

Web前端基础面试题：HTML23道+CSS42道+JS33道，共计98道题14405字

## HTML

- 1、对HTML语义化的理解
- 2、html document是干嘛的？
- 3、DOCTYPE有什么作用
- 4、什么是严格模式与混杂模式？
- 5、head 标签有什么作用，其中什么标签必不可少？
- 6、什么是WEB标准？什么是W3C标准？
- 7、HTML 的注释怎样写？
- 8、src和href的区别？
- 9、常见的块标签和内联标签以及内联块标签的区别？空(void)元素有那些？
- 10、title与h1的区别、b与strong的区别、i与em的区别？
- 11、label 的作用是什么？如何使用？
- 12、如何合并表格单元格
- 13、标签上title属性与alt属性的区别是什么？
- 14、标签应该如何合理嵌套？
- 15、png、jpg、gif 这些图片格式解释一下，分别什么时候用。有没有了解过webp？
- 16、表单提交中Get和Post方式的区别
- 17、meta标签的name属性值
- 18、sass是什么？

## HTML5

- 19、HTML5 为什么只需要写 ？
- 20、Html5 有哪些新特性、移除了哪些元素？
- 21、如何区分 HTML 和 HTML5？
- 22、HTML5的离线储存怎么使用，工作原理解释一下？
- 23、HTML5 Canvas元素有什么用？

## CSS/CSS3/移动端适配

- 1、css盒子模型，box-sizing属性的理解
- 2、引入css的方式有几种，分别是什么？
- 3、px、em、rem的区别？
- 4、怪异盒模型box-sizing？弹性盒模型|盒布局？
- 5、:before 和 ::before 区别？
- 6、display的block、inline和inline-block的区别？
- 7、隐藏元素的方法有哪些？
- 8、图片间隙问题如何解决
- 9、项目中你是如何做图片优化的？
- 10、简述rem布局原理
- 11、解释下浮动和它的工作原理？清除浮动有哪些方式？
- 12、什么是 BFC？
- 13、对 CSS 的新属性有了解过的吗？
- 14、讲一讲 CSS 的权重和优先级？
- 15、介绍 Flex 布局，flex 是什么属性的缩写？
- 16、说一说你知道的position属性，都有啥特点？
- 17、在网页中的应该使用奇数还是偶数的字体？为什么呢？
- 18、margin和padding分别适合什么场景使用？
- 19、CSS属性overflow属性定义溢出元素内容区的内容会如何处理？
- 20、对line-height是如何理解的？
- 21、元素竖向的百分比设定是相对于容器的高度吗？
- 22、transition和animation的区别是什么？
- 23、rgba()和opacity的透明效果有什么不同？
- 24、css sprite是什么,有什么优缺点？
- 25、display,float,position的关系？
- 26、css中link与@import的区别？
- 27、CSS3新特性有哪些？

- 28、display:none;与visibility:hidden的区别是什么?
- 29、画一条0.5px的直线?
- 30、html元素的id跟class什么区别
- 31、什么是响应式设计, 响应式设计的基本原理是什么
- 32、什么是外边距重叠? 重叠的结果是什么?
- 33、CSS属性content有什么作用? 有什么应用?
- 34、CSS优先级如何排序?
- 35、nth-of-type和nth-child的区别
- 36、有什么方式可以对一个DOM设置它的CSS?
- 37、CSS中, 自适应的单位都有哪些?
- 38、为什么css放在顶部而js写在后面?
- 39、z-index属性在什么情况下会失效
- 40、Flex 布局父级容器属性和子级项目属性有哪些?
- 41、flex 布局中 align-content 与 align-items 有何区别?
- 42、'+' 与 '~' 选择器有什么不同?

#### JavaScript基础篇

- 1、JavaScript 有几种数据类型
- 2、怎么进行数据类型检测
- 3、get 请求传参长度的误区
- 4、如何让事件先冒泡后捕获
- 5、说一下事件委托?
- 6、说一下图片的懒加载和预加载?
- 7、mouseover 和 mouseenter 的区别?
- 8、JS 的 new 操作符做了哪些事情?
- 9、改变函数内部 this 指针的指向函数 (bind, apply, call 的区别)
- 10、箭头函数中this 指向举例?
- 11、数组常用方法有哪些?
- 12、说说数组去重?
- 13、闭包有什么用?
- 14、闭包的特点?
- 15、JS 基本数据类型?
- 16、this 的指向哪几种
- 17、什么是暂停死区?
- 18、什么是按需加载?
- 19、什么是事件监听?
- 20、什么是 js 的闭包? 用闭包写个单例模式?
- 21、深浅拷贝的区别和实现
- 22、JS 中 string 的 startwith 和 indexOf 两种方法的区别
- 23、JS 字符串转数字的方法?
- 24、构造函数是什么?
- 25、构造函数的执行过程?
- 26、构造函数this指向?
- 27、什么是作用域?
- 28、NaN 是什么, 用 typeof 会输出什么?
- 29、说说写JavaScript的基本规范?
- 30、栈和堆的区别?
- 31、null, undefined的区别?
- 32、什么是===运算符? 说明“==”和“===”之间的区别?
- 33、break和continue语句的作用?

# HTML

## 1、对HTML语义化的理解

语义化是指根据内容的结构化（内容语义化），选择合适的标签（代码语义化）。通俗来讲就是用正确的标签做正确的事情。

语义化的优点如下：

对机器友好，带有语义的文字表现力丰富，更适合搜索引擎的爬虫爬取有效信息，有利于SEO。除此之外，语义类还支持读屏软件，根据文章可以自动生成目录；

对开发者友好，使用语义类标签增强了可读性，结构更加清晰，开发者能清晰的看出网页的结构，便于团队的开发与维护。

常见的语义化标签：

```
<header></header>  头部
<nav></nav>    导航栏
<section></section>  区块（有语义化的div）
<main></main>    主要区域
<article></article>  主要内容
<aside></aside>    侧边栏
<footer></footer>  底部
```

## 2、html document是干嘛的？

HTML即：超文本标记语言，标准通用标记语言的一个应用，“超文本”就是指页面内可以包含图片、链接、甚至音乐、程序等非文字元素。

HTML Document即：HTML Document对象，每个载入浏览器的HTML文档都会成为Document对象

由于Document对象是window对象的一部分，所以可通过window.document属性对其进行访问。

## 3、DOCTYPE有什么作用

告诉浏览器使用哪个版本的HTML规范来渲染文档。DOCTYPE不存在或形式不正确会导致HTML文档以混杂模式呈现。

## 4、什么是严格模式与混杂模式？

严格模式：是以浏览器支持的最高标准运行

混杂模式：页面以宽松向下兼容的方式显示，模拟老式浏览器的行为

## 5、head 标签有什么作用，其中什么标签必不可少？

head标签用于定义文档的头部，它是所有头部元素的容器。中的元素可以引用脚本、指示浏览器在哪里找到样式表、提供元信息等。

文档的头部描述了文档的各种属性和信息，包括文档的标题、在 Web 中的位置以及和其他文档的关系等。绝大多数文档头部包含的数据都不会真正作为内容显示给读者。

下面这些标签可用在 head 部分：<base>，<link>，<meta>，<script>，<style>，<title>。

其中 <title> 定义文档的标题，它是 head 部分中唯一必需的元素。

## 6、什么是WEB标准？什么是W3C标准？

1、Web标准不是某一个标准，而是一系列标准的集合

web标准简单来说可以分为结构、表现和行为

结构 主要是有HTML标签组成

表现 即指css样式表

行为 主要是有js、dom组成

web标准一般是将该三部分独立分开，使其更具有模块化。但一般产生行为时，就会有结构或者表现的变化，也使这三者的界限并不那么清晰。

2、W3C对于WEB标准提出了规范化的要求

1) 标签和属性名字母要小写

2) 标签要闭合

3) 标签不允许随意嵌套

4) 尽量使用外链css样式表和js脚本。让结构、表现和行为分为三块，符合规范。同时提高页面渲染速度，提高用户的体验。

5) 样式尽量少用行内样式表，使结构与表现分离

6) 标签的id和class等属性命名要做到见文知义，更利于seo，代码便于维护

## 7、HTML 的注释怎样写？

HTML 中的注释也叫"注释标签"。

"<!--"，表示注释的开始，"-->" 表示注释的结束。

## 8、src和href的区别？

src和href都是用来引用外部的资源，它们的区别如下：

**src：**表示对资源的引用，它指向的内容会嵌入到当前标签所在的位置。src会将其指向的资源下载并应用到文档内，如请求js脚本。当浏览器解析到该元素时，会暂停其他资源的下载和处理，直到将该资源加载、编译、执行完毕，所以一般js脚本会放在页面底部。

**href：**表示超文本引用，它指向一些网络资源，建立和当前元素或本文档的链接关系。当浏览器识别到它指向的文件时，就会并行下载资源，不会停止对当前文档的处理。常用在a、link等标签上。

## 9、常见的块标签和内联标签以及内联块标签的区别？空(void)元素有那些？

块级标签：单独占一行，可以设置宽高等样式h1 h2 h3 h4 h5 h6 p div ul li ol dl dt dd

内联标签（行内标签）在同一行显示，不可以设置宽度，宽高由内容撑开strong b em i span

空元素，即没有内容的HTML元素。空元素是在开始标签中关闭的，也就是空元素没有闭合标签：

常见的有：<br>、<hr>、<img>、<input>、<link>、<meta>;

## 10、title与h1的区别、b与strong的区别、i与em的区别？

strong标签有语义，是起到加重语气的效果，而b标签是没有的，b标签只是一个简单加粗标签。b标签之间的字符都设为粗体，strong标签加强字符的语气都是通过粗体来实现的，而搜索引擎更侧重strong标签。

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，对页面信息的抓取有很大的影响

i内容展示为斜体，em表示强调的文本



## 11、label 的作用是什么？如何使用？

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

使用方法1：

```
<label for="mobile">Number:</label>
<input type="text" id="mobile"/>
```

使用方法2：

```
<label>Date:<input type="text"/></label>
```

## 12、如何合并表格单元格

rowspan="2"表示从设置的td单元格开始向下合并两个单元格

colspan="2"表示自左向右合并两个单元格。

8、Thead,tbody,tfoot的作用

thead 表格的头 用来放标题之类的东西

tbody 表格的身体 放数据本体

tfoot 表格的脚 放表格的脚注之类

## 13、标签上title属性与alt属性的区别是什么？

alt属性是为了给那些不能看到你文档中图像的浏览者提供文字说明的。且长度必须少于100个英文字符或者用户必须保证替换文字尽可能的短。

这包括那些使用本来就不支持图像显示或者图像显示被关闭的浏览器的用户，视觉障碍的用户和使用屏幕阅读器的用户等。

title属性为设置该属性的元素提供建议性的信息。使用title属性提供非本质的额外信息。

## 14、标签应该如何合理嵌套？

合理嵌套HTML标签，ul和li是固定嵌套，ul的直接子元素必须是li标签，ol和li是固定嵌套，ol的直接子元素必须是li。dl,dt, dd是固定嵌套，dl的直接子元素必须是dt和dd,dt和dd是兄弟元素。

块标签可以套其他的标签，但是p标签除外，p标签不能嵌套其他的块标签，内联标签不能嵌套块标签(a标签特殊)，a标签不允许嵌套a标签和其他有交互作用的标签。块级元素与块级元素并列、内嵌元素与内嵌元素并列。

## 15、png、jpg、gif 这些图片格式解释一下，分别什么时候用。有没有了解过webp？

1. png是便携式网络图片（Portable Network Graphics）是一种无损数据压缩位图文件格式.优点是：压缩比高，色彩好。大多数地方都可以用。
2. jpg是一种针对相片使用的一种失真压缩方法，是一种破坏性的压缩，在色调及颜色平滑变化做的不错。在www上，被用来储存和传输照片的格式。
3. gif是一种位图文件格式，以8位色重现真色彩的图像。可以实现动画效果。
4. webp格式是谷歌在2010年推出的图片格式，压缩率只有jpg的2/3，大小比png小了45%。缺点是压缩的时间更久了，兼容性不好，目前谷歌和opera支持。

## 16、表单提交中Get和Post方式的区别

Get 一般用于从服务器上获取数据，Post 向服务器传送数据

Get 传输的数据是拼接在Url之后的，对用户是可见的；Post 的传输数据对用户是不可见的

Get 传送的数据量较小，不能大于 2KB。Post 传送的数据量较大，一般被默认为不受限制

Get 安全性非常低，Post 安全性较高

在 FORM 提交的时候，如果不指定 Method，则默认为 Get 请求

## 17、meta标签的name属性值

name 属性主要用于描述网页，与之对应的属性值为content，content中的内容主要是便于搜索引擎机器人查找信息和分类信息用的。

meta标签的name属性语法格式是：< meta name="参数" content="具体的参数值" >。

其中name属性主要有以下几种参数：

A、Keywords(关键字)说明：keywords用来告诉搜索引擎你网页的关键字是什么。

B、description(网站内容描述)说明：description用来告诉搜索引擎你的网站主要内容。

C、robots(机器人向导)说明：robots用来告诉搜索机器人哪些页面需要索引，哪些页面不需要索引。

content的参数有all,none,index,noindex,follow,nofollow,默认是all。

举例：< meta name="robots" content="none" > D、author(作者)

## 18、sass是什么？

sass是css预处理语言，无法直接运行在浏览器，需要其他的编译框架进行二次编译以后才可以在浏览器中运行。sass提供了许多便利的写法，大大节省了设计者的时间，使得CSS的开发，变得简单和可维护

## HTML5

### 19、HTML5 为什么只需要写？

(1) 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）；

(2) HTML4.01基于SGML，所以需要DTD进行引用，才能让浏览器知道该文档所使用的文档类型。

### 20、Html5 有哪些新特性、移除了哪些元素？

新增元素：

- 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
- 语义化更好的内容元素，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风格

## 21、如何区分 HTML 和 HTML5?

(1) 在文档类型声明上不同：

HTML是很长的一段代码，很难记住，而HTML5却只有简简单单的声明，方便记忆。

(2) 在结构语义上不同：

HTML：没有体现结构语义化的标签，通常都是这样来命名的

，这样表示网站的头部。HTML5：在语义上却有很大的优势。提供了一些新的标签，比如：

## 22、HTML5的离线储存怎么使用，工作原理解释一下？

localStorage 长期存储数据，浏览器关闭后数据不丢失；

sessionStorage 数据在浏览器关闭后自动删除。

## 23、HTML5 Canvas元素有什么用？

Canvas 元素用于在网页上绘制图形，该元素标签强大之处在于可以直接在HTML上进行图形操作，

## CSS/CSS3/移动端适配

### 1、css盒子模型，box-sizing属性的理解

在网页中，一个元素占有空间的大小由几个部分构成，其中包括元素的内容（content），元素的内边距（padding），元素的边框（border）元素的外边距（margin）四个部分。这四个部分占有的空间中，有的部分可以显示相应内容，而有的部分只用来分隔相邻的区域。

CSS盒模型：标准模型 + IE模型

#### 1. 标准模型

`div`宽度 = 内容宽度+`border`宽度+`padding`宽度 //改变`border`宽度，`div`宽度会变化  
`box-sizing:content-box;` /\*设置标准盒子\*/

#### 1. 怪异模型|IE模型

`div`宽度（定死） = 内容宽度+`border`宽度+`padding`宽度 //改变`border`宽度，`div`宽度不会变化  
`box-sizing:border-box;` /\*IE模型\*/

`box-sizing` 是一个 CSS3 属性，与盒子模型有着密切联系。即决定元素的宽高如何计算，`box-sizing` 有三个属性：

`content-box` 使得元素的宽高即为内容区的宽高(默认模式)



border-box: 计算方式  $\text{content} + \text{padding} + \text{border}$  = 本身元素大小, 即缩小了 content 大小

inherit 指定 box-sizing 属性的值, 应该从父元素继承

## 2、引入css的方式有几种, 分别是什么?

### (1) 内嵌式

通过<style></style>来书写CSS代码。

只能应用于当前网页, 不能被其它网页共享。

注意: <style>标记可以放在网页的任何地方, 但一般放在<head>。

### (2) 外联式

通过<link>标记来引入外部的CSS文件(.css)。

可以被其它网页共享。

格式: <link rel="stylesheet" href="style.css" type="text/css">

rel="stylesheet" 被引入文件和当前文件之间的关系是引入了外部样式表

href="" css文件的路径

注意: <link>标记只能放在<head>中

### (3) 行内样式

通过style的属性来书写CSS代码。写在标签的开始标签里

在开始的标签里写style="" <div style="width:100px;height:100px;"></div>

什么时候使用: 只有这么一个标签需要这个样式的时候

## 3、px、em、rem的区别?

1、px像素。绝对单位, 像素px是相对于显示器屏幕分辨率而言的, 是一个虚拟单位。是计算机系统的数字化图像长度单位, 如果px要换算成物理长度, 需要指定精度DPI。

2、em是相对长度单位, 相对于当前对象内文本的字体尺寸。如当前对行内文本的字体尺寸未被人为设置, 则相对浏览器的默认字体尺寸。它会继承父级元素的字体大小, 因此并不是一个固定的值。

rem是CSS3新增的一个相对单位(root em, 根em), 使用rem为元素设定字体大小, 仍然是相对大小但相对的只是HTML根元素。

4、区别: IE无法调用那些使用px作为单位的字体大小, 而em和rem可以缩放, rem相对的只是HTML根元素。这个单位可谓集相对大小和绝对大小的优点于一身, 通过它既可以做到只修改根元素就成比例地调整所有字体大小, 又可以避免字体大小逐层复合的连锁反应。目前, 除了IE8及更早版本外, 所有浏览器已支持rem。

## 4、怪异盒模型box-sizing? 弹性盒模型|盒布局?

在标准模式下的盒模型: 盒子总宽度/高度=width/height+padding+border+margin

在怪异模式下的盒模型下, 盒子的总宽度和高度是包含内边距padding和边框border宽度在内的,

盒子总宽度/高度=width/height + margin = 内容区宽度/高度 + padding + border + margin;

box-sizing有两个值一个是content-box, 另一个是border-box。

当设置为box-sizing:content-box时, 将采用标准模式解析计算;

当设置为box-sizing:border-box时, 将采用怪异模式解析计算。



## 5、:before 和 ::before 区别?

1. 单冒号(:)用于CSS3伪类, 双冒号(::)用于CSS3伪元素。
2. ::before就是以子元素的存在, 定义在元素主体内容之前的一个伪元素。并不存在于dom之中, 只存在于页面之中。

:before 和 :after 这两个伪元素, 是在CSS2.1里新出现的。起初, 伪元素的前缀使用的是单冒号语法, 但随着Web的进化, 在CSS3的规范里, 伪元素的语法被修改成使用双冒号, 成为::before ::after。

## 6、display的block、inline和inline-block的区别?

(1) **block**: 会独占一行, 多个元素会另起一行, 可以设置width、height、margin和padding属性;

(2) **inline**: 元素不会独占一行, 设置width、height属性无效。但可以设置水平方向的margin和padding属性, 不能设置垂直方向的padding和margin;

(3) **inline-block**: 将对象设置为inline对象, 但对象的内容作为block对象呈现, 之后的内联对象会被排列在同一行内。

对于行内元素和块级元素, 其特点如下:

### (1) 行内元素

设置宽高无效;

可以设置水平方向的margin和padding属性, 不能设置垂直方向的padding和margin;

不会自动换行;

### (2) 块级元素

可以设置宽高;

设置margin和padding都有效;

可以自动换行;

多个块状, 默认排列从上到下。

## 7、隐藏元素的方法有哪些?

display: none: 渲染树不会包含该渲染对象, 因此该元素不会在页面中占据位置, 也不会响应绑定的监听事件。

visibility: hidden: 元素在页面中仍占据空间, 但是不会响应绑定的监听事件。

opacity: 0: 将元素的透明度设置为 0, 以此来实现元素的隐藏。元素在页面中仍然占据空间, 并且能够响应元素绑定的监听事件。

position: absolute: 通过使用绝对定位将元素移除可视区域内, 以此来实现元素的隐藏。

z-index: 负值: 来使其他元素遮盖住该元素, 以此来实现隐藏。

transform: scale(0,0): 将元素缩放为 0, 来实现元素的隐藏。这种方法下, 元素仍在页面中占据位置, 但是不会响应绑定的监听事件。

## 8、图片间隙问题如何解决

两个图片之间和图片下方多出的空白间隙可以使用以下方式解决。

方法 1：将图片显示为块：解决下方间隙

```
img{
display:block;
}
```

方法 2：改变图片的 vertical-align :解决下方间隙

```
img{
vertical-align:middle;
}
```

除了 middle值，还可以设置为 top / bottom / middle

方法 3：设置图片父级标签的 font-size:0;line-height:0; 水平间隙，图片下方间隙都能解决

这个办法不仅可以解决图片下面的小空白，也能解决图片与图片之间的小空白  
缺点是让父元素里的文字的字号和行高都变成了0，文字看不见了，需要重新定义文字的字号和行高

```
.imgwrap{
font-size:0;
line-height:0;
}
```

## 9、项目中你是如何做图片优化的？

(1)、降低图片的大小、选择适当的图片宽度尺寸，压缩图片，选择更小kb的图片格式，以达到最优的显示效果。

(2)、考虑CSS Sprites(背景精灵图/雪碧图)，将同类型的图标或按钮等背景图合到一张大图中，减少HTTP的网络资源请求。

(3)、Icon Font，将图标做成字体文件。优点是图标支持多个尺寸，兼容所有浏览器，减少页面请求等。

## 10、简述rem布局原理

rem是指相对于根元素的字体大小的单位,即根据html元素的font-size来计算大小。

比如说html的font-size大小为100px,一个div的width为1rem,则div的width大小为100px。

rem布局实现步骤：

(1) 设置页面的viewport 动态计算并设置html的fontsize值

(2) 按照pc端布局方式正常布局，把px单位换算成rem（较小的长度比如1px的边框就不需要转换成rem,直接设置px

## 11、解释下浮动和它的工作原理？清除浮动有哪些方式？

浮动为该元素脱离文档流，不占据空间。浮动元素碰到包含它的边框或者浮动元素的边框停留。

浮动引起的问题：

父元素的高度无法被撑开，影响与父元素同级的元素

与浮动元素同级的非浮动元素（内联元素）会紧跟其后

若非第一个元素浮动，则该元素之前的元素也需要浮动，否则会影响页面显示的结构  
清除浮动的方法：

- (1)、父级div定义height。
- (2)、结尾处加空div标签clear:both。
- (3)、父级div定义伪类:after和zoom。
- (4)、父级div定义overflow:hidden。
- (5)、父级div定义overflow:auto。
- (6)、父级div也浮动，需要定义宽度。
- (7)、父级div定义display:table。
- (8)、结尾处加br标签clear:both

## 12、什么是 BFC?

BFC 也就是常说的块格式化上下文，这是一个独立的渲染区域，规定了内部如何布局，并且这个区域的子元素不会影响到外面的元素，其中比较重要的布局规则有内部 box 垂直放置，计算 BFC 的高度的时候，浮动元素也参与计算，触发 BFC 的规则有根元素，浮动元素，position 为 absolute 或 fixed 的元素，display 为 inline-block, table-cell, table-caption, flex, inline-flex, overflow 不为 visible 的元素

## 13、对 CSS 的新属性有了解过的吗?

CSS3 的新特性中，在布局方面新增了 flex 布局，在选择器方面新增了例如 first-of-type, nth-child 等选择器，在盒模型方面添加了 box-sizing 来改变盒模型，在动画方面增加了 animation, 2d 变换, 3d 变换等，在颜色方面添加透明, rgba 等，在字体方面允许嵌入字体和设置字体阴影，最后还有媒体查讯等

## 14、讲一讲 CSS 的权重和优先级?

权重

从0开始，一个行内样式+1000，一个id选择器+100，一个属性选择器、class或者伪类+10，一个元素选择器，或者伪元素+1，通配符+0

优先级

权重相同，写在后面的覆盖前面的

使用 !important 达到最大优先级，都使用 !important 时，权重大的优先级高

## 15、介绍 Flex 布局，flex 是什么属性的缩写?

flex布局是CSS3新增的一种布局方式，可以通过将一个元素的display属性值设置为flex从而使它成为一个flex容器，它的所有子元素都会成为它的项目。一个容器默认有两条轴：一个是水平的主轴，一个是与主轴垂直的交叉轴。可以使用flex-direction来指定主轴的方向。可以使用justify-content来指定元素在主轴上的排列方式，使用align-items来指定元素在交叉轴上的排列方式。还可以使用flex-wrap来规定当一行排列不下时的换行方式。对于容器中的项目，可以使用order属性来指定项目的排列顺序，还可以使用flex-grow来指定当排列空间有剩余的时候，项目的放大比例，还可以使用flex-shrink来指定当排列空间不足时，项目的缩小比例。

- 弹性盒布局，CSS3 的新属性，用于方便布局，比如垂直居中

- flex属性是 flex-grow、flex-shrink 和 flex-basis 的简写

## 16、说一说你知道的position属性，都有啥特点？

static：无特殊定位，对象遵循正常文档流。top, right, bottom, left等属性不会被应用。

relative：对象遵循正常文档流，但将依据top, right, bottom, left等属性在正常文档流中偏移位置。而其层叠通过z-index属性定义。

absolute：对象脱离正常文档流，使用top, right, bottom, left等属性进行绝对定位。而其层叠通过z-index属性定义。

fixed：对象脱离正常文档流，使用top, right, bottom, left等属性以窗口为参考点进行定位，当出现滚动条时，对象不会随着滚动。而其层叠通过z-index属性定义。

sticky：具体是类似 relative 和 fixed，在 viewport 视口滚动到阈值之前应用 relative，滚动到阈值之后应用 fixed 布局，由 top 决定。

## 17、在网页中的应该使用奇数还是偶数的字体？为什么呢？

使用偶数字体。偶数字号相对更容易和 web 设计的其他部分构成比例关系。Windows 自带的点阵宋体（中易宋体）从 Vista 开始只提供 12、14、16 px 这三个大小的点阵，而 13、15、17 px 时用的是小一号的点。（即每个字占的空间大了 1 px，但点阵没变），于是略显稀疏。

## 18、margin和padding分别适合什么场景使用？

何时使用margin：

1. 需要在border外侧添加空白
2. 空白处不需要背景色
3. 上下相连的两个盒子之间的空白，需要相互抵消时。

何时使用padding：

1. 需要在border内侧添加空白
2. 空白处需要背景颜色
3. 上下相连的两个盒子的空白，希望为两者之和。

兼容性的问题：在IE5 IE6中，为float的盒子指定margin时，左侧的margin可能会变成两倍的宽度。通过改变padding或者指定盒子的display：inline解决。

## 19、CSS属性overflow属性定义溢出元素内容区的内容会如何处理？

参数是scroll时候，必会出现滚动条。

参数是auto时候，子元素内容大于父元素时出现滚动条。

参数是visible时候，溢出的内容出现在父元素之外。

参数是hidden时候，溢出隐藏。

## 20、对line-height是如何理解的？

行高是指一行文字的高度，具体说是两行文字间基线的距离。CSS中起高度作用的是height和line-height，没有定义height属性，最终其表现作用一定是line-height。

单行文本垂直居中：把line-height值设置为height一样大小的值可以实现单行文字的垂直居中，其实也可以把height删除。

多行文本垂直居中：需要设置display属性为inline-block。



## 21、元素竖向的百分比设定是相对于容器的高度吗？

当按百分比设定一个元素的宽度时，它是相对于父容器的宽度计算的，但是，对于一些表示竖向距离的属性，例如 `padding-top` , `padding-bottom` , `margin-top` , `margin-bottom` 等，当按百分比设定它们时，依据的也是父容器的宽度，而不是高度。

## 22、transition和animation的区别是什么？

过渡属性 `transition` 可以在一定的事件内实现元素的状态过渡为最终状态，用于模拟一种过渡动画效果，但是功能有限，只能用于制作简单的动画效果；

动画属性 `animation` 可以制作类似 Flash 动画，通过关键帧控制动画的每一步，控制更为精确，从而可以制作更为复杂的动画。

## 23、rgba()和opacity的透明效果有什么不同？

`rgba()` 和 `opacity` 都能实现透明效果，但最大的不同是 `opacity` 作用于元素，以及元素内的所有内容的透明度，

而 `rgba()` 只作用于元素的颜色或其背景色。（设置 `rgba` 透明的元素的子元素不会继承透明效果！）

## 24、css sprite是什么,有什么优缺点？

- 概念：将多个小图片拼接到一个图片中。通过 `background-position` 和元素尺寸调节需要显示的背景图案。
- 优点：
  - 减少 HTTP 请求数，极大地提高页面加载速度
  - 增加图片信息重复度，提高压缩比，减少图片大小
  - 更换风格方便，只需在一张或几张图片上修改颜色或样式即可实现
- 缺点：
  - 图片合并麻烦
  - 维护麻烦，修改一个图片可能需要从新布局整个图片，样式

## 25、display,float,position的关系？

如果 `display` 为 `none`，那么 `position` 和 `float` 都不起作用，这种情况下元素不产生框

否则，如果 `position` 值为 `absolute` 或者 `fixed`，框就是绝对定位的，`float` 的计算值为 `none`，`display` 根据下面的表格进行调整。

否则，如果 `float` 不是 `none`，框是浮动的，`display` 根据下表进行调整

否则，如果元素是根元素，`display` 根据下表进行调整

其他情况下 `display` 的值为指定值

总结起来：绝对定位、浮动、根元素都需要调整 `display`

## 26、css中link与@import的区别？

1、`@import` 是 CSS 提供的语法规则，只有导入样式表的作用；`link` 是 HTML 提供的标签，不仅可以加载 CSS 文件，还可以定义 RSS、rel 连接属性等。

2、加载页面时，`link` 引入的 CSS 被同时加载，`@import` 引入的 CSS 将在页面加载完毕后加载。

3、link标签作为HTML元素，不存在兼容性问题，而@import是CSS2.1才有的语法，故老版本浏览器（IE5之前）不能识别。

4、可以通过JS操作DOM，来插入link标签改变样式；由于DOM方法是基于文档的，无法使用@import方式插入样式。

## 27、CSS3新特性有哪些？

- 1、颜色：新增RGBA、HSLA模式
- 2、文字阴影(text-shadow)
- 3、边框：圆角 (border-radius) 边框阴影：box-shadow
- 4、盒子模型：box-sizing
- 5、背景：background-size设置背景图片的尺寸，background-origin设置背景图片的原点，background-clip设置背景图片的裁剪区域，以“,”分隔可以设置多背景，用于自适应布局
- 6、渐变：linear-gradient、radial-gradient
- 7、过渡：transition可实现动画
- 8、自定义动画
- 9、在CSS3中唯一引入的伪元素是::selection
- 10、多媒体查询、多栏布局
- 11、border-image
- 12、2D转换：transform:translate(x,y)rotate(x,y)skew(x,y)scale(x,y)
- 13、3D转换

## 28、display:none;与visibility:hidden的区别是什么？

display:none;使用该属性后，HTML元素（对象）的宽高，高度等各种属性值都将“丢失”；

visibility:hidden;使用该属性后，HTML元素（对象）仅仅是在视觉上看不见（完全透明），而它所占据的空间位置仍然存在，也即是说它仍然具有高度，宽度等属性值。

## 29、画一条0.5px的直线？

```
height: 1px;  
transform: scale(0.5);
```

## 30、html元素的id跟class什么区别

id和class是网页中两个通用属性，他们协同工作使整个页面丰富多彩，当我们为一个元素定义样式时，二者都可用，但有区别？

- 1、在css样式表中书写时，id选择符前缀应加"#", class选择符前缀应加"."
- 2、id属性在一个页面中书写时只能使用一次，而class可以反复使用
- 3、id作为元素标签用于区分不同结构和内容，而class作为一个样式，可以应用到任何结构和内容当中去
- 4、布局上的一般原则：id先确定结构和内容再为它定义样式。而class正好相反，是先定义样式，然后在页面中根据不同需求把样式应用到不同结构和内容上

- 5、目前浏览器都允许同一个页面出现多个相同属性值的id，一般情况能正常显示，不过当javascript通过id来控制元素时就会出错
- 6、在实际应用中，class常被用到文字版块和页面修饰上，而id多被用在宏伟布局和设计包含块，或包含框的样式。

### 31、什么是响应式设计，响应式设计的基本原理是什么

响应式网站设计是一个网站能够兼容多个终端，而不是为每一个终端做一个特定的版本。基本原理是通过媒体查询检测不同的设备屏幕尺寸做处理。

### 32、什么是外边距重叠？重叠的结果是什么？

外边距重叠就是 margin-collapse在CSS中，相邻的两个盒子（可能是兄弟关系也可能是祖先关系）的外边距可以结合成一个单独的外边距。这种合并外边距的方式称为折叠，因此而结合成的外边距称为折叠外边距。

折叠结果遵循下列计算规则。

- (1) 当两个相邻的外边距都是正数时，折叠的结果是它们两者中较大的值
- (2) 当两个相邻的外边距都是负数时，折叠的结果是两者中绝对值较大的值。
- (3) 当两个外边距一正一负时，折叠的结果是两者相加的和。

### 33、CSS属性content有什么作用？有什么应用？

content属性与::before及::after伪元素配合使用，用来插入生成的内容，可以在元素之前或之后放置生成的内容。可以插入文本、图像、引号，并可以结合计数器，为页面元素插入编号。

### 34、CSS优先级如何排序？

优先级如下：

! important > style (内联) > Id (权重100) > class (权重10) > 标签 (权重1)。同类别的样式中，后面的会覆盖前面的。

### 35、nth-of-type和nth-child的区别

:nth-of-type(n) 选择器匹配属于父元素的特定类型的第 N 个子元素的每个元素  
:nth-child(n) 选择器匹配属于其父元素的第 N 个子元素，不论元素的类型

### 36、有什么方式可以对一个DOM设置它的CSS？

外链式，即通过link标签引入一个外部CSS文件中。

内嵌式，即将CSS代码写在 style 标签内。

行内式，即将CSS代码写在元素的 style 属性中。

### 37、CSS中，自适应的单位都有哪些？

自适应的单位有以下几个

百分比：%

相对于视口宽度的单位：vw

相对于视口高度的单位：vh

相对于视口宽度或者高度（取决于哪个小）的单位：Vmin



相对于父元素字体大小的单位: em

相对于根元素字体大小的单位: rem

### 38、为什么css放在顶部而js写在后面?

- 1.浏览器预先加载css后, 可以不必等待HTML加载完毕就可以渲染页面了
- 2.其实HTML渲染并不会等到完全加载完在渲染页面, 而是一边解析DOM一边渲染。
- 3.js写在尾部, 主要是因为js主要扮演事件处理的功能, 一方面很多操作是在页面渲染后才执行的。另一方面可以节省加载时间, 使页面能够更加的加载, 提高用户的良好体验

但是随着JS技术的发展, JS也开始承担页面渲染的工作。比如我们的UI其实可以分被对待, 把渲染页面的js放在前面, 时间处理的js放在后面。

### 39、z-index属性在什么情况下会失效

通常 z-index 的使用是在有两个重叠的标签, 在一定的情况下控制其中一个在另一个的上方或者下方出现。z-index值越大就越是在上层。z-index元素的position属性需要是relative, absolute或是fixed。

z-index属性在下列情况下会失效:

父元素position为relative时, 子元素的z-index失效。解决: 父元素position改为absolute或static;

元素没有设置position属性为非static属性。解决: 设置该元素的position属性为relative, absolute或是fixed中的一种;

元素在设置z-index的同时还设置了float浮动。解决: float去除, 改为display: inline-block;

### 40、Flex 布局父级容器属性和子级项目属性有哪些?

以下6个属性设置在容器上:

flex-direction属性决定主轴的方向(即项目的排列方向)。

flex-wrap属性定义, 如果一条轴线排不下, 如何换行。

flex-flow属性是flex-direction属性和flex-wrap属性的简写形式, 默认值为row nowrap。

justify-content属性定义了项目在主轴上的对齐方式。

align-items属性定义项目在交叉轴上如何对齐。

align-content属性定义了多根轴线的对齐方式。如果项目只有一根轴线, 该属性不起作用。

以下6个属性设置在项目上:

order属性定义项目的排列顺序。数值越小, 排列越靠前, 默认为0。

flex-grow属性定义项目的放大比例, 默认为0, 即如果存在剩余空间, 也不放大。

flex-shrink属性定义了项目的缩小比例, 默认为1, 即如果空间不足, 该项目将缩小。

flex-basis属性定义了再分配多余空间之前, 项目占据的主轴空间。浏览器根据这个属性, 计算主轴是否有多余空间。它的默认值为auto, 即项目的本来大小。

flex属性是flex-grow, flex-shrink和flex-basis的简写, 默认值为0 1 auto。

align-self属性允许单个项目有与其他项目不一样的对齐方式, 可覆盖align-items属性。默认值为auto, 表示继承父元素的align-items属性, 如果没有父元素, 则等同于stretch。



## 41、flex 布局中 align-content 与 align-items 有何区别？

align-content 作用于纵轴多行元素，一行元素不起作用

align-items 作用于纵轴单行元素

## 42、 '+' 与 '~' 选择器有什么不同？

+ 选择器匹配紧邻的兄弟元素

~ 选择器匹配随后的所有兄弟元素

## JavaScript 基础篇

### 1、JavaScript 有几种数据类型

基本数据类型：undefined、null、boolean、number、string、symbol(es6 的新数据类型)

引用数据类型：object、array、function

### 2、怎么进行数据类型检测

typeof 可以检测除了 null 以外的基础数据类型，typeof 对于对象来说，除了函数都会显示 object

Object.prototype.toString.call() 可以检测所有数据类型

### 3、get 请求传参长度的误区

我们经常说 get 请求参数的大小存在限制，而 post 请求的参数大小是无限制的。实际上 HTTP 协议从未规定 GET/POST 的请求长度限制是多少。对 get 请求参数的限制是来源与浏览器或 web 服务器，浏览器或 web 服务器限制了 url 的长度。为了明确这个概念，我们必须再次强调下面几点：

HTTP 协议 未规定 GET 和 POST 的长度限制

GET 的最大长度显示是因为 浏览器和 web 服务器限制了 URI 的长度

不同的浏览器和 WEB 服务器，限制的最大长度不一样

要支持 IE，则最大长度为 2083byte，若只支持 Chrome，则最大长度 8182byte

### 4、如何让事件先冒泡后捕获

在 DOM 标准事件模型中，是先捕获后冒泡。但是如果要实现先冒泡后捕获的效果，对于同一个事件，监听捕获和冒泡，分别对应相应的处理函数，监听到捕获事件，先暂缓执行，直到冒泡事件被捕获后再执行捕获之间。

### 5、说一下事件委托？

事件委托指的是，不在事件的发生地（直接 dom）上设置监听函数，而是在其父元素上设置监听函数，通过事件冒泡，父元素可以监听到子元素上事件的触发，通过判断事件发生元素 DOM 的类型，来做出不同的响应。

举例：最经典的就是 ul 和 li 标签的事件监听，比如我们在添加事件时候，采用事件委托机制，不会在 li 标签上直接添加，而是在 ul 父元素上添加。

好处：比较合适动态元素的绑定，新添加的子元素也会有监听函数，也可以有事件触发机制。

## 6、说一下图片的懒加载和预加载？

预加载：提前加载图片，当用户需要查看时可直接从本地缓存中渲染。

懒加载：懒加载的主要目的是作为服务器前端的优化，减少请求数或延迟请求数。

两种技术的本质：两者的行为是相反的，一个是提前加载，一个是迟缓甚至不加载。

懒加载对服务器前端有一定的缓解压力作用，预加载则会增加服务器前端压力。

## 7、mouseover 和 mouseenter 的区别？

mouseover：当鼠标移入元素或其子元素都会触发事件，所以有一个重复触发，冒泡的过程。对应的移除事件是 mouseout

mouseenter：当鼠标移除元素本身（不包含元素的子元素）会触发事件，也就是不会冒泡，对应的移除事件是 mouseleave

## 8、JS 的 new 操作符做了哪些事情？

new 操作符新建了一个空对象，这个对象原型指向构造函数的 prototype，执行构造函数后返回这个对象。

## 9、改变函数内部 this 指针的指向函数（bind，apply，call 的区别）

通过 apply 和 call 改变函数的 this 指向，他们两个函数的第一个参数都是一样的表示要改变指向的那个对象，第二个参数，apply 是数组，而 call 则是 arg1,arg2...这种形式。通过 bind 改变 this 作用域会返回一个新的函数，这个函数不会马上执行。

## 10、箭头函数中this 指向举例？

```
var a=11;

function test2(){
  this.a=22;

  let b=()=>{console.log(this.a)}
  b();
}

var x=new test2();

//输出 22

定义时绑定。
```

## 11、数组常用方法有哪些？

push(), pop(), shift(), unshift(), splice(), sort(), reverse(), map()等

## 12、说说数组去重？

法一：indexOf 循环去重

法二：ES6 Set 去重；Array.from(new Set(array))

法三：Object 键值对去重；把数组的值存成 Object 的 key 值，比如 Object[value1] = true，在判断另一个值的时候，如果 Object[value2]存在的话，就说明该值是重复的。

### 13、闭包有什么用？

(1) 什么是闭包：

闭包是指有权访问另外一个函数作用域中的变量的函数。

闭包就是函数的局部变量集合，只是这些局部变量在函数返回后会继续存在。闭包就是函数的“堆栈”在函数返回后并不释放，我们也可以理解为这些函数堆栈并不在栈上分配而是在堆上分配。当在一个函数内定义另外一个函数就会产生闭包。

(2) 为什么要用：

匿名自执行函数：我们知道所有的变量，如果不加上 var 关键字，则默认会添加到全局对象的属性上去，这样的临时变量加入全局对象有很多坏处，比如：别的函数可能误用这些变量；造成全局对象过于庞大，影响访问速度(因为变量的取值是需要从原型链上遍历的)。除了每次使用变量都是用 var 关键字外，我们在实际情况经常遇到这样一种情况，即有的函数只需要执行一次，其内部变量无需维护，可以用闭包。

结果缓存：我们开发中会碰到很多情况，设想我们有一个处理过程很耗时的函数对象，每次调用都会花费很长时间，那么我们就需要将计算出来的值存储起来，当调用这个函数的时候，首先在缓存中查找，如果找不到，则进行计算，然后更新缓存并返回值，如果找到了，直接返回查找到的值即可。闭包正是可以做到这一点，因为它不会释放外部的引用，从而函数内部的值可以得以保留。

封装：实现类和继承等

### 14、闭包的特点？

- 1、让外部访问函数内部变量成为可能
- 2、局部变量会常驻在内存中
- 3、可以避免使用全局变量，防止全局变量污染
- 4、会造成内存泄漏（有一块内存空间被长期占用，而不被释放）

### 15、JS 基本数据类型？

基本数据类型：undefined、null、number、boolean、string、symbol

16、JS 深度拷贝一个元素的具体实现？

```
var deepCopy = function(obj) {  
  if (typeof obj !== 'object') return;  
  var newObj = obj instanceof Array ? [] : {};  
  for (var key in obj) {  
    if (obj.hasOwnProperty(key)) {  
      newObj[key] = typeof obj[key] === 'object' ? deepCopy(obj[key]) : obj[key];  
    }  
  }  
  return newObj;  
}
```



## 16、this 的指向哪几种

默认绑定：全局环境中，this 默认绑定到 window。

隐式绑定：一般地，被直接对象所包含的函数调用时，也称为方法调用，this 隐式绑定到该直接对象。

隐式丢失：隐式丢失是指被隐式绑定的函数丢失绑定对象，从而默认绑定到 window。显式绑定：通过 call()、apply()、bind() 方法把对象绑定到 this 上，叫做显式绑定。

new 绑定：如果函数或者方法调用之前带有关键字 new，它就构成构造函数调用。对于 this 绑定来说，称为 new 绑定。

【1】构造函数通常不使用 return 关键字，它们通常初始化新对象，当构造函数的函数体执行完毕时，它会显式返回。在这种情况下，构造函数调用表达式的计算结果就是这个新对象的值。

【2】如果构造函数使用 return 语句但没有指定返回值，或者返回一个原始值，那么这时将忽略返回值，同时使用这个新对象作为调用结果。

【3】如果构造函数显式地使用 return 语句返回一个对象，那么调用表达式的值就是这个对象。

## 17、什么是暂停死区？

在代码块内，使用 let、const 命令声明变量之前，该变量都是不可用的。这在语法上，称为“暂时性死区”。

## 18、什么是按需加载？

当用户触发了动作时才加载对应的功能。触发的动作，是要看具体的业务场景而言，包括但不限于以下几个情况：鼠标点击、输入文字、拉动滚动条，鼠标移动、窗口大小更改等。加载的文件，可以是 JS、图片、CSS、HTML 等。

## 19、什么是事件监听？

addEventListener() 方法，用于向指定元素添加事件句柄，它可以更简单的控制事件，语法为

```
element.addEventListener(event, function, useCapture);
```

第一个参数是事件的类型(如 "click" 或 "mousedown")。

第二个参数是事件触发后调用的函数。

第三个参数是个布尔值用于描述事件是冒泡还是捕获。该参数是可选的。

事件传递有两种方式，冒泡和捕获

事件传递定义了元素事件触发的顺序，如果你将 P 元素插入到 div 元素中，用户点击 P 元素，在冒泡中，内部元素先被触发，然后再触发外部元素，捕获中，外部元素先被触发，在触发内部元素。

## 20、什么是 js 的闭包？用闭包写个单例模式？

MDN 对闭包的定义是：闭包是指那些能够访问自由变量的函数，自由变量是指在函数中使用的，但既不是函数参数又不是函数的局部变量的变量，由此可以看出，闭包=函数+函数能够访问的自由变量，所以从技术的角度讲，所有 JS 函数都是闭包，但是这是理论上的闭包，还有一个实践角度上的闭包，从实践角度上来说，只有满足 1、即使创建它的上下文已经销毁，它仍然存在，2、在代码中引入了自由变量，才称为闭包。

闭包的应用：

模仿块级作用域。2、保存外部函数的变量。3、封装私有变量

单例模式：



```

var Singleton = (function(){
var instance;

var CreateSingleton = function (name) {

this.name = name;

if(instance) {

return instance;

}

// 打印实例名字

this.getName();

// instance = this;

// return instance;

return instance = this;

}

// 获取实例的名字

CreateSingleton.prototype.getName = function() {

console.log(this.name)

}

return CreateSingleton;

})();

// 创建实例对象 1

var a = new Singleton('a');

// 创建实例对象 2

var b = new Singleton('b');

console.log(a===b);

```

## 21、深浅拷贝的区别和实现

数组的浅拷贝：

如果是数组，我们可以利用数组的一些方法，比如 slice，concat 方法返回一个新数组的特性来实现拷贝，但假如数组嵌套了对象或者数组的话，使用 concat 方法克隆并不完整，如果数组元素是基本类型，就会拷贝一份，互不影响，而如果是对象或数组，就会只拷贝对象和数组的引用，这样我们无论在新旧数组进行了修改，两者都会发生变化，我们把这种复制引用的拷贝方法称为浅拷贝，

深拷贝就是指完全的拷贝一个对象，即使嵌套了对象，两者也互相分离，修改一个对象的属性，不会影响另一个如何深拷贝一个数组1、这里介绍一个技巧，不仅适用于数组还适用于对象！那就是：

```

var arr = ['old', 1, true, ['old1', 'old2'], {old: 1}]

var new_arr = JSON.parse(JSON.stringify(arr));

```

```
console.log(new_arr);
```

原理是 JSON 对象中的 stringify 可以把一个 js 对象序列化为一个 JSON 字符串，parse 可以把 JSON 字符串反序列化为一个 js 对象，通过这两个方法，也可以实现对象的深复制。但是这个方法不能够拷贝函数。

浅拷贝的实现：

以上三个方法 concat,slice ,JSON.stringify 都是技巧类，根据实际项目情况选择使用，我们可以思考下如何实现一个对象或数组的浅拷贝，遍历对象，然后把属性和属性值都放在一个新的对象里即可

```
var shallowCopy = function(obj) {  
  // 只拷贝对象  
  if (typeof obj !== 'object') return;  
  // 根据 obj 的类型判断是新建一个数组还是对象  
  var newObj = obj instanceof Array ? [] : {};  
  // 遍历 obj，并且判断是 obj 的属性才拷贝  
  for (var key in obj) {  
    if (obj.hasOwnProperty(key)) {  
      newObj[key] = obj[key];  
    }  
  }  
  return newObj;  
}
```

深拷贝的实现

那如何实现一个深拷贝呢？说起来也好简单，我们在拷贝的时候判断一下属性值的类型，如果是对象，我们递归调用深拷贝函数不就好了

```
var deepCopy = function(obj) {  
  if (typeof obj !== 'object') return;  
  var newObj = obj instanceof Array ? [] : {};  
  for (var key in obj) {  
    if (obj.hasOwnProperty(key)) {  
      newObj[key] = typeof obj[key] === 'object'  
        ? deepCopy(obj[key]) : obj[key];  
    }  
  }  
  return newObj;  
}
```

## 22、JS 中 string 的 startwith 和 indexof 两种方法的区别

JS 中 startwith 函数，其参数有 3 个，stringObj,要搜索的字符串对象，str，搜索的字符串，position，可选，从哪个位置开始搜索，如果以 position 开始的字符串以搜索字符串开头，则返回 true，否则返回 false。indexof 函数，indexof 函数可返回某个指定字符串在字符串中首次出现的位置。

## 23、JS 字符串转数字的方法？

通过函数 parseInt ( )，可解析一个字符串，并返回一个整数，语法为 parseInt (string ,radix)

string: 被解析的字符串

radix: 表示要解析的数字的基数，默认是十进制，如果 radix<2 或>36,则返回 NaN

## 24、构造函数是什么？

构造函数和普通函数在定义上没有强制要求，但是大家默认构造函数的首字母大写，并且在使用构造函数的时候使用new关键字调用。

## 25、构造函数的执行过程？

在new关键字调用时会创建一个新的空间，每当创建实例时函数体内部this都会指向当前

- 1、立刻在堆内存中创建一个新的对象
- 2、将新建的对象设置为函数中的this
- 3、逐个执行函数中的代码
- 4、将新建的对象作为返回值

## 26、构造函数this指向？

this的指向在函数定义的时候是确定不了的，只有函数执行的时候才能确定this到底指向谁，实际上this的最终指向的是那个调用它的对象，在构造函数中，this指向构造实例

## 27、什么是作用域？

ES5 中只存在两种作用域：全局作用域和函数作用域。在 JavaScript 中，我们将作用域定义为一套规则，这套规则用来管理引擎如何在当前作用域以及嵌套子作用域中根据标识符名称进行变量（变量名或者函数名）查找

## 28、NaN 是什么，用 typeof 会输出什么？

Not a Number，表示非数字，typeof NaN === 'number'

## 29、说说写JavaScript的基本规范？

- 1、不要在同一行声明多个变量
- 2、使用 ===或!==来比较true/false或者数值
- 3、switch必须带有default分支
- 4、函数应该有返回值
- 5、for if else 必须使用大括号
- 6、语句结束加分号
- 7、命名要有意义，使用驼峰命名法

### 30、栈和堆的区别？

栈 (stack)：由编译器自动分配释放，存放函数的参数值，局部变量等；

堆 (heap)：一般由程序员分配释放，若程序员不释放，程序结束时可能由操作系统释放。

### 31、null，undefined的区别？

null表示一个对象被定义了，但存放了空指针，转换为数值时为0。

undefined表示声明的变量未初始化，转换为数值时为NaN。

`typeof(null) -- object;`

`typeof(undefined) -- undefined`

### 32、什么是===运算符？说明“==”和“===”之间的区别？

===被称为严格等式运算符，当两个操作数具有相同的值而没有任何类型转换时，该运算符返回true。

==仅检查值相等，而===是一个更严格的等式判定，如果两个变量的值或类型不同，则返回false。

### 33、break和continue语句的作用？

Break语句从当前循环中退出。

continue语句继续下一个循环语句。

**获取更多资源，对话微软MVP，微信扫一扫！**



微信公众号



助教小姐姐



助教小仙女