

NumPy Datatypes: Takeaways

by Dataquest Labs, Inc. - All rights reserved © 2021

Syntax

- Getting the datatype of an ndarray:

```
x.dtype
```

- Setting the datatype of an ndarray:

```
x = np.array([1, 2, 3, 4], np.float64)
```

- Casting to another datatype:

```
y = x.astype(np.int64)
```

Concepts

- NumPy arrays can only hold one type of data. This type cannot be changed once it is created.
- NumPy datatypes are based on C types and have fixed-bit length. This means that they only support a limited range of values.
- Overflow occurs when the result of a calculation exceeds the maximum value supported by the datatype. Underflow occurs when the result of a calculation is smaller than the minimum value supported by the datatype.
- Floating-point numbers use a different type of representation. This representation allows working with very small values as well as very large ones. However, there are gaps, and some numbers cannot be represented, such as `0.1` and `16777217` (for 32-bit floats).
- We can estimate the total memory required by a dataset by calculating the total number of bits based on the total number of values and the number of bits used in the datatype. To obtain the memory in gigabytes, we need to divide the number of bits by $8,000,000,000 = 8 \times 10^9$

Resources

- [NumPy datatypes](#)
- [Floating Point Arithmetic](#)