操作都可以用同样的方法进行处理。

特点：

① 自己身上有 all、reject、resolve 这几个方法

② 原型上有 then、catch 等方法

③ 一旦建立，就无法取消，这是它的缺点

18. ES6 常用语法有哪些

【参考答案】

① Let：声明的变量，只作用于使用了 let 命令的代码块

② const ：声明一个常量，声明过后，就不可修改其值(会报错)

③ 解构性赋值

④ 箭头函数：用法的两个明显的优点：函数的写法更加简洁；箭头函数内部没有自己的 this 对象，而是全部继承外面的，所以内部的 this 就是外层代码块的 this。

⑤ 字符串模板：ES6 中允许使用反引号 ` 来创建字符串，此种方法创建的字符串里面可以包含由美元符号加花括号包裹的变量${vraible}

⑥ Proxy：Proxy 可以监听对象身上发生了什么事情，并在这些事情发生后执行一些相应的操作

⑦ 循环：for of 实现数组的遍历（以及 keys(),value(),entries()等方法）

19. Jquery 在项目中常用的方法有哪些

【参考答案】

使用 JQuery 获取元素

JQ 获取元素的内容：$("xxxx").html()

JQ 中的事件操作

JQ 操作控件属性

JQ 操作 css

ajax 操作

jq 函数级插件--$.fn.extend-添加成员函数/$.extend-添加静态方法

20. jquery.extend 与 jquery.fn.extend 的区别是什么

【参考答案】

jQuery 为开发插件提拱了两个方法，分别是：

① jquery.fn.extend();用来扩展 jquery 实例

② jquery.extend();用来扩展 jquery 对象本身

21. json 字符串和和 json 对象如何相互转化?

【参考答案】

JSON.parse() 从一个字符串中解析出 json 对象
JSON.stringify() 从一个对象中解析出字符串

22. 跨域的几种解决方式

【参考答案】

① 通过 jsonp 跨域
② CORS
核心思想：在服务器端通过检查请求头部的 origin，从而决定请求应该成功还是失败。具体的方法是在服务端设置 Response Header 响应头中的 Access-Control-Allow-Origin 为对应的域名，实现了 CORS（跨域资源共享），这里出于在安全性方面的考虑就是尽量不要用 *，但对于一些不重要的数据则随意。
③ 后端配置，允许跨域

23. 前端开发过程中使用过哪些优化手段？（看雅虎 14 条性能优化原则）

【参考答案】

(1) 减少 http 请求次数：CSS Sprites，JS、CSS 源码压缩、图片大小控制合适；网页 Gzip，CDN 托管，data 缓存 ，图片服务器。
(2) 前端模板 JS+数据，减少由于 HTML 标签导致的带宽浪费，前端用变量保存 AJAX 请求结果，每次操作本地变量，不用请求，减少请求次数
(3) 用 innerHTML 代替 DOM 操作，减少 DOM 操作次数，优化 javascript 性能。
(4) 当需要设置的样式很多时设置 className 而不是直接操作 style。
(5) 少用全局变量、缓存 DOM 节点查找的结果。减少 IO 读取操作。
(6) 避免使用 CSS Expression（css 表达式）又称 Dynamic properties(动态属性)。
(7) 图片预加载，将样式表放在顶部，将脚本放在底部 加上时间戳。
(8) 避免在页面的主体布局中使用 table，table 要等其中的内容完全下载之后才会显示出来，显示比 div+css 布局慢。

四. vue

1. vue 有哪几种导航钩子

【参考答案】

① 全局导航钩子：router.beforeEach(to,from,next)，作用：跳转前进行判断拦截。

② 组件内的钩子；

③ 单独路由独享组件

2. vue 全家桶是什么

【参考答案】

vue 全家桶包括：vue-router、vuex、axios，再加上构建工具 vue-cli，sass 样式和组件库，就是一个完整的 vue 项目的核心构成。

概括起来就是：项目构建工具 + 路由 + 状态管理 + http 请求工具

3. v-show 和 v-if 的区别

【参考答案】

① v-if 是通过控制 dom 节点的存在与否来控制元素的显隐；v-show 是通过设置 DOM 元素的 display 样式，block 为显示，none 为隐藏；

② v-if 切换有一个局部编译/卸载的过程，切换过程中合适地销毁和重建内部的事件监听和子组件；v-show 只是简单的基于 css 切换；

③ v-if 是惰性的，如果初始条件为假，则什么也不做；只有在条件第一次变为真时才开始局部编译；v-show 是在任何条件下（首次条件是否为真）都被编译，然后被缓存，而且 DOM 元素保留；

④ v-if 有更高的切换消耗；v-show 有更高的初始渲染消耗；

4. 谈谈对 mvvm 的理解

【参考答案】

mvvm 就是 vm 框架视图、m 模型就是用来定义驱动的数据、v 经过数据改变后的 html、vm 就是用来实现双向绑定

双向绑定:一个变了另外一个跟着变了，例如：视图一个绑定了模型的节点有变化，模型对应的值会跟着变

5. 谈谈对于 vuex 的理解，有哪些属性，运用场景是什么

【参考答案】

理解:

① vuex 可以理解为通过状态（数据源）集中管理驱动组件的变化

② 应用级的状态集中放在 store 中； 改变状态的方式是提交 mutations，这是个同步的事物； 异步逻辑应该封装在 action 中。

属性:

有五种，分别是 State、 Getter、Mutation 、Action、 Module

6. vue 的组件通信是如何实现的

【参考答案】

组件关系可分为父子组件通信、兄弟组件通信。

① 父组件向子组件:

通过 props 属性来实现

② 子组件向父组件:

子组件用$emit （）来触发事件，父组件用$on（）来监听子组件的事件。

父组件可以直接在子组件的自定义标签上使用 v-on 来监听子组件触发的自定义事件。

③ 兄弟之间的通信:

通过实例一个 vue 实例 Bus 作为媒介，要相互通信的兄弟组件之中都引入 Bus，之后通过分别调用 Bus 事件触发和监听来实现组件之间的通信和参数传递。

7. 谈一下对 vue 双向数据绑定原理的理解

【参考答案】

vue.js 是采用数据劫持结合发布者-订阅者模式的方式，通过 Object.defineProperty()来劫持各个属性的 setter，getter，在数据变动时发布消息给订阅者，触发相应的监听回调。

具体步骤:

第一步：需要 observe 的数据对象进行递归遍历，包括子属性对象的属性，都加上 setter 和 getter，这样的 话，给这个对象的某个值赋值，就会触发 setter，那么就能监听到了数据变化。

第二步：compile 解析模板指令，将模板中的变量替换成数据，然后初始化渲染页

面视图，并将每个指令对应的节点绑定更新函数，添加监听数据的订阅者，一旦数据有变动，收到通知，更新视图。

第三步：Watcher 订阅者是 Observer 和 Compile 之间通信的桥梁，主要做的事情是:1、在自身实例化时往属性订阅器(dep)里面添加自己 2、自身必须有一个 update()方法 3、待属性变动 dep.notice()通知时，能调用自身的 update()方法，并触发 Compile 中绑定的回调，则功成身退。

第四步：MVVM 作为数据绑定的入口，整合 Observer、Compile 和 Watcher 三者，通过 Observer 来监听自己的 model 数据变化，通过 Compile 来解析编译模板指令，最终利用 Watcher 搭起 Observer 和 Compile 之间的通信 桥梁，达到数据变化 -> 视图更新；视图交互变化(input) -> 数据 model 变更的双向绑定效果。

8. 简述一下 vue 的生命周期的理解

【参考答案】

总共分为 8 个阶段创建前/后，载入前/后，更新前/后，销毁前/后。

① 在 beforeCreated 阶段，vue 实例的挂载元素$el 和数据对象 data 都为 undefined，还未初始化。

② 在 created 阶段，vue 实例的数据对象 data 有了，$el 还没有。

③ 在 beforeMount 阶段，vue 实例的$el 和 data 都初始化了，但还是挂载之前为虚拟的 dom 节点，data.message 还未替换。

④ 在 mounted 阶段，vue 实例挂载完成，data.message 成功渲染。

⑤ 当 data 变化时，会触发 beforeUpdate 和 updated 方法。

⑥ 在执行 destroy 方法后，对 data 的改变不会再触发周期函数，说明此时 vue 实例已经解除了 事件监听以及和 dom 的绑定，但是 dom 结构依然存在

9. vue 是如何做 IE 兼容

【参考答案】

安装并引入 babel-polyfill 处理器包、es6-promise 包（一般是不支持 es6 语法及 promise，无法发送 axios 请求）

10.vue 深度监听对象中新老值如何保持不一样

【参考答案】

官方术语：观察 Vue 实例变化的一个表达式或计算属性函数。回调函数得到的参数为新值和旧值。表达式只接受监督的键路径。对于更复杂的表达式，用一个函数取代

理解：对于复杂的表达式，用一个函数取代。

实现：把复杂表达式或者对象，用 computed 属性包装一下，再来监听 computed 属性

11.vue 项目中使用过哪些第三方插件及使用的步骤

【参考答案】

插件有很多，随便列举几个：
vue-clipboard2（复制插件）
Moment（时间格式处理）
export2table（table 数据打印为 excel 表格）等。
步骤：
安装 -> 导入 -> 配置 -> 使用

12.怎么在使用 vue-router 切换页面时设置过渡动画

【参考答案】

分析：根据路有深度和不同的转场设置动画
① 首先我们在路由文件（router\index.js）中，在每个路径的路由元信息 meta 中设置个 depth 属性值，用来表示其路由的深度（层级）；
② 使用 animinate.css 提供的效果来绑定不同的 class
③ 然后通过 watch 监听 $route，根据 to、from 元信息中的 depth 值的大小，设置不同的跳转动画

13.vue 状态更改时，怎么让页面局部渲染

【参考答案】

在改变 data 中已定义好的属性值或对某个变量重新赋值时，可以通过使用 v-show 或 v-if 来实现局部渲染

14.怎么解决 spa 首屏加载比较慢的问题

【参考答案】

① 使用服务器端渲染首屏
② 将 js 中的 css 单独抽离出来，放在 css 文件中
③ 极限打包压缩静态文件，首屏图片等可以使用 lazy-load

15.在 vue 项目中遇到哪些坑（难题）

【参考答案】

① 使用 keep-alive 包裹的组件/路由，打开一次后 created 只会执行一次，有两种情况，一、如果要重新渲染部分数据，可以在 activated 中做处理；二、路由/组件重新重新 created，可以使用官方推荐的:key="key"，然后去改变 key 的值，组件就会重新挂载了

② beforeRouteEnter 中的 next 函数的执行时间是在组件 mounted 之后，因此需要在此处处理的数据要注意了

③ 网页刷新时 vuex 数据会丢失，需配合 localStorage 或 sessionStorage 使用，把必须数据先存后取

④ 对于权限及不确定路由，可以使用 addRoutes()，可以避免抖动

⑤ 熟练使用 es6 的数组 map、find、filter 等方法，对解构赋值、class 继承、promise，及 es7 中的 async 和 await

⑥ 使用 computed 替代 watch，computed 依赖于 data 属性的更改，是有缓存的

⑦ 通过 props 传递的值，不要在子组件去更改。开发中，如果直接更改 props，一、基本类型的值会报错，二、引用类型的值不会报错，但是不好去追溯数据的更改，很多人不太注意引用类型，可通过 computed 或 watch 去更改

⑧ 在 data 里调用 methods 的方法，可以在 data 里定义 let self = this，然后在使用 self.xx()进行调用

16.基于 vue 的组件库有哪些

【参考答案】

iView UI 组件库、Vux UI 组件库、Element UI 组件库、Mint UI 组件库、Bootstrap-Vue UI 组件库、Ant Design Vue UI 组件库、AT-UI UI 组件库、Vant UI 组件库、cube-ui UI 组件库等

17.computed 和 watch 的区别和运用的场景?

【参考答案】

computed: 是计算属性，依赖其它属性值，并且 computed 的值有缓存，只有它依赖的属性值发生改变，下一次获取 computed 的值时才会重新计算 computed 的值；
watch: 更多的是「观察」的作用，类似于某些数据的监听回调 ，每当监听的数据变化时都会执行回调进行后续操作；
运用场景：
当我们需要进行数值计算，并且依赖于其它数据时，应该使用 computed，因为可以利用 computed 的缓存特性，避免每次获取值时，都要重新计算；
当我们需要在数据变化时执行异步或开销较大的操作时，应该使用 watch，使用 watch 选项允许我们执行异步操作 (访问一个 API)，限制我们执行该操作的频率，并在我们得到最终结果前，设置中间状态。这些都是计算属性无法做到的

五. 小程序

1. mixins 和 extends 的区别

【参考答案】

extends 是创建一个子类，最终返回一个 vue 实例。一般在单独用 js 书写组件的时候使用。

而 mixins 选项是指定要混入的代码片段，vue 代码中的 script 部分。混入则可认为是 vue 版本的全局方法库，而且不怎么影响现有 vue 逻辑的一个特殊处理方式。通常用在业务逻辑相似但又不同的兄弟组件之间。

2. 小程序的生命周期有哪些

【参考答案】

onLaunch 生命周期回调——监听小程序初始化。
onShow 生命周期回调——监听小程序启动或切前台
onHide 生命周期回调——监听小程序切后台。
onError 错误监听函数。
onPageNotFound 页面不存在监听函数等

3. 小程序的双向数据绑定和 vue 的有什么不同

【参考答案】

小程序直接 this.data 的属性是不可以同步到视图的，必须调用：
小程序：

```
Page({
    data: {
        items: []
    },
    onLoad: function(options) {
        this.setData({
            items: [1,2,3]
    })
}})
```

Vue：

```
new Vue({
    data: {
        items: []
    },
```

```
        mounted () {
            this.items = [1, 2, 3]
    }})
```

4. 小程序开发过程中遇到过哪些兼容性问题

【参考答案】

① https 部署以及设置合法域名（必须设置 https）

② post 请求，json 数据格式转换；当我们向后台进行 post 请求的时候，当我们的传输数据的格式为 json 的时候，需要设置

‘content-type’：‘application/x-www-form-urlencoded’

③ wxss 文件中不支持本地图片

④ wxml 文件中可以使用三目运算符

⑤ this.setData 不可直接赋值

⑥ 微信小程序不能直接操作 DOM 树

5. 哪些方法可以提高小程序的加载速度

【参考答案】

可以使用延迟跳转和预加载