

Part 2 CSS

1.介绍一下标准的CSS的盒子模型？与低版本IE的盒子模型有什么不同的？

CSS框模型(Box Model)规定了元素框处理元素内容、内边距、边框和外边距的方式。元素框的最内部分是实际的内容，直接包围内容的是内边距。内边距呈现了元素的背景。内边距的边缘是边框。边框以外是外边距，外边距默认是透明的，因此不会遮挡其后的任何元素。

盒模型组成：内容(content)、内边距(padding)、边框(border)、外边距(margin)。

两种盒模型：IE盒子模型、标准W3C盒子模型；IE的content部分包含了border和padding。

2.CSS选择符有哪些？哪些属性可以继承？

CSS选择符分类：9类

- (1)标签/元素/类型选择器 (div, h1, p)
- (2)通配符选择器 (*)
- (3)类选择器 (.myclassname)
- (4)ID选择器 (#myid)
- (5)属性选择器 (a[rel = "external"])
- (6)后代选择器 (li a)
- (7)子选择器 (ul > li)
- (8)相邻选择器 (h1 + p)
- (9)伪类选择器 (a: hover, li:nth-child)

CSS可继承的属性：

- (1) 字体属性(Font)系列的：font-style, font-variant, font-weight, font-size, font-family.
- (2) 文本属性(Text)：color, direction, letter-spacing, line-height, text-align, text-indent, text-transform, word-spacing, and so on.
- (3) 列表属性(List)：list-style, list-style-image, list-style-position, list-style-type.
- (4) 表格属性：border-collapse, border-spacing, caption-side, empty-cells, table-layout.
- (5) Other：visibility, cursor.

CSS不可继承的属性：

animation, background, margin, padding, border, 尺寸属性(height/width), display, float, position, overflow, transform, transition.

3.CSS优先级算法如何计算？

优先级就近原则，同权重情况下样式定义最近者为准；

载入样式以最后载入的定位为准；

优先级为：

!important > id > class > tag

!important比内联优先级高,但内联比id要高.

4.CSS3新增伪类有那些？

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。

:after css2	在元素之前添加内容,也可以用来做清除浮动。
:before css2	在元素之后添加内容
:enabled	
:disabled	控制表单控件的禁用状态。
:checked	单选框或复选框被选中。

5.如何居中div？如何居中一个浮动元素？如何让绝对定位的div居中？

盘点8种CSS实现垂直居中水平居中的绝对定位居中技术
大小不固定的图片、多行文字的水平垂直居中
为什么 CSS 3 要实现垂直居中那么的难？

6.display有哪些值？说明他们的作用。

none 此元素不会被显示。
block 此元素将显示为块级元素，此元素前后会带有换行符。
inline 默认。此元素会被显示为内联元素，元素前后没有换行符。
inline-block 行内块元素。像行内元素一样显示，但其内容像块类型元素一样显示。
list-item 此元素会作为列表显示。像块类型元素一样显示，并添加样式列表标记。
table 此元素会作为块级表格来显示（类似 `<table>`），表格前后带有换行符。
inherit 规定应该从父元素继承 display 属性的值。

7.position的值有哪些？relative和absolute定位原点是什么？

*static 默认值。没有定位，元素出现在正常的流中（忽略top, bottom, left, right z-index声明）。
元素框正常生成。块级元素生成一个矩形框，作为文档流的一部分，行内元素则会创建一个或多个行框，置于其父元素中。
*relative 生成相对定位的元素，相对于其在普通流中的位置进行定位。元素框偏移某个距离。元素仍保持其未定位前的形状，它原本所占的空间仍保留。
*absolute 生成绝对定位的元素，相对于static定位以外的第一个祖先元素进行定位。
元素框从文档流完全删除，并相对于其包含块定位。包含块可能是文档中的另一个元素或者是初始包含块。元素原先在正常文档流中所占的空间会关闭，就好像元素原来不存在一样。元素定位后生成一个块级框，而不论原来它在正常流中生成何种类型的框。
*fixed（老IE不支持）生成绝对定位的元素，相对于浏览器窗口进行定位。
*inherit 规定从父元素继承position属性的值。

8.CSS3有哪些新特性？

CSS边框

创建圆角边框(border-radius)，向矩形添加阴影(box-shadow)，使用图片来绘制边框(border-image)

CSS3 背景

background-size规定背景图片的尺寸
background-origin规定背景图片的定位区域
background-clip规定背景的绘制区域。

CSS3 文本效果

text-shadow文本阴影
word-wrap自动换行,允许对长的不可分割的单词进行分割并换行到下一行。

CSS3 @font-face 规则（自定义字体）

CSS3 2D 转换（transform）

通过 CSS3 转换，我们能够对元素进行移动、缩放、转动、拉长或拉伸。
转换是使元素改变形状、尺寸和位置的一种效果。
translate(), rotate(), scale(), skew(), matrix()。

CSS3 3D 转换

rotateX(), rotateY()

CSS3 过渡（transition）

CSS3 过渡是元素从一种样式逐渐改变为另一种的效果。

CSS3 动画（animation）

动画是使元素从一种样式逐渐变化为另一种样式的效果。
过渡和动画的区别：虽然它们两者都是通过改变元素的属性值来实现动画效果，但是transition只能通过指定属性的开始值与结束值，然后在这两者之间进行平滑过渡的方式来实现动画的效果，因此不能实现较复杂的动画效果；而animation功能可以通过定义多个关键帧以及每个关键帧中元素的属性值来实现复杂的动画效果。

CSS3 多列

column-count 规定元素应该被分隔的列数
column-gap 规定列之间的间隔
column-rule 设置列之间的宽度、样式和颜色规则

CSS3 用户界面 新的用户界面特性包括重设元素尺寸、盒尺寸以及轮廓等。

resize 规定是否可由用户调整元素尺寸
box-sizing 允许您以确切的方式定义适应某个区域的具体内容
outline-offset 对轮廓进行偏移，并在超出边框边缘的位置绘制轮廓
常见的前缀有：
• Chrome: -webkit-

- Safari: -webkit-
- Firefox: -moz-
- IE: -ms-
- Opera: -o-

透明度的设置

opacity:0.5;

filter:Alpha(opacity=50); /* IE8 以及更早的浏览器 */

渐变效果

linear-gradient;

多重背景

9.请解释一下CSS3的Flexbox（弹性盒布局模型）,以及适用场景？

[Flex 布局教程：语法篇](#)

10.用纯CSS创建一个三角形的原理是什么？

把上、左、右三条边隐藏掉（颜色设为 transparent）

```
#demo {
  width: 0;
  height: 0;
  border-width: 20px;
  border-style: solid;
  border-color: transparent transparent red transparent;
}
```

11.一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

12.常见兼容性问题？

[【web前端开发】浏览器兼容性处理大全](#)

13.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。display: inline-block;

经测试不好用，字体消失了，还有重新设置字体大小；

```
.wrap ul{letter-spacing: -5px;}
```

```
.wrap ul li{letter-spacing: normal;}
```

14.为什么要初始化CSS样式。

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。

当然，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。

最简单的初始化方法就是：*{padding: 0; margin: 0;}(不建议)

15.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)，根元素也是initial containing block

16.CSS里的visibility属性有个collapse属性值是干嘛用的？在不同浏览器下以后什么区别？

对于普通元素visibility:collapse;会将元素完全隐藏,不占据页面布局空间,与display:none;表现相同. 如果目标元素为table,visibility:collapse;将table隐藏,但是会占据页面布局空间. 仅在Firefox下起作用,IE会显示元素,Chrome会将元素隐藏,但是占据空间.

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

如果元素的display为none,那么元素不被渲染,position,float不起作用,如果元素拥有position:absolute;或者position:fixed;属性那么元素将为绝对定位,float不起作用.如果元素float属性不是none,元素会脱离文档流,根据float属性值来显示.有浮动,绝对定位,inline-block属性的元素,margin不会和垂直方向上的其他元素margin折叠.

18.对BFC规范(块级格式化上下文：block formatting context)的理解？

BFC(Block formatting context)，直译为‘块级格式化上下文’，它是一个独立的渲染区域，只有Block-level box参与，它规定了内部的Block-level Box如何布局，并且与这个区域外部毫不相干。在同一个BFC中的两个毗邻的块级盒在垂直方向（和布局方向有关系）的margin会发生折叠。

以下内容摘自《简书》：

对CSS有了解的朋友肯定都知道盒式模型这个概念，对一个元素设置CSS，首先需要知道这个元素是block还是inline类型。而BFC就是用来格式化块级盒子，同样管理inline类型的盒子还有IFC，以及其他的FC。那首先我们就来看看FC的概念。

Formatting Context：指页面中一个渲染区域，并且拥有一套渲染规则，他决定了其子元素如何定位，以及与其他元素的相互关系和作用。
BFC：块级格式化上下文，它是指一个独立的块级渲染区域，只有Block-level Box参与，该区域拥有一套渲染规则来约束块级盒子的布局，且与区域外部无关。

BFC的生成

既然上文提到BFC是一块渲染区域，那这块渲染区域到底在哪，它又是有多大，这些由生成BFC的元素决定，CSS2.1中规定满足下列CSS声明之一的元素便会生成BFC。

float的值不为none；

overflow的值不为visible；

display的值为inline-block、table-cell、table-caption；

position的值为absolute或fixed；

看到有人把display：table也认为可以生成BFC，其实这里的主要原因在于Table会默认生成一个匿名的table-cell，正是这个匿名的table-cell生成了BFC。

BFC的约束规则

浏览器对于BFC这块区域的约束规则如下：

生成BFC元素的子元素会一个接一个的放置。垂直方向上他们的起点是一个包含块的顶部，两个相邻子元素之间垂直距离取决于元素margin特性。在BFC中相邻的块级元素外边距会折叠。

生成BFC元素的子元素中，每一个子元素的外边距和包含块的左边界相接触，（对于从右到左的格式化，右边边距和右边界相接触），除非这个子元素也创建了一个新的BFC（如它自身也是一个浮动元素）。

有朋友对它做了分解，我们直接拿来：

内部的BOX会在垂直方向上一个接一个的放置；

垂直方向上的距离有margin决定。（完整的说法是：属于同一个BFC的2个相邻的BOX的margin会发生重叠，与方向无关。）

每个元素的左外边距与包含块的左边界相接触（从左到右），即使浮动元素也是如此。（这说明BFC中的子元素不会超出它的包含块，而position为absolute的元素可以超出它的包含块边界）；

BFC的区域不会与float的元素区域重叠；

计算BFC的高度时，浮动子元素也参与计算；

BFC就是页面上的一个隔离的独立容器，容器里面的子元素不会影响到外面的元素，反之亦然；

看到以上的几条约束，让我想起了学习CSS时的几条规则：

Block元素会扩展到与父元素同宽，所以block元素会垂直排列；

垂直方向上的两个相邻DIV的margin会重叠，而水平方向不会（此规则并不完全正确）；

浮动元素会尽量往左上方（或右下方）；

为父元素设置overflow：hidden或浮动父元素，则会包含浮动元素；

.....

BFC在布局中的应用

上面说了那么多，那么BFC究竟有什么用，毕竟再好的东西也要为我所用才行。

防止margin重叠：

同一个BFC中两个相邻的box才会发生重叠与方向无关，不过由于上文提到的第一条限制，我们甚少看到水平方向的margin重叠。这在IE中是个特例，IE可以设置write-mode。

要阻止margin重叠，只要将两个元素别放在一个BFC中即可（可以用上文提到的方式让相邻元素其中一个生成BFC）。阻止两个相邻元素的重叠看起来没有意义，主要用于嵌套元素。

浮动相关问题：

使得父元素包含子元素，常见的方式是为父元素设置overflow：hidden或者浮动父元素。根本原因在于创建BFC的元素，子浮动元素也会参与其高度计算，即不会产生高度塌陷问题。

多栏布局的一种方式

上文提到的一条规则：与浮动元素相邻的已生成BFC的元素不能与浮动元素互相覆盖。利用该特性可以作为多栏布局的一种实现方式。

这种布局的特点在于左右俩栏的宽度固定，中间栏可以根据浏览器宽度自适应。

IE中也有与BFC类似概念的haslayout。

总结

在我第一次接触BFC时经常有一个疑问，BFC的结构是什么样的，像DOM一样的树状结构，还是一个BFC集合。其实我们不需要关心BFC的具体结构，这要看浏览器的具体实现采用什么样的数据结构。对于BFC我们只需要知道使用一定的CSS声明可以生成BFC，浏览器对生成的BFC有一系列的渲染规则，利用这些渲染规则我们可以达到一定的布局效果，为了达到特定的布局效果我们让元素生成BFC。

对于CSS新手来说不建议涉猎BFC，还是应该去看看相应的CSS布局规则，像《CSS设计指南》、《CSS权威指南》这两本都很不错，达到一定积累再来看BFC，说不定会有一种豁然开朗的感觉。BFC中几乎涉及到CSS布局的所有重要属性，这也是BFC的难点和我们需要掌握BFC的意义所在。

19.CSS权重优先级是如何计算的？

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；

20.请解释一下为什么会出现浮动和什么时候需要清除浮动？清除浮动的方式

浮动的框可以向左或向右移动，直到它的外边缘碰到包含框或另一个浮动框的边框为止。由于浮动框不在文档的普通流中，所以文档的普通流中的块框表现得就像浮动框不存在一样。

在非IE浏览器（如Firefox）下，当容器的高度为auto，且容器的内容中有浮动（float为left或right）的元素，在这种情况下，容器的高度不能自动伸长以适应内容的高度，使得内容溢出到容器外面而影响（甚至破坏）布局的现象。这个现象叫浮动溢出，为了防止这个现象的出现而进行的CSS处理，就叫CSS清除浮动。

浮动元素引起的问题：

- （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。

清除浮动的几种方法：

- （1）通过在浮动元素的末尾添加一个空元素，设置clear：both属性。

（缺点：不过这个办法会增加额外的标签使HTML结构看起来不够简洁。）

- （2）通过设置父元素overflow：hidden/auto或者display：table或者为父元素添加float属性，或者给父元素定高。

- （3）父级div定义使用CSS的after伪元素：

原理：这种方法清除浮动是现在网上最拉风的一种清除浮动，他就是利用after和before来在元素内部插入两个元素块，从面达到清除浮动的效果。其实现原理类似于clear：both方法，只是区别在于clear在html插入一个div.clear标签，而clearfix利用其伪类clear：after在元素内部增加一个类似于div.clear的效果。下面来看看其具体的使用方法：

```
.clearfix {zoom:1;} /*==for IE6/7 Maxthon2==*/
```

```
.clearfix: after{clear: both; content: '.'; display: block; width: 0; height: 0; visibility: hidden;} /*==for FF/chrome/opera/IE8==*/
```

其中clear：both;指清除所有浮动；content：'.'; display：block;对于FF/chrome/opera/IE8不能缺少，其中content：可以取值也可以为空。

visibility：hidden;的作用是允许浏览器渲染它，但是不显示出来，这样才能实现清楚浮动。

解析原理：

- 1) display:block 使生成的元素以块级元素显示,占满剩余空间;
- 2) height:0 避免生成内容破坏原有布局的高度。
- 3) visibility:hidden 使生成的内容不可见，并允许可能被生成内容盖住的内容可以进行点击和交互;

4) 通过 `content: "."` 生成内容作为最后一个元素, 至于content里面是点还是其他都是可以的, 例如oocss里面就有经典的 `content: "."`, 有些版本可能content 里面内容为空, 一丝冰凉是不推荐这样做的, firefox直到7.0 `content: ""` 仍然会产生额外的空隙;

5) `zoom: 1` 触发IE hasLayout。

通过分析发现, 除了`clear: both`用来闭合浮动的, 其他代码无非都是为了隐藏掉content生成的内容, 这也就是其他版本的闭合浮动为什么会有`font-size: 0, line-height: 0`。

21.移动端的布局用过媒体查询吗?

[css3 media媒体查询器用法总结](#)

[CSS3 @media 查询](#)

22.使用 CSS 预处理器吗? 喜欢那个?

less (SASS、LESS没有本质区别, 只因为团队前端都是用的SASS)

23.CSS优化、提高性能的方法有哪些?

[CSS 优化、提高性能的方法有哪些?](#)

24.浏览器是怎样解析CSS选择器的?

按照从上到下,从右到左的顺序解析。

为什么排版引擎解析 CSS 选择器时一定要从右往左解析?

[CSS选择器从右向左的匹配规则](#)

25.在网页中的应该使用奇数还是偶数的字体? 为什么呢?

现在网页设计中的为什么少有人用 11px、13px、15px 等奇数的字体?

26.margin和padding分别适合什么场景使用?

margin是用来隔开元素与元素的间距; padding是用来隔开元素与内容的间隔;

margin用于布局分开元素使元素与元素互不相干;

padding用于元素与内容之间的间隔, 让内容(文字)与(包裹)元素之间有一段。

详解: [用Margin还是用Padding](#)

27.抽离样式模块怎么写, 说出思路, 有无实践经验? [阿里航旅的面试题]

一个有 15 个页面的项目怎么规范 css 样式?

28.元素竖向的百分比设定是相对于容器的高度吗?

(1) 对于height属性取值百分比, 是现对于容器高度的;

(2) 对于margin-top、margin-bottom、padding-top、padding-bottom这些竖直方向的内外边距属性的百分比取值, 参考的其实是容器的宽度而不是高低。

29.全屏滚动的原理是什么? 用到了CSS的那些属性?

[CSS-页面滚屏滚动原理](#)

30.什么是响应式设计? 响应式设计的基本原理是什么? 如何兼容低版本的IE?

[什么是响应式布局设计?](#)

[响应式Web设计中的IE6~8兼容性问题?](#)

31.视差滚动效果, 如何给每页做不同的动画? (回到顶部, 向下滑动要再次出现, 和只出现一次分别怎么做?)

两篇文章对其做了介绍:

[如何实现视差滚动效果的网页](#)

[视差滚动效果原理解析和效果实现](#)

32.::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CSS2伪类, 双冒号(::)用于CSS3伪元素。(伪元素由双冒号和伪元素名称组成);

双冒号是在当前规范中引入的, 用于区分伪类和伪元素。不过浏览器需要同时支持旧的已经存在的伪元素写法, 比如: `first-line`、`:first-`

letter、:before、:after等，而新的在CSS3中引入的伪元素则不允许再支持旧的单冒号的写法。

对于CSS2之前已有的伪元素，比如:before，单冒号和双冒号的写法::before作用是一样的。所以，如果你的网站只需要兼容webkit、firefox、opera等浏览器，建议对于伪元素采用双冒号的写法，如果不得不兼容IE浏览器，还是用CSS2的单冒号写法比较安全。

想让插入的内容出现在其它内容前，使用::before，否者，使用::after；

在代码顺序上，::after生成的内容也比::before生成的内容靠后。

如果按堆栈视角，::after生成的内容会在::before生成的内容之上

33.如何修改chrome记住密码后自动填充表单的黄色背景？

浏览器默认样式导致的，如下：

```
input:-webkit-autofill,
textarea:-webkit-autofill,
select:-webkit-autofill {
  background-color: rgb(250, 255, 189); /* #FAFFBD; */
  background-image: none;
  color: rgb(0, 0, 0);
}
```

(1) 设置>，但是无法自动记录表单；

(2) hack:

```
input:-webkit-autofill,
textarea:-webkit-autofill,
select:-webkit-autofill {
  -webkit-box-shadow: 0 0 0 1000px white inset;
}
```

34.你对line-height是如何理解的？

张鑫旭css行高line-height的一些深入理解及应用

35.设置元素浮动后，该元素的display值是多少？

自动变成display:block

36.怎么让Chrome支持小于12px 的文字？

(1) 用图片：如果是内容固定不变情况下，使用将小于12px文字内容切出做图片，这样不影响兼容也不影响美观。

(2) 使用12px及12px以上字体大小：为了兼容各大主流浏览器，建议设计美工图时候设置大于或等于12px的字体大小，如果是接单的这个时候就需要给客户讲解小于12px浏览器不兼容等事宜。

(3) 继续使用小于12px字体大小样式设置：如果不考虑chrome可以不用考虑兼容，同时在设置小于12px对象设置-webkit-text-size-adjust:none，做到最大兼容考虑。

(4) 使用12px以上字体：为了兼容、为了代码更简单 从新考虑权重下兼容性。

37.让页面里的字体变清晰，变细用CSS怎么做？

-webkit-font-smoothing: antialiased;

字体平滑,该属性定义文本的平滑属性。有关字体平滑的介绍可参考字体渲染一文。目前该属性已从W3C标准中移除，慎用！

-webkit-font-smoothing

38.font-style属性可以让它赋值为“oblique” oblique是什么意思？

倾斜的字体样式

39.position:fixed;在android下无效怎么处理？

fixed的元素是相对整个页面固定位置的，你在屏幕上滑动只是在移动这个所谓的viewport，原来的网页还好好地在那，fixed的内容也没有变过位置，所以说并不是iOS不支持fixed，只是fixed的元素不是相对手机屏幕固定的。

```
<meta name="viewport" content="width=device-width, initial-scale=1.0, maximum-scale=1.0, minimum-scale=1.0, user-scalable=no"/>
```

40.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 $1/60 * 1000\text{ms} = 16.7\text{ms}$ ；

41.如何清除display:inline-block 产生的间隙？(携程)

inline-block水平呈现的元素间，换行显示或空格分隔的情况下会有间距；
移除空格、使用margin负值、使用font-size:0、letter-spacing、word-spacing
去除inline-block元素间间距的N种方法

42.overflow: scroll时不能平滑滚动的问题怎么处理？

<http://www.jianshu.com/p/1f4693d0ad2d>

43.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。

segmentfault上的解决方案

44.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？

JPG格式也叫JPEG格式是数码相机用于储存相片的常用格式，现在也是QQ空间、新闻照片中被广泛使用的图片格式。其特点是色彩还原性好，可以在照片不明显失真的情况下，大幅度压缩图片文件，所以储存体积不是很大。其缺点是不像PNG那样支持图片的透明度。照片库，例如QQ相册，都是采用JPG有损压缩的形式储存相片。如果你想要在网页上张贴出你的拍摄的风光照片，最好都用JPG格式。

GIF 算是比较老的图片格式了，它的色彩效果最低，如果用GIF压缩储存色彩丰富的风光照片会让你的网站看上去很吓人。但是GIF有着不可忽视的特点：体积小，有着极好的压缩效果，支持动画，并且支持透明效果，但是不能像PNG那样让图片在被放大后仍然保持透明。如果你的图片只有很单调的色彩，没有渐变色，例如只有红蓝两色，那么选GIF最好不过了！例如百度的首页LOGO就是很好的例子，它的大小只有2KB，节省了资源。

可谓是最适合网络图片！PNG的优点是，清晰，无损压缩，压缩比率很高，可渐变透明，具备几乎所有GIF的优点；缺点是不如JPG的颜色丰富，同样的图片体积也比JPG略大。但是PNG应该在网站设计上被推广，它是公认的最适合网页使用的图片格式。Google就是一个很好的例子。Google所有站点几乎全部的图片资源都是PNG格式。8位的PNG完全可以替代掉GIF。

PNG有着另一个优点，那就是逐次逼近显示(progressive display)：传输图像文件的同时，可以先把整个轮廓显示出来，然后逐步显示图像的细节，即先显示低分辨率显示图像，从模糊到清晰，然后逐步提高它的分辨率。这是一个很好的用户体验。

对于一个网站来说，按照保证色彩显示良好同时文件最小的原则，我的总结是这样的：

网站的LOGO，如果色彩单调，选择PNG或者GIF，推荐前者。但是如果为了IE6的兼容，选择后者；

如果你需要动图，选择GIF；

网站上的元素背景图片（比如按钮背景、导航条背景），或者很精细的图片，还是那句话，如果色彩单调，选择PNG或者GIF，推荐前者。但是如果为了IE6的兼容，选择后者，但要考虑失真问题；少用JPG，否则会让这些东西很难看，除非你用风景图片作按钮背景；

网站内容里，你真实拍摄的图片，或者你下载的风景图片，保持JPG来获得更好的显示效果和体积；

截图都应该用PNG。当然，如果你并不追求非常完美的显示效果，只是注重内容本身而容许图片上的小瑕疵，选择JPG也可。

写在最后

不论你用哪种图片，你都应该考虑到图片的压缩性，并对图片尽可能的无损压缩。单纯想要压缩图片，例如ImageOptim For Mac就是一个很好的压缩工具。不过大多数图片编辑软件，如Photoshop、GIMP都提供了图片压缩功能。

此外，如果你的图片非常小，你甚至可以考虑Base64。

2010年谷歌推迟的图片格式,专门用来在web中使用,压缩率只有jpg的2/3或者更低；第一个版本的webp图片格式是有损的,新版本中webp图片是无损的。

相对于png图片,webp比png小了45%,但是缺点是你压缩的时候需要的时间更久了；

Webp：WebP格式，谷歌（google）开发的一种旨在加快图片加载速度的图片格式。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%。

Apng：全称是“Animated Portable Network Graphics”，是PNG的位图动画扩展，可以实现png格式的动态图片效果。04年诞生，但一直得不到各大浏览器厂商的支持，直到日前得到 iOS safari 8的支持，有望代替GIF成为下一代动态图标准。

详见：[GIF/PNG/JPG/WEBP/base64/apng 格式图片的优点和缺点整理](#)

45.什么是Cookie 隔离？（或者说：请求资源的时候不要让它带cookie怎么做）

cookie隔离技术和传统的多域名拆分请求，提高浏览器并发请求数有点类似，均是采用多域名来处理请求。

传统做法是将css, js, 图片等静态文件放在多个域名下面请求，这样就可以跨过浏览器对统一主机名并发连接数的限制，提高整体并发请求量。

cookie隔离技术则是通过使用多个非主要域名来请求静态文件，如果静态文件都放在主域名下，那静态文件请求的时候带有的cookie的数据提交给server是非常浪费的，还不如隔离开。

因为cookie有域的限制，因此不能跨域提交请求，故使用非主要域名的时候，请求头中就不会带有cookie数据，这样可以降低请求头的大小，降低请求时间，从而达到降低整体请求延时的目的。同时这种方式不会将cookie传入webserver，也减少了webserver对cookie的处理分析环节，提高了webserver的http请求的解析速度。

46.style标签写在body后与body前有什么区别？

写在head标签中利于浏览器逐步渲染（resources downloading->CSSOM+DOM->RenderTree(composite)->Layout->paint）。具体渲染过程请参考[浏览器的渲染机制](#)

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）。

47.什么是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以及代码压缩。

48.display:none和visibility:hidden的区别？

display:none 隐藏对应的元素，在文档布局中不再给它分配空间，它各边的元素会合拢，就当它从来不存在。

visibility:hidden 隐藏对应的元素，但是在文档布局中仍保留原来的空间。

49. position的absolute与fixed共同点与不同点？

共同点：

- (1)改变行内元素的呈现方式，display被置为block；
- (2)让元素脱离普通流，不占据空间；
- (3)默认会覆盖到非定位元素上。

不同点：

absolute的“根元素”是可以设置的，而fixed的“根元素”固定为浏览器窗口。（absolute相对是第一个有定位的祖元素进行定位，fixed相对于浏览器进行定位。）当你滚动网页，fixed元素与浏览器窗口之间的距离是不变的。

50.解释下CSS sprites，以及你要如何在页面或网站中使用它。

CSS Sprites其实就是把网页中一些背景图片整合到一张图片文件中，再利用CSS的“background-image”，“background-repeat”，“background-position”的组合进行背景定位，background-position可以用数字能精确的定位出背景图片的位置。

51.行内元素的padding和margin可设置吗？

宽度(width)、高度(height)、内边距的top/bottom(padding-top/padding-bottom)和外边距的top/bottom(margin-top/margin-bottom)都不可改变（也就是padding和margin的left和right是可以设置的），就是里面文字或图片的大小。

那么问题来了，浏览器还有默认的天生inline-block元素（拥有内在尺寸，可设置高宽，但不会自动换行），有哪些？

```
<input> 、 <img> 、 <button> 、 <textarea> 、 <label>
```

52.请用Css写一个简单的幻灯片效果页面

53.什么是Css Hack？ie6,7,8的hack分别是什么？

针对不同的浏览器写不同的CSS code的过程，就是CSS hack。

```
1 #test {
2     width:300px;
3     height:300px;
4
5     background-color:blue;    /*firefox*/
```

```
6 background-color:red\9; /*all ie*/
7 background-color:yellow\0; /*ie8*/
8 +background-color:pink; /*ie7*/
9 _background-color:orange; /*ie6*/ }
10 :root #test { background-color:purple\9; } /*ie9*/

11 @media all and (min-width:0px){ #test {background-color:black\0;} } /*opera*/
12 @media screen and (-webkit-min-device-pixel-ratio:0){ #test {background-color:gray;} } /*chrome and safari*/
```

54.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？

最基本的：设置display属性为none，或者设置visibility属性为hidden;

技巧性：设置宽高为0，设置透明度为0，设置z-index位置在-1000;

55.什么是外边距重叠？重叠的结果是什么？

外边距重叠就是margin-collapse。

在CSS当中，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被称为折叠，并且因而所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：

两个相邻的外边距都是正数时，折叠结果是它们两者之间较大的值。

两个相邻的外边距都是负数时，折叠结果是两者绝对值的较大值。

两个外边距一正一负时，折叠结果是两者的相加的和。

56.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！）

56.css中可以让文字在垂直和水平方向上重叠的两个属性是什么？

垂直方向：line-height

水平方向：letter-spacing

那么问题来了，关于letter-spacing的妙用知道有哪些么？

可以用于消除inline-block元素间的换行符空格间隙问题。

56.px和em的区别。

px和em都是长度单位，区别是，px的值是固定的，指定是多少就是多少，计算比较容易。em得值不是固定的，并且em会继承父级元素的字体大小。浏览器的默认字体高都是16px。所以未经调整的浏览器都符合：1em=16px。那么12px=0.75em, 10px=0.625em。

57.为什么利用多个域名来存储网站资源会更有效？

- CDN缓存更方便
- 突破浏览器并发限制
- 节约cookie带宽
- 节约主域名的连接数，优化页面响应速度
- 防止不必要的安全问题

58.在css/js代码上线之后开发人员经常会优化性能，从用户刷新网页开始，一次js请求一般情况下有哪些地方会有缓存处理？

dns缓存，cdn缓存，浏览器缓存，服务器缓存

59.一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，在页面上的未可视区域可以添加一个滚动条事件，判断图片位置与浏览器顶端的距离与页面的距离，如果前者小于后者，优先加载。
- 如果为幻灯片、相册等，可以使用图片预加载技术，将当前展示图片的前一张和后一张优先下载。
- 如果图片为css图片，可以使用CSSsprite，SVGsprite，Iconfont、Base64等技术。
- 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。
- 如果图片展示区域小于图片的真实大小，则因在服务器端根据业务需要先行进行图片压缩，图片压缩后大小与展示一致。

60.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)

61.知道css有个content属性吗？有什么作用？有什么应用？

css的content属性专门应用在 before/after 伪元素上，用于来插入生成内容。
最常见的应用是利用伪类清除浮动。

```
1 //一种常见利用伪类清除浮动的代码
2 .clearfix:after {
3     content:".";          //这里利用到了content属性
4     display:block;
5     height:0;
6     visibility:hidden;
7     clear:both; }
8
9 .clearfix {
10     *zoom:1;
11 }
```

after伪元素通过 content 在元素的后面生成了内容为一个点的块级元素，再利用clear:both清除浮动。

62.知道css计数器（序列数字字符自动递增）吗？如何通过css content属性实现css计数器？

css计数器是通过设置counter-reset、counter-increment 两个属性、及 counter()/counters()一个方法配合after / before 伪类实现。
使用CSS计数器之前，必须重置一个值，默认是0。使用counter()函数来给元素增加计数器。下面的CSS给每个h3元素的前面增加了"Section <计算器值>:"。

```
body {
    counter-reset: section;          /* 重置计数器成0 */
}
h3:before {
    counter-increment: section;      /* 增加计数器值 */
    content: "Section " counter(section) ": "; /* 显示计数器 */
}
```

bfore:

```
<h3>Introduction</h3>
<h3>Body</h3>
<h3>Conclusion</h3>
```

after:

```
Section 1: Introduction
Section 2: Body
Section 3: Conclusion
```