地址：<http://www.zlprogram.com/Show/11/AEBBFB4D.shtml>

#### 1、js 数据类型：

* 1. 基本类型：string、number、boolean、undefined、null、systembol
  2. 引用类型：object（function、object、array）

#### 2、类型检测

* 1. Typeof：检查变量的类型
  2. intanceof: 检测对象的类型
  3. Object.prototype.toString.call()：检测对象类型

#### 3、判断空数组

* 1. Arr.length === 0
  2. JSON.stringify(arr) === ‘[]’

#### 4、判读空对象

* 1. obj instanceof Object && Object.keys(obj).length === 0
  2. JSON.stringify(arr) === ‘{}’

#### 5、哪些非 boolean 值被强制转换为一个 boolean 时，它是 false ？

* 1. ""（空字符串）
  2. 0, -0, NaN （非法的 number ）
  3. null, undefined

#### 6、{} 和 [] 的 valueOf 和 toString 的结果是什么？

* 1. {} 的 valueOf 结果为 {} ，toString 的结果为 “[object Object]”
  2. [] 的 valueOf 结果为 [] ，toString 的结果为 “”

#### 7、null，undefined 的区别？

* 1. Null表示一个对象，对象值为空
  2. Undefined 表示声明一个变量，没有赋值
  3. Null typeof 是object， undefined typeof 是undefined
  4. Number(null) // 0; Number(undefined) // NaN
  5. Null 和 undefined toString 报错
  6. 检查null 和undefined的真实类型：Object.prototype.toString.call(null) // [obejct, null]

#### 8、offsetWidth/offsetHeight，clientWidth/clientHeight 与 scrollWidth/scrollHeight 的区别

* 1. offsetWidth/offsetHeight 返回值包含 content + padding + border，包括滚动条；
  2. clientWidth/clientHeight 可视区的尺寸，返回值只包含 content + padding，如果有滚动条，也不包含滚动条；
  3. scrollWidth/scrollHeight 实际尺寸，返回值包含 content + padding + 溢出内容的尺寸。

#### 9、什么是 Virtual DOM，为何要用 Virtual DOM？

* 1. Virtual DOM 是轻量级的js对象，模拟真实DOM元素及其属性
  2. 当组件状态改变时，Virtual DOM会被重新构建，与上一版本进行比较（diff算法），从而确定实际DOM需要更新的部分，然后将此部分更新到真实DOM。目的在于减少直接操作DOM带来的开销。

#### 10、对象的原生方法

* 1. Object.assign(): 复制对象可枚举属性
  2. Object.is()： 判断两个值是否一致
  3. Object.keys/values/entries: 返回键名、键值、键值对的数组

#### 数组的遍历方法

地址：<https://blog.csdn.net/qq_38261819/article/details/121415798>

* 1. 标准for循环；使用break （中断for循环，继续执行后面语句）和continue（跳出当前循环，继续执行下一轮循环），return （中断循环，不再执行后面语句）
  2. forEach((当前值, 当前索引,当前数组)=>{})；无法中途退出循环，只能用 return 退出本次回调，进行下一次回调。它总是返回 undefined 值,即使你 return 了一个值。（使用try catch，可以跳出循环）
  3. for-in（不推荐）会把继承链的对象属性都会遍历一遍，而且数组遍历不一定按次序； for-in 循环返回的是所有能通过对象访问的、可枚举的属性。
  4. for (variable of iterable)（ES6）可迭代 Array ，Map，Set，String 等（迭代的是值 value ）；在 for-of 中如果遍历中途要退出，可以使用 break 退出循环。
  5. map : 不能中断，返回修改后的数组
  6. Every： 遍历每一项，给定函数，判断是否都为true，则返回true，反之有一项为false，则返回false;
  7. Some：遍历每一项，给定函数，若有一项返回true，则返回true，反之，返回false；
  8. Reduce：遍历数组，每一项给定reducer函数（升序执行），将其结果汇总成单个返回
  9. Filter： 每一项给定函数，返回满足条件的数据
  10. Find: 找出第一个符合条件的数组元素，并返回该数组元素，如果没有符合，则返回undefined
  11. 备注：标准for循环性能最好

#### 12、为什么for循环比forEach性能高？

* 1. 循环控制：for循环可控制循环变量和逻辑，foreach迭代器函数是通过遍历数组每个元素来执行回调函数，无法在迭代过程中控制循环逻辑
  2. 在原型链上的调用次数：foreach 迭代器函数是Array对象的方法，每次迭代都需要通过原型链来调用；而for循环直接拖过索引获取数据元素，避免原型链上的多次调用，因此性能更加高效
  3. 中断、跳出：for循环可以会用break、continue跳出循环，foreach不行，只能通过抛异常的方式跳出

#### 13、改变原数组的方法

* 1. splice() 添加 / 删除数组元素

splice() 方法向/从数组中添加/删除项目，然后返回被删除的项目；

Eg: array.splice(index,howmany,item1,…,itemX)

index：必需。整数，规定添加/删除项目的位置，使用负数可从数组结尾处规定位置。

howmany：可选。要删除的项目数量。如果设置为 0，则不会删除项目。

item1, …, itemX： 可选。向数组添加的新项目。

返回值: 如果有元素被删除,返回包含被删除项目的新数组。

* 1. sort: 排序，参数排序的比较函数
  2. pop: 删除数组最后一个元素，并返回这个元素
  3. shift：删除数组的第一个元素，并返回这个元素
  4. push: 向数组的末尾添加一个或多个元素，并返回新的长度
  5. unshift: 可向数组的开头添加一个或更多元素，并返回新的长度
  6. reverse: 颠倒数组中元素的顺序

#### 14、不改变原数组的方法

* 1. slice： 浅拷贝数组，array.slice(begin, end)
  2. join: 数组转换成字符串，默认以逗号分隔；对象转字符串：JSON.stringify(obj)
  3. concat：合并数组，参数可以是具体的值、数组对象
  4. 扩展运算符 ： c = [...a, ...b]
  5. indexOf: 查找数组是否存在某个元素，返回下标，不存在返回-1，不能识别NaN;
  6. lastIndexOf：查找指定元素在数组中最后一个位置，返回下标，返回下标，不存在返回-1
  7. Includes: 数组是否包含某个值，返回boolean

#### 15、字符串方法

* 1. charAt：从字符串返回指定字符，charAt(下标)
  2. substring：返回字符串子集
  3. replace：替换字符 replace(正则表达式|string，function|string）
  4. toLowerCase / toUpperCase: 转换成小写/大写
  5. includes：查找是否包含指定值
  6. repeat：获取字符串重复n次
  7. startsWith：参数字符串是否在目标字符串的头部，返回布尔值
  8. endsWith：参数字符串是否在目标字符串的尾部，返回布尔值
  9. padStart/padEnd：允许将空字符串或其他字符串添加到原始字符串的开头或结尾

#### 16、闭包

* 1. 闭包就是能够读取其他函数内部变量的函数
  2. 用途：保存：缓存数据，延长作用域链

保护：形成私有作用域，保护里面私有变量不受外界干扰，避免全局污染

缺点：耗内存，耗性能，函数中的变量不能及时释放

c） 应用场景：循环li标签，onclick 每个节点的下标

#### 17、this的理解

#### 18、apply, call, bind 的区别

* 1. 均可以改变this的指向
  2. apply(对象，参数数组)，call，bind(对象，参数列表）
  3. Apply和call立即返回结果，bind返回的是函数，需要调用

#### 19、原型：

实例对象中有个属性 \_\_proto\_\_ ，是个对象，叫原型，不是标准的属性，浏览器使用的----->可以叫原型对象

构造函数中有一个属性 prototype ，也是个对象，叫原型，是标准属性，程序员使用—>可以叫原型对象

实例对象的 \_\_proto\_\_ 和构造函数中的 prototype 相等—> true

又因为实例对象是通过构造函数来创建的，构造函数中有原型对象prototype

实例对象的 \_\_proto\_\_ 指向了构造函数的原型对象 prototype

作用：共享数据和继承

#### 20、原型链

当对象查找一个属性的时候，如果没有在自身找到，那么就会查找自身的原型，如果原型还没有找到，那么会继续查找原型的原型，直到找到 Object.prototype 的原型时，此时原型为 null，查找停止。这种通过 原型链接的逐级向上的查找被称为原型链

#### 21、原型链和class继承

* 1. 组合继承：原型链+构造函数
     1. 单纯原型链继承：多个初始对象，传值初始值一致，无法向父类传递参数；
     2. 单纯借助构造函数继承：只能继承父类构造函数里面的属性和方法【Person.call(this, name, age)】，但父类的 prototype（原型）上的属性和方法不能继承。
     3. 缺陷：调用了两次父类的构造函数，造成性能上的浪费
  2. 寄生组合继承：
  3. Class继承：extends， super

#### 22、原型链，proto 和 prototype 的区别

对象拥有 \_\_proto\_\_ 属性，函数拥有 prototype 属性。所以：实例对象的 \_\_proto\_\_ 指向了构造函数的原型对象 prototype

函数是对象，对象不一定是函数。

#### 23、new操作符具体干啥的

创建一个新对象，新对象的 \_\_proto\_\_ 属性指向构造函数的原型对象prototype。

#### 24、事件的处理机制：捕获事件和冒泡事件

#### 25、事件流：三个阶段：事件捕获、处于目标、事件冒泡

#### 26、DOM事件模型：

* 1. 事件捕获
  2. 事件冒泡
  3. DOM事件流：先捕获，后冒泡；

#### 27、事件绑定的三种方式

* 1. Dom标签添加onclick // ← 绑定在事件冒泡阶段
  2. document.getElementById("xxx").onclick = function(){} // ← 绑定在事件冒泡阶段
  3. document.getElementById("xxx").addEventListener("click", function(e){}, false)

// ---> 为 false 时，绑定在事件冒泡阶段（默认下是绑定在冒泡阶段）

// ---> 为 true 时，绑定在捕获阶段

#### 28、阻止事件传播：stopPropagation(); 取消默认事件：preventDefault()

事件代理：通过事件冒泡给父元素添加事件监听，e.target指向引发触发事件的元素

#### 29、节流和防抖

* 1. 节流（throttle）:在用户动作时每隔一定时间（如200ms）执行一次回调（相当于技能刷新时间）
  2. 防抖（debounce）：在让在用户动作停止后延迟x ms再执行回调（相当于回城技能）

应用场景：

debounce 应用在搜索框的即时搜索（input 事件），避免用户狂按键盘导致的频繁请求

throttle 应用在监听 resize 改变布局或 onscroll 滚动

#### 30、mouseover 和 mouseenter 的区别

* 1. mouseover：

不论鼠标指针穿过被选元素或其子元素，都会触发 mouseover 事件。

支持事件冒泡。相对应 mouseout 事件

* 1. mouseenter：

只有在鼠标指针穿过被选元素时，才会触发 mouseenter 事件。

不支持事件冒泡。

相对应 mouseleave 事件。

#### 31、ES6 用到过吗，新增了哪些东西，你用到过什么？

* 1. let 和 const
  2. 模板字符串
  3. 箭头函数（自己没有 this ，从自己的作用域链的上一层继承 this ）
     1. 没有构造函数，不能使用new
     2. 不绑定arguments， 是使用...rest参数
     3. 自己没有 this ，从自己的作用域链的上一层继承 this
  4. for-of（用来遍历数据—例如数组中的值）e.g. Array，String，Set，Map
     1. Set 是有序列表，类似于数组，但是没有重复值
        1. 操作方法：

add、delete、has、clear

* + - 1. 遍历

Keys/values/entries/foreach

* + - 1. 用途

数组去重

* + 1. Map 是存储许多键值对的有序列表，key 和 value 支持所有数据类型
       1. 操作方法：

Set(key,value)、get(key)、delete、has、clear、size

* + - 1. 遍历

Keys/values/entries/foreach

* 1. arguments 对象可被不定参数和默认参数完美代替
  2. Promise
     1. Promise 是异步编程的一种解决方案，解决异步请求的回调地狱
     2. 使用场景
     3. 三种状态
        1. Pedding/fulfilled/reject
     4. 优缺点
        1. Promise 的优点:

一旦状态改变，就不会再变，任何时候都可以得到这个结果；

可以将异步操作以同步操作的流程表达出来，避免了层层嵌套的回调函数。

* + - 1. Promise 的缺点:

无法取消 Promise，错误需要通过回调函数来捕获；

当处于 pending 状态时，无法得知目前进展到哪一个阶段。

* + 1. Async/await
       1. Async/Await 是一种用于处理 JS 异步操作的语法糖,async 返回是promise对象，await返回promise对象或者其他值
  1. 数组的拓展
     1. 解构赋值
        1. 数组
        2. 对象
        3. ...扩展运算符
        4. 函数中剩余参数：rest参数
  2. 引入 module 模块的概念
     1. 模块化发展：无模块化 --> CommonJS规范 --> AMD规范 --> CMD规范 --> ES6模块化
     2. CommonJS规范（nodeJs）:CommonJS 用同步的方式加载模块
     3. AMD规范（RequireJS）：异步方式加载模块，必须要**提前加载所有的依赖**，然后才可以使用，不能按需加载
     4. CMD规范（SeaJS）：依赖就近、延迟执行
     5. ES6模块化：使用import、export关键字，按需加载

#### 32、Ajax 是什么? 如何创建一个Ajax？

使用 JavaScript 异步获取数据，而且页面不会发生整页刷新的，提高了用户体验

创建 XMLHttpRequest 对象,也就是创建一个异步调用对象

创建一个新的 HTTP 请求,并指定该 HTTP 请求的方法、URL 及验证信息

设置响应 HTTP 请求状态变化的函数

发送 HTTP 请求

获取异步调用返回的数据

使用 JavaScript 和 DOM 实现局部刷新

var xhr = new XMLHttpRequest()

xhr.open('GET', 'news/list?type=gn')

xhr.send(null)

// 4. 指定 xhr 状态变化事件处理函数 —— 相当于处理网页呈现后的操作

xhr.onreadystatechange = function () {

// 通过 xhr 的 readyState 判断此次请求的响应是否接收完成

// 4代表done

if (this.readyState === 4) {

// 通过 xhr 的 responseText 获取到响应的响应体

console.log(this)

}

}

#### 33、单线程和任务队列

单线程：

任务队列：

任务（消息）队列是一个先进先出的队列，它里面存放着各种任务（消息）。

在 JavaScript 中任务有两种，一种是同步任务，一种是异步任务。

同步任务：各个任务按照文档定义的顺序一一推入“执行栈”中，当前一个任务执行完毕，才会开始执行下一个任务。

异步任务：各个任务推入“任务队列”中，只有在当前的所有同步任务执行完毕，才会将队列中的任务“出队”执行。

当线程中没有执行任何同步代码的前提下才会执行异步代码。

#### vue中子组件可以修改接父组件数据（.sync-update）

* 1. Vue单向流，使用.sync, 通过update事件$emit
  2. 地址：<https://blog.csdn.net/weixin_44178305/article/details/121315440>

#### vue-simple-uploader 大文件分片上传

地址：<https://www.cnblogs.com/xiahj/p/vue-simple-uploader.html>

#### localStorage 保存的字符串格式，例如保存[ 1,2], 获取时，数据变成”1,2”

解决：保存时采用JSON.stringify()，获取时采用JSON.parse()

#### 使用公共事件总线Bus, 在组件mounted中使用bus.$on(“test”), 必须在beforeDestroy中执行bus.$off(“test”), 否则组件使用时，会调两次$on事件

参考地址： https://blog.csdn.net/qq\_45327886/article/details/129568400

#### 38、安装telnet

Ctrl + R => 输入control => 程序和功能 => 启用或关闭windows功能 => 勾选Telnet客户端 => 确定

#### 39、vue文本识别"\n"换行问题的解决方式

地址：<https://www.jb51.net/article/268122.htm>

设置 white-space: pre-wrap; 代码如下：

<div style=”white-space: pre-wrap”> {{ 这是一段文本\n 换行}} </div>

#### 40、vue el-dialog 嵌套问题

地址：<https://blog.csdn.net/weixin_42322454/article/details/117418976>

解决方案：1、内层dialog 添加append-to-body

1. 尽量不嵌套，同平级展示，影响会更小

#### vue--表单检验、常用到的一些正则表达校验、身份证15位转18位

地址：<https://www.cnblogs.com/shenpeng/p/11654382.html>

#### 使用Ant Design of Vue中的Checkbox 多选框实现多组全选和全不选

地址：<https://blog.csdn.net/weixin_45531945/article/details/120487569>

43、webpack 配置和vue2和Vue3区别