1. Perl, Python negatif integer subscript kabul edebiliyor( A[-2] ). Perl, JavaScript, Python, C, PHP get all the types as subscript in associative arrays.
2. Python, PHP check the range. JavaScript, Perl, C don't check the range size in page 277. JavaScript print "undefined". Perl does not write anything. C prints “zero".
3. In C , char\_type a[] = "string literal”; is an example of fixed-sized static and it is bound in run time however char\_type a[N] = "string literal”; burada arrayin buyuklugu (N) compile time da belirlenir ama "string literal”degeri run time da atanir. Array tanimi basında “static" varsa ve initialization yapilmamissa tum elemanlar sıfıra eşitlenir(pointer ise NULL) ve butun array islemleri compile time'da yapilir ( static float samples[kFrameCountSample] = { 1, 2, 3 };). T \*p = malloc( sizeof \*p \* N ); bu ornek de runtime da belirlenir.

In JavaScript, var points = new Array(40, 100);  example for opening heap-dynamic because there is no range checking. Array is perfectly dynamic and flexible. var points = [40, 100, 1, 5, 25, 10]; is an example of stack-dynamic. These allocations all happens during run time also the bounding happens in run time with the life time of array. (The static array is not allowed)?

Python does not have built-in support for Arrays, but Python lists can be used instead. List is a heap-dynamic array. It’s size can be changed during the list’s lifetime by the help of some methods such as append(), pop(). It binds the variable of the memory and allocation in the heap at run time.

In Perl, arrays are heap-dynamic so all the operations made with arrays happen during execution time. Both the bind of variables with memory allocation and memory allocation are done in run time.

@ages = (25, 30, 40);

@names = ("John Paul", "Lisa", “Kumar”); .

The stack implementation has been become with the reserve word “my” such as my @names = ('Foo', 'Bar', 'Baz');

print “@names\n"; . In the stack implementation, each operation such as binding or memory allocation is done compile time.(Sor).

In PHP, static kelimesine izin veriyor ve static ile tanımlanan arrayi run time siransinda değiştirmeye izin veriyor.(?)

<?php //php 7.0.8

static $cars = array("Volvo", "BMW", "Toyota");

$cars[3] = "firat";

//static public $array = new array();

echo "I like " . $cars[0] . ", " . $cars[1] . " and " . $cars[3] . ".";

?>

$array = array(); bu kod heapde mi?

$myArr[] = 'New Element to Array’; it defines and bound in run time in stack or heap(?).

$array = array(1); it initialize the first element element of the array as “1” and all the operations is done during run time. And it is heap-dynamic because you can initialize the 2nd element as “12" Iike in :

<?php //php 7.0.8

static $cars = array("Volvo", "BMW", "Toyota");

$cars[3] = "firat";

$array = array(1);

$array[2] = 12;

echo "I like " . $array[0] . ", " . $cars[1] . " and " . $cars[3] . ".";

?>

6)

The property length of arrays is always an integer value l in the range

0 ≤ l ≤ 232−1 (32 bit)

7) Perl @other\_array = (0,0,0,1,2,2,3,3,3,4); Yes.

Python arr = [1, 2, 3, 4, 5, 6]: Yes.

Php $array=array("chemistry", “physics”,”maths”); Yes

Javascript var cars = ["Saab", "Volvo", “BMW"]; Yes

C int myArray[10] = { 5, 5, 5, 5, 5, 5, 5, 5, 5, 5 }; Yes

8)In C slicing is not allowed. We cannot take a dimension of multi-dimensional array or defined group of elements inside of the array whose indexes are defined.

int main()

{

int a[3][4] = {

{0, 1, 2, 3} , /\* initializers for row indexed by 0 \*/

{4, 5, 6, 7} , /\* initializers for row indexed by 1 \*/

{8, 9, 10, 11} /\* initializers for row indexed by 2 \*/

};

int b[4] = a[1];

printf("%d", b[1]);

}

Error.

In Javascript, there is a specific method(slice()) to slice arrays.

var fruits = ["Banana", "Orange", "Lemon", "Apple", "Mango"];

var citrus = fruits.slice(1, 3);

This method can be used for multi-dimensional arrays.

function myFunction() {

var items = [

[1, 2],

[3, 4],

[5, 6]

];

var citrus = items.slice(1);

document.getElementById("demo").innerHTML = citrus;

}

In php, there is a specific method(array\_slice()) to slice arrays.

<!DOCTYPE html>

<html>

<body>

<?php

$a=array("red","green","blue","yellow","brown");

print\_r(array\_slice($a,2));

?>

</body>

</html>

This method can be used for multi-dimensional arrays.

<!DOCTYPE html>

<html>

<body>

<?php

$cars = array(array("Volvo",22,18),array("BMW",15,13),array("Saab",5,2),array("Land Rover",17,15));

print\_r(array\_slice($cars,2));

?>

</body>

</html>

In Python:

vector = [2,4,6,8,10,12,14,16]

mat = [[1,2,3],[4,5,6],[7,8,9]]

We can slice a dimension of multi dimensional array by using mat[1].

One dimensional array can be sliced like vector[3:6]. For example mat[0][0:2] refers to the first and second element of the first row of the mat which is 1,2.

There is more complex slicing operations such as vector[0:7:2] references every other element of the vector up to but not including the element with subscript 7 by adding 2. [2,6,10,14]

In Perl:

@list[1..5] = @list2[3,5,7,9,13] which assigns the elements of list between 1 and 5(not included) to list2’s 3., 5., 7., 9., 13. Elements.

Different types of slicing.

1. my @kings = ('Baldwin', 'Melisende', 'Fulk', 'Amalric', 'Guy', 'Conrad');
3. my @names = ($kings[2], $kings[4], $kings[1]);
4. say join ', ', @names; # Fulk, Guy, Melisende

7. my @slice = @kings[2,4,1]
8. say join ', ', @slice; # Fulk, Guy, Melisende