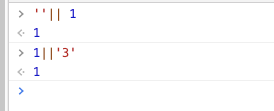
**1.**[**jQuery选择器的返回值**](http://nferzhuang.com/jquery%E9%80%89%E6%8B%A9%E5%99%A8%E7%9A%84%E8%BF%94%E5%9B%9E%E5%80%BC/)

jQuery选择器的返回值并不是一个js的Array数组

**2.一些技巧**

**1. ||**



**2.&&**

f && f()

**3.**[**Git-命令行-删除本地和远程分支**](https://blog.csdn.net/qq_32452623/article/details/54340749)

**4.**[**git同步远程已删除的分支和删除本地多余的分支**](https://www.cnblogs.com/diffx/p/10898504.html)

**5.**[**git命令之git fetch的用法**](https://blog.csdn.net/QH_JAVA/article/details/77969010)

**6.**[**HTML5 移动端自适应方案与踩坑**](https://juejin.im/post/5c8870c1e51d453ce668cc7f)

**7.**[**如何获取网页默认字体的字号大小?**](https://segmentfault.com/q/1010000008605788)

字体大小不仅是由行内style属性决定，还取决于外部link引入的样式，style标签里写的样式，还有继承的样式，浏览器默认样式。

你使用document.getElementsByTagName("p")[0].style获取的仅仅是style标签的样式，因为你没有设定，所以得到的是空。

如果你要得到计算后的样式，可以同getComputedStyle(document.getElementsByTagName("p")[0],undefined)来得到全部的最终渲染样式。

getComputedStyle(document.getElementsByTagName("p")[0],undefined).fontSize

**8.H5移动端屏幕自适应原理**

根节点字体大小 RFS，屏幕宽度 SW，浏览器默认字体大小DFS，对html font-size的缩放比rate = RFS/DFS，使用的长度a rem = p / RFS rem。选择一个标准：iPhone6，SW0=375px，RFS0设为50px。

自适应原理，内容随屏幕宽度进行变换。即当SW变换时，通过改变rate， 让 p进行变换，而a不变。SW1 / SW0 = p1 / p0 = (a \* RFS1) / (a \* RFS0) = RFS1 / RFS0 = rate1 / rate0。

一般直接给html的font-size设个百分数rate1 = SW1 / SW0 \* rate0 = SW1 / SW0 \* RFS0 / DFS. （DFS是无法改变的）

**8. png & jpg**

PNG是[无损压缩](https://www.baidu.com/s?wd=%E6%97%A0%E6%8D%9F%E5%8E%8B%E7%BC%A9&tn=SE_PcZhidaonwhc_ngpagmjz&rsv_dl=gh_pc_zhidao)，保存了原始图片的完整数据，JPG是[有损压缩](https://www.baidu.com/s?wd=%E6%9C%89%E6%8D%9F%E5%8E%8B%E7%BC%A9&tn=SE_PcZhidaonwhc_ngpagmjz&rsv_dl=gh_pc_zhidao)，只是大概看上去和原始图像一样，其实损失了一些细节，或者多了噪点等等，损失的程度主要看压缩的质量，文件越小质量越低，损失就越大。

还有，PNG和JPG所采用的[色彩空间](https://www.baidu.com/s?wd=%E8%89%B2%E5%BD%A9%E7%A9%BA%E9%97%B4&tn=SE_PcZhidaonwhc_ngpagmjz&rsv_dl=gh_pc_zhidao)不一样，PNG大部分应该都是RGB[色彩空间](https://www.baidu.com/s?wd=%E8%89%B2%E5%BD%A9%E7%A9%BA%E9%97%B4&tn=SE_PcZhidaonwhc_ngpagmjz&rsv_dl=gh_pc_zhidao)（或者RGBA，多一个[透明通道](https://www.baidu.com/s?wd=%E9%80%8F%E6%98%8E%E9%80%9A%E9%81%93&tn=SE_PcZhidaonwhc_ngpagmjz&rsv_dl=gh_pc_zhidao)），如果不压缩的话，每一个像素要占24位或32位或者更高，而JPEG大部分使用的yuv[色彩空间](https://www.baidu.com/s?wd=%E8%89%B2%E5%BD%A9%E7%A9%BA%E9%97%B4&tn=SE_PcZhidaonwhc_ngpagmjz&rsv_dl=gh_pc_zhidao)本来就比RGB占的空间少

不过也并不是所有的JPG都比PNG小，比如纯色图像，或者颜色比较少，比较不复杂的图像，PNG就可以比JPG小，这个可以扯到压缩算法上，我不懂不能乱讲……

**9.移动端使用图片尺寸问题**

一般手机端以375px的宽度为准，所以一般用750px宽度的图片就可以了

**10.iview form表单的一些坑**

props: {
order: {
type: Object,
required: true,
}
},
data() {
return {
formItem: {
user: {
uname: '',
user\_id: '',
mobile: '',
...this.order.user,
},
orderId: this.order.order\_id,
...
ruleValidate: {
refundAmount: [
{
required: true,
message: '请输入正确的退款金额',
type: 'integer',
min: 1,
max: this.order.money
},
],

如果props里order的值会改变，则不应该让formItem的值初始化为order的值，也不应该在ruleValidate里使用order的值

1.当调用resetFields时，会将formItem的值重置为初始化的值而不是新的order的值；

2.当调用validate时，会根据初始化时ruleValidate的值来校验，而不是新的order的值。