附录：help 以及文档帮助

2023 年 9 月 18 日

help 以及文档帮助

1. help()

如果想要了解 Python 里面变量、函数等未知的信息，Python 有一个内置的 help() 函数，可以帮助您访问这些信息。这意味着几乎在任何需要更多信息的时候，您都可以使用 help() 来快速查找所需的信息。

[4]:
```
help(max)
```

```
Help on built-in function max in module builtins:

max(…)
    max(iterable, *[, default=obj, key=func]) -> value
    max(arg1, arg2, *args, *[, key=func]) -> value

    With a single iterable argument, return its biggest item. The
    default keyword-only argument specifies an object to return if
    the provided iterable is empty.
    With two or more arguments, return the largest argument.
```

[6]:
```
import numpy as np
help(np.max)
```

```
Help on function amax in module numpy:

amax(a, axis=None, out=None, keepdims=<no value>, initial=<no value>, where=<no
value>)
```

```
Return the maximum of an array or maximum along an axis.

Parameters
----------
a : array_like
    Input data.
axis : None or int or tuple of ints, optional
    Axis or axes along which to operate.  By default, flattened input is
    used.

    .. versionadded:: 1.7.0

    If this is a tuple of ints, the maximum is selected over multiple axes,
    instead of a single axis or all the axes as before.
out : ndarray, optional
    Alternative output array in which to place the result.  Must
    be of the same shape and buffer length as the expected output.
    See :ref:`ufuncs-output-type` for more details.

keepdims : bool, optional
    If this is set to True, the axes which are reduced are left
    in the result as dimensions with size one. With this option,
    the result will broadcast correctly against the input array.

    If the default value is passed, then `keepdims` will not be
    passed through to the `amax` method of sub-classes of
    `ndarray`, however any non-default value will be.  If the
    sub-class' method does not implement `keepdims` any
    exceptions will be raised.

initial : scalar, optional
    The minimum value of an output element. Must be present to allow
    computation on empty slice. See `~numpy.ufunc.reduce` for details.

    .. versionadded:: 1.15.0

where : array_like of bool, optional
```

```
    Elements to compare for the maximum. See `~numpy.ufunc.reduce`
    for details.

    .. versionadded:: 1.17.0

Returns
-------
amax : ndarray or scalar
    Maximum of `a`. If `axis` is None, the result is a scalar value.
    If `axis` is given, the result is an array of dimension
    ``a.ndim - 1``.

See Also
--------
amin :
    The minimum value of an array along a given axis, propagating any NaNs.
nanmax :
    The maximum value of an array along a given axis, ignoring any NaNs.
maximum :
    Element-wise maximum of two arrays, propagating any NaNs.
fmax :
    Element-wise maximum of two arrays, ignoring any NaNs.
argmax :
    Return the indices of the maximum values.

nanmin, minimum, fmin

Notes
-----
NaN values are propagated, that is if at least one item is NaN, the
corresponding max value will be NaN as well. To ignore NaN values
(MATLAB behavior), please use nanmax.

Don't use `amax` for element-wise comparison of 2 arrays; when
``a.shape[0]`` is 2, ``maximum(a[0], a[1])`` is faster than
``amax(a, axis=0)``.
```

```
Examples
--------
>>> a = np.arange(4).reshape((2,2))
>>> a
array([[0, 1],
       [2, 3]])
>>> np.amax(a)           # Maximum of the flattened array
3
>>> np.amax(a, axis=0)   # Maxima along the first axis
array([2, 3])
>>> np.amax(a, axis=1)   # Maxima along the second axis
array([1, 3])
>>> np.amax(a, where=[False, True], initial=-1, axis=0)
array([-1,  3])
>>> b = np.arange(5, dtype=float)
>>> b[2] = np.NaN
>>> np.amax(b)
nan
>>> np.amax(b, where=~np.isnan(b), initial=-1)
4.0
>>> np.nanmax(b)
4.0

You can use an initial value to compute the maximum of an empty slice, or
to initialize it to a different value:

>>> np.max([[-50], [10]], axis=-1, initial=0)
array([ 0, 10])

Notice that the initial value is used as one of the elements for which the
maximum is determined, unlike for the default argument Python's max
function, which is only used for empty iterables.

>>> np.max([5], initial=6)
6
>>> max([5], default=6)
5
```

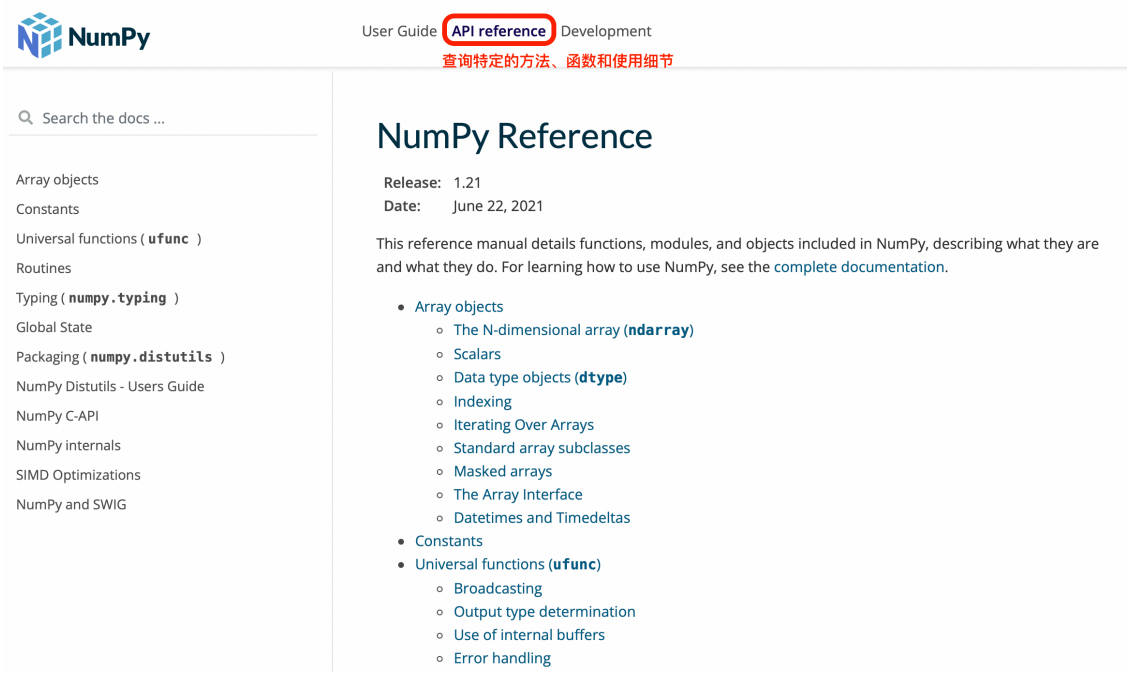
2. 文档帮助

有时候我们需要系统性得了解某一个知识点，光是看 help() 下面的解释已经不够了，这时候需要向这个知识和工具的创造者 (creater) 求助。这些知识和工具的创造者一般会有博客或者官方网站，提供给使用者参考。

我们以 numpy 的官方网站为例：https://numpy.org/doc/stable/user/index.html

点击 User Guide 用户指南，进入后，能看到关于 Numpy 的介绍、安装、新手指南和基础知识。

如果是想要直接查询 Numpy 里面的特定函数，请访问 API reference。


NumPy
User Guide
API reference
Development
查询特定的方法、函数和使用细节
Search the docs ...
Array objects
Constants
Universal functions (ufunc)
Routines
Typing (numpy.typing)
Global State
Packaging (numpy.distutils)
NumPy Distutils - Users Guide
NumPy C-API
NumPy internals
SIMD Optimizations
NumPy and SWIG
NumPy Reference
Release: 1.21
Date: June 22, 2021
This reference manual details functions, modules, and objects included in NumPy, describing what they are and what they do. For learning how to use NumPy, see the complete documentation.
Array objects
The N-dimensional array (ndarray)
Scalars
Data type objects (dtype)
Indexing
Iterating Over Arrays
Standard array subclasses
Masked arrays
The Array Interface
Datetimes and Timedeltas
Constants
Universal functions (ufunc)
Broadcasting
Output type determination
Use of internal buffers
Error handling

[ ]: