

简答题

1. JavaScript的作用和特点:

主要用于开发交互式的Web页面，使网页的互动性更强，用户体验更好；是内嵌于HTML网页中，通过浏览器内置的JavaScript引擎直接编译，把一个原本只用来显示的页面，转变成支持用户交互的页面程序。

- (1) 是脚本语言
- (2) 可以跨平台
- (3) 支持面向对象

2. 网页制作涉及的技术及作用:

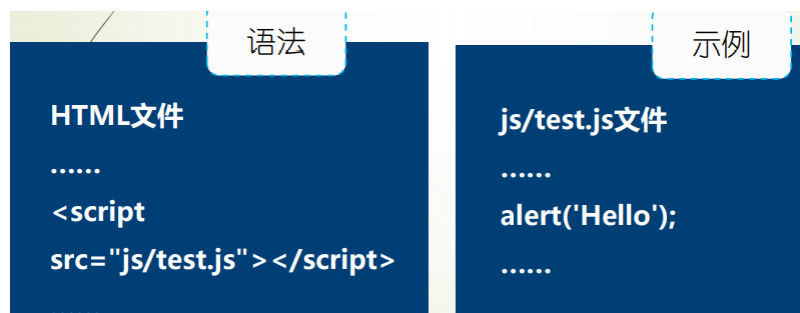
- HTML代表了结构，结构是网页的骨架，从语义的角度，描述页面结构。
- CSS代表了样式，样式是网页的外观，从审美的角度，美化页面。
- JavaScript代表行为，行为是网页的交互逻辑，从交互的角度，提升用户体验。

3. JavaScript引入方式有哪些?

- 嵌入式：使用标签包裹JavaScript代码，直接编写到HTML文件中。



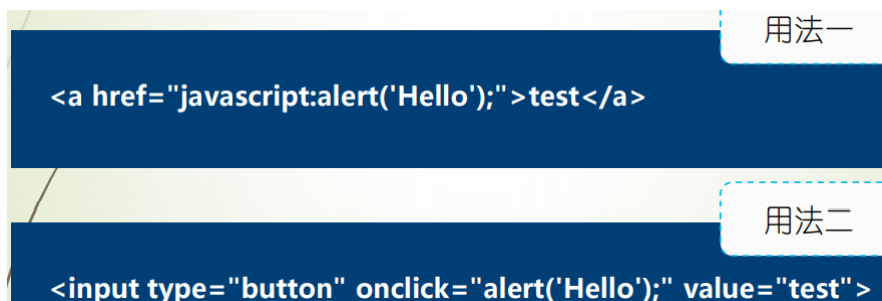
- 外链式：将JavaScript代码保存到一个单独的文件中，通常使用“js”作为文件的扩展名，然后使用标签的src属性引入文件中。



- ① 相对路径：“js/test.js”引入当前目录下的js子目录中的text.js文件；
- ② 绝对路径：“js/test.js”引入网站根目录下的js子目录中的text.js文件；

相比嵌入式，外链式的优势可以总结为以下3点：

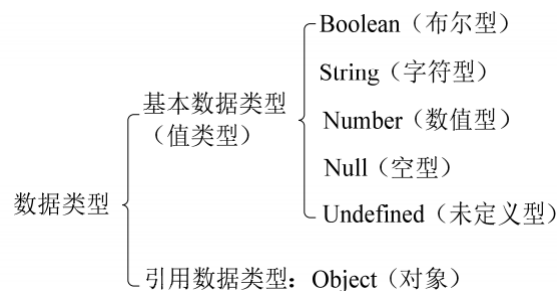
- ① 嵌入式会导致HTML与JavaScript代码混合在一起，不利用修改和维护。
 - ② 嵌入式会增加HTML文件的体积，影响网页本身的加载速度，而外链式可以利用浏览器缓存提高速度。
 - ③ 外链式有利于分布式部署。网页中链接的js、css、图片等静态文件可以部署到CDN服务器上，利用CDN的优势加快下载速度。
- 行内式：是将JavaScript代码作为HTML标签的属性值使用。



4.事件：指可以被JavaScript侦测到的交互行为。如：在网页中滑动、点击鼠标，滚动屏幕，敲击键盘等
事件处理程序：指的就是JavaScript为响应用户行为所执行的程序代码。

用处：当发生事件以后，可以利用JavaScript编程来执行一些特定的代码，从而实现 网页的交互效果。

5.数据类型分类：



6.面向过程与面向对象：

面向过程思想注重的是具体的步骤，只有按照步骤一步一步的执行，才能够完成这件事情。（执行者）

面向对象思想注重的是一个个对象，这些对象各司其职，我们只要发号施令，即可指挥这些对象帮我们完成任务。（指挥官）

7.面向对象的特征：

封装：指的是隐藏内部的实现细节，只对外开放操作接口。接口：就是对象的方法，无论对象的内部多么复杂，用户只需知道这些接口怎么 使用即可。优势：无论一个对象内部的代码经过了多少次修改，只要不改变接口，就不会影 响到使用这个对象时编写的代码。

继承：是指一个对象继承另一个对象的成员，从而在不改变另一个对象的前提下 进行扩展。优势：可在保持接口兼容的前提下对功能进行扩展；增强了代码的复用性，为程序的修改和补充提供便利。

多态：指的是同一个操作作用于不同的对象，会产生不同的执行结果。

8.函数中的this指向：

- new关键字将函数作为构造函数调用时，构造函数内部的this指向新创建的对象。
- 直接通过函数名调用函数时， this指向全局对象（浏览器中表示window对象）。
- 如果将函数作为对象的方法调用， this将会指向该对象。

9.如何让两个对象共同拥有一份相同的属性和方法？

可通过原型对象来解决。利用原型对象可以保存一些公共的属性和方法。当访问某个对象中的一个 不存在的属性或方法时，会自动调用原型中的属性和方法。也就是说，基于原型 创建的对象会自动拥有原型的属性和方法。

10.实现继承的方式有哪些？

利用原型对象实现继承；替换原型对象实现继承；利用 Object.create() 实现继承；混入继承

11.原型链的结构：

(1) 自定义函数以及String、Number、Object等内置构造函数的构造函数都是Function函数。Function函数的构造函数是Function自身

(2) 每个构造函数都有一个原型对象，构造函数通过prototype属性指向原型对象，原型对象通过constructor属性指向构造函数。

(3)由构造函数创建的实例对象，继承自构造函数的原型对象。通过实例对象的 proto__属性可以直接访问原型对象。

(4)构造函数的原型对象，继承自Object的原型对象，而Object的原型对象的 proto__属性为null。

12.什么是BOM, 什么是DOM?

BOM: 指的是浏览器对象模型。作用: 操作浏览器窗口及窗口上的控件, 实现用户和页面的动态交互。

document (文档对象): 也称为DOM对象, 是HTML页面当前窗体的内容
也是JavaScript重要组成部分之一。

history (历史对象): 主要用于记录浏览器的访问历史记录, 也就是浏览器的前进与后退功能。

location (地址栏对象): 用于获取当前浏览器中URL地址栏内的相关数据

navigator (浏览器对象): 用于获取浏览器的相关数据, 例如浏览器的名称和版本等, 也称为浏览器的嗅探器。

screen (屏幕对象): 可获取与屏幕相关的数据, 例如屏幕的分辨率等。

DOM: 文档对象模型; 作用: 是一套规范文档内容的通用型标准。

方法	说明
setTimeout()	在指定的毫秒数后调用函数或执行一段代码
setInterval()	按照指定的周期 (以毫秒计) 来调用函数或执行一段代码
clearTimeout()	取消由setTimeout()方法设置的定时器
clearInterval()	取消由setInterval()设置的定时器

13.

setTimeout()和setInterval()方法区别:

- 相同点: 都可以在一个固定时间段内执行JavaScript程序代码。
- 不同点: **setTimeout()**只执行一次代码, **setInterval()**会在指定的时间后, 自动重复执行代码。

14.HTML文档根据节点作用可以分为哪几个? 各个节点之间的关系又可以分为哪几个方面?

分为标签节点、文本节点、属性节点和注释节点。

- 根节点: 标签是整个文档的根节点, 有且仅由一个。
- 子节点: 指的是某一个节点的下级节点。
- 父节点: 指的是某一个节点的上级节点。
- 兄弟节点: 两个节点同属于一个父节点。

15.HTMLCollection与NodeList对象的区别:

- HTMLCollection对象用于元素操作。

□ NodeList对象用于节点操作。

16.事件流：事件发生时，会在发生事件的元素节点与DOM树根节点之间按照特定的顺序进行传播，这个事件传播的过程就是事件流。

事件流的传播顺序解决方案：

网景提出了“事件捕获方式”：事件流传播的顺序应该是从DOM树的根节点到发生事件的元素节点。

微软提出了“事件冒泡方式”：事件流传播的顺序应该是从发生事件的元素节点到DOM树的根节点。

17.事件的绑定方式：

行内绑定式：通过HTML标签的属性设置实现的。

动态绑定式：事件的处理程序一般都是匿名函数或有名的函数。

事件监听式：给同一个DOM对象的同一个事件添加多个事件处理程序。

事件绑定指的是为某个元素对象的事件绑定事件处理程序。

行内绑定式与动态绑定式的异同

不同点：

- 实现语法不同，前者通过HTML标签的属性设置，后者在JS中处理DOM对象。
- 事件处理程序中关键字this的指向也不同。前者指向window对象，后者指向当前正在操作的DOM元素对象。

相同点：

- 同一个DOM对象的同一个事件只能有一个事件处理程序。

事件监听式的两种不同实现方式的区别

- 实现的语法不同。
- 事件处理程序的触发顺序也不相同，同一个对象的相同事件，早期版本IE浏览器的事件处理程序按照添加的顺序倒序执行。而标准浏览器的事件处理程序按照添加顺序正序执行。

行内绑定式与动态绑定式的异同

不同点：

- ❑ 实现语法不同，前者通过HTML标签的属性设置，后者在JS中处理DOM对象。
- ❑ 事件处理程序中关键字this的指向也不同。前者指向window对象，后者指向当前正在操作的DOM元素对象。

相同点：

- ❑ 同一个DOM对象的同一个事件只能有一个事件处理程序。

事件监听式的两种不同实现方式的区别

- ❑ 实现的语法不同。
- ❑ 事件处理程序的触发顺序也不相同，同一个对象的相同事件，早期版本IE浏览器的事件处理程序按照添加的顺序倒序执行。而标准浏览器的事件处理程序按照添加顺序正序执行。