**一、答出下面代码运行结果：**

**for** (**var** i **=** 0; i **<** 5; i**++**) {

setTimeout(**function**() {

console.log(**new** Date, i);

}, 1000);

}

console.log(**new** Date, i);

答案：5 -> 5,5,5,5,5，即第 1 个 5 直接输出，1 秒之后，输出 5 个 5；

**【提升】如何修改上面的代码，使其打印出5->0,1,2,3,4,5：**

**答案：**

**var** output **=** **function** (i) {

setTimeout(**function**() {

console.log(**new** Date, i);

}, 1000);

};

**for** (**var** i **=** 0; i **<** 5; i**++**) {

output(i); *// 这里传过去的 i 值被复制了*

}

console.log(**new** Date, i);

（JS 中基本类型（Primitive Type）的参数传递是按值传递（Pass by Value）的特征）

**【再提升】如何让代码打印出0->1->2->3->4->5，这里考虑使用ES6属性：**

**答案：**

**const** tasks **=** []; *// 这里存放异步操作的 Promise*

**const** output **=** (i) **=>** **new** Promise((resolve) **=>** {

setTimeout(() **=>** {

console.log(**new** Date, i);

resolve();

}, 1000 **\*** i);

});

*// 生成全部的异步操作*

**for** (**var** i **=** 0; i **<** 5; i**++**) {

tasks.push(output(i));

}

*// 异步操作完成之后，输出最后的 i*

Promise.all(tasks).then(() **=>** {

setTimeout(() **=>** {

console.log(**new** Date, i);

}, 1000);

});

使用Promise可以达到效果；

**二、HTML5 为什么只需要写 ？**

答案： HTML5 不基于 SGML，因此不需要对DTD进行引用，但是需要doctype来规范浏览器的行为（让浏览器按照它们应该的方式来运行）；而HTML4.01基于SGML,所以需要对DTD进行引用，才能告知浏览器文档所使用的文档类型。

**三、介绍一下你对浏览器内核的理解？**

答案：主要分成两部分：渲染引擎(layout engineer或Rendering Engine)和JS引擎。渲染引擎：负责取得网页的内容（HTML、XML、图像等等）、整理讯息（例如加入CSS等），以及计算网页的显示方式，然后会输出至显示器或打印机。浏览器的内核的不同对于网页的语法解释会有不同，所以渲染的效果也不相同。所有网页浏览器、电子邮件客户端以及其它需要编辑、显示网络内容的应用程序都需要内核。JS引擎则：解析和执行javascript来实现网页的动态效果。最开始渲染引擎和JS引擎并没有区分的很明确，后来JS引擎越来越独立，内核就倾向于只指渲染引擎。

**四、常见的浏览器内核有哪些？**

Trident内核：IE,MaxThon,TT,The World,360,搜狗浏览器等。[又称MSHTML]

Gecko内核：Netscape6及以上版本，FF,MozillaSuite/SeaMonkey等

Presto内核：Opera7及以上。 [Opera内核原为：Presto，现为：Blink;]

Webkit内核：Safari,Chrome等。 [ Chrome的：Blink（WebKit的分支）]

**五、html5有哪些新特性、移除了那些元素？如何处理HTML5新标签的浏览器兼容问题？如何区分 HTML 和 HTML5？**

\* HTML5 现在已经不是 SGML 的子集，主要是关于图像，位置，存储，多任务等功能的增加。

绘画 canvas;

用于媒介回放的 video 和 audio 元素;

本地离线存储 localStorage 长期存储数据，浏览器关闭后数据不丢失;

sessionStorage 的数据在浏览器关闭后自动删除;

语意化更好的内容元素，比如 article、footer、header、nav、section;

表单控件，calendar、date、time、email、url、search;

新的技术webworker, websocket, Geolocation;

移除的元素：

纯表现的元素：basefont，big，center，font, s，strike，tt，u;

对可用性产生负面影响的元素：frame，frameset，noframes；

\* 支持HTML5新标签：

IE8/IE7/IE6支持通过document.createElement方法产生的标签，

可以利用这一特性让这些浏览器支持HTML5新标签，

浏览器支持新标签后，还需要添加标签默认的样式。

当然也可以直接使用成熟的框架、比如html5shim;

<!--[if lt IE 9]>

<script> src="http://html5shim.googlecode.com/svn/trunk/html5.js"</script>

<![endif]-->

\* 如何区分HTML5： DOCTYPE声明\新增的结构元素\功能元素

**六、简述一下你对HTML语义化的理解？**

用正确的标签做正确的事情。

html语义化让页面的内容结构化，结构更清晰，便于对浏览器、搜索引擎解析;

即使在没有样式CSS情况下也以一种文档格式显示，并且是容易阅读的;

搜索引擎的爬虫也依赖于HTML标记来确定上下文和各个关键字的权重，利于SEO;

使阅读源代码的人对网站更容易将网站分块，便于阅读维护理解。

**七、请描述一下 cookies，sessionStorage 和 localStorage 的区别？**

cookie是网站为了标示用户身份而储存在用户本地终端（Client Side）上的数据（通常经过加密）。

cookie数据始终在同源的http请求中携带（即使不需要），记会在浏览器和服务器间来回传递。

sessionStorage和localStorage不会自动把数据发给服务器，仅在本地保存。

存储大小：

cookie数据大小不能超过4k。

sessionStorage和localStorage 虽然也有存储大小的限制，但比cookie大得多，可以达到5M或更大。

有期时间：

localStorage 存储持久数据，浏览器关闭后数据不丢失除非主动删除数据；

sessionStorage 数据在当前浏览器窗口关闭后自动删除。

cookie 设置的cookie过期时间之前一直有效，即使窗口或浏览器关闭

**八、Label的作用是什么？是怎么用的？**

label标签来定义表单控制间的关系,当用户选择该标签时，浏览器会自动将焦点转到和标签相关的表单控件上。

<label for="Name">Number:</label>

<input type=“text“name="Name" id="Name"/>

<label>Date:<input type="text" name="B"/></label>

**九、HTML5的form如何关闭自动完成功能？**

给不想要提示的 form 或某个 input 设置为 autocomplete=off。

**十、如何实现浏览器内多个标签页之间的通信? (阿里)**

WebSocket、SharedWorker；

也可以调用localstorge、cookies等本地存储方式；

localstorge另一个浏览上下文里被添加、修改或删除时，它都会触发一个事件，

我们通过监听事件，控制它的值来进行页面信息通信；

注意quirks：Safari 在无痕模式下设置localstorge值时会抛出 QuotaExceededError 的异常；

**十一、webSocket如何兼容低浏览器？(阿里)**

Adobe Flash Socket 、

ActiveX HTMLFile (IE) 、

基于 multipart 编码发送 XHR 、

基于长轮询的 XHR

**十二、如何在页面上实现一个圆形的可点击区域？**

1、map+area或者svg

2、border-radius

3、纯js实现 需要求一个点在不在圆上简单算法、获取鼠标坐标等等

**十三、实现不使用 border 画出1px高的线，在不同浏览器的标准模式与怪异模式下都能保持一致的效果**

<div style="height:1px;overflow:hidden;background:red"></div>

**十四、网页验证码是干嘛的，是为了解决什么安全问题？**

区分用户是计算机还是人的公共全自动程序。可以防止恶意破解密码、刷票、论坛灌水；

有效防止黑客对某一个特定注册用户用特定程序暴力破解方式进行不断的登陆尝试。

**十五、title与h1的区别、b与strong的区别、i与em的区别？**

title属性没有明确意义只表示是个标题，H1则表示层次明确的标题，对页面信息的抓取也有很大的影响；

strong是标明重点内容，有语气加强的含义，使用阅读设备阅读网络时：<strong>会重读，而<B>是展示强调内容。

i内容展示为斜体，em表示强调的文本；

Physical Style Elements -- 自然样式标签

b, i, u, s, pre

Semantic Style Elements -- 语义样式标签

strong, em, ins, del, code

应该准确使用语义样式标签, 但不能滥用, 如果不能确定时首选使用自然样式标签。

**十六、请解释一下CSS3的Flexbox（弹性盒布局模型）,以及适用场景？**

一个用于页面布局的全新CSS3功能，Flexbox可以把列表放在同一个方向（从上到下排列，从左到右），并让列表能延伸到占用可用的空间。

较为复杂的布局还可以通过嵌套一个伸缩容器（flex container）来实现。

采用Flex布局的元素，称为Flex容器（flex container），简称"容器"。

它的所有子元素自动成为容器成员，称为Flex项目（flex item），简称"项目"。

常规布局是基于块和内联流方向，而Flex布局是基于flex-flow流可以很方便的用来做局中，能对不同屏幕大小自适应。

在布局上有了比以前更加灵活的空间。

**十七、一个满屏 品 字布局 如何设计?**

简单的方式：

上面的div宽100%，

下面的两个div分别宽50%，

然后用float或者inline使其不换行即可

**十八、css多列等高如何实现？**

利用padding-bottom|margin-bottom正负值相抵；

设置父容器设置超出隐藏（overflow:hidden），这样子父容器的高度就还是它里面的列没有设定padding-bottom时的高度，

当它里面的任 一列高度增加了，则父容器的高度被撑到里面最高那列的高度，

其他比这列矮的列会用它们的padding-bottom补偿这部分高度差。

**十九、css定义的权重?**

以下是权重的规则：标签的权重为1，class的权重为10，id的权重为100，以下例子是演示各种定义的权重值：

/\*权重为1\*/

div{

}

/\*权重为10\*/

.class1{

}

/\*权重为100\*/

#id1{

}

/\*权重为100+1=101\*/

#id1 div{

}

/\*权重为10+1=11\*/

.class1 div{

}

/\*权重为10+10+1=21\*/

.class1 .class2 div{

}

**二十、什么是外边距合并？**

外边距合并指的是，当两个垂直外边距相遇时，它们将形成一个外边距。

合并后的外边距的高度等于两个发生合并的外边距的高度中的较大者。

**二十一、如何修改chrome记住密码后自动填充表单的黄色背景 ？**

input:-webkit-autofill, textarea:-webkit-autofill, select:-webkit-autofill {

background-color: rgb(250, 255, 189); /\* #FAFFBD; \*/

background-image: none;

color: rgb(0, 0, 0);

}

**二十二、让页面里的字体变清晰，变细用CSS怎么做？**

-webkit-font-smoothing: antialiased;

**二十三、如果需要手动写动画，你认为最小时间间隔是多久，为什么？（阿里）**

多数显示器默认频率是60Hz，即1秒刷新60次，所以理论上最小间隔为1/60＊1000ms ＝ 16.7ms

**二十四、display:inline-block 什么时候会显示间隙？(携程)**

移除空格、使用margin负值、使用font-size:0、letter-spacing、word-spacing

**二十五、什么是CSS 预处理器 / 后处理器？**

- 预处理器例如：LESS、Sass、Stylus，用来预编译Sass或less，增强了css代码的复用性，

还有层级、mixin、变量、循环、函数等，具有很方便的UI组件模块化开发能力，极大的提高工作效率。

- 后处理器例如：PostCSS，通常被视为在完成的样式表中根据CSS规范处理CSS，让其更有效；目前最常做的

是给CSS属性添加浏览器私有前缀，实现跨浏览器兼容性的问题。

**二十六、介绍js的基本数据类型。**

Undefined、Null、Boolean、Number、String、

ECMAScript 2015 新增:Symbol(创建后独一无二且不可变的数据类型 )

**二十七、介绍js有哪些内置对象？**

Object 是 JavaScript 中所有对象的父对象

数据封装类对象：Object、Array、Boolean、Number 和 String

其他对象：Function、Arguments、Math、Date、RegExp、Error

**二十八、javascript创建对象的几种方式？**

javascript创建对象简单的说,无非就是使用内置对象或各种自定义对象，当然还可以用JSON；但写法有很多种，也能混合使用。

1、对象字面量的方式

person={firstname:"Mark",lastname:"Yun",age:25,eyecolor:"black"};

2、用function来模拟无参的构造函数

function Person(){}

var person=new Person();//定义一个function，如果使用new"实例化",该function可以看作是一个Class

person.name="Mark";

person.age="25";

person.work=function(){

alert(person.name+" hello...");

}

person.work();

3、用function来模拟参构造函数来实现（用this关键字定义构造的上下文属性）

function Pet(name,age,hobby){

this.name=name;//this作用域：当前对象

this.age=age;

this.hobby=hobby;

this.eat=function(){

alert("我叫"+this.name+",我喜欢"+this.hobby+",是个程序员");

}

}

var maidou =new Pet("麦兜",25,"coding");//实例化、创建对象

maidou.eat();//调用eat方法

4、用工厂方式来创建（内置对象）

var wcDog =new Object();

wcDog.name="旺财";

wcDog.age=3;

wcDog.work=function(){

alert("我是"+wcDog.name+",汪汪汪......");

}

wcDog.work();

5、用原型方式来创建

function Dog(){

}

Dog.prototype.name="旺财";

Dog.prototype.eat=function(){

alert(this.name+"是个吃货");

}

var wangcai =new Dog();

wangcai.eat();

5、用混合方式来创建

function Car(name,price){

this.name=name;

this.price=price;

}

Car.prototype.sell=function(){

alert("我是"+this.name+"，我现在卖"+this.price+"万元");

}

var camry =new Car("凯美瑞",27);

camry.sell();