**1、** javascript里面的基本数据类型（primitive data types）和复合数据类型（complex type）有哪些（基础）

Primitive: \_\_Undefined ,Null ,Boolean ,Number ,String\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Complext:\_\_Function ,Object ,Array ,Map, Set\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**2、** 简要说明undefined,null,Nan的区别（基础）

undefined: \_\_ 未赋值的变量为undefined\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

null： \_\_未赋值的Number为NaN\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Nan： \_\_未赋值的Object为Null\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**3、** var,const,let的区别(es2015,context)

var： \_\_\_没有块级作用域，只有函数作用域\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

const： \_\_一旦定义，无法更改\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

let： \_\_\_有块级作用域，不能重复定义\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**4、** 填写打印输出结果（context, highter function）

console.log(x);

var x;

function fnTest(){

x = 10;

this.x = 'fnText.x';

return function(x) {

console.log(x, x.x, x['x']);

};

}

fnTest()('test');

console.log(x);

fnTest.apply({});

console.log(x);

输出1：\_\_\_undefined \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

输出2：\_\_\_text,undefined,undefined\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

输出3：\_\_\_fnText.x\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

输出4：\_\_\_fnText.x\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**5、** 填写打印输出结果(es2015)

const objA = {

x: 1,

arrInner: [1,2,3],

name: 'a',

};

const objB = {

x: 100,

arrInner: [4,5,6],

name: 'b',

}

console.log(Object.assign({}, objB, objA))

console.log({ ...objA, ...objB })

function fnArray([...arr] = []){

return arr.map(item => item + 1);

}

const [a, b , ...rest] = fnArray([2.1314,1.7182]);

console.log(a, b, rest);

输出1：\_\_\_\_x:1 ,arrlnner: [1,2,3] ,name: ‘a’\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

输出2：\_\_\_\_x:100 ,arrlnner:[4,5,6] ,name:’b’ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

输出3：\_\_\_\_[]\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**6、** 请在文件名为a.js的文件中(es2015)，

a 导出一个默认函数用于生成一个0至100的随机整数（要求每次生产的随机数不能存在于已生成的随机数）

b 导出一个数组对象，该对象包含该脚本累计生成的随机数

**7、** 请在文件名为b.js的文件中(es2015,algorithm)，

a 生产一个数组大小为6的数组并使用文件a.js中定义函数随机填充满

b 使用刚才生成的数组中的元素，计算&打印(console.log)出填满一个长度为2的数组的所有可能

**8、** 使用过的前端框架有哪些

jQuery ,Bootstrap ,react

**9、** 通过何种方式进行技术学习或帮助

看书，谷歌或百度自己的问题，在github提issue，同一起学习的

小伙伴谈论最近学习的知识点，逛论坛

**10、** 聊聊webpack、nodejs、react、react-native的认知（附加题）

Webpack：是一种前端模块化构建工具，我用来打包代码，生成服务，它的热替换功能在实际开发中相当实用。他牛逼在，不仅仅是模块化js，连同css，图片等都一同模快化，它的各种loader使用方便，功能强大。Webpack配合react用，很爽。让整体都变得有序起来。

Node.js常被人们评论为性能多么出众，可能我没有做过大项目来体会。让我印象深刻的是它为我们Web前端er带来的处理服务的能力，我可以用node来起服务，处理路由，完成一整套ajax请求，处理cookie等等，因为node，我们能建立一套MVC模型。另外让我印象深刻的就是Node的非阻塞机制，它的事件轮询和线程池机子让它能高性能得处理I/O事物。

React只是属于MVC的V层，这是我用得最爽的框架。它完全用JS来驱动视图的构建，事件绑定和数据流。用props和state管理数据流的思路很清晰，结构明确。在了解了它mount和update的生命周期后，它于各个周期的事件带来了很大的方便。后来用react-router实现了一个单页面的Web应用，我感觉这是一种实现单页面应用很好的形式。Redux我正在学习中，还未在项目中使用过。

React-native没有了解过，只知道是用来构建移动端应用的，对性能有一定要求。

**11、**如果你之前未了解或学习ES2015,花上2-3天时间去了解学习，之后发送自己的学习成果至邮箱([25359952@qq.com](mailto:25359952@qq.com))

**12、**如果你之前未了解或学习REACT,花上2-3天时间去了解学习，之后发送自己的学习成果至邮箱(25359952@qq.com)