

struct

Python提供了一个struct模块来解决bytes和其他二进制数据类型的转换。

```
--struct.pack(fmt, (v1, v2...))
```

根据fmt，将v1, v2... 转换为一个字符串

格式字符有下面的定义：

'>' 是网络序的意思

Format	C Type	Python	字节数
x	pad byte	no value	1
c	char	bytes of length 1	1
b	signed char	integer	1
B	unsigned char	integer	1
?	_Bool	bool	1
h	short	integer	2
H	unsigned short	integer	2
i	int	integer	4
I	unsigned int	integer	4
l	long	integer	4
L	unsigned long	integer	4
q	long long	integer	8
Q	unsigned long long	integer	8
f	float	float	4
d	double	float	8
s	char[]	bytes	1
p	char[]	bytes	1
P	void *	integer	

- 注意：
- 1. c, s和p按照bytes对象执行转码操作，但是在使用UTF-8编码时，也支持str对象。
 - 2. ‘?’ 按照C99中定义的_Bool类型转码。如果该类型不可用，可使用一个char冒充。
 - 3. ‘q’ 和 ‘Q’ 仅在64位系统上有用。

FE:

```
>>> import struct
>>> struct.pack('I', 5644)
b'\x0c\x16\x00\x00'
>>> struct.pack('>I', 5644)
b'\x00\x00\x16\x0c'
```

```
--struct.unpack(fmt, bytes)
```

FE:

```
>>> struct.unpack('>I', b'\x00\x00\x16\x0c')
(5644,)
>>> struct.unpack('>I', b'\x0c\x16\x00\x00')
(202768384,)
```

拓:

Windows的位图文件（.bmp）是一种非常简单的文件格式，我们用来用struct分析一下。

首先找一个bmp文件，没有的话用“画图”画一个。

读入前30个字节来分析：

```
>>> s =
b'\x42\x4d\x38\x8c\xa0\x00\x00\x00\x00\x36\x00\x00\x00\x28\x00\x00\x80\x02\x00\x00\x68\x01\x00\x00\x01\x00\x18\x00'
```

BMP格式采用小端方式存储数据，文件头的结构按顺序如下：

两个字节：'BM'表示Windows位图，'BA'表示OS/2位图；

一个4字节整数：表示位图大小；

一个4字节整数：保留位，始终为0；

一个4字节整数：实际图像的偏移量；

一个4字节整数：Header的字节数；

一个4字节整数：图像宽度；

一个4字节整数：图像高度；

一个2字节整数：始终为1；

一个2字节整数：颜色数。

所以，组合起来用`unpack`读取：

```
>>> struct.unpack('<ccIIIIIIHH', s)
(b'B', b'M', 691256, 0, 54, 40, 640, 360, 1, 24)
```

结果显示，`b'B'`、`b'M'`说明是Windows位图，位图大小为640x360，颜色数为24。

请编写一个`bmpinfo.py`，可以检查任意文件是否是位图文件，如果是，打印出图片大小和颜色数。