**简要描述 JavaScript 的数据类型？**

参考答案：

JavaScript 的数据类型可以分为原始类型和对象类型。

原始类型包括 string、number 和 boolean 三种。其中，字符串是使用一对单引号或者一对双引号括起来的任意文本；而数值类型都采用 64 位浮点格式存储，不区分整数和小数；布尔（逻辑）只能有两个值：true 或 false。

复杂类型指其他对象，如 Array、Date、Object 等。

除此之外，JavaScript 中还有两个特殊的原始值：null（空）和 undefined（未定义），它们代表了各自特殊类型的唯一成员。

**简要描述 null 和 undefined 的区别**

参考答案：

null：是 JavaScript 的关键字，用于描述“空值”，对其执行 typeof 操作，返回“object”，即为一个特殊的对象值，可以表示数字、字符串和对象是“无值”的。

undefined：是预定义的全局变量，其值为“未定义”，它是变量的一种取值，表示变量没有初始化。当查询对象属性、数组元素的值时，如果返回 undefined 则表示属性或者元素不存在；如果函数没有任何返回值，也返回 undefined。

需要注意的是，虽然 null 和 undfined 是不同的，但是因为都表示“值的空缺”，两者可以互换。因此，使用“＝＝”认为二者是相等的，需要使用“＝＝＝”来区分它们。

**解释一下 JavaScript 中的局部变量与全局变量的区别**

参考答案：

全局变量拥有全局作用域，在 JavaScript 代码的任何地方都可以访问；在函数内声明的变量只在函数体内有定义，即为局部变量，其作用域是局部性的。

需要注意的是，在函数体内声明局部变量时，如果不使用 var 关键字，则将声明全局变量。

**什么是 JavaScript 中的函数作用域**

参考答案：

JavaScript 中的函数作用域是指：变量在声明它的函数体以及这个函数体嵌套的任意函数体内都是有定义的。这意味着，在函数体内声明的所有变量在整个函数体内始终是可见的，这种特性也被称为“声明提前”，即，函数内声明的所有变量（不涉及到赋值）都被提前至函数的顶部声明。

**简述 arguments 对象的作用**

参考答案：

在函数代码中，使用特殊对象 arguments 可以访问函数的参数。即，开发者在定义函数时，无需明确的为方法声明参数，也可以在方法体中使用 arguments 来访问参数。这是因为， arguments 是一种特殊对象，在函数代码中，表示函数的参数数组。正因为 arguments 表示参数组成的数组，因此，首先可以使用 arguments.length检测函数的参数个数，其次，可以通过下标（arguments[index]）来访问某个参数。这样，可以用 arguments 对象判断传递给函数的参数个数并获取参数，适用于函数参数无法确定个数的情况下。

**简要描述 JavaScript 中定义函数的几种方式**

参考答案：

JavaScript 中，有三种定义函数的方式：

1、函数语句：即使用 function 关键字显式定义函数。如：

Function f(x){ return x+1; }

2、函数定义表达式：也称为“函数直接量”。形如：

var f = function (x){ return x+1; };

3、使用 Function() 构造函数定义，形如：

Var f = new Function(“x”,”return x+1;”);

**列举几个 JavaScript 中常用的全局函数，并描述其作用**

参考答案

JavaScript 中常用的全局函数及其作用如下：

1. parseInt：解析一个字符串并返回一个整数；

2. parseFloat：解析一个字符串并返回一个浮点数；

3. isNaN：检查某个值是否是数字，返回 true 或者 false；

4. encodeURI ：把字符串作为 URI 进行编码；

5. decodeURI ：对 encodeURI() 函数编码过的 URI 进行解码；

6. eval：计算某个字符串，以得到结果，或者用于执行其中的 JavaScript 代码。

**什么是栈？在 JavaScript 中，如何模拟栈操作？**

参考答案：

栈（stack）是一种运算受限的线性表，其限制是仅允许在表的一端进行插入和删除运算。这一端被称为栈顶，相对的把另一端称为栈底。向一个栈插入新元素又称作进栈、入栈或压栈，它是把新元素放到栈顶元素的上面，使之成为新的栈顶元素；从一个栈删除元素又称作出栈或退栈，它是把栈顶元素删除掉，使其相邻的元素成为新的栈顶元素。在 JavaScript 中，可以使用数组及其相关操作来模拟栈操作。首先，使用数组存储一列元素，然后使用数组的 push() 方法在数组的尾部添加指定的元素，类似于在栈顶添加元素，即顶部入；然后使用数组的 pop() 删除并返回数组尾部的元素，类似于顶部出栈，即后入的元素先出。

**什么是正则表达式？在 JavaScript 中，如何应用正则表达式？**

参考答案：

正则表达式(Regular Expression) 本身就是一个字符串，由一些普通字符和特殊字符组成的，用以描述一种特定的字符规则的表达式。

正则表达式常用于在一段文本中搜索、匹配或替换特定形式的文本。如：词语出现频率统计、验证字符串是否符合邮箱格式、屏蔽一篇帖子中的限制性词语等。许多程序设计语言都支持利用正则表达式进行字符串操作。

在 JavaScript 中，正则表达式的应用分为两种：

1. 结合 String 对象的 replace、search 和 match 方法，实现对字符串的替换、查找和匹配；
2. 定义正则表达式对象，实现对字符串的复杂匹配操作。

**简要描述 JavaScript 中的匿名函数**

参考答案：

匿名函数是指在定义时没有指定名字的函数，且定义后往往直接调用。如：

function(num1, num2){

console.log( num1 + num2 );

}

这种方式所定义的匿名函数，往往需要直接调用，如：

(function (num1, num2) {

console.log(num1 + num2);

})(10,20);

匿名函数常用于定义不需要重复使用的函数，用完即释放。另外，对于直接调用的匿名函数而言，可以看成是一个临时的命名空间，其区域内定义的所有变量，不会污染到全局命名空间。

**简要描述 JavaScript 中的作用域链**

参考答案：

任何一段 JavaScript 代码都对应一个作用域链，作用域链中存放一系列对象，代码中声明的变量将作为对象的属性存放。

在 JavaScript 的顶层代码中，作用域链由一个全局对象组成；当定义一个函数时，它保存一个作用域链，作用域链上有两个对象，一个是函数对象，一个是全局对象。

每当一个函数被调用时，会创建一个活动对象（也叫上下文对象），函数中的局部变量将作为该对象的属性存放。

当需要使用一个变量时，将从作用域链中逐个查找对象的属性。比如：要使用变量 a，将先查找作用域中的第一个对象是否有属性 a，如果有就使用；如果没有就查找作用域链中下一个对象的属性，以此类推。如果作用域链上没有任何一个对象含有属性 x，则认为这段代码的作用域链上不存在 x，将抛出引用错误异常。

当函数调用完成后，如果没有其他引用指向为此次调用所创建的上下文对象，该对象将被回收。

**简要描述你对闭包的理解**

参考答案：

函数对象可以通过作用域链相互关联起来，函数体内部的变量都可以保存在函数作用域内，这种特性称为闭包。这意味着函数变量可以隐藏于作用域链之内，看起来好像是函数将变量包裹了起来。这种方式常用于共享函数内的私有变量。

闭包有如下应用特征：

1、局部变量：在函数中定义有共享意义（如：缓存、计数器等）的局部变量（注：定义成全局变量会对外造成污染）；

2、内嵌函数：在函数中声明有内嵌函数，内嵌函数对函数中的局部变量进行访问；

3、外部使用：函数向外返回此内嵌函数，外部可以通过此内嵌函数持有并访问声明在函数中的局部变量，而此变量在外部是通过其他途径无法访问的。

**简述 JavaScript 中创建自定义对象的方式**

参考答案：

自定义对象（user-defined object）指由用户创建的对象，兼容性问题需要由编写者注意。创建自定义对象的方式有：

1、对象直接量

2、new Object()

3、function 对象模板

4、Object.create()

**JavaScript 中，this 关键字的作用是什么？**

参考答案：

笼统的说，关键字 this 指向当前对象。比如，顶级代码中的 this 指向全局对象；在指定元素事件的时候，this 指定当前发生事件的元素对象。

对于嵌套函数，如果嵌套函数作为方法被调用，其 this 指向调用它的对象；如果作为函数调用，this 是全局对象或者为 undefined（严格模式下）。

**简述你对 JavaScript 中原型的理解**

参考答案：

在 JavaScript 中，函数本身就是一个包含了方法和属性的对象。每个函数中都有一个prototype 属性，该属性所存储的就是原型对象。

原型对象用来保存共享属性和方法，可以通过原型来实现为对象扩展属性，实现继承。

**简要描述 JavaScript 中的自有属性和原型属性**

参考答案：

自有属性是指，通过对象的引用添加的属性，此时，其它对象可能无此属性。对于自有属性，是各个对象所特有的、彼此独立的属性。比如：

emp1.job = 'Coder';

原型属性是指从原型对象中继承来的属性，一旦原型对象中属性值改变，所有继承自该原型的对象属性均改变。比如：

Emp.prototype.dept = '研发部';

当需要检测对象的自有属性时，可以使用 hasOwnProperty()方法。另，还可以使用 in操作检测对象及其原型链中是否具备指定属性。

需要注意的是，在检测对象属性时，先检测自有属性，再检测原型属性。

**简要描述你对原型链的理解**

参考答案：

每个函数中都有 prototype 属性，该函数被 new 操作符用于创建对象。将一个函数的prototype 属性指向某个对象，由此形成了一条链，称之为原型链。

可以使用isPrototypeOf()方法判定一个prototype对象是否存在于另一个对象的原型链中。如果是，返回 true，否则返回 false。

**简要描述 JavaScript 中的继承**

参考答案：

在 JavaScript 中，继承都是源于原型，有多种实现方式。比如：

1. 修改构造函数的原型，为该构造函数创建的对象指定统一的父级对象。代码如下：

B.prototype = new A();

2. 只继承于原型（尽可能地将可重用的属性和方法添加到原型中），代码如下：

B.prototype = A.prototype;

3. 单独修改一个对象的原型，而不影响其他对象的原型。代码如下：

Var b1 = new B();

Object.setPrototypeOf(b1, new A());

4. 修改构造函数，这将影响使用该构造函数创建的所有对象。代码如下：

function B() {

Object.setPrototypeOf(this, new A());

}

**简要描述 Call 和 apply 的区别**

参考答案：

call()和 apply()都用于间接调用函数。

call 方法用于调用一个对象的一个方法，并以另一个对象替换当前对象。即，任何函数可以作为任何对象的方法来调用，哪怕这个函数并非那个对象的方法。语法如：

call([thisObj[,arg1[, arg2[, [,.argN]]]]])

其中，第一个参数 thisObj 要调用函数的上下文，即将被用作当前对象的对象。其他参数为可选参数，表示将被传递方法参数序列。

apply()和 call()在作用上是相同的，但两者在参数上有区别的。它俩的第一个参数相同，不同的是第二个参数。对于 apply()，第二个参数是一个参数数组，也就是将多个参数组合

成为一个数组传入。如：

func.call(func1,var1,var2,var3)

func.apply(func1,[var1,var2,var3])

## DOM

**DOM 操作中，如何获取元素的属性值？**

参考答案：

对于元素节点，获取其某属性的值有多种方式，如下所示：

1、element.attributes[下标].value

2、element.attributes['属性名'].value

3、element.getAttributeNode('属性名').value

4、element.getAttribute('属性名')

**简要描述 DOM 操作中查找元素的方式**

参考答案：

1、 通过 HTML 中的信息选取元素，比如：

a) getElementById() 方法：根据元素的 id 属性值查询单个节点；

b) getElementsByTagName() 方法：根据元素标签的名称查询节点；

c) getElementsByName() 方法：根据元素 name 属性的值查询节点 。

2、 通过 CSS 类选取元素

a) getElementsByClassName('className')方法：根据 class 名称选取元素；

b) querySelector('selector') 和 querySelectorAll('selector')方法：根据 CSS 选择器选取元素。

3、 通过 document 对象选取，如 document.all、document.body 等；

4、 通过节点遍历选取节点，如 parentNode、firstChild 等。

**为 html 页面上的一个按钮添加 onclick 事件处理，有几种方法？**

参考答案：

1、 直接在 HTML 代码中添加，如：

<input type=”button” onclick=”funcA();” />

其中，funcA() 为一个有效函数。

2、 在 js 代码中添加，如：

btn.onclick = funcA; 或者 btn.onclick = function(){};

其中，btn 表示按钮对象。

3、 定义监听函数，代码如：

btn.addEventListener('click', function( ){ });

## HTML5

**简要描述 HTML5 中 Canvas 的作用**

参考答案：

Canvas 是 HTML5 出现的新标签，拥有自己的属性、方法和事件，其中就有绘图的方法，JavaScript 能够调用它在网页上完成绘图。

Canvas 也是 HTML5 中最强大的特性之一，允许开发者使用动态和交互式可视化方法在 Web 上实现桌面应用程序的功能。

**除了音频和视频，HTML5 还支持其他什么新的媒体元素？**

参考答案：

HTML5 对媒体支持很强，除了 audio 和 video 外，还提供：

 <embed>：作为外部应用的容器；

 <track>：定义媒体的文本跟踪；

 <source>：多种媒体源的支持。

**简要描述 HTML5 中新增的表单元素**

参考答案：

Datalist

Datetime

Output

Date

Month

Week

Time

Color

Number

Range

Email

Url

search 等。

## JQUERY

**body 中的 onload()函数和 jQuery 中的$(document).ready()有什么区别**

参考答案：

onload（）和 document.ready（）的区别如下：

 可以在页面中使用多个 document.ready()，但只能使用一次 onload()；

 document.ready()函数在页面 DOM 元素加载完成以后就会被调用，而

onload()函数则要在所有的关联资源（包括图像，音频）加载完毕才会调用。

**jQuery 中有哪几种类型的选择器**

参考答案：

有 3 种类型的选择器，如下：

 基本选择器：直接根据 id，css 类名，元素名返回 dom 元素；

 层次选择器：也叫做路径选择器，可以根据路径层次来选择相应的dom元素；

 过滤选择器：在前面的基础上过滤相关条件，得到匹配的 dom 元素。

**jQuery 的美元符号$有什么作用？**

参考答案：

美元符号 $ 是'jQuery'的别名，它是 jQuery 选择器，查看如下代码：

$(document).ready(function(){});

也可以用 jQuery 来替代

jQuery(document).ready(function(){});

**jQuery 中的 Delegate（）函数有什么作用？**

参考答案：

delegate()会在以下两个情况下使用到：

 当需要给父元素其下的子元素添加事件时，代码如下：

Html 代码

$("ul").delegate("li", "click", function(){ $(this).hide(); });

 当元素在当前页面中不可用时，可以使用 delegate()

## AJAX

**请写出至少 5 种常见的 http 状态码以及代表的意义**

参考答案：

5 种常见的 http 状态码以及代表的意义如下：

 200（OK）：请求已成功，请求所希望的响应头或数据体将随此响应返回。

 303（See Other）：告知客户端使用另一个 URL 来获取资源。

 400（Bad Request）：请求格式错误。1)语义有误，当前请求无法被服务器理解。除非进行修改，否则客户端不应该重复提交这个请求；2)请求参数有误。

 404（Not Found）：请求失败，请求所希望得到的资源未被在服务器上发现。

 500（Internal Server Error）：服务器遇到了一个未曾预料的状况，导致了它无法完成对请求的处理。

**简要描述你对 AJAX 的理解**

参考答案：

AJAX 的全称是“Asynchronous JavaScript and XML”，即异步的 JavaScript 和XML。它是指一种创建交互式网页应用的网页开发技术，可以实现页面的异步请求和局部刷新。

AJAX 包含下列技术：

 基于 web 标准 XHTML+CSS 的表示

 使用 DOM（Document ObjectModel）进行动态显示及交互

 使用 XML 和 XSLT 进行数据交换及相关操作

 使用 XMLHttpRequest 进行异步数据查询、检索

**请介绍一下 XMLHttprequest 对象**

参考答案：

AJAX 的核心是 JavaScript 对象 XmlHttpRequest。该对象在 Internet Explorer 5 中首次引入，它是一种支持异步请求的技术。简而言之，XmlHttpRequest 可以使用 JavaScript向服务器提出请求并处理响应，而不阻塞用户。通过 XMLHttpRequest 对象，Web 开发人员可以在页面加载以后进行页面的局部更新。

**AJAX 应用和传统 Web 应用有什么不同**

参考答案：

在传统的 Javascript 编程中，如果想得到服务器端数据库或文件上的信息，或者发送客户端信息到服务器，需要建立一个 HTML form 然后 GET 或者 POST 数据到服务器端。

用户需要点击“Submit”按钮来发送或者接受数据信息，然后等待服务器响应请求，页面重新加载。因为服务器每次都会返回一个新的页面， 所以传统的 web 应用有可能很慢而且用户交互较差。

使用 AJAX 技术， 就可以使 Javascript 通过 XMLHttpRequest 对象直接与服务器进行交互。通过 HTTP Request， 一个 web 页面可以发送一个请求到 web 服务器并且接受web 服务器返回的信息(不用重新加载页面)，展示给用户的还是同一个页面，但是只实现页面的局部刷新，从而提高体验度。

**介绍一下 XMLHttpRequest 对象的常用方法和属性**

参考答案：

 open(“method”,”URL”) ：建立对服务器的调用，第一个参数是 HTTP请求方式（可以为 GET，POST 或任何服务器所支持的您想调用的方式）， 第二个参数是请求页面的 URL；

 send()方法：发送具体请求；

 abort()方法：停止当前请求；

 readyState 属性：请求的状态，有 5 个可取值（0=未初始化，1=正在加载，2=以加载，3=交互中，4=完成）；

 responseText 属性：服务器的响应，表示为一个串；

 reponseXML 属性：服务器的响应，表示为 XML；

 status 属性：服务器的 HTTP 状态码。

**简述 AJAX 的交互模型，以及同步和异步的区别**

参考答案：

AJAX 主要用于实现从服务器获取数据并局部刷新页面。其交互模型为，AJAX 在浏览器端引入一个执行引擎，它一边接收 user 的请求，一边传送数据给服务器，并把服务器端返回的结果展现给 user。

同步：脚本会停留并等待服务器发送回复然后继续。

异步：脚本不停留并处理可能的回复。

**现需要使用 jQuery 代码实现 ajax 请求，详细信息如下：**

请求的 url ：carinfo.do

发送方式：post

发送数据：id 为 s1 的文本框的值

返回数据：json 格式

请求成功：将返回数据写入 id 为 s2 的文本框中

请求失败：弹出文本“系统错误”

**请提供代码实现上述功能。**

参考答案：

$.ajax({

url:"carinfo.do",

type:"post",

data:$("#s1").val(),

dataType:"json",

success:function(resText){

$("#s2").val(resText);

},error:function(){

window.alert("系统错误");

}

});

**AJAX 都有哪些优点和缺点**

参考答案：

优点：

 页面局部刷新，提高用户体验度；

 使用异步方式与服务器通信，具有更加迅速的响应能力；

 减轻服务器负担；

 基于标准化的并被广泛支持的技术，不需要下载插件或者小程序。

缺点：

 不支持浏览器 back 按钮；

 安全问题；

 对搜索引擎的支持比较弱。