(2020.11.02)

说说浏览器解析CSS选择器的过程？为什么这么解析

浏览器从右往左进行解析。因为在遍历 DOM Tree 时，对于每个 DOM 元素，必须在所有 Style Rules 中找到符合的 selector 并将对应的规则进行合并，而所有样式规则可能数量很大，而且绝大多数不会匹配到当前的 DOM 元素，所以需要高效的解析顺序。而从左往右一层层匹配的话，会产生不必要的遍历和回溯，造成效率很低；从右往左的话，可以将匹配到不相同的元素直接排除，只有匹配到相同元素时，才会进行验证，从而减少不必要的回溯，尽早排除无关规则和元素，提高整体效率。

CSS专题（2020.11.03）

说说CSS里面有哪些选择器类型， CSS权重是怎么计算的

CSS选择器，属性和属性值区分大小写吗？

class 和 id 选择器有什么区别

1)

id选择器(#id),类选择器（.className）,元素选择器(E) ,群组选择器（selector1,selector2,...,selectorN）,通用兄弟选择器（Ｅ〜F）,相邻兄弟元素选择器(E + F) ,子元素选择器(E>F) ,后代选择器（Ｅ Ｆ）,通配符选择器（＊）,伪类选择器,伪元素选择器,属性选择器

权重计算规则:

第一等：代表内联样式，如: style=””，权值为1000。

第二等：代表ID选择器，如：#content，权值为0100。

第三等：代表类，伪类和属性选择器，如.content，权值为0010。

第四等：代表类型选择器和伪元素选择器，如div p，权值为0001。

通配符、子选择器、相邻选择器等的。如\*、>、+,权值为0000。

继承的样式没有权值。

2) CSS样式中的选择器严格区分大小写,属性和值可以不区分大小写

3)不同元素可以用相同的class，一个元素只能有一个id且不同元素之间id值不能相同。即：class选择器对应一个class相同的多个元素设置布局样式，id选择器只能给一个元素设置布局样式

CSS专题 （2020.11.04）

CSS里面的单位有哪些？（提示： 相对单位 && 绝对单位）

这些单位的定位原则

响应式Web设计中会使用到的单位有哪些

绝对长度单位是一个固定的值，它反应一个真实的物理尺寸。绝对长度单位视输出介质而定，不依赖于环境（显示器、分辨率、操作系统等）如：

pt 磅 (1 pt 等于 1/72 in)

pc 12 点活字 (1 pc =12 pt)

px 像素 (计算机屏幕上的一个点,1/96th of 1 in )

in 英寸（1 in=96px=2.54cm）

cm 厘米

mm 毫米

相对长度单位指定了一个长度相对于另一个长度的属性。对于不同的设备相对长度更适用。如：

em 基准点为父节点字体的大小（父元素）改变font-size的大小时，它的尺寸就会发生变化。

rem值根据（html）根元素的大小而定，同样可以作为宽度，高度等单位

vw 可视口宽度，将可视口宽度的百分之一（viewport-width)

vh 可视口高度，将可视口高度的百分之一（viewport-height)

vmin 比较可视口宽度和高度 中较小的那个

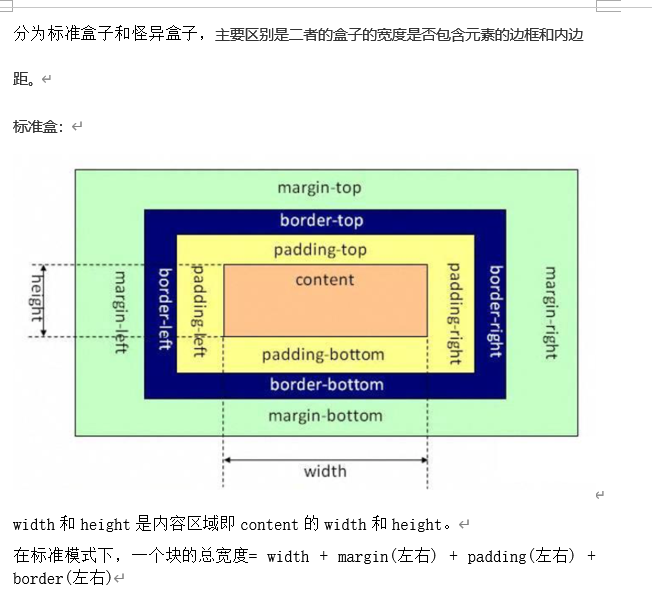
vmax 比较可视口宽度和高度 中较大的那个

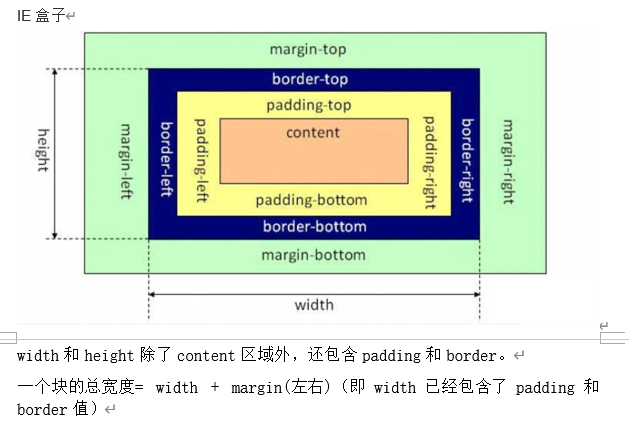
% 百分比是一个相对长度单位，相对于包含块（containing block）的高宽或字体大小。

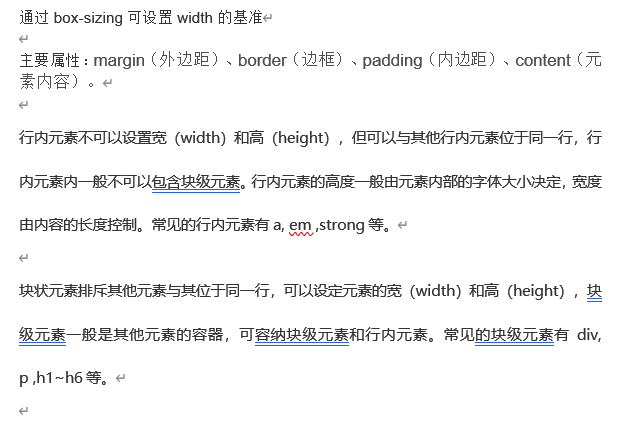
响应式Web设计中会使用到的单位有：vw,vh,vmin,vmax,%

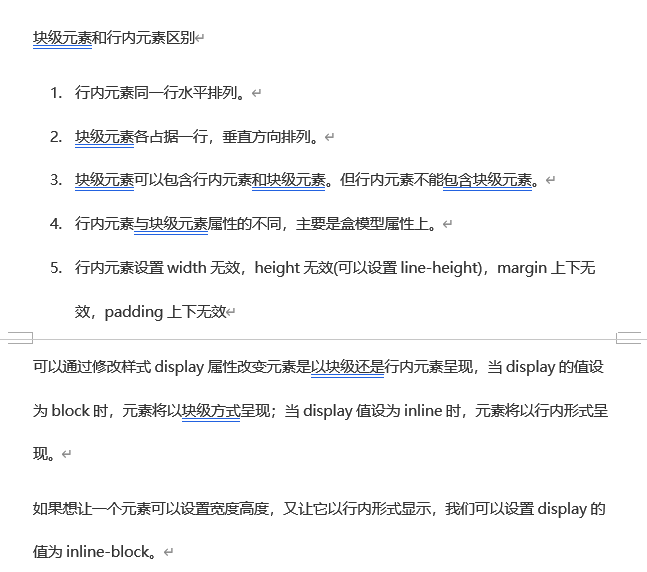
CSS专题 （2020.11.05）

了解CSS的盒模型，为什么存在两种，这两种有什么区别， 通过哪个CSS属性可以设置width的基准， 盒模型主要有哪些属性，这些属性对块级元素和行内元素分别有什么效果



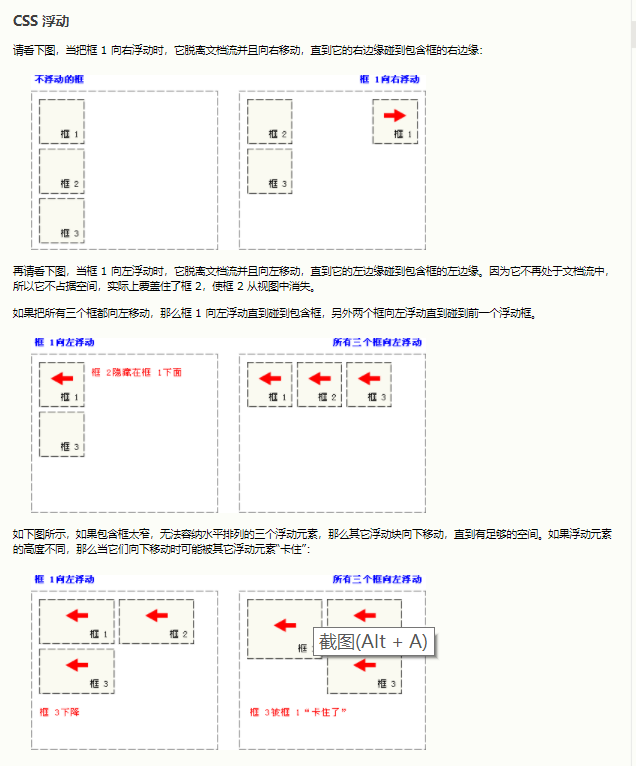






2020.11.06

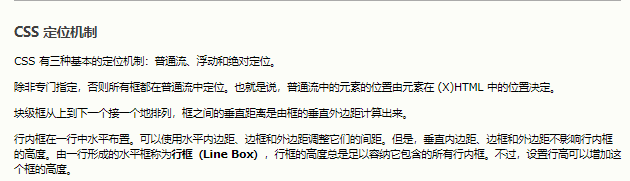
了解CSS浮动，清除浮动的方法，clear清除浮动的原理。

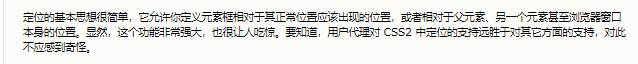


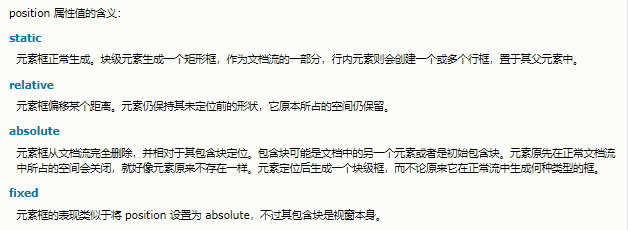




了解CSS定位，每种定位的定位基准，使用场景











了解float和absolute的区别

absolute：脱离文档流，覆盖其他元素

float：会占据位置

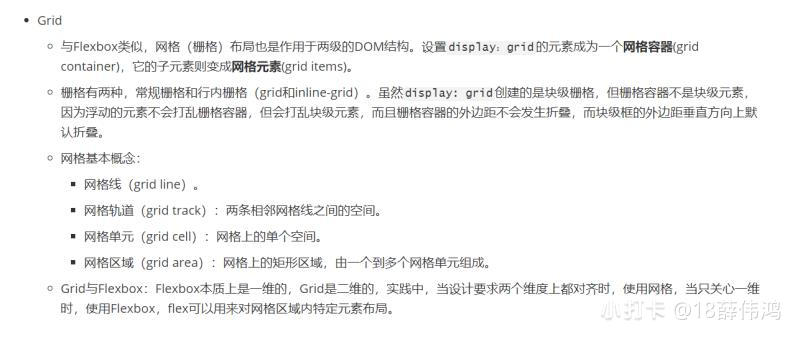
2020.11.07

了解display的几种定位类型，重点在Flex, Grid这两个上面









2020.11.08

学习CSS中的transform && transition

transform用来移动，比例化，反转，旋转，和拉伸元素。transition为元素从一种样式变成另一种样式的过程添加过渡动画，transition的作用在于，指定状态变化所需要的时间。





### transition的使用注意

（1）目前，各大浏览器（包括IE 10）都已经支持无前缀的transition，所以transition已经可以很安全地不加浏览器前缀。

（2）不是所有的CSS属性都支持transition，完整的列表查看[这里](http://oli.jp/2010/css-animatable-properties/" \t "_blank)，以及具体的[效果](http://leaverou.github.io/animatable/)。

（3）transition需要明确知道，开始状态和结束状态的具体数值，才能计算出中间状态。比如，height从0px变化到100px，transition可以算出中间状态。但是，transition没法算出0px到auto的中间状态，也就是说，如果开始或结束的设置是height: auto，那么就不会产生动画效果。类似的情况还有，display: none到block，background: url(foo.jpg)到url(bar.jpg)等等。

### 1.6 transition的局限

transition的优点在于简单易用，但是它有几个很大的局限。

（1）transition需要事件触发，所以没法在网页加载时自动发生。

（2）transition是一次性的，不能重复发生，除非一再触发。

（3）transition只能定义开始状态和结束状态，不能定义中间状态，也就是说只有两个状态。

（4）一条transition规则，只能定义一个属性的变化，不能涉及多个属性。

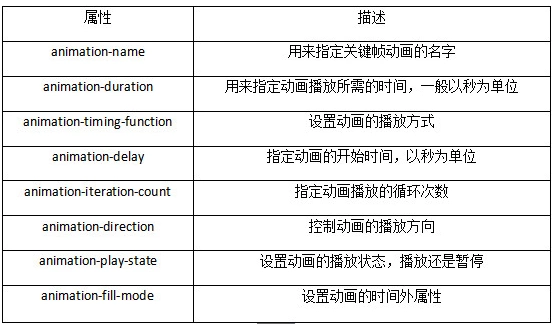
2020.11.09

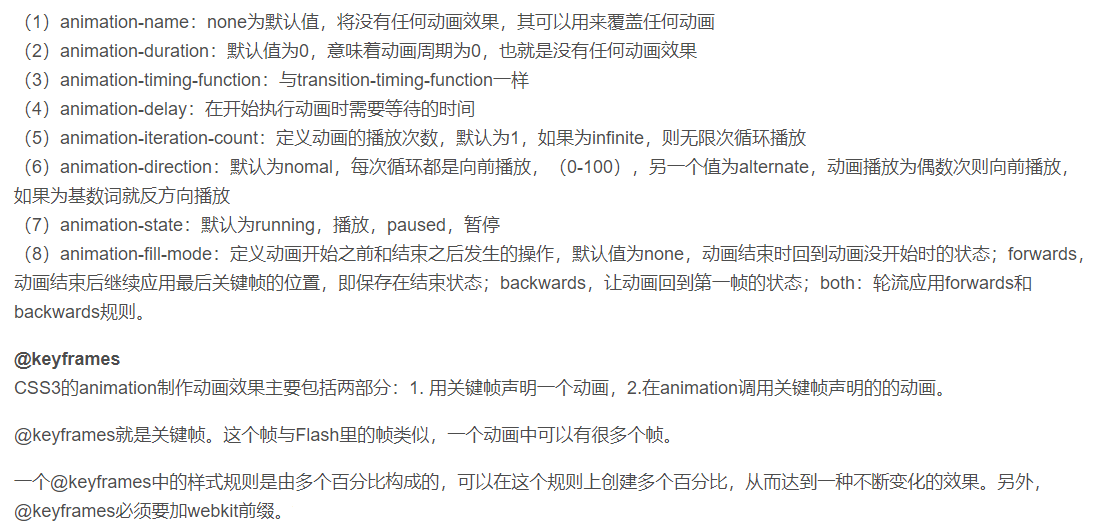
学习CSS中的animation动画

CSS3的animation属性可以像Flash制作动画一样，通过控制关键帧来控制动画的每一步，实现更为复杂的动画效果。ainimation实现动画效果主要由两部分组成：

1）通过类似Flash动画中的帧来声明一个动画；  
2）在animation属性中调用关键帧声明的动画。\*\*

注：animation属性到目前位置得到了大多数浏览器的支持，但是，需要添加浏览器前缀

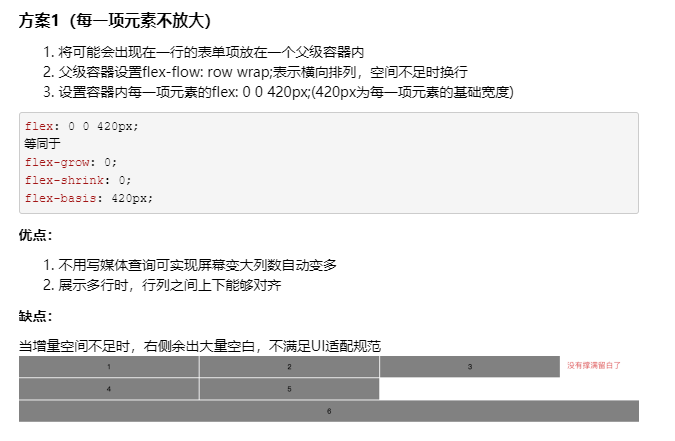




2020.11.10

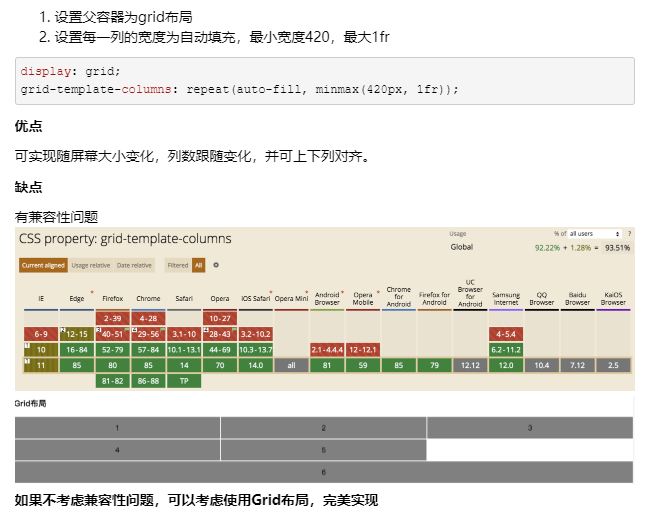
学习响应式Web开发， 重点了解CSS相对单位，flex， gird布局如何在响应式开发中使用的，了解媒体查询，查阅谷歌开发者博客中关于响应式Web开发的博客，了解相关知识。

flex：





grid





js基础 & css面试题 --2020.11.11

js基础

1.了解什么是Javascript, 什么是ECMAScript，两者之间的关系

2.了解HTML文件中引入JS的方式

3.了解JS中常见的几种数据类型

css面试题

1.RGBA() 与 opacity 在透明效果上有什么区别？

1.

什么是JavaScript？

1、JavaScript是一种在网络浏览器上运行的编程语言。

它主要添加到使用HTML和CSS构建的网站中，并在实现各种页面动态效果方面发挥作用。 例如，网页上显示的轮播模式和在咨询平台上输入内容后显示的提示信息不正确等。另外，在商场网站上购物时，购物车的使用和成本估算也 需要JavaScript。 因此，即使不是很明显，它仍然是编码人员经常使用的语言。

2、JavaScript是一种多功能开发语言。

除了向网页添加操作的有用性之外，JavaScript还可以用于实现许多其他功能。 例如，它可以开发各种网站应用程序，例如“聊天工具”和“货币图表”，它们可以实时修改，这是JavaScript的主要魅力之一。

JavaScript的特点是什么？

1，脚本语言。 JavaScript是一种解释型脚本语言。首先编译和执行C，C ++和其他语言，同时在程序运行期间逐行解释JavaScript。

2.基于对象。 JavaScript是一种基于对象的脚本语言，不仅可以创建对象，还可以使用现有对象。

3.简单。 JavaScript语言使用弱类型的变量类型。它对使用的数据类型没有严格要求。它是一种基于Java基本语句和控件的脚本语言。它的设计既简单又紧凑。

4，动态。 JavaScript是一种事件驱动的脚本语言，无需通过Web服务器即可响应用户输入。访问网页时，鼠标可以在网页上单击鼠标，或上下移动并移动窗口。 JavaScript可以直接响应这些事件。

5.跨平台。 JavaScript脚本语言不依赖于操作系统，仅需要浏览器支持。因此，只要机器上的浏览器支持JavaScript脚本语言，那么编写后就可以在任何机器上使用JavaScript脚本。目前，大多数浏览器都支持JavaScript。

什么是ECMAScript、什么又是ECMA?

Ecma国际（Ecma International）是一家国际性会员制度的信息和电信标准组织。1994年之前，名为[欧洲计算机制造商协会](http://baike.baidu.com/view/2233504.htm)（European Computer Manufacturers Association）。因为计算机的国际化，组织的标准牵涉到很多其他国家，因此组织决定改名表明其国际性。现名称已不属于首字母缩略字。

ECMAScript是一种由[Ecma国际](http://baike.baidu.com/view/3986646.htm)（前身为[欧洲计算机制造商协会](http://baike.baidu.com/view/2233504.htm),英文名称是European Computer Manufacturers Association）通过ECMA-262标准化的脚本[程序设计语言](http://baike.baidu.com/view/128511.htm)。这种语言在[万维网](http://baike.baidu.com/view/7833.htm)上应用广泛，它往往被称为[JavaScript](http://baike.baidu.com/view/16168.htm)或[JScript](http://baike.baidu.com/view/40829.htm)，但实际上后两者是ECMA-262标准的实现和扩展。

什么是TypeScript？

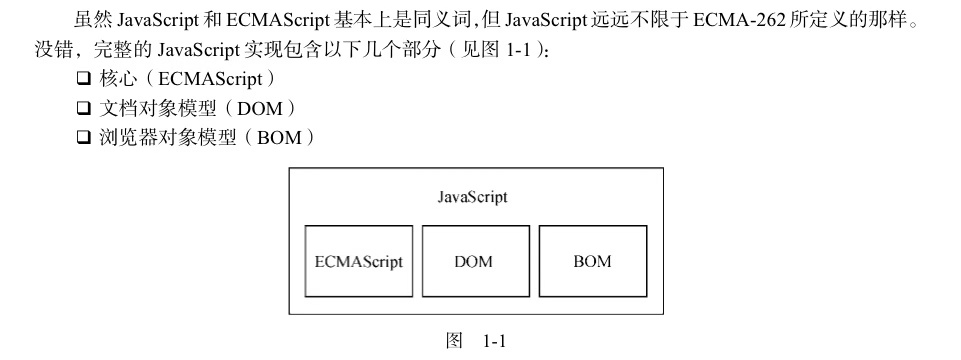
TypeScript是一种由微软开发的自由和[开源](http://baike.baidu.com/view/9664.htm" \t "_blank)的编程语言。

它是[JavaScript](http://baike.baidu.com/view/16168.htm" \t "_blank)的一个超集，而且本质上向这个语言添加了可选的静态类型和基于类的[面向对象编程](http://baike.baidu.com/view/10110.htm" \t "_blank)。TypeScript扩展了[JavaScript](http://baike.baidu.com/view/16168.htm" \t "_blank)的 语法，所以任何现有的JavaScript程序可以不加改变的在TypeScript下工作。TypeScript是为大型应用之开发而设计，而编译时它产生 JavaScript 以确保兼容性。它与JavaScript相比进步的地方包括：加入注释，让编译器理解所支持的对象和函数，编译器会移除注释，不会增加开销。 而JavaScript只是一个脚本语言，并非设计用于开发大型 Web 应用，*JavaScript* 没有提供类和模块的概念，而TypeScript扩展实现了这些特性。

*TypeScript* 最大的特点就是类型化，因此才叫做TypeScript。比起弱类型的JavaScript，类型化的TypeScript显得更加容易维护。

Javascript, ECMAScript两者之间的关系？

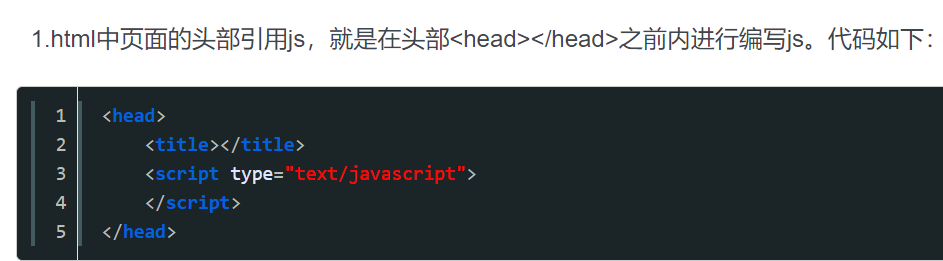
简单来说，ECMAScript是JavaScript语言的国际标准，JavaScript是ECMAScript的实现。



2. HTML文件中引入JS的方式

在html中引用一些js特效，我们可以通过四种方式去实现：

1. html页面的头部head标签中去引用



1. html页面的主体body标签内去引用



1. 元素事件中去引用



4.直接引用js文件

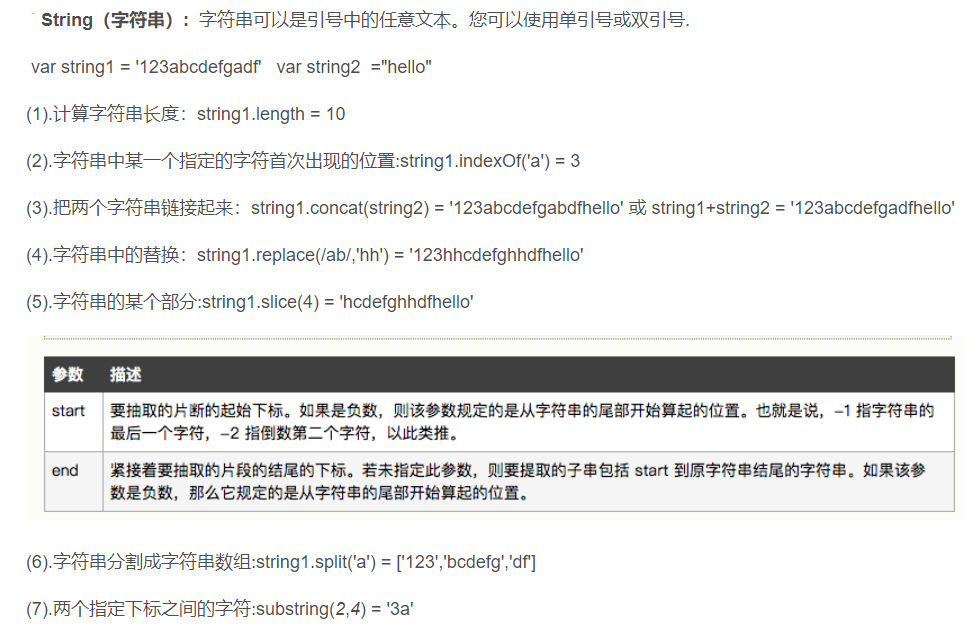


1. JS中常见的几种数据类型

五种基本数据类型（Number,String,Boolean,Undefined,Null）,和一种复杂数据类型（Object）



Boolean类型有两个字面值：true和false。true不一定等于1，false不一定等于0。只有Number（true）Number(false)才是1或0，注意的是true和false是区分大小写的。





数值类型（Number）

1.一切数字都是数值类型(包括:十进制,二进制,十六进制,八进制等)

2.NAN（not a number），一个非数字。常用于判断是否是数字。任何涉及NaN的操作都会返回NaN，NaN与任何值都不相等

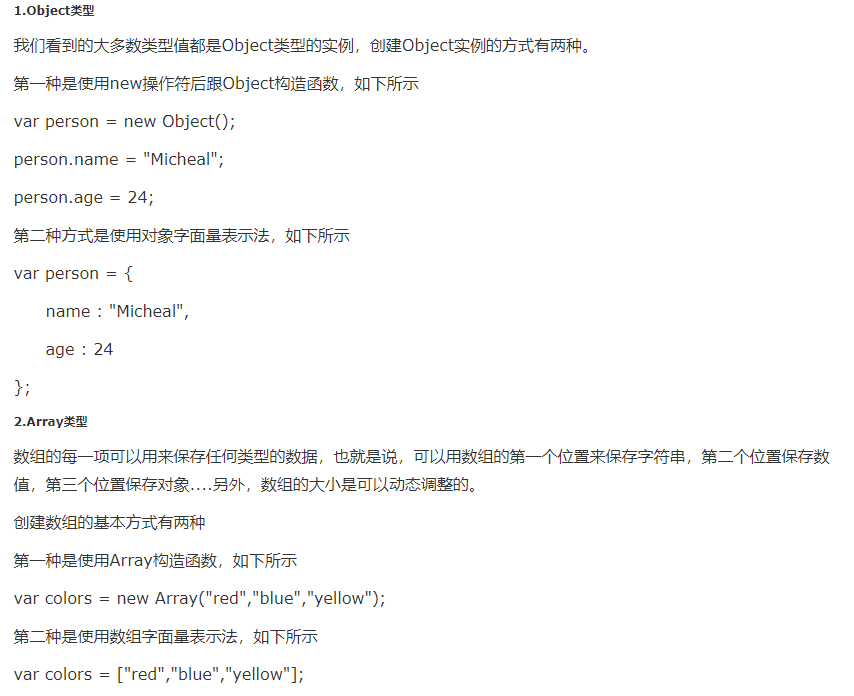
Undefined类型

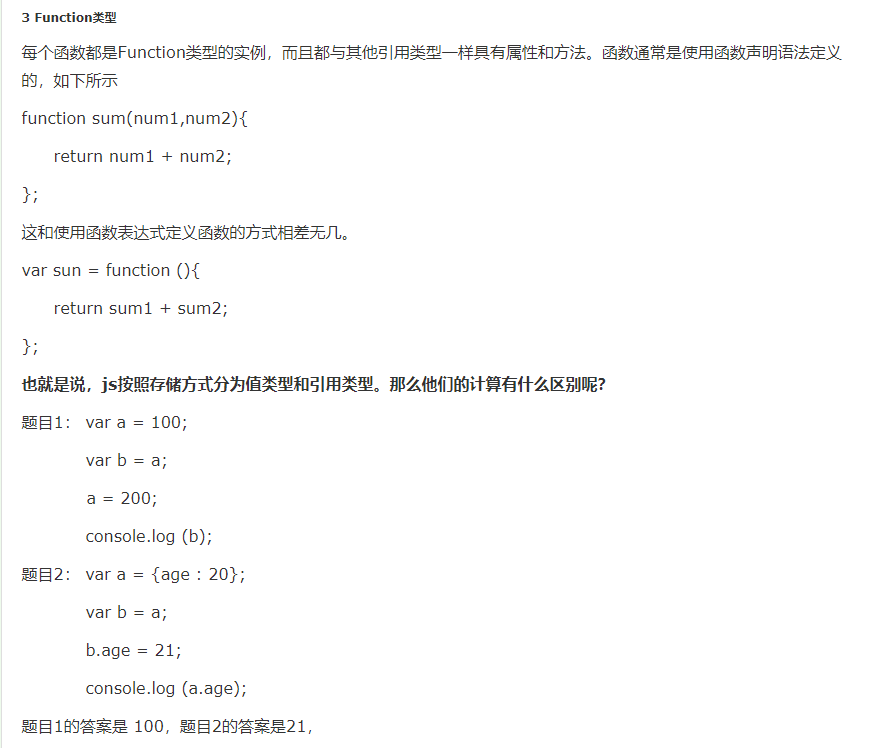
对未初始化和未声明的变量执行typeof操作符都会返回undefined

Null类型：null值表示一个空指针对象

注：没有必要把一个变量的值初始化为undefined，但对于意在保存对象的变量则应明确的初始化为null值

三大引用类型：





RGBA() 与 opacity 在透明效果上有什么区别

rgba：

　　　　使用方式：rgba(255, 255, 255, .5) 最后一个参数表示透明度取值范围 0 ~1

　　　   只作用于元素的颜色或其背景色。

opacity :

　　　　使用方式：opacity : .5 ;  参数表示透明度取值范围 0 ~1

　　　　作用于元素及元素内的所有内容。（即可以被其子元素继承）

总体来说：

1.opacity 是属性，rgba()是函数，计算之后是个属性值；

2.opacity 作用于元素和元素的内容，内容会继承元素的透明度，取值0-1；

3.rgba() 一般作为背景色 background-color 或者颜色 color 的属性值，透明度由其中的 alpha 值生效，取值0-1；

DISPLAY 和VISIBILITY有什么区别?

1.主要在于visibility:hidden占据空间，无法点击，另一个display:none;不占据空间，无法点击。有一句话是这么说的，display：none会真的把你弄没，而visibility：hidden则仅仅是为你披上一件隐身衣而已，你还在原来的地方呆着。设置透明值(opcity)，占据空间，可以点击。 该方法并非真正意义上的隐藏，只是将元素设为透明，但是只适用于文字类的元素，不适用于input，因为即使看不到，可以点击相应input。

2.就是displayde回流与渲染visibility没有这个影响前端性能的问题。回流则是重新使该页面渲染一次，从而影响性能。

3.株连性,所谓“株连性”，就是如果祖先元素遭遇某祸害，则其子子孙孙无一例外也要遭殃。display:none就是“株连性”明显的声明：一旦父节点元素应用了display:none，父节点及其子孙节点元素全部不可见，而且无论其子孙元素如何不屈地挣扎都无济于事.我们给一个父元素应用visibility:hidden，则其子孙后代也都会全部不可见。如果子孙元素应用了visibility:visible，那么这个子孙元素又会刘谦般地显现出来。

js基础 && CSS面试题 - 2020.11.12

js基础

了解JS中由+ , == 导致的数据类型转换规则（Javascript高级程序设计 && Javascript权威指南中有讲）

css面试题

css哪些属性是可继承的

**+ ：**

可用于将变量转换为数字：

var y = "5";      // y 是一个字符串  
var x = + y;      // x 是一个数字

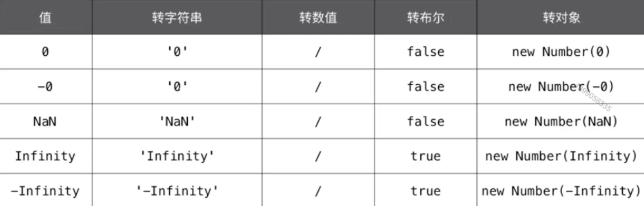
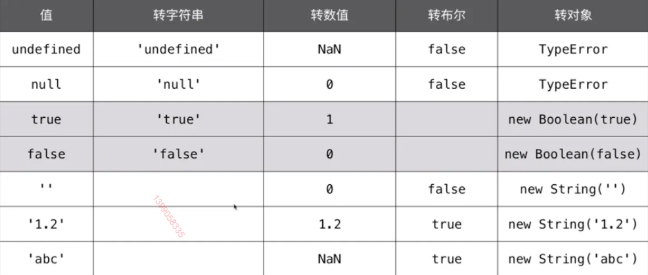
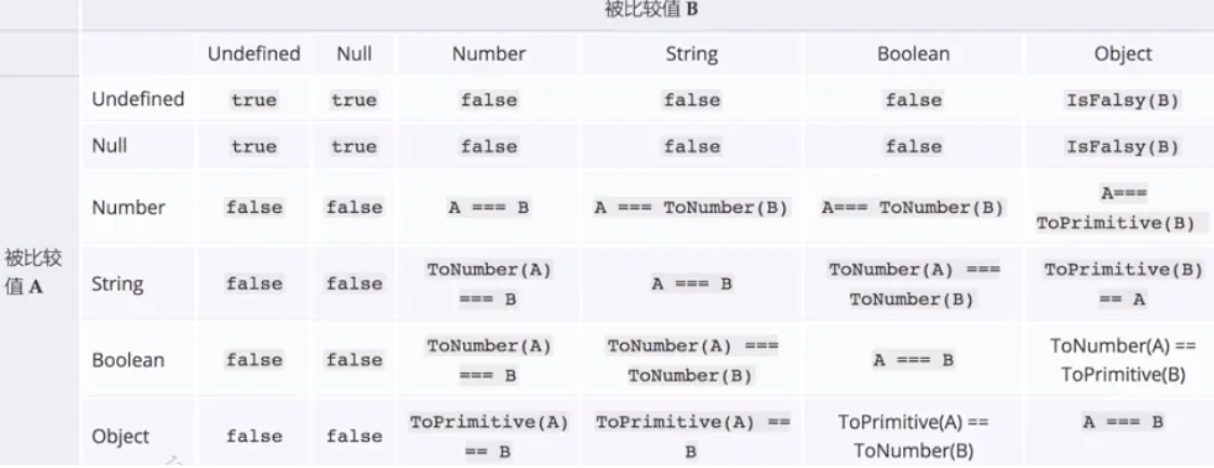
如果变量不能转换，它仍然会是一个数字，但值为 NaN (不是一个数字):

var y = "John";   // y 是一个字符串  
var x = + y;      // x 是一个数字 (NaN)

num  +  ""，当 + 两边一个操作符是字符串类型，一个操作符是其它类型的时候，会先把其它类型转换成字符串再进行字符串拼接，返回字符串

==：

当我们使用**==**进行比较时，如果两边的类型不同，JS会进行类型转换，然后再比较，**===**则不会进行类型转换，如果**===**两边的数据类型不同，直接返回**false**。



css哪些属性是可继承的

css中有继承性的属性之字体属性中的：  
font：字体。font-family：规定元素的字体。font-weight：设置字体的粗细。  
font-size：设置字体的尺寸。font-style：定义字体的风格。  
font-variant：设置小型大写字母的字体显示文本，这意味着所有的小写字母均会被转换为大写，但是所有使用小型大写字体的字母与其余文本相比，其字体尺寸更小。  
font-stretch：对当前的 font-family 进行伸缩变形。所有主流浏览器都不支持。  
font-size-adjust：为某个元素规定一个 aspect 值，这样就可以保持首选字体的 x-height。  
2、css中有继承性的属性之文本属性中的：  
text-indent：文本缩进。text-align：文本水平对齐。line-height：行高。word-spacing：增加或减少单词间的空白（即字间隔）。  
letter-spacing：增加或减少字符间的空白（字符间距）。text-transform：控制文本大小写。direction：规定文本的书写方向。  
color：文本颜色  
3、css中有继承性的属性之元素可见性：  
visibility：规定元素是否可见。  
4、css中有继承性的属性之表格布局属性：  
caption-side：设置表格标题的位置。border-collapse：设置是否将表格边框折叠为单一边框。  
border-spacing：设置分隔单元格边框的距离。empty-cells：设置是否显示表格中的空单元格。  
table-layout：设置显示单元、行和列的算法。  
5、css中有继承性的属性之列表布局属性：  
list-style-type：修改用于列表项的标志类型。list-style-image：对各标志使用一个图像。  
list-style-position：可以确定标志出现在列表项内容之外还是内容内部。  
list-style：简写列表样式，用于把所有用于列表的属性设置于一个声明中。  
6、css中有继承性的属性之生成内容属性：  
quotes：设置嵌套引用的引号类型。  
7、css中有继承性的属性之光标属性：  
cursor：规定要显示的光标的类型（形状）  
8、css中有继承性的属性之页面样式属性：  
page：规定元素应该被显示的页面特定类型。page-break-inside：设置在表格元素内部避免进行分页的分页行为。  
windows：设置当元素内部发生分页时必须在页面顶部保留的最少行数。  
orphans：设置当元素内部发生分页时必须在页面底部保留的最少行数。  
9、css中有继承性的属性之声音样式属性  
speak：设置或检索声音是否给出。speak-punctuation：设置或检索标点符号如何发音。  
speak-numeral：设置或检索数字如何发音。speak-header：设置或检索表格头与其后的一系列单元格发生多少次关系。  
speech-rate：设置或检索发音速度。volume：设置或检索音量。  
voice-family：设置或检索当前声音类型。pitch：设置或检索音高。  
pitch-range：设置或检索声音的平滑程度。stress：和pitch-range相似。设置或检索当前声音波形的最高峰值。  
richness：设置或检索当前声音的音色。azimuth：设置或检索当前声音的音场角度。  
elevation：设置或检索当前声音的音源仰角。  
10、所有元素可以继承的属性：visibility、cursor  
11、内联元素可以继承的属性：  
（1）字体系列属性  
（2）除text-indent、text-align之外的文本系列属性  
12、块级元素可以继承的属性：text-indent、text-align

JS基础 && CSS面试题 - 2020.11.13

JS基础

了解JS中的对象： 如何创建对象，如何获取对象中的属性（. 和 [] 获取方式有什么不同），如何修改&&增加对象的属性，拷贝对象时拷贝的是整个对象还是对象的引用， 如何判断对象中是否存在某一个属性，如何查看对象中存在的所有属性，如何对对象中的所有属性做一个遍历，如何删除对象中的属性

css面试题

元素居中方案（包括行内、块级元素，定宽、不定宽元素，水平、垂直居中），方案尽可能多

如何创建对象

**1、new 操作符 + Object 创建对象**

**2、字面式创建对象**

以上两种方法在使用同一接口创建多个对象时，会产生大量重复代码，为了解决此问题，工厂模式被开发。

**3、工厂模式：**工厂模式解决了重复实例化多个对象的问题，但没有解决对象识别的问题（但是工厂模式却无从识别对象的类型，因为全部都是Object，不像Date、Array等，本例中，得到的都是o对象，对象的类型都是Object，因此出现了构造函数模式）。

**4、构造函数模式**

对比工厂模式有以下不同之处：

1、没有显式地创建对象

2、直接将属性和方法赋给了 this 对象

3、没有 return 语句

以此方法调用构造函数步骤：

1、创建一个新对象

2、将构造函数的作用域赋给新对象（将this指向这个新对象）

3、执行构造函数代码（为这个新对象添加属性）

4、返回新对象 ( 指针赋给变量person ？？？ )

可以看出，构造函数知道自己从哪里来（通过 instanceof 可以看出其既是Object的实例，又是Person的实例）

构造函数也有其缺陷，每个实例都包含不同的Function实例（ 构造函数内的方法在做同一件事，但是实例化后却产生了不同的对象，方法是函数 ，函数也是对象）

**5、原型模式：**原型模式的好处是所有对象实例共享它的属性和方法（即所谓的共有属性），此外还可以设置实例自己的属性（方法）（即所谓的私有属性），可以覆盖原型对象上的同名属性（方法）。

**6、混合模式（**构造函数模式+原型模式）构造函数模式用于定义实例属性，原型模式用于定义方法和共享的属性。混合模式共享着对相同方法的引用，又保证了每个实例有自己的私有属性。最大限度的节省了内存。

如何获取对象中的属性（. 和 [] 获取方式有什么不同）

.运算符:  右侧必须是一个属性名称命名的简单标识符

[] :  右侧必须是一个计算结果为字符串的表达式

相同点：都可以获取到obj的属性值

不同点：

     1 [ ]运算符可以使用字符串变量的内容作为属性名，点运算符不能

     2 [ ]运算符可以用纯数字作为属性名，点运算符不能

    3 [ ]运算符可以用js关键字和保留字作为属性名，点运算符不能

1、对象内置属性方法:Object.keys()；该方法返回一个数组，数组内包括对象内可枚举属性以及方法名称。

1 var keys= Object.keys(testObj);

2 　　console.log(keys); // 输出 keys ["name", "age", "action"]

2、Object.getOwnPropertyNames():方法返回一个指定对象所有自身属性的属性名（包括不可枚举属性但不包括Symbol值作为名称的属性）组成的数组.（不可枚举: 比如属性是一个js方法）

1 var keys = Object.getOwnPropertyNames(testObj);

2 　　console.log(keys);　　// 输出 keys ["name", "age", "action"]

3、最原始的 for...in 循环。

1 var keys =[];

2 　　for(var i in testObj){

3 　　keys.push(i);

4 　　}

5 　　console.log(keys); // keys ["name", "age", "action"]

拷贝对象时拷贝的是整个对象还是对象的引用

1. 在对象中赋值运算符 = 实现的是浅拷贝，只拷贝对象的引用值
2. 递归是做复杂深拷贝比较合理的方法
3. JSON深拷贝只能是对象中没有function时可以使用
4. 数组的深拷贝方法较多，但是大多是只能进行第一层的深拷贝

如何判断对象中是否存在某一个属性

使用in关键字

该方法可以判断对象的自有属性和继承来的属性是否存在。

使用对象的hasOwnProperty()方法

该方法只能判断自有属性是否存在，对于继承属性会返回false。

使用undefined判断

自有属性和继承属性均可判断。

在条件语句中判断

如何删除对象中的属性

delete 是删除对象属性唯一真正的方法。已声明的对象不可删除, 对象中的对象属性可以删除，已声明的变量windows下的变量可以删除, 未声明的变量不可删除

css元素居中方案

