



NumPy Data Types

[< Previous](#)[Next >](#)

Data Types in Python

By default Python have these data types:

- **strings** - used to represent text data, the text is given under quote marks. e.g. "ABCD"
- **integer** - used to represent integer numbers. e.g. -1, -2, -3
- **float** - used to represent real numbers. e.g. 1.2, 42.42
- **boolean** - used to represent True or False.
- **complex** - used to represent complex numbers. e.g. 1.0 + 2.0j, 1.5 + 2.5j

Data Types in NumPy

NumPy has some extra data types, and refer to data types with one character, like **i** for integers, **u** for unsigned integers etc.

Below is a list of all data types in NumPy and the characters used to represent them.

- **i** - integer
- **b** - boolean
- **u** - unsigned integer
- **f** - float
- **c** - complex float
- **m** - timedelta
- **M** - datetime

- **O** - object
- **S** - string
- **U** - unicode string
- **V** - fixed chunk of memory for other type (void)

Checking the Data Type of an Array

The NumPy array object has a property called **dtype** that returns the data type of the array:

Example

Get the data type of an array object:

```
import numpy as np

arr = np.array([1, 2, 3, 4])

print(arr.dtype)
```

[Try it Yourself »](#)

Example

Get the data type of an array containing strings:

```
import numpy as np

arr = np.array(['apple', 'banana', 'cherry'])

print(arr.dtype)
```

[Try it Yourself »](#)

Creating Arrays With a Defined Data Type

We use the `array()` function to create arrays, this function can take an optional argument: `dtype` that allows us to define the expected data type of the array elements:

Example

Create an array with data type string:

```
import numpy as np

arr = np.array([1, 2, 3, 4], dtype='S')

print(arr)
print(arr.dtype)
```

[Try it Yourself »](#)

For `i`, `u`, `f`, `S` and `U` we can define size as well.

Example

Create an array with data type 4 bytes integer:

```
import numpy as np
```

```
arr = np.array([1, 2, 3, 4], dtype='i4')

print(arr)
print(arr.dtype)
```

[Try it Yourself »](#)

What if a Value Can Not Be Converted?

If a type is given in which elements can't be casted then NumPy will raise a `ValueError`.

ValueError: In Python `ValueError` is raised when the type of passed argument to a function is unexpected/incorrect.

Example

A non integer string like 'a' can not be converted to integer (will raise an error):

```
import numpy as np

arr = np.array(['a', '2', '3'], dtype='i')
```

[Try it Yourself »](#)

Converting Data Type on Existing Arrays

The best way to change the data type of an existing array, is to make a copy of the array with the `astype()` method.

The `astype()` function creates a copy of the array, and allows you to specify the data type as a parameter.

The data type can be specified using a string, like `'f'` for float, `'i'` for integer etc. or you can use the data type directly like `float` for float and `int` for integer.

Example

Change data type from float to integer by using `'i'` as parameter value:

```
import numpy as np

arr = np.array([1.1, 2.1, 3.1])

newarr = arr.astype('i')

print(newarr)
print(newarr.dtype)
```

[Try it Yourself »](#)

Example

Change data type from float to integer by using `int` as parameter value:

```
import numpy as np

arr = np.array([1.1, 2.1, 3.1])

newarr = arr.astype(int)

print(newarr)
print(newarr.dtype)
```

[Try it Yourself »](#)

Example

Change data type from integer to boolean:

```
import numpy as np

arr = np.array([1, 0, 3])

newarr = arr.astype(bool)
```

```
print(newarr)
print(newarr.dtype)
```

[Try it Yourself »](#)

Test Yourself With Exercises

Exercise:

NumPy uses a character to represent each of the following data types, which one?

```
i = integer
  = boolean
  = unsigned integer
  = float
  = complex float
  = timedelta
  = datetime
  = object
  = string
```

[Submit Answer »](#)

[Start the Exercise](#)

[◀ Previous](#)

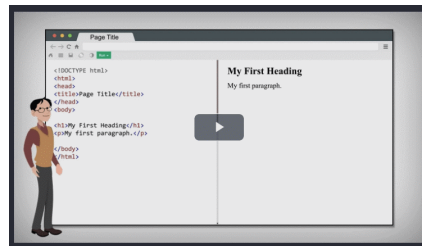
[Next >](#)





NEW

We just launched
W3Schools videos



[Explore now](#)

COLOR PICKER



Get certified
by completing
a Python
course today!



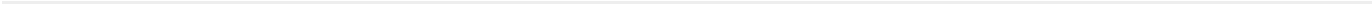
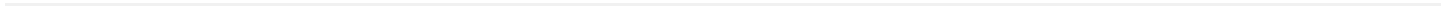
Get started

CODE GAME



Play Game

ADVERTISEMENT

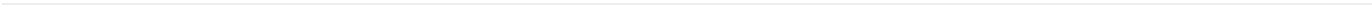


[Report Error](#)

[Spaces](#)

[Pro](#)

[Get Certified](#)



Top Tutorials

[HTML Tutorial](#)
[CSS Tutorial](#)
[JavaScript Tutorial](#)
[How To Tutorial](#)
[SQL Tutorial](#)
[Python Tutorial](#)
[W3.CSS Tutorial](#)
[Bootstrap Tutorial](#)
[PHP Tutorial](#)
[Java Tutorial](#)
[C++ Tutorial](#)
[jQuery Tutorial](#)

Top References

[HTML Reference](#)
[CSS Reference](#)
[JavaScript Reference](#)
[SQL Reference](#)
[Python Reference](#)
[W3.CSS Reference](#)
[Bootstrap Reference](#)
[PHP Reference](#)
[HTML Colors](#)
[Java Reference](#)
[Angular Reference](#)
[jQuery Reference](#)

Top Examples

[HTML Examples](#)
[CSS Examples](#)
[JavaScript Examples](#)
[How To Examples](#)
[SQL Examples](#)
[Python Examples](#)
[W3.CSS Examples](#)
[Bootstrap Examples](#)
[PHP Examples](#)
[Java Examples](#)
[XML Examples](#)
[jQuery Examples](#)

Get Certified

[HTML Certificate](#)
[CSS Certificate](#)
[JavaScript Certificate](#)
[Front End Certificate](#)
[SQL Certificate](#)
[Python Certificate](#)
[PHP Certificate](#)
[jQuery Certificate](#)
[Java Certificate](#)
[C++ Certificate](#)
[C# Certificate](#)
[XML Certificate](#)

W3Schools is optimized for learning and training. Examples might be simplified to improve reading and learning. Tutorials, references, and examples are constantly reviewed to avoid errors, but we cannot warrant full correctness of all content. While using W3Schools, you agree to have read and accepted our [terms of use](#), [cookie](#) and [privacy policy](#).

Copyright 1999-2022 by Refsnes Data. All Rights Reserved.

W3Schools is Powered by W3.CSS.

