资源汇总

1.百度interview问题引导
https://github.com/fex-team/interview-questions

知识点 —— js篇

1.语法：parseInt(string, radix)

1.radix可以指定使用几进制进行类型转换,默认为十进制。parseFloat()只应用于解析十进制数字

```
Number.parseInt('0110', 2) // 6
Number.parseInt('0110', 10); // 110
```

2.单元运算符+可以把数字字符串转换成数值

```
+ "42";   // 42
+ "010";  // 10
+ "0x10"; // 16
```

3.一个字符串加上一个数字（或其他值），操作数都会被转换成字符串。

（实用技巧：通过与空字符串相加，可以将某个变量快速转换成字符串类型。）

```
""+ 123; // "123" string
```

4.null和undefined

1.null表示一个空值，undefined表示一个未定义的值。

```
false、0、空字符串("")、NaN、null 和 undefined 被转换为 false
所有其他值被转换为 true
```

5.常用数组方法

```
【删除元素】
splice(start,deleteCount,item1,item2)
//start:指定修改的开始位置 deleteCount:要删除的个数 item1:要添加的值
//返回值：由被删除的元素组成的一个数组
var months = ['Jan', 'March', 'April', 'June'];
months.splice(1, 0, 'Feb'); //['Jan', 'Feb', 'March', 'April', 'June']
months.splice(4, 1, 'May'); //['Jan', 'Feb', 'March', 'April', 'May']
```

```
【数组首尾增删元素】
push()增     pop()删      //数组尾部
unshift()增  shift()删    //数组头部
```

```
【数组--字符串互相转换】
----数组转字符串----
1.arr.join(sep)   //返回一个包含数组中所有元素的字符串，每个元素通过指定的 sep 分隔。
["Jan", "Feb", "March", "haha", "June"].join(",") //"Jan,Feb,March,haha,June"
2.arr.toString()
----字符串转数组----
string.split(sep,limit) //limit 限定返回分割片段数量
"Hello World. How are you doing?".split(" ", 3); //["Hello", "World.", "How"]
```

6.展开语法

知识梳理

Javascript

1. 【this的理解】

```
this的绑定规则：
默认绑定 ： this指向全局对象（非严格模式），this指向undefined(严格模式)，undefined上没有this对象，会抛出错误。
隐式绑定 ： XXX.fun() 函数调用是在某个对象上触发的。对象属性链中只有最后一层会影响到调用位置。
显式绑定 ： call(),apply(),bind()的第一个参数，就是对应函数的this所指向的对象。
new绑定 ： 将构造函数的作用域赋值给新对象，即this指向这个新对象。
绑定规则的优先级 new绑定 > 显式绑定 > 隐式绑定 > 默认绑定
```

2. 【变量提升】

函数和变量的声明都会被提升到函数的最顶部，变量的初始化不会被提升。

```
var x; // 声明 x      能被提升
var x = 5; // 初始化 x        不能被提升
```

3. 【Promise的理解】

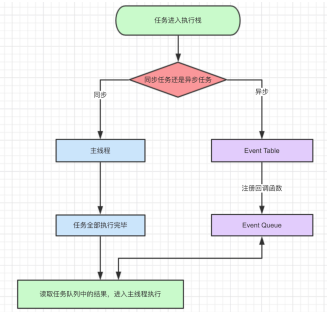
4. 【闭包】

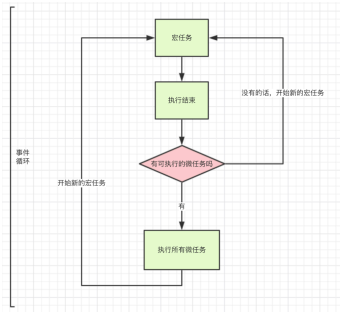
5. 【原型和原型链】

6. 【js的执行机制】

js是单线程解释型语言，同一时间只能执行一条命令。（不设计成多线程是因为渲染网页的时候多线程容易引起死锁或资源冲突，比如，DOM节点上添加内容，另一个线程删除了这个节点这时浏览器应该以哪个线程为准？）

任务队列　　事件循环（Event Loop）






```
macro-task(宏任务): 包括整体代码script, setTimeout, setInterval, setImmediate, I/O,  UI rendering
micro-task(微任务): Promise, process.nextTick,  MutationObserver(html5新特性)
```

浏览器

1. 【浏览器渲染机制】

1. 处理 HTML 标记并构建 DOM 树。

2. 处理 CSS 标记并构建 CSSOM 树。
3. 将 DOM 与 CSSOM 合并成一个渲染树。
4. 根据渲染树来布局，以计算每个节点的几何信息。
5. 将各个节点绘制到屏幕上。

2. 【进程和线程】

进程：操作系统进行资源分配和调度的一个独立单位，是应用程序运行的载体。

线程：程序执行流的最小单元，线程是进程中的一个实体，是被系统独立调度和分派的基本单位。

区别：

a. 一个进程由一个或多个线程组成。
b. 进程之间相互独立，但是同一进程下各个线程共享程序的内存空间及进程级资源。
c. 线程上下文切换比进程快得多

浏览器的线程：

i. GUI渲染线程：用于渲染页面
ii. JS引擎线程：执行js任务
iii. 事件触发线程：控制交互，相应用户
iv. 异步http请求线程：处理请求
v. EventLoop轮询的处理线程：用于轮询消息队列

解决方案类

1. C端布局适配方案 （rem结合项目）

2. 微信相关（小程序，会员卡）

工具类

1. 抓包工具

2. 打包编译工具