

CS 315 - 02

Homework 1

**Arrays in Dart, Javascript, PHP,
Python and Rust**

Dilay Yigit

21602059

DART

- What types are legal for subscripts?

Array subscript type of Dart is integer and index syntax is brackets.

```
var nums = [1, 2, 3, 4];  
print(nums[0]);
```

- Are subscripting expressions in element references range checked?

Dart is checking index range in arrays. If we try to print the index that is not exist, we will see index out of range error.

```
print(nums[-8]); ---> IndexOutOfRangeException  
print(nums[20]); ---> IndexOutOfRangeException
```

- When are subscript ranges bound?
- When does allocation take place?

Dart supports heap-dynamic arrays. So that, binding array subscript and allocating storage is dynamic and done at run time, they can change during execution time.

- Are ragged or rectangular multidimensional arrays allowed, or both?

Both of them is allowed.

```
List<List<int>> arr1 = [  
  [1, 2, 3, 4, 5],  
  [6, 7],  
  [8, 9, 0],  
];
```

```
List<List<int>> arr2 = [  
  [1, 2, 3],
```

```
[4, 5, 6],  
[7, 8, 9],  
];
```

- Can array objects be initialised?

Yes.

```
var money = [100, 200, 5000, 35, 42]; // dynamic-size  
var vowels = new List.filled(5, '', growable:  
false); // fixed-size
```

- Are any kind of slices supported?

Dart supports sublist.

```
var emotes = ["Kappa", "KEKW", "OmegaLUL", "PogChamp"];  
print(emotes.sublist(1, 3));
```

- Which operators are provided?

+ and == operators.

```
var nums1 = [1, 2, 3, 4, 5];  
var nums2 = [6, 7, 8, 9];  
var result = nums1 + nums2;  
print(nums1 == nums2);
```

JAVASCRIPT

- What types are legal for subscripts?

Array subscript type of JavaScript is integer and index syntax is brackets.

```
var friends = ['Dilay', 'Arda', 'Ozgur', 'Eylul'];  
console.log(friends[0]);
```

- Are subscripting expressions in element references range checked?

Javascript is checking index range in arrays. However it does not print an error message. It does not support negative index. Also, if we try to print the index that is not exist, we will see undefined.

```
console.log(friends[-1]); //---> undefined
console.log(friends[10]); //---> undefined
console.log(friends.length);
```

- When are subscript ranges bound?
- When does allocation take place?

JavaScript supports heap-dynamic arrays. So that, binding array subscript and allocating storage is dynamic and done at run time, they can change during execution time.

- Are ragged or rectangular multidimensional arrays allowed, or both?
Both of them is allowed. It supports array of array.

```
var arr1 = [[5, 10, 15],
            [20, 25, 30],
            [35, 40, 45]]

var arr2 = [['A', 'B', 'C'],
            ['D', 'E'],
            ['F']]
```

- Can array objects be initialised?

Yes.

```
var emotes = ["Kappa", "KEKW", "OmegaLUL",
"PogChamp"];
console.log(emotes);
```

- Are any kind of slices supported?

Yes.

```
var emotes_vol2 = ["FailFish", "CoolStoryBob",
"NotLikeThis", "SeemsGood"];
console.log(emotes_vol2.slice(1,3));
console.log(emotes_vol2.slice(2,4));
```

- Which operators are provided?

[] and []= operators.

```
emotes_vol2[0] = "Keepo";
var firstData = emotes_vol2[0];
```

PHP

- What types are legal for subscripts?

Array subscript type of PHP can be integer, string and index syntax is brackets.

```
$books = array("lotr1", "lotr2", "lotr3");
echo $books[0];

$stringSubscript = array(
'Turkey' => 'Ankara',
'England' => 'London',
'Italy' => 'Rome',
'France' => 'Paris'
);
```

- Are subscripting expressions in element references range checked?

PHP is checking index range in arrays. However it does not print an error message. It does not support negative index. Also, if we try to print the index that is not exist, we will see undefined.

```
echo $friends[23]; //--> undefined
echo $friends[-1]; //--> undefined
```

- When are subscript ranges bound?
- When does allocation take place?

PHP supports heap-dynamic arrays. So that, binding array subscript and allocating storage is dynamic and done at run time, they can change during execution time.

- Are ragged or rectangular multidimensional arrays allowed, or both?

Both of them is allowed.

```
$menu = array(
    array("water", "soda", "lemonade"),
    array("pancake", "sufle"),
    array("salad")
);

$num = array(
    array(10, 20),
    array(30, 40),
);
```

- Can array objects be initialised?

Yes.

```
$lectures = ["CS315", "CS202", "CS224"];
echo $lectures[0];
```

- Are any kind of slices supported?

PHP supports slices.

```
$vowel = array("a","e","i","o","u");
print_r(array_slice($vowel,0,5));
```

- Which operators are provided?

+ and == operators.

```
$first1 = array('firstNum' => 23);
$second1 = array('secondNum' => 9);
$result = $first1 + $second1;
$first2 = array("key" => 23);
$second2 = array("key" => 9);
var_dump($first2 == $second2);
```

PYTHON

- What types are legal for subscripts?

Array subscript type of Python is integer and index syntax is brackets.

```
students = np.array("Dilay", "Arda", "Ozgur", "Eylul"]
print(students[0])
```

- Are subscripting expressions in element references range checked?

Python is checking index range in arrays. When we use negative index, Python will convert it if that index number is less than length. If we try to print the index that does not exist, we will see the IndexError.

```
print(students[-1])
print(students[-2])
print(students[-3])
print(students[-4])
print(len(students))
# print(students[9]) ---> IndexError: list index out of
range
```

- When are subscript ranges bound?
- When does allocation take place?

Python supports heap-dynamic arrays. So that, binding array subscript and allocating storage is dynamic and done at run time, they can change during execution time.

- Are ragged or rectangular multidimensional arrays allowed, or both?

Both of them is allowed.

```
arr1 = np.array([[5, 10, 15],
                 [20, 25, 30],
                 [35, 40, 45]])
```

```
arr2 = np.array(['A', 'B', 'C'],
                 ['D', 'E'],
                 ['F'])
```

- Can array objects be initialised?

Yes.

```
colors = np.array("blue", "pink", "white", "black")
print(colors)
```

- Are any kind of slices supported?

Yes.

```
colors_vol2 = np.array("red", "green", "orange",
"yellow")
print(colors_vol2[1:3])
print(colors_vol2[2:4])
print(colors_vol2[1::2])
```


- Which operators are provided?

Identity operators which are "is", "is not" and also "+", "-", "*", "/" operators are provided.

```
nums1 = np.array[1, 2, 3, 4, 5]
nums2 = np.array[6, 7, 8, 9]
print(nums1 + nums2)
print(nums2 - nums1)
print(nums1 * nums2)
print(nums1 / nums2)
print(nums1 is nums2)
print(nums1 is not nums2)
```

RUST

- What types are legal for subscripts?

Array subscript type of Rust is integer and index syntax is brackets.

```
let array: [i32; 5] = [1, 2, 3, 4, 5];
println!("1. element of the array: {}", array[0]);
```

- Are subscripting expressions in element references range checked?

Rust is checking index range in arrays. When we use negative index, Python will convert it if that index number is less than length. If we try to print the index that does not exist, we will see the `IndexError`.

```
println!("{:?}", array[10]); ---> index out of range
println!("{:?}", array[4]);
```

- When are subscript ranges bound?
- When does allocation take place?

Rust supports stack-allocated arrays. So that, binding array subscript is static. The storage allocation is done at compile time.

- Are ragged or rectangular multidimensional arrays allowed, or both?

Both of them is allowed.

```
let mut raggedArray = [[0u8; 2]; 2];
raggedArray[0][0] = 5;
raggedArray[0][1] = 10;
raggedArray[1][0] = 15;
raggedArray[1][1] = 20;

let mut multiArray = [[0u8; 2]; 3] = [[1,2],[4,5],
[6,7]];
```

- Can array objects be initialised?

Yes.

```
let arrExample: [i32; 3] = [1, 2, 3];
```

- Are any kind of slices supported?

Yes.

```
let array_vol2: [i32; 5] = [1, 2, 3, 4, 5];
let slice = &array_vol2[1..3];
println!("{:?}", &slice[..]);
```

- Which operators are provided?

Identity operators which are "is", "is not" and also "+", "-", "*", "/" operators are provided.

```
let nums1: [i32; 5] = [1, 2, 3, 4, 5];
```

Question 1:

The most functional programming language in my opinion is Python. Because its readability and writability is very good. Multidimensional arrays can be created and arrays can be easily modified and manipulated by arithmetic operators. In addition to this, the feature that I would like to have in Dart but also in Python is that arrays can be defined as fixed size and dynamic. In addition, the feature of reading negative indexes in Python, not in other languages, makes this programming language more functional. The fact that the basic commands are not difficult to write increases its functionality. The two languages I would not prefer for array operations are Rust and Dart. I think the readability of the language is not good enough. In addition to all of these ideas, slicing methods in Python language are more flexible than the other programming languages. Because a programmer can use “::” sign to shift the elements or take a block of the array and manipulate them with ease. For example, it took a lot of time to understand the slicing methods in Dart and Rust. In my opinion, when I want to write more complicated array operations, it is better to use Python language.

Question 2:

I didn't know any of the languages we needed to write and learn in this assignment until I started it. For this reason, in order to start from scratch, I first repeated the 6th chapter on the slides and did research on the links on the course site. My learning strategy is all about trial and error. After writing the code by just thinking and reasoning, I will most likely get an error on every line. Then I copy the errors I get in each line and search on the internet. With the answers I read in various forums, a logic starts to form in my head and I understand the reason for the error I get. I'll try a few more cases to make sure the error is actually caused by what I'm thinking. In this way, I try to reach the right result by doing line by line analysis. Of course, I don't just

use the search engine when I do my analysis. In this process, I review the notes I took in class and take a quick glance at our textbook. If it is a subject that I usually start from scratch, I first start the research from the internet. Trying to understand from the book when there is no shaping in my head usually confuses me even more. In cases where I cannot make any progress, I consult a friend or relative who knows the subject. For example, I consulted my friend because I couldn't understand the reason for the error I got in Dart in this homework, despite thinking for hours. No matter how much I read and research, my best way of learning is to learn by discussing and analysing with someone. But since consulting with someone while I'm working on the code also prevents me from developing my ideas, I usually complete my codes by trying to write it on my own. Here are the links of some sites whose information I have benefited from in this assignment:

- <https://doc.rust-lang.org/rust-by-example/primitives/array.html>
- <https://www.educative.io/edpresso/how-to-create-an-array-in-dart>
- <https://en.wikipedia.org/wiki/>

Comparison_of_programming_languages_%28array%29

- <https://www.tutorialsteacher.com/javascript/javascript-array>
- <https://numpy.org/doc/stable/reference/arrays.ndarray.html>
- <https://www.askpython.com/python/array/initialize-a-python-array>
- https://www.tutorialspoint.com/php/php_arrays.htm