1.GIT常用的命令有哪些？

Git status 查看仓库状态

Git init 初始化仓库

Git add 添加

Git commit -m “注释”

Git push 推送到github上

Git diff 查看修改的内容

Git log 提交的历史纪录

1. Javascript中的变量类型有哪几种？

ES6之前

基本数据类型5种

Number 数值

Null 空

Undefined 未定义

String 字符串

Boolean 布尔

复杂数据类型3种

Object 对象

Array 数组

Function 函数

ES6中新增了一个数据类型symbol类型 symbol是解决属性名冲突问题的

1. 引用类型有哪些？

Object 对象

Array 数组

Function 函数

4.数据类型之间如何相互转换？

字符串转数值

ParseInt

数值转字符串

Tostring

Set转数组

...

数组转Set

...

字符串常用的方法有哪些？

charAt 返回index所指定的字符

endsWith 测试字符串是否以str结尾

Length 返回字符串长度

toLowerCase()  将字符串内的字符改写成小写

toUpperCase()  将字符串内的字符改写成大写

Object常用方法有哪些？

Object.assign() // 通过复制一个或多个对象来创建一个新的对象

Object.create() //使用指定的原型对象和属性创建一个新对象

Object.defineProperty()//给对象添加属性并指定该属性的特征

Object.defineProperties()//添加多个属性并分别指定它们的配置

Object.values() //返回给定对象自身可枚举值的数组。

Object.entries() //返回给定对象自身可枚举属性的键值对

Object.getOwnPropertyDescriptor() // 返回对象属性配置。

Object.getOwnPropertyNames()//返回一个数组，它包含了指定对象所有的可枚举或不可枚举的属性名

Object.getOwnPropertySymbols() //返回一个数组，它包含了指定对象自身所有的符号属性。

Object.getPrototypeOf() //返回指定对象的原型对象。

Object.setPrototypeOf() // 设置对象的原型（即内部 [[Prototype]] 属性）。

Object.is() //比较两个值是否相同。所有 NaN 值都相等（这与==和===不同）。

Object.keys() //返回一个包含所有给定对象"自身"可枚举属性名称的数组。

Array常用方法有哪些？

Array.length // [属性] 获取数组长度

Array.form() //从类数组对象或者可迭代对象中创建一个新的数组实例

Array.isArray() //用来判断某个变量是否是一个数组对象。

Array.of() //根据一组参数来创建新的数组实例，支持任意的参数数量和类型

// 修改器方法

pop() // 删除数组的最后一个元素，并返回这个元素

push() //在数组的末尾增加一个或多个元素，并返回数组的新长度

reverse() //颠倒数组中元素的排列顺序

shift() // 删除数组的第一个元素，并返回这个元素

unshift() //在数组的开头增加一个或多个元素，并返回数组的新长度

sort() // 对数组元素进行排序，并返回当前数组

splice() //在任意的位置给数组添加或删除任意个元素

//访问器方法

concat() //返回一个由当前数组和其它若干个数组或者若干个非数组值组合而成的新数组

join() // 连接所有数组元素组成一个字符串

slice() // 抽取当前数组中的一段元素组合成一个新数组。

toString() //返回一个由所有数组元素组合而成的字符串。遮蔽了原型链上的 Object.prototype.toString() 方法

indexOf() //返回数组中第一个与指定值相等的元素的索引，如果找不到这样的元素，则返回 -1。

lastIndexOf() //返回数组中最后一个（从右边数第一个）与指定值相等的元素的索引，如果找不到这样的元素，则返回 -1。

Set常用的方法有哪些？

* Set.prototype.add(value)：添加某个值，返回 Set 结构本身。
* Set.prototype.delete(value)：删除某个值，返回一个布尔值，表示删除是否成功。
* Set.prototype.has(value)：返回一个布尔值，表示该值是否为Set的成员。
* Set.prototype.clear()：清除所有成员，没有返回值。
* Set.prototype.keys()：返回键名的遍历器
* Set.prototype.values()：返回键值的遍历器
* Set.prototype.entries()：返回键值对的遍历器
* Set.prototype.forEach()：使用回调函数遍历每个成员

Map常用的方法有哪些？

* Map.prototype.keys()：返回键名的遍历器。
* Map.prototype.values()：返回键值的遍历器。
* Map.prototype.entries()：返回所有成员的遍历器。
* Map.prototype.forEach()：遍历 Map 的所有成员。

解构赋值如何使用？

  //数组的解构赋值

    var [a,b,c] = [1,2,3]

    console.log(a);

    console.log(b);

    console.log(c);

    //对象的解构赋值

    var {a1,b1,c1} = {a1:"zs",b1:"ls",c1:"zl"}

    console.log(a1);

    console.log(b1);

    console.log(c1);

扩展运算符的作用？

ES6的扩展运算符可以说是非常使用的，在给多参数函数传参，替代Apply，合并数组，和解构配合进行赋值方面提供了很好的便利性。

扩展运算符就是三个点“...”，就是将实现了Iterator 接口的对象中的每个元素都一个个的迭代并取出来变成单独的被使用。

rest参数如何使用？

es6中引入了rest参数，样式形如...xxx，用于获取函数的多余参数，这样就不需要使用arguments对象了。rest参数搭配的一个变量是一个数组，该变量将多余的参数放入数组中。

原型是什么？有什么作用？

原型就是对象  
作用是共享方法，通过原型链可以找到Object上的对象，不要开辟新的内存空间。公共属性定义再构造函数里。公共方法放在原型对象里。

ES6之前是如何实现继承的？

ES6之前是使用prototype来继承

ES6中如何定义一个类?构造方法有什么作用？类如何继承？

Class

它可以用来对其初始化,但最主要的还是构造出这个东西。

extends

如何声明类的静态属性？静态属性有什么特征？

Static

特征是：不用实例化即可调用

ES6中模块如何使用？

·导入 export

·导出 import

PHP的运行模式有哪两种？

php一共分为五大运行模式：包括cgi 、fast-cgi、cli、isapi、apache

什么是AJAX?核心对象是什么？

XHRHttpRequest

实现AJAX的步骤是什么？

function ajax(obj){

            // 1、实例化 xhr

        var xhr = new XMLHttpRequest()

        // 2、监听函数

        xhr.onreadystatechange = function(){

            //console.log(xhr.readyState)

            if(xhr.readyState==4 && xhr.status==200){

                // 接收服务器返回的数据

                //console.log(typeof xhr.responseText)

                //console.log(xhr.responseText)

                var res = JSON.parse(xhr.responseText)

                // 请求成功后的处理函数

                obj.success(res)

            }

        }

        // 3、open

        xhr.open(obj.method,obj.url)

        // 4、send

XHR对象中常用的属性和方法有哪些？

1、open( ) -方法

作用：创建请求

语法：open(“method”,”url”,isAsyn)

1. method 请求方式：POST或GET

2. url 请求地址

3. isAsyn 使用 同步(false) 还是 异步(true) 的方式发送请求

2、readyState - 属性

作用：

0：请求发送还未发送

1：已经打开到WEB服务器的连接

2：xhr对象已经接受响应消息的头部

3：xhr对象已经接受响应消息的主体

4：xhr已经接受完成响应消息

注意：当readyState为4的时候，表示所有的响应消息都接受完毕

3、status -属性

作用：表示服务器相应状态码

值： 1、200 正常接受请求以及给出响应

2、404 要请求的资源不存在

3、500 服务器内部错误

注意：当status值是200的时候

4、onreadystatechange - 事件

当xhr的readyState发生改变时，要自动激发的操作。用于监听readyState状态改变的

xhr.onreadystatechange=function(){

//每当readyState状态码发生改变的时候要执行的操作

if(xhr.readyState==4 && xhr.status==200){

//当xhr已经完全接收响应了，并且服务器也正常给出响应的时候要做的事

var txt=xhr.responseText;//服务器端响应回来的数据

}

}

5、send( ) -方法

作用：发送请求

语法：xhr.send(body);

body：发送的请求数据。如果是get提交方式，此处为null；如果是post提交方式，此处就是提交的数据

在Javascript中：

如何将JSON字符串转为Javascript对象？

parseJSON()

如何将Javascript对象转为JSON字符串？

JSON.stringify()