1. **http协议**

**HTTP**和**HTTPS**

HTTP协议通常承载于TCP协议之上，在HTTP和TCP之间添加一个安全协议层（SSL或TSL），这个时候，就成了我们常说的HTTPS。

默认HTTP的端口号为80，HTTPS的端口号为443。

为什么HTTPS安全

因为网络请求需要中间有很多的服务器路由器的转发。中间的节点都可能篡改信息，而如果使用HTTPS，密钥在你和终点站才有。https之所以比http安全，是因为他利用ssl/tls协议传输。它包含证书，卸载，流量转发，负载均衡，页面适配，浏览器适配，refer传递等。保障了传输过程的安全性。

**2、promise是什么？**

\*\*\*\*\*Promise:

问题: 回调地狱 callback hell

为了保证多个异步任务必须顺序执行，而形成了很深的嵌套调用结构

解决: Promise

何时: 只要多个异步任务，希望顺序执行时

如何:

1. 定义Promise对象封装要执行的异步任务

new Promise(function(open){

//异步任务...

//在异步任务最后一步之后，打开开关: open()

})

强调:

1. new Promise在创建对象同时，就自动执行内部的异步任务

2. 如果不open(),则后续的then不执行！

2. 将多个Promise对象，用.then连起来：

Promise1

.then(function(){

return Promise2

})

...

.then(function(){

最后一项任务

})

错误处理:

1. 在new Promise中，如果出错，则调用err开关:

new Promise(function(open,err){

//异步任务

//如果正常执行完，调用open()

//如果出错，调用err("错误消息")

})

强调: 调用err()后，后续then不执行

2. 在顺序调用的结尾，只添加一个.catch()

.catch(function(err){ ...err变量接住"错误消息" })

前后两个异步调用间传参

前一个异步调用，通过open(参数)方式传参

后一个异步调用，通过.then(function(参数){ ... })方式接受

简化: ES7

ES6的Promise的问题: .then()依然使用嵌套方式：

.then(function(){

return 异步调用的函数

})

解决: await:

前提: await修饰的函数必须也是return new Promise()的

如何:

(async function(){ //async说明大任务整体是异步的

try{ //代替.catch(function(err){}) 做错误处理

var result=await 异步调用1

//其中: await是等待当前函数执行完的意思

// 阻塞程序执行

// 如果异步调用中通过open(参数)返回了数据，可用var result=来接住异步任务返回的数据，在后续步骤中继续使用。

var result=await 异步调用2

...

}catch(err){

...

}

})()

等待多个异步任务执行后才执行:

Promise.all([

异步调用1,

异步调用2,

... ...

])

.then(function(){

...

})

.catch(function(err){ ... })

3、同步和异步的区别？

[**https://www.jianshu.com/p/561db8ff3e7a**](https://www.jianshu.com/p/561db8ff3e7a)

**4、有一个导航栏，数据是从后端动态获取的，如何实现跳转？**

AJAX获取数据之后存放在data中，拿spli按照“=”进行分割以后，在实行跳转

**5、微信聊天对话框怎么实现？**

订阅号测试帐户开发->聊天工具(不是聊天机器人)

开发者工具->公众平台测试帐户

(1)创建服务器并且合法域名[内网穿透工具]

ngrok.exe http 3000

https://743f7eb4.ngrok.io

(2)下载第三方微信聊天模块

(3)使用聊天模块调用模块方法完成聊天操作

let config = {

appid:"wx31c2f70c58017a68 "

token:"weixin"

encodingAESKey:""

};

app.post("/",wechat(config,(req,res)=>{

req.weixin.Content 服务器发送文字

res.reply(返回文字);

}));

**6.字符串API**

字符串API

**toUpperCase()** 将英文字符串转为大写

**toLowerCase()**  将英文字符串转为小写

**charAt(参数)** 查找下标对应的字符

**charCodeAt(参数)** 查找下标对应字符的Unicode码

**indexOf(参数1,[参数2])** 查找某个字符对应的下标，参数2开始查找的下标，找不到返回-1

**lastIndexOf(参数1)** 查找某个字符最后一次出现位置的下标

**slice(start, end)** 截取字符串，start开始的下标，end是结束的下标；范围start~end-1。如果end为空，从start到结尾；负数表示倒数第几个。

**substring(start, end)** 和slice作用几乎一致，区别在于不支持负数写法，如果参数为负数，自动转成0。

**split(参数)** 将字符串按照指定的字符转换为数组——切割字符串，参数指定的字符。

**7.公司现在需要用框架完成一个项目，你的思路是什么？**

先问一下公司的项目需求是什么，然后根据项目需求选择合适的框架，再结合框架官网完成相应的项目

**8、说一下为什么要重构项目？**

首先，要看看你的项目能不能重构

其次，要选择合适的框架进行重构项目

重构项目最主要的原因是因为客户的需求越来越多，为了满足客户的需求，所以就需要对项目进行重构，从而实现更多的功能，满足客户的需求。