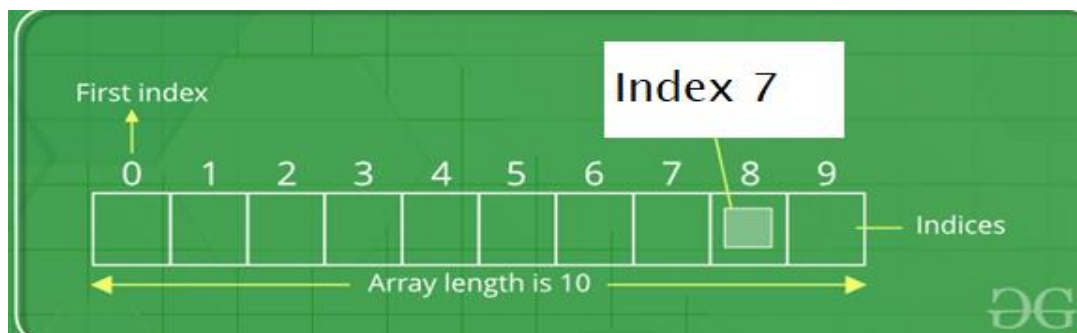


Python Arrays

An array is a collection of items stored at contiguous memory locations. The idea is to store multiple items of the same type together. This makes it easier to calculate the position of each element by simply adding an offset to a base value, i.e., the memory location of the first element of the array (generally denoted by the name of the array).

For simplicity, we can think of an array a fleet of stairs where on each step is placed a value (let's say one of your friends). Here, you can identify the location of any of your friends by simply knowing the count of the step they are on. Array can be handled in Python by a module named array. They can be useful when we have to manipulate only a specific data type value. A user can treat lists as arrays. However, user cannot constraint the type of elements stored in a list. If you create arrays using the array module, all elements of the array must be of the same type.



Syntax:

```
class array.array(typecode[, initializer])
```

Type Code	C Type
'b'	signed char
'B'	unsigned char
'u'	Py_UNICODE
'h'	signed short
'H'	unsigned short
'i'	signed int
'I'	unsigned int
'l'	signed long
'L'	unsigned long
'q'	signed long long
'Q'	unsigned long long
'f'	float
'd'	double

numpy Module

NumPy is a Python library.

NumPy is used for working with arrays.

NumPy is short for "Numerical Python".

NumPy Faster Than Lists :

NumPy arrays are stored at one continuous place in memory unlike lists, so processes can access and manipulate them very efficiently.

This behavior is called locality of reference in computer science.

This is the main reason why NumPy is faster than lists. Also it is optimized to work with latest CPU architectures.

NumPy Data Types

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)

Creating Arrays With a Defined Data Type :

```
import numpy as np
```

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

```
print(arr)
```

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

Math Methods

Method	Description
<code>math.acos()</code>	Returns the arc cosine of a number
<code>math.acosh()</code>	Returns the inverse hyperbolic cosine of a number
<code>math.asin()</code>	Returns the arc sine of a number
<code>math.asinh()</code>	Returns the inverse hyperbolic sine of a number
<code>math.atan()</code>	Returns the arc tangent of a number in radians
<code>math.atan2()</code>	Returns the arc tangent of y/x in radians
<code>math.atanh()</code>	Returns the inverse hyperbolic tangent of a number
<code>math.ceil()</code>	Rounds a number up to the nearest integer
<code>math.comb()</code>	Returns the number of ways to choose k items from n items without repetition and order
<code>math.copysign()</code>	Returns a float consisting of the value of the first parameter and the sign of the second parameter
<code>math.cos()</code>	Returns the cosine of a number
<code>math.cosh()</code>	Returns the hyperbolic cosine of a number
<code>math.degrees()</code>	Converts an angle from radians to degrees
<code>math.dist()</code>	Returns the Euclidean distance between two points (p and q), where p and q are the coordinates of that point
<code>math.erf()</code>	Returns the error function of a number
<code>math.erfc()</code>	Returns the complementary error function of a number
<code>math.exp()</code>	Returns E raised to the power of x
<code>math.expm1()</code>	Returns $e^x - 1$
<code>math.fabs()</code>	Returns the absolute value of a number

<code>math.factorial()</code>	Returns the factorial of a number
<code>math.floor()</code>	Rounds a number down to the nearest integer
<code>math.fmod()</code>	Returns the remainder of x/y
<code>math.frexp()</code>	Returns the mantissa and the exponent, of a specified number
<code>math.fsum()</code>	Returns the sum of all items in any iterable (tuples, arrays, lists, etc.)
<code>math.gamma()</code>	Returns the gamma function at x
<code>math.gcd()</code>	Returns the greatest common divisor of two integers
<code>math.hypot()</code>	Returns the Euclidean norm
<code>math.isclose()</code>	Checks whether two values are close to each other, or not
<code>math.isfinite()</code>	Checks whether a number is finite or not
<code>math.isinf()</code>	Checks whether a number is infinite or not
<code>math.isnan()</code>	Checks whether a value is NaN (not a number) or not
<code>math.isqrt()</code>	Rounds a square root number downwards to the nearest integer
<code>math.ldexp()</code>	Returns the inverse of <code>math.frexp()</code> which is $x * (2^i)$ of the given numbers x and i
<code>math.lgamma()</code>	Returns the log gamma value of x
<code>math.log()</code>	Returns the natural logarithm of a number, or the logarithm of number to base
<code>math.log10()</code>	Returns the base-10 logarithm of x
<code>math.log1p()</code>	Returns the natural logarithm of $1+x$
<code>math.log2()</code>	Returns the base-2 logarithm of x
<code>math.perm()</code>	Returns the number of ways to choose k items from n items with order and without repetition
<code>math.pow()</code>	Returns the value of x to the power of y

<code>math.prod()</code>	Returns the product of all the elements in an iterable
<code>math.radians()</code>	Converts a degree value into radians
<code>math.remainder()</code>	Returns the closest value that can make numerator completely divisible by the denominator
<code>math.sin()</code>	Returns the sine of a number
<code>math.sinh()</code>	Returns the hyperbolic sine of a number
<code>math.sqrt()</code>	Returns the square root of a number
<code>math.tan()</code>	Returns the tangent of a number
<code>math.tanh()</code>	Returns the hyperbolic tangent of a number
<code>math.trunc()</code>	Returns the truncated integer parts of a number