

使用 `font-size:0`  
`letter-spacing`  
`word-spacing`

## 17 PNG\GIF\JPG的区别及如何选

- GIF
  - 8 位像素, 256 色
  - 无损压缩
  - 支持简单动画
  - 支持 `boolean` 透明
  - 适合简单动画
- JPEG
  - 颜色限于 256
  - 有损压缩
  - 可控制压缩质量
  - 不支持透明
  - 适合照片
- PNG
  - 有 `PNG8` 和 `truecolor PNG`
  - `PNG8` 类似 `GIF` 颜色上限为 256 , 文件小, 支持 `alpha` 透明度, 无动画
  - 适合图标、背景、按钮

## 18 行内元素`float:left`后是否变为块级元素?

行内元素设置成浮动之后变得更加像是 `inline-block` (行内块级元素, 设置成这个属性的元素会同时拥有行内和块级的特性, 最明显的不同是它的默认宽度不是 100% ), 这时候给行内元素设置 `padding-top` 和 `padding-bottom` 或者 `width` 、 `height` 都是有效果的

## 19 在网页中的应该使用奇数还是偶数的字体? 为什么呢?

- 偶数字号相对更容易和 web 设计的其他部分构成比例关系

## 20 ::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
- 用于区分伪类和伪元素

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

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

## 22 CSS合并方法

- 避免使用 @import 引入多个 css 文件， 可以使用 CSS 工具将 CSS 合并为一个 CSS 文件，例如使用 Sass\Compass 等

## 23 CSS不同选择器的权重(CSS层叠的规则)

- ! important 规则最重要，大于其它规则
- 行内样式规则，加 1000
- 对于选择器中给定的各个 ID 属性值，加 100
- 对于选择器中给定的各个类属性、属性选择器或者伪类选择器，加 10
- 对于选择其中给定的各个元素标签选择器，加1
- 如果权值一样，则按照样式规则的先后顺序来应用，顺序靠后的覆盖靠前的规则

## 24 列出你所知道可以改变页面布局的属性

- position 、 display 、 float 、 width 、 height 、 margin 、 padding 、 top 、 left 、 right 、 、

## 25 CSS在性能优化方面的实践

- css 压缩与合并、 Gzip 压缩
- css 文件放在 head 里、不要用 @import
- 尽量用缩写、避免用滤镜、合理使用选择器

## 26 CSS3动画（简单动画的实现，如旋转等）

- 依靠 CSS3 中提出的三个属性： `transition` 、 `transform` 、 `animation`
- `transition` ：定义了元素在变化过程中是怎么样，包含 `transition-property` 、 `transition-duration` 、 `transition-timing-function` 、 `transition-delay` 。
- `transform` ：定义元素的变化结果，包含 `rotate` 、 `scale` 、 `skew` 、 `translate` 。
- `animation` ：动画定义了动作的每一帧（`@keyframes`）有什么效果，包括 `animation-name` , `animation-duration` 、 `animation-timing-function` 、 `animation-delay` 、 `animation-iteration-count` 、 `animation-direction`

## 27 base64的原理及优缺点

- 优点可以加密，减少了 HTTP 请求
- 缺点是需要消耗 CPU 进行编解码

## 28 几种常见的CSS布局

### 流体布局

```
.left {  
    float: left;  
    width: 100px;  
    height: 200px;  
    background: red;  
}  
.right {  
    float: right;  
    width: 200px;  
    height: 200px;  
    background: blue;  
}  
.main {  
    margin-left: 120px;  
    margin-right: 220px;  
    height: 200px;  
    background: green;  
}
```

CSS

html

```
<div class="container">
  <div class="left"></div>
  <div class="right"></div>
  <div class="main"></div>
</div>
```

## 圣杯布局

CSS

```
.container {
  margin-left: 120px;
  margin-right: 220px;
}
.main {
  float: left;
  width: 100%;
  height: 300px;
  background: green;
}
.left {
  position: relative;
  left: -120px;
  float: left;
  height: 300px;
  width: 100px;
  margin-left: -100%;
  background: red;
}
.right {
  position: relative;
  right: -220px;
  float: right;
  height: 300px;
  width: 200px;
  margin-left: -200px;
  background: blue;
}
```

html

```
<div class="container">
  <div class="main"></div>
  <div class="left"></div>
  <div class="right"></div>
</div>
```

## 双飞翼布局

```
.content {  
    float: left;  
    width: 100%;  
}  
.main {  
    height: 200px;  
    margin-left: 110px;  
    margin-right: 220px;  
    background: green;  
}  
.main::after {  
    content: '';  
    display: block;  
    font-size: 0;  
    height: 0;  
    zoom: 1;  
    clear: both;  
}  
.left {  
    float: left;  
    height: 200px;  
    width: 100px;  
    margin-left: -100%;  
    background: red;  
}  
.right {  
    float: right;  
    height: 200px;  
    width: 200px;  
    margin-left: -200px;  
    background: blue;  
}
```

CSS

```
<div class="content">  
    <div class="main"></div>  
</div>  
<div class="left"></div>  
<div class="right"></div>
```

html

## 29 stylus/sass/less区别

- 均具有“变量”、“混合”、“嵌套”、“继承”、“颜色混合”五大基本特性
- `Scss` 和 `LESS` 语法较为严谨，`LESS` 要求一定要使用大括号“{}”，`Scss` 和 `Stylus` 可以通过缩进表示层次与嵌套关系
- `Scss` 无全局变量的概念，`LESS` 和 `Stylus` 有类似于其它语言的作用域概念
- `Sass` 是基于 `Ruby` 语言的，而 `LESS` 和 `Stylus` 可以基于 `NodeJS` `NPM` 下载相应库后进行编译；

## 30 postcss的作用

- 可以直观的理解为：它就是一个平台。为什么说它是一个平台呢？因为我们直接用它，感觉不能干什么事情，但是如果让一些插件在它上面跑，那么将会很强大
- `PostCSS` 提供了一个解析器，它能够将 `CSS` 解析成抽象语法树
- 通过在 `PostCSS` 这个平台上，我们能够开发一些插件，来处理我们的 `CSS`，比如热门的：`autoprefixer`
- `postcss` 可以对`sass`处理过后的 `css` 再处理 最常见的就是 `autoprefixer`

## 31 css样式（选择器）的优先级

- 计算权重确定
- `!important`
- 内联样式
- 后写的优先级高

## 32 自定义字体的使用场景

- 宣传/品牌/ `banner` 等固定文案
- 字体图标

## 33 如何美化CheckBox

- `<label>` 属性 `for` 和 `id`
- 隐藏原生的 `<input>`
- `:checked + <label>`

## 34 伪类和伪元素的区别

- 伪类表状态
- 伪元素是真的有元素
- 前者单冒号，后者双冒号

## 35 base64 的使用

- 用于减少 HTTP 请求
- 适用于小图片
- base64 的体积约为原图的 4/3

## 36 自适应布局

思路：

- 左侧浮动或者绝对定位，然后右侧 margin 撑开
- 使用 <div> 包含，然后靠负 margin 形成 bfc
- 使用 flex

## 37 请用CSS写一个简单的幻灯片效果页面

知道是要用 CSS3 。使用 animation 动画实现一个简单的幻灯片效果

```
/**css**/  
.ani{  
  width:480px;  
  height:320px;  
  margin:50px auto;  
  overflow: hidden;  
  box-shadow:0 0 5px rgba(0,0,0,1);  
  background-size: cover;  
  background-position: center;  
  -webkit-animation-name: "loops";  
  -webkit-animation-duration: 20s;  
  -webkit-animation-iteration-count: infinite;  
}
```

CSS