

NumPy 数学函数

NumPy 包含大量的各种数学运算的函数，包括三角函数，算术运算的函数，复数处理函数等。

三角函数

NumPy 提供了标准的三角函数：`sin()`、`cos()`、`tan()`。

实例

```
import numpy as np
a = np.array([0,30,45,60,90])
print ('不同角度的正弦值：')
# 通过乘 pi/180 转化为弧度
print (np.sin(a*np.pi/180))
print ('\n')
print ('数组中角度的余弦值：')
print (np.cos(a*np.pi/180))
print ('\n')
print ('数组中角度的正切值：')
print (np.tan(a*np.pi/180))
```

输出结果为：

不同角度的正弦值：

```
[0.          0.5          0.70710678  0.8660254   1.          ]
```

数组中角度的余弦值：

```
[1.00000000e+00  8.66025404e-01  7.07106781e-01  5.00000000e-01
 6.12323400e-17]
```

数组中角度的正切值：

```
[0.00000000e+00  5.77350269e-01  1.00000000e+00  1.73205081e+00
 1.63312394e+16]
```

`arcsin`，`arccos`，和 `arctan` 函数返回给定角度的 `sin`，`cos` 和 `tan` 的反三角函数。

这些函数的结果可以通过 `numpy.degrees()` 函数将弧度转换为角度。

实例

```
import numpy as np
a = np.array([0,30,45,60,90])
print ('含有正弦值的数组：')
sin = np.sin(a*np.pi/180)
print (sin)
print ('\n')
print ('计算角度的反正弦，返回值以弧度为单位：')
```

```
inv = np.arcsin(sin)
print (inv)
print ('\n')
print ('通过转化为角度制来检查结果: ')
print (np.degrees(inv))
print ('\n')
print ('arccos 和 arctan 函数行为类似: ')
cos = np.cos(a*np.pi/180)
print (cos)
print ('\n')
print ('反余弦: ')
inv = np.arccos(cos)
print (inv)
print ('\n')
print ('角度制单位: ')
print (np.degrees(inv))
print ('\n')
print ('tan 函数: ')
tan = np.tan(a*np.pi/180)
print (tan)
print ('\n')
print ('反正切: ')
inv = np.arctan(tan)
print (inv)
print ('\n')
print ('角度制单位: ')
print (np.degrees(inv))
```

输出结果为：

含有正弦值的数组：

```
[0.          0.5          0.70710678 0.8660254  1.          ]
```

计算角度的反正弦，返回值以弧度为单位：

```
[0.          0.52359878 0.78539816 1.04719755 1.57079633]
```

通过转化为角度制来检查结果：

```
[ 0. 30. 45. 60. 90.]
```

arccos 和 arctan 函数行为类似：

```
[1.00000000e+00 8.66025404e-01 7.07106781e-01 5.00000000e-01
 6.12323400e-17]
```

反余弦：

```
[0.          0.52359878 0.78539816 1.04719755 1.57079633]
```

角度制单位：

```
[ 0. 30. 45. 60. 90.]
```

tan 函数：

```
[0.00000000e+00  5.77350269e-01  1.00000000e+00  1.73205081e+00  
 1.63312394e+16]
```

反正切：

```
[0.          0.52359878  0.78539816  1.04719755  1.57079633]
```

角度制单位：

```
[ 0. 30. 45. 60. 90.]
```

舍入函数

`numpy.around()` 函数返回指定数字的四舍五入值。

```
numpy.around(a,decimals)
```

参数说明：

- a: 数组
- decimals: 舍入的小数位数。默认值为0。如果为负，整数将四舍五入到小数点左侧的位置

实例

```
import numpy as np  
a = np.array([1.0,5.55, 123, 0.567, 25.532])  
print ('原数组: ')  
print (a)  
print ('\n')  
print ('舍入后: ')  
print (np.around(a))  
print (np.around(a, decimals = 1))  
print (np.around(a, decimals = -1))
```

输出结果为：

原数组：

```
[ 1.    5.55 123.    0.567 25.532]
```

舍入后：

```
[ 1.    6. 123.    1. 26.]
```

```
[ 1.   5.6 123.   0.6 25.5]
[ 0.  10. 120.   0.  30.]
```

numpy.floor()

numpy.floor() 返回数字的下舍整数。

实例

```
import numpy as np
a = np.array([-1.7, 1.5, -0.2, 0.6, 10])
print ('提供的数组: ')
print (a)
print ('\n')
print ('修改后的数组: ')
print (np.floor(a))
```

输出结果为：

提供的数组：

```
[-1.7  1.5 -0.2  0.6 10. ]
```

修改后的数组：

```
[-2.  1. -1.  0. 10.]
```

numpy.ceil()

numpy.ceil() 返回数字的上入整数。

实例

```
import numpy as np
a = np.array([-1.7, 1.5, -0.2, 0.6, 10])
print ('提供的数组: ')
print (a)
print ('\n')
print ('修改后的数组: ')
print (np.ceil(a))
```

输出结果为：

提供的数组：

```
[-1.7  1.5 -0.2  0.6 10. ]
```

修改后的数组：

```
[-1.  2. -0.  1. 10.]
```

 点我分享笔记

