

PNG 图片压缩原理解析

png 的图片我们每天都在用，可是 png 到底是什么，它的压缩原理是什么？很好，接下来我将会给大家一一阐述。

什么是 PNG

PNG 的全称叫便携式网络图型（Portable Network Graphics）是目前最流行的网络传输和展示的图片格式，原因有如下几点：

无损压缩：PNG 图片采取了基于 LZ77 派生算法对文件进行压缩，使得它压缩比率更高，生成的文件体积更小，并且不损失数据。

体积小：它利用特殊的编码方法标记重复出现的数据，使得同样格式的图片，PNG 图片文件的体积更小。网络通讯中因受带宽制约，在保证图片清晰、逼真的前提下，优先选择 PNG 格式的图片。

支持透明效果：PNG 支持对原图像定义 256 个透明层次，使得图像的边缘能与任何背景平滑融合，这种功能是 GIF 和 JPEG 没有的。

PNG 类型

PNG 图片主要有三个类型，分别为 PNG 8/ PNG 24 / PNG 32。

PNG8：PNG 8 中的 8，其实指的是 8bits，相当于用 2^8 （2 的 8 次方）大小来存储一张图片的颜色种类， 2^8 等于 256，也就是说 PNG 8 能存储 256 种颜色，一张图片如果颜色种类很少，将它设置成 PNG 8 得图片类型是非常适合的。

PNG24：PNG 24 中的 24，相当于 3 乘以 8 等于 24，就是用三个 8bits 分别去表示 R（红）、G（绿）、B（蓝）。R(0~255),G(0~255),B(0~255)，可以表达 256 乘以 256 乘以 256=16777216 种颜色的图片，这样 PNG 24 就能比 PNG 8 表示色彩更丰富的图片。但是所占用的空间相对就更大了。

PNG32：PNG 32 中的 32，相当于 PNG 24 加上 8bits 的透明颜色通道，就相当于 R（红）、G（绿）、B（蓝）、A（透明）。R(0~255),G(0~255),B(0~255),A(0~255)。比 PNG 24 多了一个 A（透明），也就是说 PNG 32 能表示跟 PNG 24 一样多的色彩，并且还支持 256 种透明的颜色，能表示更加丰富的图片颜色类型。

怎么说呢，总的来说，PNG 8/ PNG 24 / PNG 32 就相当于我们屌丝心中，把女神分为三类：

一类女神=PNG8：屌丝舔狗们见到第一类女神，顿时会觉得心情愉悦、笑逐颜开，屌丝发黑的印堂逐渐舒展，确认过眼神，是心动的感觉。

二类女神=PNG24：第二类女神开始厉害了，会给屌丝们一种菊花一紧、振聋发聩的心弦震撼，接触多了第二类女神能让屌丝每天精神抖擞，延年益寿。

三类女神=PNG32：在第三类女神面前，所有的语言都显得苍白无力。那是一种看了让屌丝上下通透、手眼通天的至尊级存在。超凡脱俗、天神下凡都不足以描摹她美色的二分之一。我曾经只有在梦里才见到过。

哎。。。我的初恋，看着她现在的照片，应该是触及 PNG 24 这一等级了。

PNG 图片数据结构

PNG 图片的数据结构其实跟 http 请求的结构很像，都是一个数据头，后面跟着很多的数据块，如下图所示：



如果你用 vim 的查看编码模式打开一张 png 图片，会是下面这个样子：

```
2. vi -b output_1.png (vim)
00000000: 8950 4e47 0d0a 1a0a 0000 000d 4948 4452 .PNG.....IHDR
00000010: 0000 02bc 0000 03a5 0802 0000 0000 76a3 .....v.
00000020: f300 0000 0970 4859 7300 000e c400 000e ....pHYs.....
00000030: c401 952b 0e1b 0000 2000 4944 4154 7801 ...+....IDATx.
00000040: 5cdd e793 245d 76df f7ee aeb6 639f 3500 \...$]v.....c.5.
00000050: 2188 1400 c204 2228 12a4 0811 6e61 0990 !....."(....na..
00000060: 1415 4185 def0 0f97 420a 4a94 b804 b0bb ..A.....B.J.....
00000070: 58ec 3e66 66da 7755 75eb f33d b766 8150 X.>ff.wUu...=.f.P
```

握草，第一眼看到这一坨坨十六进制编码是不是感觉和女神的心思一样晦涩难懂？

老弟 莫慌，讲实话，如果撩妹纸有那一坨坨乱码那么简单，哥哥我早就妻妾成群啦。

接下来我就一一讲解这一堆十六进制编码的含义。

89504e470d0a1a0a: 这个是 PNG 图片的头，所有的 PNG 图片的头都是这一串编码，图片软件通过这串编码判定这个文件是不是 PNG 格式的图片。

0000000d: 是 iHDR 数据块的长度，为 13。

49484452: 是数据块的 type,为 IHDR，之后紧跟着是 data。

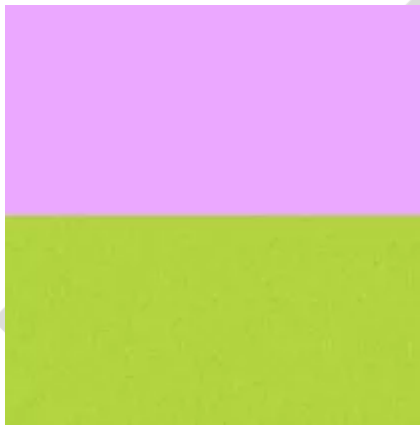
000002bc: 是图片的宽度。

000003a5: 是高度。

以此类推，每一段十六进制编码就代表着一个特定的含义。下面其他的就不一一分析了，太多了，小伙伴们自己去查吧。

什么样的 PNG 图片更适合压缩

常规的 png 图片，颜色越单一，颜色值越少，压缩率就越大，比如下面这张图：



它仅仅由红色和绿色构成，如果用 0 代表红色，用 1 代表绿色，那用数字表示这张图就是下面这个样子：

000000000000000000

000000000000000000

000000000000000000

11111111111111111111

11111111111111111111

11111111111111111111

我们可以看到，这张图片是用了大量重复的数字，我们可以将重复的数字去掉，直接用数组形式的[0, 1]就可以直接表示出这张图片了，仅仅用两个数字，就能表示出一张很大的图片，这样就极大的压缩了一张 png 图片。

所以！颜色越单一，颜色值越少，颜色差异越小的 png 图片，压缩率就越大，体积就越小。

PNG 的压缩

PNG 图片的压缩，分两个阶段：

预解析 (Prediction)：这个阶段就是对 png 图片进行一个预处理，处理后让它更方便后续的压缩。说白了，就是一个女神，在化妆前，会先打底，先涂乳液和精华，方便后续上妆、美白、眼影、打光等等。

压缩 (Compression)：执行 Deflate 压缩，该算法结合了 LZ77 算法和 Huffman 算法对图片进行编码。

预解析 (Prediction)

png 图片用差分编码 (Delta encoding) 对图片进行预处理，处理每一个的像素点中每条通道的值，差分编码主要有几种：

- 不过滤
- X-A
- X-B
- $X-(A+B)/2$ (又称平均值)
- Paeth 推断 (这种比较复杂)

假设，一张 png 图片如下：



这张图片是一个红色逐渐增强的渐变色图，它的红色从左到右逐渐加强，映射成数组的值为[1,2,3,4,5,6,7,8]，使用 X-A 的差分编码的话，那就是：

[2-1=1, 3-2=1, 4-3=1, 5-4=1, 6-5=1, 7-6=1, 8-7=1]

得到的结果为

[1,1,1,1,1,1,1]

最后的[1,1,1,1,1,1,1]这个结果出现了大量的重复数字，这样就非常适合进行压缩。

这就是为什么渐变色图片、颜色值变化不大并且颜色单一的图片更容易压缩的原理。

差分编码的目的，就是尽可能的将 png 图片数据值转换成一组重复的、低值，这样的值更容易被压缩。

最后还要注意的，差分编码处理的是每一个的像素点中每条颜色通道的值，R（红）、G（绿）、B（蓝）、A（透明）四个颜色通道的值分别进行处理。

压缩 (Compression)

压缩阶段会将预处理阶段得到的结果进行 **Deflate** 压缩，它由 **Huffman** 编码和 **LZ77** 压缩构成。

如前面所说，**Deflate** 压缩会标记图片所有的重复数据，并记录数据特征和结构，会得到一个压缩比最大的 **png** 图片 编码数据。

Deflate 是一种压缩数据流的算法。任何需要流式压缩的地方都可以用。

还有就是我们前面说过，一个 **png** 图片，是由很多的数据块构成的，但是数据块里面的一些信息其实是没有用的，比如用 **Photoshop** 保存了一张 **png** 图片，图片里就会有一个区块记录“这张图片是由 **photoshop** 创建的”，很多类似这些信息都是无用的，如果用 **photoshop** 的“导出 **web** 格式”就能去掉这些无用信息。导出 **web** 格式前后对比效果如下图所示：



可以看到，导出 **web** 格式，去除了很多无用信息后，图片明显小了很多。