

## 3.4 二八十六进制整数¶

### 问题¶

你需要转换或者输出使用二进制，八进制或十六进制表示的整数。

### 解决方案¶

为了将整数转换为二进制、八进制或十六进制的文本串，可以分别使用 `bin()`、`oct()` 或 `hex()` 函数：

```
>>> x = 1234
>>> bin(x)
'0b10011010010'
>>> oct(x)
'0o2322'
>>> hex(x)
'0x4d2'
>>>
```

另外，如果你不想输出 `0b`、`0o` 或者 `0x` 的前缀的话，可以使用 `format()` 函数。比如：

```
>>> format(x, 'b')
'10011010010'
>>> format(x, 'o')
'2322'
>>> format(x, 'x')
'4d2'
>>>
```

整数是有符号的，所以如果你在处理负数的话，输出结果会包含一个负号。比如：

```
>>> x = -1234
>>> format(x, 'b')
'-10011010010'
>>> format(x, 'x')
'-4d2'
>>>
```

如果你想产生一个无符号值，你需要增加一个指示最大位长度的值。比如为了显示32位的值，可以像下面这样写：

```
>>> x = -1234
>>> format(2**32 + x, 'b')
'111111111111111111111111111111110100101110'
>>> format(2**32 + x, 'x')
'fffffb2e'
>>>
```

为了以不同的进制转换整数字符串，简单的使用带有进制的 `int()` 函数即可：

```
>>> int('4d2', 16)
1234
>>> int('10011010010', 2)
1234
>>>
```

### 讨论¶

大多数情况下处理二进制、八进制和十六进制整数是很简单的。只要记住这些转换属于整数和其对应的文本表示之间的转换即可。永远只有一种整数类型。

最后，使用八进制的程序员有一点需要注意下。Python指定八进制数的语法跟其他语言稍有不同。比如，如果你像下面这样指定八进制，会出现语法错误：

```
>>> import os
```

```
>>> os.chmod('script.py', 0755)
      File "<stdin>", line 1
        os.chmod('script.py', 0755)
                                ^
SyntaxError: invalid token
>>>
```

需确保八进制数的前缀是 0o，就像下面这样：

```
>>> os.chmod('script.py', 0o755)
>>>
```