

# JavaScript 如何检测文件的类型？

原创 阿宝哥 全栈修仙之路 4月29日



全栈修仙之路  
专注分享 TS、Vue3、前端架构和源码解析等技术干货。  
125篇原创内容

公众号

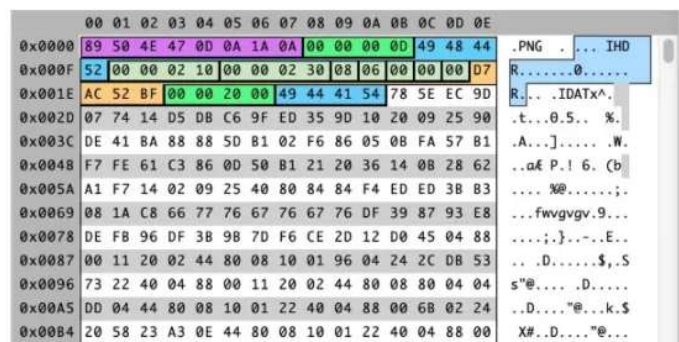
在日常工作中，文件上传是一个很常见的功能。在某些情况下，我们希望能限制文件上传的类型，比如限制只能上传 PNG 格式的图片。针对这个问题，我们会想到通过 `input` 元素的 `accept` 属性来限制上传的文件类型：

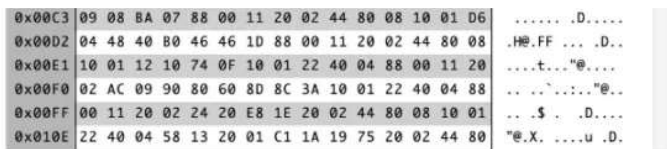
```
<input type="file" id="inputFile" accept="image/png" />
```

这种方案虽然可以满足大多数场景，但如果用户把 JPEG 格式的图片后缀名更改为 `.png` 的话，就可以成功突破这个限制。那么应该如何解决这个问题呢？其实我们可以通过读取文件的二进制数据来识别正确的文件类型。在介绍具体的实现方案前，阿宝哥先以图片类型的文件为例，来介绍一下相关的知识。

## 一、如何查看图片的二进制数据

要查看图片对应的二进制数据，我们可以借助一些现成的编辑器，比如 Windows 平台下的 **WinHex** 或 macOS 平台下的 **Synalyze It! Pro** 十六进制编辑器。这里我们使用 **Synalyze It! Pro** 这个编辑器，以十六进制的形式来查看阿宝哥头像对应的二进制数据。





计算机并不是通过图片的后缀名来区分不同的图片类型，而是通过“魔数”（**Magic Number**）来区分。对于某一些类型的文件，起始的几个字节内容都是固定的，根据这几个字节的内容就可以判断文件的类型。

文件类型	文件后缀	魔数
JPEG	jpg/jpeg	0xFF D8 FF
PNG	png	0x89 50 4E 47 0D 0A 1A 0A
GIF	gif	0x47 49 46 38 (GIF8)
BMP	bmp	0x42 4D

Start	End	Length	Content
0x00	0x03	0x04	89 50 4E 47

由上图可知，PNG 类型的图片前 8 个字节是 **0x89 50 4E 47 0D 0A 1A 0A**。当你把 `abao.png` 文件修改为 `abao.jpeg` 后，再用编辑器打开查看图片的二进制内容，你会发现文件的前 8 个字节还是保持不变。但如果使用 `input[type="file"]` 输入框的方式来读取文件信息的话，将会输出以下结果：

```
▼ File ⓘ
  lastModified: 1591854920765
  ▶ lastModifiedDate: Thu Jun 11 2020 13:55:20 GMT+0800 (中国标准时间) {}
  name: "abao.jpeg"
  size: 46884
  type: "image/jpeg"
  webkitRelativePath: ""
  ▶ __proto__: File
```

很明显通过 文件后缀名或文件的 **MIME** 类型 并不能识别出正确的文件类型。接下来，阿宝哥将介绍在上传图片时，如何通过读取图片的二进制信息来确保正确的图片类型。

### 三、如何检测图片的类型

#### 3.1 定义 `readBuffer` 函数

在获取文件对象后，我们可以通过 `FileReader` API 来读取文件的内容。因为我们并不需要读取文件的完整信息，所以阿宝哥封装了一个 `readBuffer` 函数，用于读取文件中指定范围的二进制数据。

```
function readBuffer(file, start = 0, end = 2) {
  return new Promise((resolve, reject) => {
    const reader = new FileReader();
    reader.onload = () => {
      resolve(reader.result);
    };
    reader.onerror = reject;
    reader.readAsArrayBuffer(file.slice(start, end));
  });
}
```

对于 PNG 类型的图片来说，该文件的前 8 个字节是 **0x89 50 4E 47 0D 0A 1A 0A**。因此，我们在检测已选择的文件是否为 PNG 类型的图片时，只需要读取前 8 个字节的数据，并逐一判断每个字节的内容是否一致。

### 3.2 定义 `check` 函数

为了实现逐字节比对并能够更好地实现复用，阿宝哥定义了一个 `check` 函数：

```
function check(headers) {  
  return (buffers, options = { offset: 0 }) =>  
    headers.every(  
      (header, index) => header === buffers[options.offset + index]  
    );  
}
```

### 3.3 检测 PNG 图片类型

基于前面定义的 `readBuffer` 和 `check` 函数，我们就可以实现检测 PNG 图片的功能：

#### 3.3.1 html 代码

```
<div>  
  选择文件: <input type="file" id="inputFile" accept="image/*"  
    onchange="handleChange(event)" />  
  <p id="realFileType"></p>  
</div>
```

#### 3.3.2 JS 代码

```
const isPNG = check([0x89, 0x50, 0x4e, 0x47, 0x0d, 0x0a, 0x1a, 0x0a]); // PNG图片对应的魔数  
const realFileElement = document.querySelector("#realFileType");  
  
async function handleChange(event) {
```

```
const file = event.target.files[0];

const buffers = await readBuffer(file, 0, 8);

const uint8Array = new Uint8Array(buffers);

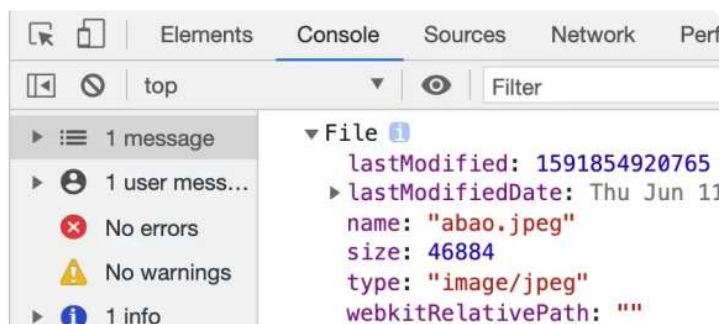
realFileElement.innerText = `${file.name}文件的类型是: ${
  isPNG(uint8Array) ? "image/png" : file.type
}`;
}
```

以上示例成功运行后，对应的检测结果如下图所示：

## JS 检测图片类型

选择文件：  abao.jpeg

abao.jpeg文件的类型是：image/png



由上图可知，我们已经可以成功地检测出正确的图片格式。如果你要检测 JPEG 文件格式的话，你只需要定义一个 `isJPEG` 函数：

```
const isJPEG = check([0xff, 0xd8, 0xff])
```

然而，如果你要检测其他类型的文件，比如 PDF 文件的话，应该如何处理呢？这里我们先使用 **Synalyze It! Pro** 编辑器来浏览一下 PDF 文件的二进制内容：



观察上图可知，PDF 文件的头 4 个字节的是 `0x25 50 44 46`，对应的字符串是 `%PDF`。为了让用户能更直观地辨别出检测的类型，阿宝哥定义了一个 `stringToBytes` 函数：

```
function stringToBytes(string) {
```



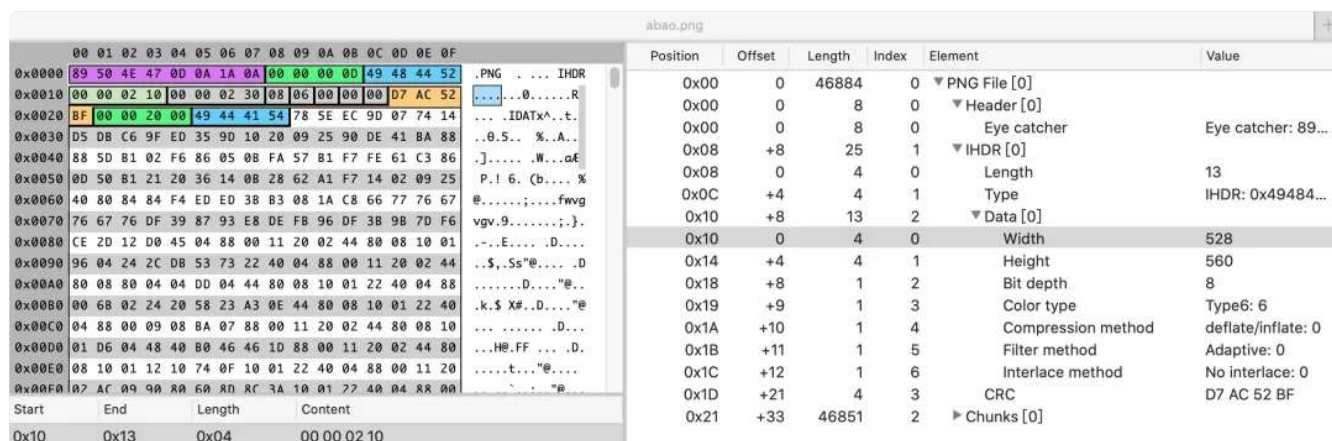
```
return [...string].map((character) => character.charCodeAt(0));
}
```

基于 `stringToBytes` 函数，我们就可以很容易的定义一个 `isPDF` 函数，具体如下所示：

```
const isPDF = check(stringToBytes("%PDF"));
```

有了 `isPDF` 函数，你就实现 PDF 文件检测的功能了。但在实际工作中，遇到的文件类型是多种多样的，针对这种情形，你可以使用现成的第三库来实现文件检测的功能，比如 `file-type` 这个库。

其实基于文件的二进制数据，除了可以检测文件的类型之外，我们还可以读取文件相关的元信息，比如图片的尺寸、位深度、色彩类型和压缩算法等，我们继续以阿宝哥的头像（`abao.png`）为例，来看一下实际的情况：



The image shows a hex editor on the left and a metadata viewer on the right for the file 'abao.png'.

**Hex Editor (Left):** Displays the raw binary data of the PNG file. The first few bytes are highlighted in yellow and green, corresponding to the PNG signature and IHDR chunk.

**Metadata Viewer (Right):** A table showing the structure of the PNG file.

Position	Offset	Length	Index	Element	Value
0x00	0	46884	0	▼ PNG File [0]	
0x00	0	8	0	▼ Header [0]	
0x00	0	8	0	Eye catcher	Eye catcher: 89...
0x08	+8	25	1	▼ IHDR [0]	
0x08	0	4	0	Length	13
0x0C	+4	4	1	Type	IHDR: 0x49484...
0x10	+8	13	2	▼ Data [0]	
0x10	0	4	0	Width	528
0x14	+4	4	1	Height	560
0x18	+8	1	2	Bit depth	8
0x19	+9	1	3	Color type	Type6: 6
0x1A	+10	1	4	Compression method	deflate/inflate: 0
0x1B	+11	1	5	Filter method	Adaptive: 0
0x1C	+12	1	6	Interlace method	No interlace: 0
0x1D	+21	4	3	CRC	D7 AC 52 BF
0x21	+33	46851	2	► Chunks [0]	

好的，在前端如何检测文件类型就介绍到这里。在实际项目中，对于文件上传的场景，出于安全考虑，建议小伙伴们在开发过程中，都限制一下文件上传的类型。对于更严格的场景来说，就可以考虑使用阿宝哥介绍的方法来做文件类型的校验。此外，如果你对前端如何处理二进制数据感兴趣可以阅读 [玩转前端二进制](#)。

#### 四、参考资源

- [玩转前端二进制](#)
- [MDN - FileReader](#)



全栈修仙之路

专注分享 TS、Vue3、前端架构和源码解析等技术干货。

125篇原创内容

公众号

关注公众号  
获取PDF电子书

分别回复 **ym**、**ts** 或 **pdf** 关键字

喜欢此内容的人还喜欢

5kb 的 Vue：尤雨溪发布新作 petite-vue

前端宇宙

Monorepo 的这些坑，我们帮你踩过了！

ELab团队

Promise必备知识汇总和面试情况

全栈前端精选