

CSS面试题合集

web标准及语义化

对WEB标准以及W3C的理解与认识？

请谈一下你对网页标准和标准制定机构重要性的理解？

什么是语义化的HTML?有何意义？为什么要做到语义化？日常工作中怎样遵循标签语义化？

你如何理解HTML结构的语义化？

对WEB标准以及W3C的理解与认识？

HTML4/5标签

HTML与XHTML 二者有什么区别

img的alt与title有何异同？strong与em的异同？

b标签和strong标签,i标签和em标签的区别？

举例并简述块级元素与行内元素的区别？/ 块级和行内元素有哪些，他们有什么区别？/ 行内元素的padding...

我想让行内元素跟上面的元素距离10px，加margin-top和padding-top可以吗？

每个HTML文件里开头都有个很重要的东西，Doctype知道这是干什么的吗？你知道多少种文档类型

严格模式与混杂模式的区别？严格模式与混杂模式-如何触发这两种模式，区分它们有何意义？

Quirks模式是什么？它和Standards模式有什么区别？/ 浏览器标准模式和怪异模式之间的区别是什么？

前端页面有哪三层构成，分别是什么？作用是什么？

HTML5常用新增标签以及API / 列举三项html5有那些新特征 / HTML5中新增的属性有哪些 / html5 你用过...

CSS2/3

为什么要初始化样式？/ 描述一个“reset”的CSS文件并如何使用它。知道normalize.css吗？你了解他们的不...

元素在不同的浏览器中默认样式不尽相同，通常如何解决？

使用 CSS 预处理器吗？/ Sass、LESS是什么？大家为什么要使用他们？

CSS中link和@import的区别是？

使用CSS,让一个div消失在视野中；发挥想象力？/ CSS中可以通过哪些属性定义，使得一个DOM元素不显示...

至少写出3种css3新增的属性？

CSS3有哪些新特性？/ CSS3常用新增方法？Canvas和svg的区别？

超链接访问过后hover样式就不出现的问题是什么？如何解决？

rgba()和opacity的透明效果有什么不同？

说display属性有哪些？可以做什么？

display:none与visibility:hidden的区别是什么？

视差滚动效果？

你对line-height是如何理解的？

怎么让Chrome支持小于12px 的文字？

让页面里的字体变清晰，变细用CSS怎么做？

li与li之间有看不见的空白间隔是什么原因引起的？有什么解决办法？

display:inline-block 什么时候会显示间隙？

style标签写在body后与body前有什么区别？

CSS属性overflow属性定义溢出元素内容区的内容会如何处理？

盒子模型

简述盒子模型？ CSS的盒模型由什么组成？ / css盒模型 / css有哪些盒子模型？ 它们对应的样式是什么？ 如...

如何用css绘制一个三角(不允许使用图片)？

一个满屏品字布局如何设计？

什么是外边距重叠？ 重叠的结果是什么？ 上下margin重合的问题

margin和padding分别适合什么场景使用？

元素竖向的百分比设定是相对于容器的高度吗？

浮动

请罗列常见的清除浮动方案 / CSS清除浮动的方法（至少两种） / HTML中浮动塌陷问题如何解决 / 清除浮...

BFC是什么？ / 对BFC规范的理解？

设置元素浮动后，该元素的display值是多少？

定位

CSS中那个属性用来定位元素；

position跟display、margin collapse、overflow、float这些特性相互叠加后会怎么样？

Position和display的取值和各自的意思和用法

position的值，relative，absolute，fixed分别是相对于谁进行定位的(定位原点是)？

请说明z-index的工作原理，适用范围？

absolute的containing block计算方式跟正常流有什么不同？

有一个高度自适应的div，里面有两个div，一个高度100px，希望另一个填满剩下的高度

背景

解释CSS sprites(雪碧图)如何使用？

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？

动画

请用CSS写一个简单的幻灯片效果页面？

列举三项transform的效果？

CSS中的animation动画如何使用 / transition animation 什么区别？

如果需要手动写动画，你认为最小时间间隔是多久，为什么？

全屏滚动的原理是什么？用到了CSS的哪些属性？

性能优化

知道的网页制作会用到的图片格式有哪些？

简述一下src与href的区别？知道什么是微格式吗？谈谈理解。在前端构建中应该考虑微格式吗？

一个页面上有大量的图片（大型电商网站），加载很慢，你有哪些方法优化这些图片的加载，给用户更好的...

谈谈以前端角度出发做好SEO需要考虑什么？CSS优化、提高性能的方法有哪些？

居中

块元素（div）水平垂直居中，文字内容在div中始终水平垂直居中 / 请列出你所知道的使用css让元素上下居...

CSS中可以让文字在垂直和水平方向上居中的两个属性是什么？ / text-align:center和line-height有什么区...

单位

CSS中表示长度的单位有哪些； px和em的区别？ / px,em,rem的区别？

布局

div+css的布局较table布局有什么优点？

实现页面的三栏布局，左右均为定宽，高度可以自适应；（双飞翼布局、圣杯布局）

响应式网页主要原理？

了解过flex吗？

请解释一下CSS3的flexbox（弹性盒布局模型），以及适用场景？

移动端的布局用过媒体查询吗？ / 如何适配不同大小的手机屏幕；

什么是响应式设计？响应式设计的基本原理是什么？如何兼容低版本的IE？

选择器及优先级

请描述css选择器的优先级原则？ / 有哪些选择符，优先级的计算公式是什么？行内样式和! important哪个...

写一个表格以及对应的CSS,使表格奇数行为白色背景，偶数行为灰色，鼠标一上去为黄色背景。

样式的层级关系，选择器优先级，样式冲突，以及抽离样式模块怎么写，说出思路，有无实践经验

CSS选择符有哪些？哪些属性可以继承？优先级算法如何计算？css3新增伪类有哪些？

浏览器和兼容

你做的页面在哪些浏览器测试过？这些浏览器的内核分别是什么？

IE的双边距BUG：块级元素float后设置横向margin，ie6显示的margin比设置的较大？

如何处理浏览器兼容性？什么是优雅降级开发模式？ / 你能描述一下渐进增强和优雅降级之间的不同吗？

web标准及语义化

对WEB标准以及W3C的理解与认识？

web标准简单来说可以分为结构、表现和行为。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

W3C对web标准提出了规范化的要求，也就是在实际编程中的一些代码规范：包含如下几点

对于结构要求：（标签规范可以提高搜索引擎对页面的抓取效率，对SEO很有帮助）

- 标签字母要小写
- 标签要闭合
- 标签不允许随意嵌套

对于css和js来说

- 尽量使用外链css样式表和js脚本。是结构、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。
- 样式尽量少用行间样式表，使结构与表现分离，标签的id和class等属性命名要做到见文知义，标签越少，加载越快，用户体验提高，代码维护简单，便于改版
- 不需要变动页面内容，便可提供打印版本而不需要复制内容，提高网站易用性。

请谈一下你对网页标准和标准制定机构重要性的理解？

网页标准和标准制定机构都是为了让web发展的更‘健康’

首先约束浏览器开发者遵循统一的标准，其次约束网站开发者，这样降低开发难度，开发成本，SEO也会更好做，也不会因为滥用代码导致各种BUG、安全问题，最终提高网站易用性。

什么是语义化的HTML？有何意义？为什么要做到语义化？日常工作中怎样遵循标签语义化？

什么是语义化的HTML？

语义化的HTML就是写出的HTML代码，符合内容的结构化（内容语义化），选择合适的标签（代码语义化），也就是说用正确的标签做正确的事情，能够便于开发者阅读和写出更优雅的代码的同时让网络爬虫很好地解析。

为什么要做到语义化？

- 有利于SEO，有利于搜索引擎爬虫更好的理解我们的网页，从而获取更多的有效信息，提升网页的权重。
- 在没有CSS的时候能够清晰的看出网页的结构，增强可读性。
- 便于团队开发和维护，语义化的HTML可以让开发者更容易的看明白，从而提高团队的效率和协调能力。

- 支持多终端设备的浏览器渲染。

语义化的HTML该怎么做？

在做前端开发的时候要记住：HTML 告诉我们一块内容是什么（或其意义），而不是它长的什么样子，它的样子应该由CSS来决定。（结构与样式分离！）

写语义化的 HTML 结构其实很简单，首先掌握 HTML 中各个标签的语义，在看到内容的时候想想用什么标签能更好的描述它，是什么就用什么标签。

- `<h1>~<h6>`，作为标题使用，并且依据重要性递减，`<h1>` 是最高的等级。
- `<p>` 段落标记，知道了 `<p>` 作为段落，你就不会再使用 `
` 来换行了，而且不需要 `
` 来区分段落与段落。`<p>` 中的文字会自动换行，而且换行的效果优于 `
`。段落与段落之间的空隙也可以利用 CSS 来控制，很容易而且清晰的区分出段落与段落。
- ``、``、``，`` 无序列表，这个被大家广泛的使用，`` 有序列表也挺常用。在 web 标准化过程中，`` 还被更多的用于导航条，本来导航条就是个列表，这样做是完全正确的，而且当你的浏览器不支持 CSS 的时候，导航链接仍然很好使，只是美观方面差了一点而已。
- `<dl>`、`<dt>`、`<dd>`，`<dl>` 就是“定义列表”。比如说词典里面的词的解释、定义就可以用这种列表。
- ``、``，`` 是用作强调，`` 是用作重点强调。
- `<q>` 也不仅仅只是为文字增加双引号，而是代表这些文字是引用来的。
- `<table>`、`<td>`、`<th>`、`<caption>`，(X)HTML中的表格不再是用来布局。

补充：网络爬虫，SEO等概念

SEO：Search Engine Optimization——搜索引擎优化，这是一种利用搜索引擎的搜索规则，采取优化策略或程序，提高网站在搜索结果中的排名。

网络爬虫：又称网络蜘蛛、网络机器人，是一种搜索引擎用于自动抓取网页资源的程序或者说叫机器人。从某一个网址为起点，去访问，然后把网页存回到数据库中，如此不断循环，一般认为搜索引擎爬虫都是靠链接爬行的，所以管他叫爬虫。这个只有开发搜索引擎才会用到。对于网站而言，只要有链接指向我们的网页，爬虫就会自动提取我们的网页。

你如何理解HTML结构的语义化？

HTML的语义化，就是在你写的HTML结构，是用相对应的有一定予以的英文字母来表示标记的。不仅对自己来说，更加容易阅读书写，别人再看你代码和结构的时候也会非常容易理解。因此在[前端](#)开发中，尽量使用官方的有予以的标签。

语义化之后文档会有什么效果呢？

- 去掉或样式丢失的时候能让页面呈现清晰的结构：
- html 本身是没有表现的，我们看到例如`<h1>`是粗体，字体大小 2em，加粗；``是加粗的，不要认为这是 html 的表现，这些其实 html 默认的 css 样式在起作用，所以去掉或样式丢失的时候能让页面呈现清晰的结构不是语义化的 HTML 结构的优点，但是浏览器都有默认样式，默认样式

的目的也是为了更好的表达 html 的语义，可以说浏览器的默认样式和语义化的 HTML 结构是不可分割的。

- 屏幕阅读器（如果访客有视障）会完全根据你的标记来“读”你的网页。例如,如果你使用的含语义的标记,屏幕阅读器就会“逐个拼出”你的单词,而不是试着去对它完整发音。
- PDA、手机等设备可能无法像普通电脑的浏览器一样来渲染网页（通常是因为这些设备对 CSS的支持较弱）。
- 使用语义标记可以确保这些设备以一种有意义的方式来渲染网页.理想情况下,观看设备的任务是符合设备本身的条件来渲染网页。
- 语义标记为设备提供了所需的相关信息,就省去了你自己去考虑所有可能的显示情况（包括现有的或者将来新的设备）。例如,一部手机可以选择使一段标记了标题的文字以粗体显示.而掌上电脑可能会以比较大的字体来显示.无论哪种方式一旦你对文本标记为标题,您就可以确信读取设备将根据其自身的条件来合适地显示页面。

搜索引擎的爬虫也依赖于标记来确定上下文和各个关键字的权重

- 过去你可能还没有考虑搜索引擎的爬虫也是网站的“访客”,但现在它们他们实际上是极其宝贵的用户.没有他们的话,搜索引擎将无法索引你的网站,然后一般用户将很难过来访问。你的页面是否对爬虫容易理解非常重要,因为爬虫很大程度上会忽略用于表现的标记,而只注重语义标记。
- 因此,如果页面文件的标题被标记,否则,那么这个页面在搜索结果的位置可能会比较靠后。除了提升易用性外,语义标记有利于正确使用 CSS 和 JavaScript,因为其本身提供了许多“钩钩”来应用页面的样式与行为。
- 同时SEO 主要靠你网站结构来定义内容和外部链接的。

以上就是对于HTML结构的语义初步理解，也可以说是为什么我要采用HTML的语义化。W3C 给我们制定了很好的标准，如果在团队开发中大家都遵循这个标准，就可以大大减少差异化内容，从而方便开发和维护，提高开发效率，甚至实现模块化开发。

对WEB标准以及W3C的理解与认识？

语义化的HTML就是写出的HTML代码，符合内容的结构化（内容语义化），选择合适的标签（代码语义化），能够便于开发者阅读和写出更优雅的代码的同时让浏览器的爬虫和机器很好地解析。

标签语义化

- 有利于SEO，有利于搜索引擎爬虫更好的理解我们的网页，从而获取更多的有效信息，提升网页的权重。
- 在没有CSS的时候能够清晰的看出网页的结构，增强可读性。
- 便于团队开发和维护，语义化的HTML可以让开发者更容易的看明白，从而提高团队的效率和协调能力。
- 支持多终端设备的浏览器渲染
- 合理的标签做合理的事情

遵循方法

- 尽量减少使用无意义标签，例如span和div;
- 尽量不使用标签本身的css属性，例如b、font、s等标签，如果需要这些样式，那么使用CSS样式来进行添加;
- 在需要强调的部分，使用strong、em，但是样式尽量使用css样式来描述;
- 表格搭建时，使用表格头部 表格身体 表格尾部`;
- 列表搭建时，使用无序列表有序列表 定义列表;

HTML4/5标签

HTML与XHTML 二者有什么区别

HTML与XHTML之间的差别，主要分为功能上的差别和书写习惯的差别两方面。

关于功能上的差别，主要是XHTML可兼容各大浏览器、手机以及PDA，并且浏览器也能快速正确地编译网页。

由于XHTML的语法较为严谨，所以如果你是习惯松散结构的HTML编写者，那需要注意XHTML的规则。

- 所有标签都必须小写：在XHTML中，所有的标签都必须小写，不能大小写穿插其中，也不能全部都是大写。
- 标签必须成双成对：像是<p>...</p>、<a>...、<div>...</div>标签等，当出现一个标签时，必须要有对应的结束标签，缺一不可，就像在任何程序语言中的括号一样
- 标签顺序必须正确：标签由外到内，一层层包覆着，所以假设你先写div后写h1，结尾就要先写h1后写div。只要记住一个原则“先进后出”，先弹出的标签要后结尾。
- 所有属性都必须使用双引号：在XHTML 1.0中规定连单引号也不能使用，所以全程都得用双引号。
- 不允许使用target="_blank"：从XHTML 1.1开始全面禁止target属性，如果想要有开新窗口的功能，就必须改写为rel="external"，并搭配JavaScript实现此效果。

最主要的不同：

- XHTML 元素必须被正确地嵌套。
- XHTML 元素必须被关闭。
- 标签名必须用小写字母。
- XHTML 文档必须拥有根元素。

img的alt与title有何异同？ strong与em的异同？

img的alt与title有何异同

- alt是在图片不能正常加载时候显示的提示语
- title属性是鼠标划上去显示的内容

strong与em的异同

- em 表示强调， strong 表示更加强烈的强调。 em 显示为斜体， strong 显示为黑体。
- em表示局部的强调， strong表示全局的强调。举个例子，当使用em标签时，阅读一篇文章读到某处才会注意到这里使用了斜体，才会看到所要强调的部分；当使用strong标签时，看到一篇文章立马就能注意到作者想要强调的部分。
- 使用em标签会引起语义上的变化，而strong则不会

b标签和strong标签,i标签和em标签的区别？

- b、i属于修饰类标签；strong、em属于内容类标签（strong和em是逻辑标签，b和i是物理标签）；
- b、strong标签表现为加粗样式；em、i表现为倾斜样式；
- strong、em表强调；strong比em语气更强烈；strong在html是页面上的强调，而em是句子中的强调；（物理逻辑就是说告诉浏览器我哪里加粗了，没有别的作用，而逻辑标签是强调语气，它强调语气是通过文本加粗来体现的，也可以通过别的样式来强调语气）；
- strong和em真正做到了结构与样式分离，而b、i没有做到结构与样式的分离；
- 在搜索引擎优化strong和em比b和i重要的多。

举例并简述块级元素与行内元素的区别？ / 块级和行内元素有哪些，他们有什么区别？ / 行内元素的padding和margin可设置吗？

块级元素

含义：块级元素是指本身属性为display:block;的元素。因为它自身的特点，我们通常使用块级元素来进行大布局（大结构）的搭建

- 独占一行，每一个块级元素都会从新的一行重新开始，从上到下排布；
- 可以直接控制宽度、高度以及盒子模型的相关css属性(width/height/border/margin/padding)；
- 在不设置宽度的情况下，块级元素的宽度是它父级元素内容的宽度；
- 在不设置高度的情况下，块级元素的高度是它本身内容的高度；
- 可以嵌套其他元素；
- ul li、ol li 、dl dt dd 成组出现，p标签不能嵌套它自己本身；

标签	作用
div	常用块级容器，也是css layout的主要标签
h1	大标题
h2	副标题
h3	三级标题
h4	四级标题
h5	五级标题
h6	六级标题
hr	水平分隔线

menu	菜单列表
ol	有序列表
ul	无序列表
li	列表项
dl	定义列表
dt	定义标题
dd	定义描述
table	表格
p	段落
form	交互表单

行内元素

行内(内联)元素是指本身属性为display:inline;的元素。因为它自身的特点，我们通常使用块级元素来进行文字、小图标（小结构）的搭建；

- 和其他内联元素从左到右在一行显示,从左到右，达到父级元素的最大宽度时，会自动折行；
- 不能直接控制宽度、高度以及盒子模型的相关css属性，可以设置margin-left (padding)、margin-right(padding),对上下无效。
- 内联元素的宽高是由本身内容的大小决定（文字、图片等）
- 内联元素只能容纳文本或者其他内联元素（此处请注意，不要在内联元素中嵌套块级元素）
- 在编辑代码时，元素之间有回车或者换行的时候，元素和元素之间有间隙；

标签	作用
span	常用内联容器，定义文本内区块
a	锚点
b	加粗
strong	加粗强调
i	斜体
em	斜体强调
s	中划线（不推荐使用）
strike	中划线
del	文档中已被删除的文本
br	强制换行

u	下划线
textarea	多行文本输入框
input	输入框
select	下拉列表
label	input 元素定义标注（标记）
img	插入图片
sub	下标
sup	上标
big	大字体文本
small	小字体文本

行内块元素

- 在一行排布，从左到右，达到父级元素的最大宽度时，会自动折行；
- 设置盒子模的CSS属性；
- 默认不设置宽度和高度的时候，宽高都是由本身内容决定的；
- 在编辑代码时，元素之间有回车或者换行的时候，元素和元素之间有间隙；

标签	作用
textarea	文本域
input	输入框
img	图片

我想让行内元素跟上面的元素距离10px，加margin-top和padding-top可以吗？

不可以；内联元素可以设置margin-left (padding)、margin-right(padding),对上下无效。

每个HTML文件里开头都有个很重要的东西，Doctype知道这是干什么的吗？你知道多少种文档类型

- DOCTYPE是Document Type（文档类型）的简写，在页面中，用来指定页面所使用的XHTML（或者HTML）的版本。要想制作符合标准的页面，一个必不可少的关键组成部分就是DOCTYPE声明。只有确定了一个正确的DOCTYPE，XHTML里的标识和CSS才能正常生效。
- Doctype描述了html文档的类型，对不同的Doctype类型，浏览器会使用不同的方法来解析该标签告知浏览器文档所使用的HTML规范。告诉浏览器按照何种规范解析页面；

该标签可声明三种 DTD 类型，分别表示严格版本、过渡版本以及基于框架的 HTML 文档。

- HTML 4.01 规定了三种文档类型：Strict、Transitional 以及 Frameset。
- XHTML 1.0 规定了三种 XML 文档类型：Strict、Transitional 以及 Frameset。
- Standards （标准）模式（也就是严格呈现模式）用于呈现遵循最新标准的网页，而Quirks（包容）模式（也就是松散呈现模式或者兼容模式）用于呈现为传统浏览器而设计的网页。

严格模式与混杂模式的区别？严格模式与混杂模式–如何触发这两种模式，区分它们有何意义？

- Doctype声明于文档最前面，告诉浏览器以何种方式来渲染页面，这里有两种模式，严格模式和混杂模式。
- 严格模式的排版和 JS 运作模式是 以该浏览器支持的最高标准运行。
- 混杂模式，向后兼容，模拟老式浏览器，防止浏览器无法兼容页面。

Quirks模式是什么？它和Standards模式有什么区别？ / 浏览器标准模式和怪异模式之间的区别是什么？

W3C标准中，如果设置一个元素的宽度和高度，指的是元素内容的宽度和高度，而在Quirks（怪异盒子模型）模式下，IE的宽度和高度还包含了padding和border。

设置行内元素的高宽：在Standards模式下，给行内元素设置width和height都不会生效，而在quirks模式下，则会生效。

标准模式下的盒模型

- 元素的实际宽度 = width + padding-left / right + border -left/right
- 元素的实际高度 = height + padding -top/bottom + border-top/bottom

IE盒模型计算公式

- 元素的实际宽度 = width
- 元素的实际高度 = height

前端页面有哪三层构成，分别是什么？作用是什么？

最准确的网页设计思路是把网页分成三个层次，即：结构层、表示层、行为层。

网页的结构层（structural layer）由 HTML 或 XHTML 之类的标记语言负责创建。标签，也就是那些出现在尖括号里的单词，对网页内容的语义含义做出了描述，但这些标签不包含任何关于如何显示有关内容的信息。例如，P 标签表达了这样一种语义：“这是一个文本段。”

网页的表示层（presentation layer）由 CSS 负责创建。CSS 对“如何显示有关内容”的问题做出了回答。

网页的行为层（behavior layer）负责回答“内容应该如何对事件做出反应”这一问题。这是 Javascript 语言和 DOM 主宰的领域。

HTML5常用新增标签以及API / 列举三项html5有那些新特征 / HTML5中新增的属性有哪些 / html5 你用过哪些？

<https://juejin.im/entry/595e43ce6fb9a06bc23a99ef>

CSS2/3

为什么要初始化样式？ / 描述一个“reset”的CSS文件并如何使用它。知道normalize.css吗？你了解他们的不同之处？

因为浏览器的兼容问题，不同浏览器对有些标签的默认值是不同的，如果没对CSS初始化往往会出现浏览器之间的页面显示差异。

Reset样式的目的就是清除某些浏览器的默认样式,方便css的书写：例如：`*{margin:0;padding:0;list-style:none;}`

normalize的理念与reset的不同，他并不是清除浏览器的默认样式，而是尽量将所有的浏览器的默认样式统一；

元素在不同的浏览器中默认样式不尽相同，通常如何解决？

CSS初始化是指重设浏览器的样式。由于浏览器的兼容的问题，不同的浏览器默认的样式可能不尽相同，所以开发时的第一件事可能就是如何把它们统一。如果没对CSS初始化往往会出现浏览器之间的页面差异。每次新开发网站或新网页时候通过初始化CSS样式的属性，为我们将用到的CSS或html标签更加方便准确，使得我们开发网页内容时更加方便简洁，同时减少CSS代码量，节约网页下载时间。

弊端：初始化样式会对SEO有一定的影响，但鱼和熊掌不可兼得，但力求影响最小的情况下初始化。

使用 CSS 预处理器吗？ / Sass、LESS是什么？大家为什么要使用他们？

SASS是一种CSS开发工具，是一种动态样式语言，Sass语法属于缩排语法，比css比多出好些功能(如变量、嵌套、运算,混入(Mixin)、继承、颜色处理，函数等)，更容易阅读。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，后缀名.scss

Less是一门CSS预处理语言，也是一种动态样式语言. 对CSS赋予了动态语言的特性，如变量，继承，运算， 函数. Less 既可以在客户端上运行（支持IE 6+, Webkit, Firefox），也可在服务端运行（借助Node.js）。它扩展了CSS语言，增加了变量，Mixin，函数等特性，使CSS更易维护和扩展。Less可以运行在Node或者浏览器端。后缀名.less

为什么使用它们？

- 结构清晰，便于扩展。
- 可以方便地屏蔽浏览器私有语法差异。这个不用多说，封装对浏览器语法差异的重复处理，减少无意义的机械劳动。
- 可以轻松实现多重继承。
- 完全兼容 CSS 代码，可以方便地应用到老项目中。LESS 只是在 CSS 语法上做了扩展，所以老的CSS 代码也可以与 LESS 代码一同编译。

CSS中link和@import的区别是？

- link和@import虽然都是引入外部的css文件，但是他们是有天差地别的区别的；
- link是html标签，@import完全是css提供的方-式，要写在css文件或者style标签中；
- 他们的加载顺序也是有区别的，当一个页面被加载的时候，link引用的css文件会被同时加载，而@import引入的css文件会等页面全部下载完后再加载；
- 当使用javascript控制DOM去改变css样式的时候，只能使用link标签，因为@import是不能被DOM控制的；

使用CSS,让一个div消失在视野中；发挥想象力？ / CSS中可以通过哪些属性定义，使得一个DOM元素不显示在浏览器可视范围内？

- display: none; 隐藏不占位
- visibility: hidden; 隐藏占位
- transform: translate(-9999px);
- margin-top: -9999px;
- position: absolute; top: -9999px;
- opacity: 0;
- width: 0; overflow: hidden;
- height: 0; overflow: hidden;
- background-color: transparent; 透明
- transform: scale(0); 缩放
- clip-path: polygon(100% 100%); 裁剪

至少写出3种css3新增的属性？

- border-radius: 圆角
- box-shadow: 阴影
- border-image: 边框图片
- text-shadow: 文字阴影
- transform
- translate 位移
- rotate 旋转
- scale 缩放
- skew 转动时给定的角度
- rotateX/rotateY/rotateZ: 3D旋转属性
- animation 动画
- box-sizing 父级宽高不变，内容随便折腾

CSS3有哪些新特性？ / CSS3常用新增方法？ Canvas和svg的区别？

- RGBA和透明度
- background-image background-origin(content-box/padding-box/border-box)
background-size background-repeat

- word-wrap (对长的不可分割单词换行) word-wrap: break-word
- 文字阴影: text-shadow: 5px 5px 5px #FF0000; (水平阴影, 垂直阴影, 模糊距离, 阴影颜色)
- font-face属性: 定义自己的字体
- 圆角 (边框半径): border-radius 属性用于创建圆角
- 边框图片: border-image: url(border.png) 30 30 round
- 盒阴影: box-shadow: 10px 10px 5px #888888
- 媒体查询: 定义两套css, 当浏览器的尺寸变化时会采用不同的属性

canvas和svg的区别

- Canvas 是基于像素的即时模式图形系统, 最适合较小的表面或较大数量的对象, Canvas不支持鼠标键盘等事件。
- SVG 是基于形状的保留模式图形系统, 更加适合较大的表面或较小数量的对象。Canvas和SVG在修改方式上还存在着不同。绘制Canvas对象后, 不能使用脚本和 CSS 对它进行修改。因为 SVG 对象是文档对象模型的一部分, 所以可以随时使用脚本和 CSS 修改它们。

超链接访问过后hover样式就不出现的问题是什么? 如何解决?

被点击访问过的超链接样式不在具有hover和active了,解决方法是改变CSS属性的排列顺序: L-V-H-A (link,visited,hover,active)

rgba()和opacity的透明效果有什么不同?

- opacity 是属性, rgba()是函数, 计算之后是个属性值;
- 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;
- 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

说display属性有哪些? 可以做什么?

inline (默认) --内联
 none--隐藏
 block--块显示
 table--表格显示
 list-item--项目列表
 inline-block

display:none与visibility:hidden的区别是什么?

display: none 不显示对应的元素, 在文档布局中不再分配空间 (回流+重绘)
 visibility: hidden 隐藏对应元素, 在文档布局中仍保留原来的空间 (重绘)

CSS里的visibility属性有个collapse属性值? 在不同浏览器下以后什么区别?

当一个元素的visibility属性被设置成collapse值后, 对于一般的元素, 它的表现跟hidden是一样的。

1. chrome中，使用collapse值和使用hidden没有区别。

firefox，opera和IE，使用collapse值和使用display: none没有什么区别。

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。

视差滚动效果？

视差滚动（Parallax Scrolling）通过在网页向下滚动的时候，控制背景的移动速度比前景的移动速度慢来创建出令人惊叹的3D效果。

- CSS3实现； 优点：开发时间短、性能和开发效率比较好，缺点是不能兼容到低版本的浏览器
- jQuery实现；通过控制不同层滚动速度，计算每一层的时间，控制滚动效果。
优点：能兼容到各个版本的，效果可控性好
缺点：开发起来对制作者要求高
- 插件实现方式；例如：parallax-scrolling，兼容性十分好

你对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。

怎么让Chrome支持小于12px 的文字？

```
p{font-size:10px;-webkit-transform:scale(0.8);} //0.8是缩放比例
```

让页面里的字体变清晰，变细用CSS怎么做？

-webkit-font-smoothing在window系统下没有起作用，但是在IOS设备上起作用-webkit-font-smoothing: antialiased是最佳的，灰度平滑。

li与li之间有看不见的空白间隔是什么原因引起的？有什么解决办法？

行框的排列会受到中间空白（回车空格）等的影响，因为空格也属于字符,这些空白也会被应用样式，占据空间，所以会有间隔，把字符大小设为0，就没有空格了。

解决方法：

1. 可以将代码全部写在一排
2. 浮动li中float: left
3. 在ul中用font-size: 0（谷歌不支持）；可以使用letter-space: -3px

display:inline-block 什么时候会显示间隙？

- 有空格时候会有间隙 解决：移除空格
- margin正值的时候 解决：margin使用负值
- 使用font-size时候 解决：font-size:0、letter-spacing、word-spacing

style标签写在body后与body前有什么区别？

页面加载自上而下 当然是先加载样式。

写在body标签后由于浏览器以逐行方式对HTML文档进行解析，当解析到写在尾部的样式表（外联或写在style标签）会导致浏览器停止之前的渲染，等待加载且解析样式表完成之后重新渲染，在windows的IE下可能会出现FOUC现象（即样式失效导致的页面闪烁问题）

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。

盒子模型

简述盒子模型？ CSS的盒模型由什么组成？ / css盒模型 / css有哪些盒子模型？它们对应的样式是什么？如何让IE以标准模型渲染？ / 简要手绘盒子模型，并对边框和边距进行标注？ / css中box-sizing的作用

CSS盒子模型：由四个属性组成的外边距(margin)、内边距(padding)、边界(border)、内容区(width和height)；

标准的CSS盒子模型和低端IE CSS盒子模型不同：宽高不一样

- 标准的css盒子模型宽高就是内容区宽高；
- 低端IEcss盒子模型宽高 内边距 + 边界 + 内容区；

box-sizing属性可以为三个值之一：content-box（default），border-box，padding-box。

- content-box，border和padding不计算入width之内
- padding-box，padding计算入width内
- border-box，border和padding计算入width之内，其实就是怪异模式了~

如何用css绘制一个三角(不允许使用图片)？

绘制左右三角形的时候适当调整height:0

border-width 边框线宽度

border-style 边框线的类型 solid 实线 dashed 虚线 dotted 点状线

border-color 颜色 需要什么方向 保留当前方向的颜色 其他方向为透明

```
1 <style type="text/css">
```



```

2 .trangle{
3 width: 0;
4 border-width: 100px 100px 100px 100px;
5 border-style: solid;
6 border-color: transparent transparent red transparent;
7 }
8 </style>
9 <div class="trangle">

```

一个满屏品字布局如何设计？

第一种真正的品字：

1. 三块高宽是确定的；
2. 上面那块用margin: 0 auto;居中；
3. 下面两块用float或者inline-block不换行；
4. 用margin调整位置使他们居中。

第二种全屏的品字布局:上面的div设置成100%，下面的div分别宽50%，然后使用float或者inline使其不换行。

什么是外边距重叠？重叠的结果是什么？上下margin重合的问题

```

1 在重合元素外包裹一层容器，并触发该容器生成一个BFC。
2 例子：
3 <div class="aside"></div>
4 <div class="text">
5     <div class="main"></div>
6 </div>
7 <!--下面是css代码-->
8 .aside {
9     margin-bottom: 100px;
10    width: 100px;
11    height: 150px;
12    background: #f66;
13 }
14 .main {
15     margin-top: 100px;
16     height: 200px;
17     background: #fcc;
18 }
19 .text{
20     /*盒子main的外面包一个div，通过改变此div的属性使两个盒子分属于两个不同的BFC，

```

```
    以此来阻止margin重叠*/
21     overflow: hidden;    //此时已经触发了BFC属性。
22 }
```

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。

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。

浮动

请罗列常见的清除浮动方案 / CSS清除浮动的方法（至少两种） / HTML中浮动塌陷问题如何解决 / 清除浮动,什么时候需要清除浮动,清除浮动都有那些方法？

浮动元素碰到包含它的边框或者浮动元素的边框停留。由于浮动元素不在文档流中，所以文档流的块框表现得就像浮动框不存在一样。浮动元素会漂浮在文档流的块框上。

浮动带来的问题：

1. 父元素的高度无法被撑开，影响与父元素同级的元素
2. 与浮动元素同级的非浮动元素（内联元素）会跟随其后
3. 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。

清除浮动的方式：

1. 父级div定义height
2. 最后一个浮动元素后加空div标签 并添加样式clear:both。
3. 包含浮动元素的父标签添加样式overflow为hidden或auto。
4. 父级div定义zoom

```
1 {display:block;
2  content:"";
3  clear:both;
4  overflow:hidden
5  font-size:0;
6  height:0;
7  visibility:hidden
8 }
9 .clearfix{
10 *zoom=1
11 }
```

BFC是什么？ / 对BFC规范的理解？

- BFC 英文解释： block formatting context。中文意思：块级格式化上下文；
- formatting context 意思是：页面中一个渲染区域，有自己的一套渲染规则，决定其子元素如何定位，以及和其他兄弟元素的关系和作用。

BFC有如下特性

- 内部的box会在垂直方向，从顶部开始一个接一个地放置；
- box垂直方向的距离由margin决定。属于同一个BFC的两个相邻的box的margin会发生折叠；
- 每一个元素的margin box的左遍，与包含块border box的左边相接触。即使浮动元素也是如此；
- BFC区域不会与float box叠加；
- 计算 BFC 的高度时，浮动子元素也参与计算；
- 文字层不会被浮动层覆盖，环绕于周围

产生BFC

- float 除了none以外的值；
- overflow 除了visible 以外的值（hidden, auto, scroll ）；
- display (table-cell, table-caption, inline-block, flex, inline-flex)；
- position值为（absolute, fixed） 这些属性值得元素都会自动创建 BFC；

BFC作用

- BFC 最大的一个作用就是：在页面上有一个独立隔离容器，容器内的元素 和 容器 外的元素布局不会相互影响。
- 解决上外边距重叠；重叠的两个box都开启BFC
- 解决浮动引起高度塌陷；容器盒子开启BFC
- 解决文字环绕图片；左边图片div，右边文本容器p，将p容器开启BFC

设置元素浮动后，该元素的display值是多少？

论行内元素还是块元素，被设置浮动之后，display属性值都变为block；

定位

CSS中那个属性用来定位元素；

用来position属性来定位元素；

static:默认值不脱离文档流；

relative：相对定位，不脱离文档流，结合四个方位相对于自己的左上角进行定位；

absolute：绝对定位，脱离文档流，结合四个值相对于离他最近的元素进行定位，**注意：脱离文档流**

fixed：固定定位，脱离文档流，相对于可视窗口进行定位，出现滚动条也不会随之滚动；

z-index：提高层级（这个不是定位的属性）；

position跟display、margin collapse、overflow、float这些特性相互叠加后会怎么样？

先看看"display"、"position" 和 "float" 的相互关系

- "display" 的值为 "none" ,如果 "display"的值为 "none"，那么"position" 和"float"不起作用。在这种情况下，元素不产生框。因此浮动和定位无效,就是说，当元素是绝对定位时，浮动失效，"display"会被按规则重置
- "float"的值不是"none"。 按照规则，span 是行内元素，因此不能够设置其宽度和高度。但是浮动后"display" 成为block, position跟display、overflow、float下的margin collapse。
- margin collapse我觉得这里的意思应该是Collapsing margins，即外边距折叠，指的是毗邻的两个或多个外边距 (margin) 会合并成一个外边距。

其中所说的 margin 毗邻，可以归结为以下两点：

- 这两个或多个外边距没有被非空内容、padding、border 或 clear 分隔开。
- 这些 margin 都处于普通流中。

折叠问题

1.两个或多个毗邻的普通流中的块元素垂直方向上的 margin 会折叠。折叠后 margin 的计算

- 参与折叠的 margin 都是正值：取其中 margin 较大的值为最终 margin 值
- 参与折叠的 margin 都是负值：当 margin 都是负值的时候，取的是其中绝对值较大的，然后，从 0 位置，负向位移

- 参与折叠的 margin 中有正值，有负值：毗邻的 margin 中有正值，同时存在负值会怎样呢？有正有负，先取出负 margin 中绝对值中最大的，然后，和正 margin 值中最大的 margin 相加。
- 相邻的 margin 要一起参与计算，不得分步计算；要注意，相邻的元素不一定非要是兄弟节点，父子节点也可以，即使不是兄弟父子节点也可以相邻。而且，在计算时，相邻的 margin 要一起参与计算，不得分步计算。

2.浮动元素、inline-block 元素、绝对定位元素的 margin 不会和垂直方向上其他元素的 margin 折叠。

3.创建了块级格式化上下文的元素，不和它的子元素发生 margin 折叠

4.元素自身的 margin-bottom 和 margin-top 相邻时也会折叠

Position和display的取值和各自的意思和用法

position

- position属性取值：static(默认)、relative、absolute、fixed、inherit。
- position: static；始终处于文档流给予的位置。看起来好像没有用，但它可以快速取消定位，让 top, right, bottom, left的值失效。在切换的时候可以尝试这个方法。
- 除了static值，在其他三个值的设置下，z-index才会起作用。（确切地说z-index只在定位元素上有效）
- position: relative和absolute都可以用于定位，区别在于前者的div还属于正常的文档流，后者已经是脱离了正常文档流，不占据空间位置，不会将父类撑开。定位原点relative是相对于它在正常流中的默认位置偏移，它原本占据的空间任然保留；absolute相对于第一个position属性值不为static的父类。所以设置了position: absolute，其父类的该属性值要注意，而且overflow: hidden也不能乱设置，因为不属于正常文档流，不会占据父类的高度，也就不会有滚动条。
- position: fixed 旧版本IE不支持，定位原点相对于浏览器窗口，而且不能变。常用于header，footer，或者一些固定的悬浮div，随滚动条滚动又稳定又流畅，比JS好多了。fixed可以有很多创造性的布局和作用，兼容性是问题。
- position: inherit。规定从父类继承position属性的值。但是任何版本的IE都不支持该属性值。

display

- display属性取值：none、inline、inline-block、block、flex、inherit。
- display属性规定元素应该生成的框的类型。文档内任何元素都是框，块框或行内框。
- display: none和visibility: hidden都可以隐藏div，区别有点像absolute和relative，前者不占据文档的空间，后者还是占据文档的位置。
- display: inline和block，又叫行内元素和块级元素。表现出来的区别就是block独占一行，在浏览器中通常垂直布局，可以用margin来控制块级元素之间的间距；而inline以水平方式布局，垂直方向的margin和padding都是无效的，大小跟内容一样，且无法设置宽高。inline就像塑料袋，内容怎么样，就长得怎么样；block就像盒子，有固定的宽和高。
- inline-block就介于两者之间。
- display: flex 意为"弹性盒布局模型"，用来为盒状模型提供最大的灵活性。任何一个容器都可以指定为flex布局。设为Flex布局以后，子元素的float、clear和vertical-align属性将失效。采用flex布局的元素，称为flex容器。它的所有子元素自动成为容器成员，称为flex项目（flex item）。

容器的属性：以下6个属性设置在容器上。

- flex-direction 决定主轴的方向（项目的排列方向）。
 - row（默认值）：主轴为水平方向，起点在左端。
 - row-reverse：主轴为水平方向，起点在右端。
 - column：主轴为垂直方向，起点在上沿。
 - column-reverse：主轴为垂直方向，起点在下沿。
- flex-wrap 默认情况下，项目都排在一条线上。flex-wrap属性定义，如果一条轴线排不下，如何换行。
 - nowrap（默认）：不换行。
 - wrap：换行，第一行在上方。
 - wrap-reverse：换行，第一行在下方。
- flex-flow 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 定义了项目在主轴上的对齐方式。假设主轴为从左到右。
 - flex-start（默认值）：左对齐
 - flex-end：右对齐
 - center：居中
 - space-between：两端对齐，项目之间的间隔都相等。
 - space-around：每个项目两侧的间隔相等。所以，项目之间的间隔比项目与边框的间隔大一倍。
- align-items 定义项目在交叉轴上如何对齐。假设交叉轴从上到下。
 - flex-start：交叉轴的起点对齐。
 - flex-end：交叉轴的终点对齐。
 - center：交叉轴的中点对齐。
 - baseline：项目的第一行文字的基线对齐。
 - stretch（默认值）：如果项目未设置高度或设为auto，将占满整个容器的高度。
- align-content 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。
 - flex-start：与交叉轴的起点对齐。
 - flex-end：与交叉轴的终点对齐。
 - center：与交叉轴的中点对齐。
 - space-between：与交叉轴两端对齐，轴线之间的间隔平均分布。
 - space-around：每根轴线两侧的间隔都相等。所以，轴线之间的间隔比轴线与边框的间隔大一倍。
 - stretch（默认值）：轴线占满整个交叉轴。

项目的属性：以下6个属性设置在项目上。

- order：定义项目的排列顺序。数值越小，排列越靠前，默认为0。
- flex-grow：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。如果所有项目的flex-grow属性都为1，则它们将等分剩余空间（如果有的话）。如果一个项目的flex-grow属性为

2, 其他项目都为1, 则前者占据的剩余空间将比其他项多一倍。

- flex-shrink: 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: 定义了分配多余空间之前, 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。可以设为跟width或height属性一样的值(比如350px), 则项目将占据固定空间。
- flex: flex属性是flex-grow, flex-shrink 和 flex-basis的简写, 默认值为0 1 auto。后两个属性可选。该属性有两个快捷值: auto (1 1 auto) 和 none (0 0 auto)。建议优先使用这个属性, 而不是单独写三个分离的属性
- align-self: 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。该属性可能取6个值, 除了auto, 其他都与align-items属性完全一致。

position的值, relative, absolute, fixed分别是相对于谁进行定位的(定位原点是)?

static (默认): 按照正常文档流进行排列;

relative (相对定位): 不脱离文档流, 参考自身静态位置通过 top, bottom, left, right 定位;

absolute(绝对定位): 参考距其最近一个不为static的父级元素通过top, bottom, left, right 定位;

fixed(固定定位): 所固定的参照对象是可视窗口。

请说明z-index的工作原理, 适用范围?

- z-index属性指定两件事:
 - 当前元素的堆叠顺序
 - 当前元素是否建立新的堆叠上下文
- 该属性仅适用于定位元素。即拥有 relative, absolute, fixed属性的 position 元素。
- z-index: auto | <integer> | inherit

absolute的containing block计算方式跟正常流有什么不同?

无论属于哪种, 都要先找到其祖先元素中最近的 position 值不为 static 的元素, 然后再判断:

1. 若此元素为 inline 元素, 则 containing block 为能够包含这个元素生成的第一个和最后一个 inline box 的 padding box (除 margin, border 外的区域) 的最小矩形;
2. 否则, 则由这个祖先元素的 padding box 构成。

如果都找不到, 则为 initial containing block。

补充:

1. static(默认的)/relative: 简单说就是它的父元素的内容框 (即去掉padding的部分)
2. absolute: 向上找最近的定位为absolute/relative的元素
3. fixed: 它的containing block一律为根元素(html/body)

有一个高度自适应的div, 里面有两个div, 一个高度100px, 希望另一个填满剩下的高度

外层div使用position: relative; 高度要求自适应的div使用position: absolute; top: 100px; bottom: 0; left: 0

背景

解释CSS sprites(雪碧图)如何使用?

CSS Sprites是CSS图像合成技术, 有的称CSS精灵有的称雪碧图, 他的作用就是把网页上很多小图标放到一张图片里面, 然后通过CSS里面的background-position来控制每个图片的坐标, 再利用CSS的“background-image”, “background-repeat”的组合来精确的定位出背景图片的位置。

使用场景

- 1.静态图片, 不随用户信息的变化而变化
- 2.小图片, 图片容量比较小
- 3.加载量比较大

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?

- 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。
- 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。
- 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。

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。

动画

请用CSS写一个简单的幻灯片效果页面?

```
1 <!DOCTYPE html>
2 <html lang="zh-CN">
3 <head>
4     <meta charset="UTF-8">
5     <title>css3实现幻灯片效果</title>
6     <style type="text/css">
7         .huanDengPic {
8             width: 600px;
9             height: 600px;
10            margin: 50px auto 0;
11            overflow: hidden;
```



```

12         box-shadow: 0 0 5px rgba(0,0,0,1);
13         background-size: cover;
14         background-position: center;
15         -webkit-animation:loops 10s infinite;
16         animation:loops 10s infinite;
17     }
18
19     @-webkit-keyframes loops {
20         0% {
21             background: url(http://t1.mmonly.cc/uploads/tu/zyf/t
22             t/20160520/kewnk2ermbe.jpg) no-repeat;
23         }
24         50% {
25             background: url(http://t1.mmonly.cc/uploads/tu/zyf/t
26             t/20160520/j0qoejfukbu.jpg) no-repeat;
27         }
28         100% {
29             background: url(http://t1.mmonly.cc/uploads/tu/zyf/t
30             t/20160520/oajlgqc2nud.jpg) no-repeat;
31         }
32     }
33 </style>
34 </head>
35 <body>
36     <div class="huanDengPic"></div>
37 </body>
38 </html>

```

列举三项transform的效果？

none 定义不进行转换。

matrix(n,n,n,n,n,n) 定义 2D 转换，使用六个值的矩阵。

matrix3d(n,n,n,n,n,n,n,n,n,n,n,n,n,n,n,n) 定义 3D 转换，使用 16 个值的 4x4 矩阵。

translate(x,y) 定义 2D 转换。

translate3d(x,y,z) 定义 3D 转换。

translateX(x) 定义转换，只是用 X 轴的值。

translateY(y) 定义转换，只是用 Y 轴的值。

translateZ(z) 定义 3D 转换，只是用 Z 轴的值。

scale(x,y) 定义 2D 缩放转换。

scale3d(x,y,z) 定义 3D 缩放转换。

scaleX(x) 通过设置 X 轴的值来定义缩放转换。

scaleY(y) 通过设置 Y 轴的值来定义缩放转换。

scaleZ(z) 通过设置 Z 轴的值来定义 3D 缩放转换。

rotate(angle) 定义 2D 旋转，在参数中规定角度。

rotate3d(x,y,z,angle) 定义 3D 旋转。

rotateX(angle) 定义沿着 X 轴的 3D 旋转。

rotateY(angle) 定义沿着 Y 轴的 3D 旋转。

rotateZ(angle) 定义沿着 Z 轴的 3D 旋转。

skew(x-angle,y-angle) 定义沿着 X 和 Y 轴的 2D 倾斜转换。

skewX(angle) 定义沿着 X 轴的 2D 倾斜转换。

skewY(angle) 定义沿着 Y 轴的 2D 倾斜转换。

perspective(n) 为 3D 转换元素定义透视视图。

CSS中的animation动画如何使用 / transition animation 什么区别？

使用方法

由”@keyframes”开头，后面是“动画的名称”加上“{}”，括号中就是一些不同时间段样式规则。可以在这个规则中创建多个百分比，我们分别给每一个百分比中给需要有动画效果的元素加上不同的属性，比如说移动，改变元素颜色，位置，大小，形状等

```
1 div{
2  animation: box 1s ease 2s infinite alternate forwards;
3 }
4 @keyframes box {
5  0%{transform: translateX(0)}
6  100%{transform: translateX(1000px)}
7 }
```

区别

a、transition需要一个触发事件(hover,checked,focus)才会随时间改变其CSS属性；animation在不需要触发任何事件的情况下，也可以显式的随时间变化来改变元素CSS属性。

b、transition是一次性的，除非不停的触发；animation:可以设置循环次数或(infinite)不停的执行动画

c、transition只能定义开始状态和结束状态，不能定义中间状态，只有两个状态。

animation通过控制keyframes(关键帧)来控制动画的每一步，类似flash的关键帧来声明动画。

(0%,10%,70%,100%)，或(from,to)

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

如果需要手动写动画，你认为最小时间间隔是多久，为什么？

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 $1/60 \times 1000\text{ms} = 16.7\text{ms}$ 。

全屏滚动的原理是什么？用到了CSS的哪些属性？

- 原理：有点类似于轮播，整体的元素一直排列下去，假设有5个需要展示的全屏页面，那么高度是500%，只是展示100%，剩下的可以通过transform进行y轴定位，也可以通过margin-top实现
- overflow: hidden; transition: all 1000ms ease;

性能优化

知道的网页制作会用到的图片格式有哪些？

png-8, png-24, jpeg, gif, svg。

WebP格式，谷歌（google）新推出的影像技术，它可让网页图档有效进行压缩，同时又不影响图片格式兼容与实际清晰度，进而让整体网页下载速度加快。图片压缩体积大约只有JPEG的2/3，并能节省大量的服务器带宽资源和数据空间。Facebook Ebay等知名网站已经开始测试并使用WebP格式。在质量相同的情况下，WebP格式图像的体积要比JPEG格式图像小40%

简述一下src与href的区别？知道什么是微格式吗？谈谈理解。在前端构建中应该考虑微格式吗？

src与href的区别

- href 是指向网络资源所在位置，建立和当前元素（锚点）或当前文档（链接）之间的链接，用于超链接；如果我们在文档中添加 `<link href="min.css" rel="stylesheet"/>` 那么浏览器会识别该文档为css文件，就会并行下载资源并且不会停止对当前文档的处理,这也是为什么建议使用 link 方式来加载css，而不是使用 @import 方式。
- src 是指向外部资源的位置，指向的内容将会嵌入到文档中当前标签所在位置（替换当前元素）；在请求src资源时会将其指向的资源下载并应用到文档内，例如js脚本，img图片和frame等元素。当浏览器解析到该元素时，会暂停其他资源的下载和处理（同步操作），直到将该资源加载、编译、执行完毕，图片和框架等元素也如此，类似于将所指向资源嵌入当前标签内。这也是为什么将js脚本放在底部而不是头部。

微格式

微格式（Microformats）是一种让机器可读的语义化 XHTML 词汇的集合，是结构化数据的开放标准。是为特殊应用而制定的特殊格式。

优点：将智能数据添加到网页上，让网站内容在搜索引擎结果界面可以显示额外的提示。

一个页面上有大量的图片（大型电商网站），加载很慢，你有哪些方法优化这些图片的加载，给用户更好的体验？

- 图片懒加载，滚动到相应位置才加载图片。
- 图片预加载，如果为幻灯片、相册等，将当前展示图片的前一张和后一张优先下载。
- 使用CSSsprite, SVGsprite, Iconfont、Base64等技术，如果图片为css图片的话。
- 如果图片过大，可以使用特殊编码的图片，加载时会先加载一张压缩的特别厉害的缩略图，以提高用户体验。

谈谈以前端角度出发做好SEO需要考虑什么？CSS优化、提高性能的方法有哪些？

- 避免过度约束
- 避免后代选择符
- 避免链式选择符
- 使用紧凑的语法
- 避免不必要的命名空间
- 避免不必要的重复
- 最好使用表示语义的名字。一个好的类名应该是描述他是什么而不是像什么
- 避免！important，可以选择其他选择器
- 尽可能的精简规则，你可以合并不同类里的重复规则

居中

块元素（div）水平垂直居中，文字内容在div中始终水平垂直居中 / 请列出你所知道的使用css让元素上下居中方案(最好包含flex的方法) / 如何垂直居中一个浮动元素？ / 如何保持浮层水平垂直居中

<https://jingyan.baidu.com/article/3a2f7c2e26041a26aed61150.html>

CSS中可以让文字在垂直和水平方向上居中的两个属性是什么？ / text-align:center和line-height有什么区别？

文字水平居中： text-align: center

文字垂直居中: `line-height = height`

单位

CSS中表示长度的单位有哪些; px和em的区别? / px,em,rem的区别?

css中长度单位有很多, 可谓五花八门, 但可分为这两类:

- 绝对长度单位: 彼此固定, 不会因为其他元素的尺寸变化而变化。主要有 `cm` `mm` `Q` `in` `pc` `pt`
`px`
- 相对长度单位: 指定相对于另一长度的长度。主要有 `em` `ex` `ch` `rem` `%` 和可视区百分比长度单位
`vm` `vh` `vmin` `vmax`

px

- 表示“绝对尺寸”(并非真正的绝对), 实际上就是css中定义的像素, 利用px设置字体大小及元素宽高等比较稳定和精确。
- 它是绝对单位, 不会因为其他元素的尺寸变化而变化。
- px的缺点是其不能适应浏览器缩放时产生的变化, 因此一般不用于响应式网站。

em

- 表示相对尺寸, 值并不是固定的, 其相对于当前对象内文本的font-size (如果当前对象内文本的font-size计量单位也是em, 则当前对象内文本的font-size的参考对象为父元素文本font-size)。
- 使用em可以较好的相应设备屏幕尺寸的变化, 但是在进行元素设置时都需要知道父元素文本的font-size及当前对象内文本的font-size, 如有遗漏可能会导致错误。
- 当使用em单位时, 像素值将是em值乘以使用em单位的元素的字体大小。

rem

- 也表示相对尺寸, 其参考对象为根元素的font-size, 因此只需要确定这一个font-size。
- 当使用 rem 单位, 他们转化为像素大小取决于页根元素的字体大小, 即 html 元素的字体大小。根元素字体大小乘以你 rem 值。

布局

div+css的布局较table布局有什么优点?

div+css布局的好处:

- 符合W3C标准, 代码结构清晰明了, 结构、样式和行为分离, 带来足够好的可维护性。
- 布局精准, 网站版面布局修改简单。
- 加快了页面的加载速度 (最重要的) (在IE中要将整个table加载完了才显示内容)。

4.节约站点所占的空间和站点的流量。

5.用只包含结构化内容的HTML代替嵌套的标签，提高另外搜索引擎对网页的搜索效率。

table布局的好处（table布局也不是一点用的没有，这点是毋庸置疑的）

1.容易上手。

2.可以形成复杂的变化，简单快速。

3.表现上更加“严谨”，在不同浏览器中都能得到很好的兼容。

实现页面的三栏布局，左右均为定宽，高度可以自适应；（双飞翼布局 、圣杯布局）

https://blog.csdn.net/cinderella_hou/article/details/52156333

<https://www.cnblogs.com/woodk/p/5147085.html>

这个是渔人码头的博客，还是很好理解的

响应式网页主要原理？

CSS3中的Media Query（媒体查询）：通过查询screen的宽度来指定某个宽度区间的网页布局。

@media screen and (判断条件){属性样式}

了解过flex吗？

文章参考链接：

http://www.ruanyifeng.com/blog/2015/07/flex-grammar.html?utm_source=tuicool（语法篇）

<http://www.ruanyifeng.com/blog/2015/07/flex-examples.html>（实例篇）

Flex是Flexible Box的缩写，意为“弹性布局”，用来为盒状模型提供最大的灵活性。布局的传统解决方案，基于盒状模型，依赖 display属性 + position属性 + float属性。它对于那些特殊布局非常不方便，比如，垂直居中就不容易实现。

简单的分为容器属性和元素属性容器的属性：

- flex-direction：决定主轴的方向（即子item的排列方法） .box { flex-direction: row | row-reverse | column | column-reverse; }
- flex-wrap：决定换行规则 .box{ flex-wrap: nowrap | wrap | wrap-reverse; }
- flex-flow： .box { flex-flow: || ; }
- justify-content：对其方式，水平主轴对齐方式
- align-items：对齐方式，垂直轴线方向

项目的属性（元素的属性）：

- order属性：定义项目的排列顺序，顺序越小，排列越靠前，默认为0
- flex-grow属性：定义项目的放大比例，即使存在空间，也不会放大

- flex-shrink属性：定义了项目的缩小比例，当空间不足的情况下会等比例的缩小，如果定义个item的flex-shrink为0，则为不缩小
- flex-basis属性：定义了再分配多余的空间，项目占据的空间。
- flex：是flex-grow和flex-shrink、flex-basis的简写，默认值为0 1 auto。
- align-self：允许单个项目与其他项目不一样的对齐方式，可以覆盖align-items，默认属性为auto，表示继承父元素的align-items

比如说，用flex实现圣杯布局

请解释一下CSS3的flexbox（弹性盒布局模型），以及适用场景？

该布局模型的目的是提供一种更加高效的方式来对容器中的条目进行布局、对齐和分配空间。在传统的布局方式中，block 布局是把块在垂直方向从上到下依次排列的；而 inline 布局则是在水平方向来排列。弹性盒布局并没有这样内在的方向限制，可以由开发人员自由操作。

试用场景：弹性布局适合于移动前端开发，在Android和ios上也完美支持。

移动端的布局用过媒体查询吗？ / 如何适配不同大小的手机屏幕；

通过媒体查询可以为不同大小和尺寸的媒体定义不同的css，适应相应的设备的显示。

1. <head>里边

```
<link rel="stylesheet" type="text/css" href="xxx.css" media="only screen and (max-device-width:480px)">
```

2. CSS : @media only screen and (max-device-width:480px) {/css样式/}

什么是响应式设计？响应式设计的基本原理是什么？如何兼容低版本的IE？

响应式网站设计(Responsive Web design)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。

页面头部必须有meta声明的viewport。

```
<meta name='viewport' content="width=device-width, initial-scale=1. maximum-scale=1,user-scalable=no">
```

选择器及优先级

请描述css选择器的优先级原则？ / 有哪些选择符，优先级的计算公式是什么？行内样式和! important哪个优先级高？ / CSS都有哪些选择器？ / CSS 选择符有哪些？哪些属性可以继承？ / 优先级算法如何计算？

<https://www.cnblogs.com/zxjwlh/p/6213239.html>

CSS选择符：id选择器(#myid)、类选择器(.myclassname)、标签选择器(div, h1, p)、相邻选择器(h1 + p)、子选择器 (ul > li)、后代选择器 (li a)、通配符选择器 (*)、属性选择器

(a[rel="external"])、伪类选择器 (a:hover, li:nth-child

可继承的属性: font-size, font-family, color

不可继承的样式: border, padding, margin, width, height

优先级 (就近原则): !important > [id > class > tag]

!important 比内联优先级高

元素选择符: 1

class选择符: 10

id选择符: 100

元素标签: 1000

1. !important声明的样式优先级最高, 如果冲突再进行计算。

2. 如果优先级相同, 则选择最后出现的样式。

3. 继承得到的样式的优先级最低。

总结排序: !important > 行内样式 > ID选择器 > 类选择器 > 标签 > 通配符 > 继承 > 浏览器默认属性

写一个表格以及对应的CSS,使表格奇数行为白色背景, 偶数行为灰色, 鼠标一上去为黄色背景。

奇数行 :nth-child(odd) 或者 :nth-child(2n+1)

偶数行:nth-child(even) 或者 :nth-child(2n)

```
table tr:nth-child(odd){background:white;}
```

```
table tr:nth-child(even){background:gray;}
```

```
table tr:hover{background:yellow;}
```

样式的层级关系, 选择器优先级, 样式冲突, 以及抽离样式模块怎么写, 说出思路, 有无实践经验

<http://blog.csdn.net/zengyonglan/article/details/53337879>

CSS选择符有哪些? 哪些属性可以继承? 优先级算法如何计算? css3新增伪类有哪些?

CSS 选择符:

1.id选择器(# myid)

2.类选择器(.myclassname)

3.标签选择器(div, h1, p)

4.相邻选择器(h1 + p)

5.子选择器(ul > li)

- 6.后代选择器(li a)
- 7.通配符选择器(*)
- 8.属性选择器(a[rel = "external"])
- 9.伪类选择器(a: hover, li:nth-child)

可继承的样式：

- 1.font-size
- 2.font-family
- 3.color
- 4.text-indent

不可继承的样式：

- 1.border
- 2.padding
- 3.margin
- 4.width
- 5.height

优先级算法：

- 1.优先级就近原则，同权重情况下样式定义最近者为准；
- 2.载入样式以最后载入的定位为准；
- 3.important > id > class > tag
- 4.important 比 内联优先级高，但内联比 id 要高

CSS3新增伪类举例：

p:first-of-type 选择属于其父元素的首个 <p> 元素的每个 <p> 元素。

p:last-of-type 选择属于其父元素的最后 <p> 元素的每个 <p> 元素。

p:only-of-type 选择属于其父元素唯一的 <p> 元素的每个 <p> 元素。

p:only-child 选择属于其父元素的唯一子元素的每个 <p> 元素。

p:nth-child(2) 选择属于其父元素的第二个子元素的每个 <p> 元素。

:enabled :disabled 控制表单控件的禁用状态。

:checked 单选框或复选框被选中。

::before 和 :after中双冒号和单冒号有什么区别？解释一下这2个伪元素的作用

- 单冒号(:)用于CSS3伪类，双冒号(::)用于CSS3伪元素。
- ::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

浏览器和兼容

你做的页面在哪些浏览器测试过？这些浏览器的内核分别是什么？

Webkit内核(v8引擎)：谷歌浏览器(chrome)、safari、大部分国产浏览器(360、搜狗、QQ、UC、猎豹、百度...)、安卓和IOS大部分手机浏览器...

Gecko内核：火狐浏览器(firefox)

Presto内核：欧朋浏览器(opera)

Trident内核：IE浏览器 (IE6 – IE11)

KHTML排版引擎：大部分网站都已放弃IE8及更低版本浏览器（例如：淘宝）

IE的双边距BUG：块级元素float后设置横向margin，ie6显示的margin比设置的较大？

```
#box{ float:left; width:10px; margin:0 0 0 10px;}
```

这种情况之下IE会产生20px的距离；

解决方案：在float的标签样式控制中加入 _display:inline; 将其转化为行内属性。（_ 这个符号只有ie6会识别）

如何处理浏览器兼容性？什么是优雅降级开发模式？/ 你能描述一下渐进增强和优雅降级之间的不同吗？

<https://blog.csdn.net/OBKoro1/article/details/70141882>

由于低级浏览器不支持 CSS3，但是 CSS3 特效太优秀不忍放弃，所以产生了一种解决方式在高级浏览器中使用CSS3，而在低级浏览器只保证最基本的功能。

渐进增强 (progressive enhancement)

针对低版本浏览器进行构建页面，保证最基本的功能，然后再针对高级浏览器进行效果、交互等改进和追加功能达到更好的用户体验。就是以最低要求，实现最基础功能为基本，向上兼容。以css为例，以下这种写法就是渐进增强

```
1 .box{
2   -webkit-border-radius:10px;
3   -moz- border- radius: 10px;
```

```
4 border- radius: 10px;
5 }
```

优雅降级 (graceful degradation)

一开始就构建完整的功能，然后再针对低版本浏览器进行兼容。也就是以高要求，高版本为基准，向下兼容。同样以css为例，优雅降级的写法如下

```
1 .box{
2 border- radius: 10px;
3 -moz- border- radius: 10px;
4 -webkit- border- radius: 10px;
5 }
```

区别：优雅降级是从复杂的现状开始，并试图减少用户体验的供给，而渐进增强则是从一个非常基础的、能够起作用的版本开始，并不断扩充，以适应未来环境的需要。

经常遇到的浏览器的兼容性有哪些？原因，解决方法是什么？ / 常用hack的技巧？ / 常见的兼容性问题？ / 常见的CSS hack问题？什么是Css Hack？ie6,7,8的hack分别是什么？
HTML常见兼容性问题？

<https://www.zhangxinxu.com/wordpress/2013/05/css3-box-shadow-animation-loading-ie10-css-hack/>

1. 不同浏览器的标签默认的margin和padding不一样。

```
*{margin:0;padding:0;}
```

2. IE6双边距bug：块属性标签float后，又有横行的margin情况下，在IE6显示margin比设置的大。
hack: display:inline;将其转化为行内属性。

3. 渐进识别的方式，从总体中逐渐排除局部。首先，巧妙的使用“9”这一标记，将IE浏览器从所有情况中分离出来。接着，再次使用“+”将IE8和IE7、IE6分离开来，这样IE8已经独立识别。

```
4. {
```

```
5. background-color:#f1ee18;/*所有识别*/
```

```
6. .background-color:#00deff\9; /*IE6、7、8识别*/
```

```
7. +background-color:#a200ff;/*IE6、7识别*/
```

```
8. _background-color:#1e0bd1;/*IE6识别*/
```

```
9. }
```

10. 设置较小高度标签（一般小于10px），在IE6，IE7中高度超出自己设置高度。hack：给超出高度的标签设置overflow:hidden;或者设置行高line-height 小于你设置的高度。

11. IE下，可以使用获取常规属性的方法来获取自定义属性,也可以使用getAttribute()获取自定义属性；Firefox下，只能使用getAttribute()获取自定义属性。解决方法:统一通过getAttribute()获取自定义属性。

12. Chrome 中文界面下默认会将小于 12px 的文本强制按照 12px 显示,可通过加入 CSS 属性 - webkit-text-size-adjust: none; 解决。

13. 超链接访问过后hover样式就不出现了，被点击访问过的超链接样式不再具有hover和active了。解决方法是改变CSS属性的排列顺序:L-V-H-A (love hate): a:link {} a:visited {} a: hover {} a:active {}

