

## 3.7 无穷大与NaN

### 问题

你想创建或测试正无穷、负无穷或NaN(非数字)的浮点数。

### 解决方案

Python并没有特殊的语法来表示这些特殊的浮点值，但是可以使用 `float()` 来创建它们。比如：

```
>>> a = float('inf')
>>> b = float('-inf')
>>> c = float('nan')
>>> a
inf
>>> b
-inf
>>> c
nan
>>>
```

为了测试这些值的存在，使用 `math.isinf()` 和 `math.isnan()` 函数。比如：

```
>>> math.isinf(a)
True
>>> math.isnan(c)
True
>>>
```

### 讨论

想了解更多这些特殊浮点值的信息，可以参考IEEE 754规范。然而，也有一些地方需要你特别注意，特别是跟比较和操作符相关的时候。

无穷大数在执行数学计算的时候会传播，比如：

```
>>> a = float('inf')
>>> a + 45
inf
>>> a * 10
inf
>>> 10 / a
0.0
>>>
```

但是有些操作时未定义的并会返回一个NaN结果。比如：

```
>>> a = float('inf')
>>> a/a
nan
>>> b = float('-inf')
>>> a + b
nan
>>>
```

NaN值会在所有操作中传播，而不会产生异常。比如：

```
>>> c = float('nan')
>>> c + 23
nan
>>> c / 2
nan
>>> c * 2
nan
```

```
>>> math.sqrt(c)
nan
>>>
```

NaN值的一个特别的地方是它们之间的比较操作总是返回False。比如：

```
>>> c = float('nan')
>>> d = float('nan')
>>> c == d
False
>>> c is d
False
>>>
```

由于这个原因，测试一个NaN值得唯一安全的方法就是使用 `math.isnan()`，也就是上面演示的那样。

有时候程序员想改变Python默认行为，在返回无穷大或NaN结果的操作中抛出异常。`fpectl` 模块可以用来改变这种行为，但是它在标准的Python构建中并没有被启用，它是平台相关的，并且针对的是专家级程序员。可以参考在线的Python文档获取更多的细节。