PRIMARY TYPES

ARRAY

Array is a collection of similar data items. Each array element can be accessed using its index. Array indexes are numbers (and sometimes strings) starting from zero. Array elements are ordered and can be of any type.

ARRAY LIST

Array list is a collection of similar data items. Each array element can be accessed using its index. Array indexes are numbers (and sometimes strings) starting from zero. Array elements are ordered and can be of any type. Array list is a type of array with additional functionality, such as dynamic size, insertion and deletion of elements, searching and sorting.

BIT

Bit is the smallest unit of data in computer. It can have two possible values: 0 and 1. Bit is used to represent binary data, such as yes/no, true/false, on/off, etc.

BIT ARRAY

Bit array is a collection of bits. Each bit can be accessed using its index. Bit array indexes are numbers (and sometimes strings) starting from zero. Bit array elements are ordered and can be of any type. Bit array is a type of array with additional functionality, such as dynamic size, insertion and deletion of elements, searching and sorting.

BIT VECTOR

Bit vector is a collection of bits. Each bit can be accessed using its index. Bit vector indexes are numbers (and sometimes strings) starting from zero. Bit vector elements are ordered and can be of any type. Bit vector is a type of array with additional functionality, such as dynamic size, insertion and deletion of elements, searching and sorting.

BOOLEAN

Boolean is a data type that can have two possible values: true and false. Boolean is used to represent binary data, such as yes/no, true/false, on/off, etc.

BYTE

Byte is a unit of data in computer. It can have 256 possible values: 0 to 255. Byte is used to represent binary data, such as yes/no, true/false, on/off, etc.

BYTE ARRAY

Byte array is a collection of bytes. Each byte can be accessed using its index. Byte array indexes are numbers (and sometimes strings) starting from zero. Byte array elements are ordered and can be of any type. Byte array is a type of array with additional functionality, such as dynamic size, insertion and deletion of elements, searching and sorting.

BYTE VECTOR

Byte vector is a collection of bytes. Each byte can be accessed using its index. Byte vector indexes are numbers (and sometimes strings) starting from zero. Byte vector elements are ordered and can be of any type. Byte vector is a type of array with additional functionality, such as dynamic size, insertion and deletion of elements, searching and sorting.

CHARACTER

Character is a data type that can have a single character as its value. Character is used to represent binary data, such as yes/no, true/false, on/off, etc.

CHARACTER ARRAY

Character array is a collection of characters. Each character can be accessed using its index. Character array indexes are numbers (and sometimes strings) starting from zero. Character array elements are ordered and can be of any type. Character array is a type of array with additional functionality, such as dynamic size, insertion and deletion of elements, searching and sorting.

CHARACTER VECTOR

Character vector is a collection of characters. Each character can be accessed using its index. Character vector indexes are numbers (and sometimes strings) starting from zero. Character vector elements are ordered and can be of any type. Character vector is a type of array with additional functionality, such as dynamic size, insertion and deletion of elements, searching and sorting.

DATE

Date is a data type that can have a date as its value. Date is used to represent binary data, such as yes/no, true/false, on/off, etc.

DATE ARRAY

Date array is a collection of dates. Each date can be accessed using its index. Date array indexes are numbers (and sometimes strings) starting from zero. Date array elements are ordered and can be of any type. Date array is a type of array with additional functionality, such as dynamic size, insertion and deletion of elements, searching and sorting.

DATE VECTOR

Date vector is a collection of dates. Each date can be accessed using its index. Date vector indexes are numbers (and sometimes strings) starting from zero. Date vector elements are ordered and can be of any type. Date vector is a type of array with additional functionality, such as dynamic size, insertion and deletion of elements, searching and sorting.

DATETIME

Datetime is a data type that can have a date and time as its value. Datetime is used to represent binary data, such as yes/no, true/false, on/off, etc.

DATETIME ARRAY

Datetime array is a collection of datetimes. Each datetime can be accessed using its index. Datetime array indexes are numbers (and sometimes strings) starting from zero. Datetime array elements are ordered and can be of any type. Datetime array is a type of array with additional functionality, such as dynamic size, insertion and deletion of elements, searching and sorting.

DATETIME VECTOR

Datetime vector is a collection of datetimes. Each datetime can be accessed using its index. Datetime vector indexes are numbers (and sometimes strings) starting from zero. Datetime vector elements are ordered and can be of any type. Datetime vector is a type of array with additional functionality, such as dynamic size, insertion and deletion of elements, searching and sorting.

DOUBLE

Double is a data type that can have a double as its value. Double is used to represent binary data, such as yes/no, true/false, on/off, etc.

DOUBLE ARRAY

Double array is a collection of doubles. Each double can be accessed using its index. Double array indexes are numbers (and sometimes strings) starting from zero. Double array elements are ordered and can be of any type. Double array is a type of array with additional functionality, such as dynamic size, insertion and deletion of elements, searching and sorting.

DOUBLE VECTOR

Double vector is a collection of doubles. Each double can be accessed using its index. Double vector indexes are numbers (and sometimes strings) starting from zero. Double vector elements are ordered and can be of any type. Double vector is a type of array with additional functionality, such as dynamic size, insertion and deletion of elements, searching and sorting.

FLOAT

Float is a data type that can have a float as its value. Float is used to represent binary data, such as yes/no, true/false, on/off, etc.

FLOAT ARRAY

Float array is a collection of floats. Each float can be accessed using its index. Float array indexes are numbers (and sometimes strings) starting from zero. Float array elements are ordered and can be of any type. Float array is a type of array with additional functionality, such as dynamic size, insertion and deletion of elements, searching and sorting.

FLOAT VECTOR

Float vector is a collection of floats. Each float can be accessed using its index. Float vector indexes are numbers (and sometimes strings) starting from zero. Float vector elements are ordered and can be of any type. Float vector is a type of array with additional functionality, such as dynamic size, insertion and deletion of elements, searching and sorting.

INTEGER

Integer is a data type that can have an integer as its value. Integer is used to represent binary data, such as yes/no, true/false, on/off, etc.

INTEGER ARRAY

Integer array is a collection of integers. Each integer can be accessed using its index. Integer array indexes are numbers (and sometimes strings) starting from zero. Integer array elements are ordered and can be of any type. Integer array is a type of array with additional functionality, such as dynamic size, insertion and deletion of elements, searching and sorting.

INTEGER VECTOR

Integer vector is a collection of integers. Each integer can be accessed using its index. Integer vector indexes are numbers (and sometimes strings) starting from zero. Integer vector elements are ordered and can be of any type. Integer vector is a type of array with additional functionality, such as dynamic size, insertion and deletion of elements, searching and sorting.

LONG

Long is a data type that can have a long as its value. Long is used to represent binary data, such as yes/no, true/false, on/off, etc.

LONG ARRAY

Long array is a collection of longs. Each long can be accessed using its index. Long array indexes are numbers (and sometimes strings) starting from zero. Long array elements are ordered and can be of any type. Long array is a type of array with additional functionality, such as dynamic size, insertion and deletion of elements, searching and sorting.

LONG VECTOR

Long vector is a collection of longs. Each long can be accessed using its index. Long vector indexes are numbers (and sometimes strings) starting from zero. Long vector elements are ordered and can be of any type. Long vector is a type of array with additional functionality, such as dynamic size, insertion and deletion of elements, searching and sorting.

NUMBER

Number is a data type that can have a number as its value. Number is used to represent binary data, such as yes/no, true/false, on/off, etc.

NUMBER ARRAY

Number array is a collection of numbers. Each number can be accessed using its index. Number array indexes are numbers (and sometimes strings) starting from zero. Number array elements are ordered and can be of any type. Number array is a type of array with additional functionality, such as dynamic size, insertion and deletion of elements, searching and sorting.

NUMBER VECTOR

Number vector is a collection of numbers. Each number can be accessed using its index. Number vector indexes are numbers (and sometimes strings) starting from zero. Number vector elements are ordered and can be of any type. Number vector is a type of array with additional functionality, such as dynamic size, insertion and deletion of elements, searching and sorting.

SHORT

Short is a data type that can have a short as its value. Short is used to represent binary data, such as yes/no, true/false, on/off, etc.

SHORT ARRAY

Short array is a collection of shorts. Each short can be accessed using its index. Short array indexes are numbers (and sometimes strings) starting from zero. Short array elements are ordered and can be of any type. Short array is a type of array with additional functionality, such as dynamic size, insertion and deletion of elements, searching and sorting.

SHORT VECTOR

Short vector is a collection of shorts. Each short can be accessed using its index. Short vector indexes are numbers (and sometimes strings) starting from zero. Short vector elements are ordered and can be of any type. Short vector is a type of array with additional functionality, such as dynamic size, insertion and deletion of elements, searching and sorting.

STRING

String is a data type that can have a string as its value. String is used to represent binary data, such as yes/no, true/false, on/off, etc.

STRING ARRAY

String array is a collection of strings. Each string can be accessed using its index. String array indexes are numbers (and sometimes strings) starting from zero. String array elements are ordered and can be of any type. String array is a type of array with additional functionality, such as dynamic size, insertion and deletion of elements, searching and sorting.

STRING VECTOR

String vector is a collection of strings. Each string can be accessed using its index. String vector indexes are numbers (and sometimes strings) starting from zero. String vector elements are ordered and can be of any type. String vector is a type of array with additional functionality, such as dynamic size, insertion and deletion of elements, searching and sorting.

TIME

Time is a data type that can have a time as its value. Time is used to represent binary data, such as yes/no, true/false, on/off, etc.

TIME ARRAY

Time array is a collection of times. Each time can be accessed using its index. Time array indexes are numbers (and sometimes strings) starting from zero. Time array elements are ordered and can be of any type. Time array is a type of array with additional functionality, such as dynamic size, insertion and deletion of elements, searching and sorting.

TIME VECTOR

Time vector is a collection of times. Each time can be accessed using its index. Time vector indexes are numbers (and sometimes strings) starting from zero. Time vector elements are ordered and can be of any type. Time vector is a type of array with additional functionality, such as dynamic size, insertion and deletion of elements, searching and sorting.

TIMESTAMP

Timestamp is a data type that can have a timestamp as its value. Timestamp is used to represent binary data, such as yes/no, true/false, on/off, etc.

TIMESTAMP ARRAY

Timestamp array is a collection of timestamps. Each timestamp can be accessed using its index. Timestamp array indexes are numbers (and sometimes strings) starting from zero. Timestamp array elements are ordered and can be of any type. Timestamp array is a type of array with additional functionality, such as dynamic size, insertion and deletion of elements, searching and sorting.

TIMESTAMP VECTOR

Timestamp vector is a collection of timestamps. Each timestamp can be accessed using its index. Timestamp vector indexes are numbers (and sometimes strings) starting from zero. Timestamp vector elements are ordered and can be of any type. Timestamp vector is a type of array with additional functionality, such as dynamic size, insertion and deletion of elements, searching and sorting.

UNIQUE ARRAY

Unique array is a collection of unique data items. Each array element can be accessed using its index. Array indexes are numbers (and sometimes strings) starting from zero. Array elements are ordered and can be of any type. Unique array is a type of array with additional functionality, such as dynamic size, insertion and deletion of elements, searching and sorting.

UNIQUE ARRAY LIST

Unique array list is a collection of unique data items. Each array element can be accessed using its index. Array indexes are numbers (and sometimes strings) starting from zero. Array elements are ordered and can be of any type. Unique array list is a type of array with additional functionality, such as dynamic size, insertion and deletion of elements, searching and sorting.

UNIQUE BIT ARRAY

Unique bit array is a collection of unique bits. Each bit can be accessed using its index. Bit array indexes are numbers (and sometimes strings) starting from zero. Bit array elements are ordered and can be of any type. Unique bit array is a type of array with additional functionality, such as dynamic size, insertion and deletion of elements, searching and sorting.

UNIQUE BIT VECTOR

Unique bit vector is a collection of unique bits. Each bit can be accessed using its index. Bit vector indexes are numbers (and sometimes strings) starting from zero. Bit vector elements are ordered and can be of any type. Unique bit vector is a type of array with additional functionality, such as dynamic size, insertion and deletion of elements, searching and sorting.

UNIQUE BOOLEAN

Unique boolean is a data type that can have two possible values: true and false. Boolean is used to represent binary data, such as yes/no, true/false, on/off, etc.

UNIQUE BYTE

Unique byte is a data type that can have a byte as its value. Byte is used to represent binary data, such as yes/no, true/false, on/off, etc.

UNIQUE BYTE ARRAY

Unique byte array is a collection of unique bytes. Each byte can be accessed using its index. Byte array indexes are numbers (and sometimes strings) starting from zero. Byte array elements are ordered and can be of any type. Unique byte array is a type of array with additional functionality, such as dynamic size, insertion and deletion of elements, searching and sorting.

UNIQUE BYTE VECTOR

Unique byte vector is a collection of unique bytes. Each byte can be accessed using its index. Byte vector indexes are numbers (and sometimes strings) starting from zero. Byte vector elements are ordered and can be of any type. Unique byte vector is a type of array with additional functionality, such as dynamic size, insertion and deletion of elements, searching and sorting.

UNIQUE CHARACTER

Unique character is a data type that can have a character as its value. Character is used to represent binary data, such as yes/no, true/false, on/off, etc.

UNIQUE CHARACTER ARRAY

Unique character array is a collection of unique characters. Each character can be accessed using its index. Character array indexes are numbers (and sometimes strings) starting from zero. Character array elements are ordered and can be of any type. Unique character array is a type of array with additional functionality, such as dynamic size, insertion and deletion of elements, searching and sorting.

UNIQUE CHARACTER VECTOR

Unique character vector is a collection of unique characters. Each character can be accessed using its index. Character vector indexes are numbers (and sometimes strings) starting from zero. Character vector elements are ordered and can be of any type. Unique character vector is a type of array with additional functionality, such as dynamic size, insertion and deletion of elements, searching and sorting.

UNIQUE DATE

Unique date is a data type that can have a date as its value. Date is used to represent binary data, such as yes/no, true/false, on/off, etc.

UNIQUE DATE ARRAY

Unique date array is a collection of unique dates. Each date can be accessed using its index. Date array indexes are numbers (and sometimes strings) starting from zero. Date array elements are ordered and can be of any type. Unique date array is a type of array with additional functionality, such as dynamic size, insertion and deletion of elements, searching and sorting.

UNIQUE DATE VECTOR

Unique date vector is a collection of unique dates. Each date can be accessed using its index. Date vector indexes are numbers (and sometimes strings) starting from zero. Date vector elements are ordered and can be of any type. Unique date vector is a type of array with additional functionality, such as dynamic size, insertion and deletion of elements, searching and sorting.

UNIQUE DATETIME

Unique datetime is a data type that can have a date and time as its value. Datetime is used to represent binary data, such as yes/no, true/false, on/off, etc.

UNIQUE DATETIME ARRAY

Unique datetime array is a collection of unique datetimes. Each datetime can be accessed using its index. Datetime array indexes are numbers (and sometimes strings) starting from zero. Datetime array elements are ordered and can be of any type. Unique datetime array is a type of array with additional functionality, such as dynamic size, insertion and deletion of elements, searching and sorting.

UNIQUE DATETIME VECTOR

Unique datetime vector is a collection of unique datetimes. Each datetime can be accessed using its index. Datetime vector indexes are numbers (and sometimes strings) starting from zero. Datetime vector elements are ordered and can be of any type. Unique datetime vector is a type of array with additional functionality, such as dynamic size, insertion and deletion of elements, searching and sorting.

UNIQUE DOUBLE

Unique double is a data type that can have a double as its value. Double is used to represent binary data, such as yes/no, true/false, on/off, etc.

UNIQUE DOUBLE ARRAY

Unique double array is a collection of unique doubles. Each double can be accessed using its index. Double array indexes are numbers (and sometimes strings) starting from zero. Double array elements are ordered and can be of any type. Unique double array is a type of array with additional functionality, such as dynamic size, insertion and deletion of elements, searching and sorting.

UNIQUE DOUBLE VECTOR

Unique double vector is a collection of unique doubles. Each double can be accessed using its index. Double vector indexes are numbers (and sometimes strings) starting from zero. Double vector elements are ordered and can be of any type. Unique double vector is a type of array with additional functionality, such as dynamic size, insertion and deletion of elements, searching and sorting.

UNIQUE FLOAT

Unique float is a data type that can have a float as its value. Float is used to represent binary data, such as yes/no, true/false, on/off, etc.

UNIQUE FLOAT ARRAY

Unique float array is a collection of unique floats. Each float can be accessed using its index. Float array indexes are numbers (and sometimes strings) starting from zero. Float array elements are ordered and can be of any type. Unique float array is a type of array with additional functionality, such as dynamic size, insertion and deletion of elements, searching and sorting.

UNIQUE FLOAT VECTOR

Unique float vector is a collection of unique floats. Each float can be accessed using its index. Float vector indexes are numbers (and sometimes strings) starting from zero. Float vector elements are ordered and can be of any type. Unique float vector is a type of array with additional functionality, such as dynamic size, insertion and deletion of elements, searching and sorting.

UNIQUE INTEGER

Unique integer is a data type that can have an integer as its value. Integer is used to represent binary data, such as yes/no, true/false, on/off, etc.

UNIQUE INTEGER ARRAY

Unique integer array is a collection of unique integers. Each integer can be accessed using its index. Integer array indexes are numbers (and sometimes strings) starting from zero. Integer array elements are ordered and can be of any type. Unique integer array is a type of array with additional functionality, such as dynamic size, insertion and deletion of elements, searching and sorting.

UNIQUE INTEGER VECTOR

Unique integer vector is a collection of unique integers. Each integer can be accessed using its index. Integer vector indexes are numbers (and sometimes strings) starting from zero. Integer vector elements are ordered and can be of any type. Unique integer vector is a type of array with additional functionality, such as dynamic size, insertion and deletion of elements, searching and sorting.

UNIQUE LONG

Unique long is a data type that can have a long as its value. Long is used to represent binary data, such as yes/no, true/false, on/off, etc.

UNIQUE LONG ARRAY

Unique long array is a collection of unique longs. Each long can be accessed using its index. Long array indexes are numbers (and sometimes strings) starting from zero. Long array elements are ordered and can be of any type. Unique long array is a type of array with additional functionality, such as dynamic size, insertion and deletion of elements, searching and sorting.

UNIQUE LONG VECTOR

Unique long vector is a collection of unique longs. Each long can be accessed using its index. Long vector indexes are numbers (and sometimes strings) starting from zero. Long vector elements are ordered and can be of any type. Unique long vector is a type of array with additional functionality, such as dynamic size, insertion and deletion of elements, searching and sorting.

UNIQUE NUMBER

Unique number is a data type that can have a number as its value. Number is used to represent binary data, such as yes/no, true/false, on/off, etc.

UNIQUE NUMBER ARRAY

Unique number array is a collection of unique numbers. Each number can be accessed using its index. Number array indexes are numbers (and sometimes strings) starting from zero. Number array elements are ordered and can be of any type. Unique number array is a type of array with additional functionality, such as dynamic size, insertion and deletion of elements, searching and sorting.

UNIQUE NUMBER VECTOR

Unique number vector is a collection of unique numbers. Each number can be accessed using its index. Number vector indexes are numbers (and sometimes strings) starting from zero. Number vector elements are ordered and can be of any type. Unique number vector is a type of array with additional functionality, such as dynamic size, insertion and deletion of elements, searching and sorting.

UNIQUE SHORT

Unique short is a data type that can have a short as its value. Short is used to represent binary data, such as yes/no, true/false, on/off, etc.

UNIQUE SHORT ARRAY

Unique short array is a collection of unique shorts. Each short can be accessed using its index. Short array indexes are numbers (and sometimes strings) starting from zero. Short array elements are ordered and can be of any type. Unique short array is a type of array with additional functionality, such as dynamic size, insertion and deletion of elements, searching and sorting.

UNIQUE SHORT VECTOR

Unique short vector is a collection of unique shorts. Each short can be accessed using its index. Short vector indexes are numbers (and sometimes strings) starting from zero. Short vector elements are ordered and can be of any type. Unique short vector is a type of array with additional functionality, such as dynamic size, insertion and deletion of elements, searching and sorting.

UNIQUE STRING

Unique string is a data type that can have a string as its value. String is used to represent binary data, such as yes/no, true/false, on/off, etc.

UNIQUE STRING ARRAY

Unique string array is a collection of unique strings. Each string can be accessed using its index. String array indexes are numbers (and sometimes strings) starting from zero. String array elements are ordered and can be of any type. Unique string array is a type of array with additional functionality, such as dynamic size, insertion and deletion of elements, searching and sorting.

UNIQUE STRING VECTOR

Unique string vector is a collection of unique strings. Each string can be accessed using its index. String vector indexes are numbers (and sometimes strings) starting from zero. String vector elements are ordered and can be of any type. Unique string vector is a type of array with additional functionality, such as dynamic size, insertion and deletion of elements, searching and sorting.

UNIQUE TIME

Unique time is a data type that can have a time as its value. Time is used to represent binary data, such as yes/no, true/false, on/off, etc.

UNIQUE TIME ARRAY

Unique time array is a collection of unique times. Each time can be accessed using its index. Time array indexes are numbers (and sometimes strings) starting from zero. Time array elements are ordered and can be of any type. Unique time array is a type of array with additional functionality, such as dynamic size, insertion and deletion of elements, searching and sorting.

UNIQUE TIME VECTOR

Unique time vector is a collection of unique times. Each time can be accessed using its index. Time vector indexes are numbers (and sometimes strings) starting from zero. Time vector elements are ordered and can be of any type. Unique time vector is a type of array with additional functionality, such as dynamic size, insertion and deletion of elements, searching and sorting.

UNIQUE TIMESTAMP

Unique timestamp is a data type that can have a timestamp as its value. Timestamp is used to represent binary data, such as yes/no, true/false, on/off, etc.

UNIQUE TIMESTAMP ARRAY

Unique timestamp array is a collection of unique timestamps. Each timestamp can be accessed using its index. Timestamp array indexes are numbers (and sometimes strings) starting from zero. Timestamp array elements are ordered and can be of any type. Unique timestamp array is a type of array with additional functionality, such as dynamic size, insertion and deletion of elements, searching and sorting.

UNIQUE TIMESTAMP VECTOR

Unique timestamp vector is a collection of unique timestamps. Each timestamp can be accessed using its index. Timestamp vector indexes are numbers (and sometimes strings) starting from zero. Timestamp vector elements are ordered and can be of any type. Unique timestamp vector is a type of array with additional functionality, such as dynamic size, insertion and deletion of elements, searching and sorting.

UNIQUE VECTOR

Unique vector is a collection of unique data items. Each array element can be accessed using its index. Array indexes are numbers (and sometimes strings) starting from zero. Array elements are ordered and can be of any type. Unique vector is a type of array with additional functionality, such as dynamic size, insertion and deletion of elements, searching and sorting.

VECTOR

Vector is a collection of similar data items. Each array element can be accessed using its index. Array indexes are numbers (and sometimes strings) starting from zero. Array elements are ordered and can be of any type. Vector is a type of array with additional functionality, such as dynamic size, insertion and deletion of elements, searching and sorting.