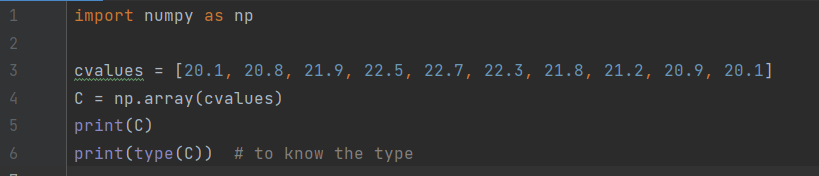
**Numpy package in python:**NumPy and Pandas are two popular Python libraries used for data manipulation, analysis, and exploration. While they serve different purposes, they often work together seamlessly, and Pandas is built on top of NumPy.

Our first simple Numpy example deals with temperatures. Given is a list with values, e.g. temperatures in Celsius. We will turn our list "cvalues" into a one-dimensional numpy array.

****

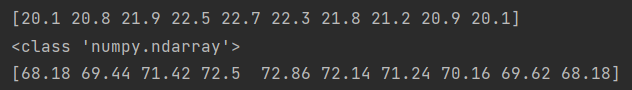
**Output:**

****

Now, we want to turn the values into degrees Fahrenheit, so we add this line to the above code. This expression has no impact on the array.

****

**Output:**

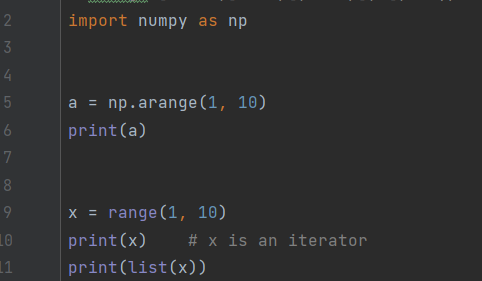
****

**Creation of Arrays with Evenly Spaced Values**

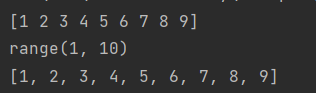
**Arange():**

`arange` generates evenly spaced values within a specified half-open interval '[start, stop)'. It is akin to the Python built-in `range` function when used with integers but differs in that it returns a NumPy ndarray rather than a list iterator. If 'start' is not provided, it defaults to 0. The end of the interval is defined by the 'stop' parameter. The spacing between consecutive values is determined by the optional 'step' parameter, which defaults to 1. If 'step' is specified, 'start' must also be provided.

Additionally, you can set the data type of the output array using the 'dtype' parameter. If 'dtype' is not specified, it is automatically inferred from the other input arguments. Overall, `arange` offers flexibility in generating arrays with specified start, stop, and step values in a concise and efficient manner.

****

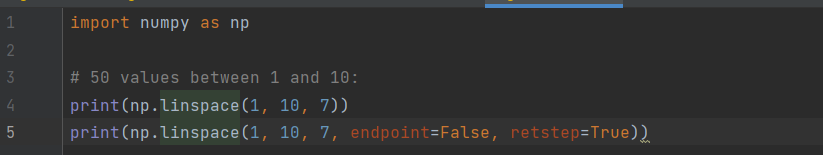
**Output:**

****

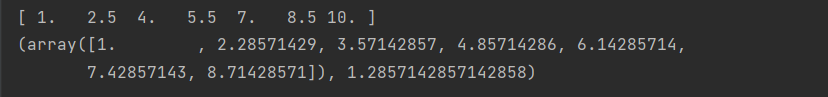
More examples are done in the python file.

**Linspace():**

Linspace returns a NumPy ndarray that contains 'num' evenly spaced samples within the specified interval [start, stop]. The interval can be either closed, including both start and stop, or half-open, including start but excluding stop, depending on the 'endpoint' parameter (True for closed, False for half-open). The starting value of the sequence is defined by the 'start' parameter, and 'stop' is the end value unless 'endpoint' is set to False, in which case the sequence will consist of 'num + 1' samples, excluding the last one. When 'endpoint' is False, the step size adjusts accordingly. The default number of samples is 50, but this can be modified using the 'num' parameter. If 'endpoint' is set to True (the default), 'stop' is the last sample; otherwise, it is not included in the sequence.

****

**Output:**

****