ST0523 Fundamentals of Programming

Topic 5 Arrays

Topic 5 Arrays

- Purpose of using array in programming
- Declaring array reference variables and creating arrays
- Initialize values in an array
- Store and process an array
- Using methods with array arguments and return type

Why Use Arrays?

Compute Average Daily Temperature Using Variables

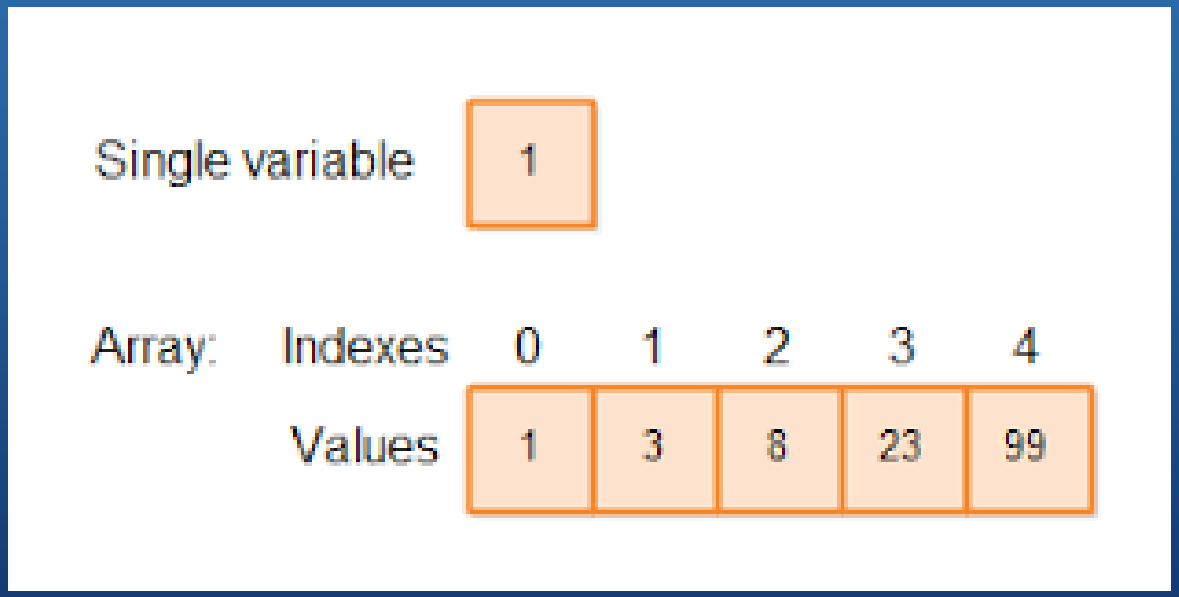
```
var temp1 = 28.5;
var temp2 = 30.6;
var temp3 = 29.2;
var temp4 = 30.8;
var temp5 = 31.2;
var temp6 = 30.2;
var temp7 = 29.5;
var avgTemp;

avgTemp = (temp1 + temp2 + temp3 + temp4 +
           temp5 + temp6 + temp 7) / 7;

console.log("Avg temp for the week: " +
            avgTemp)
```

What if you have to find the average temp for 365 days?

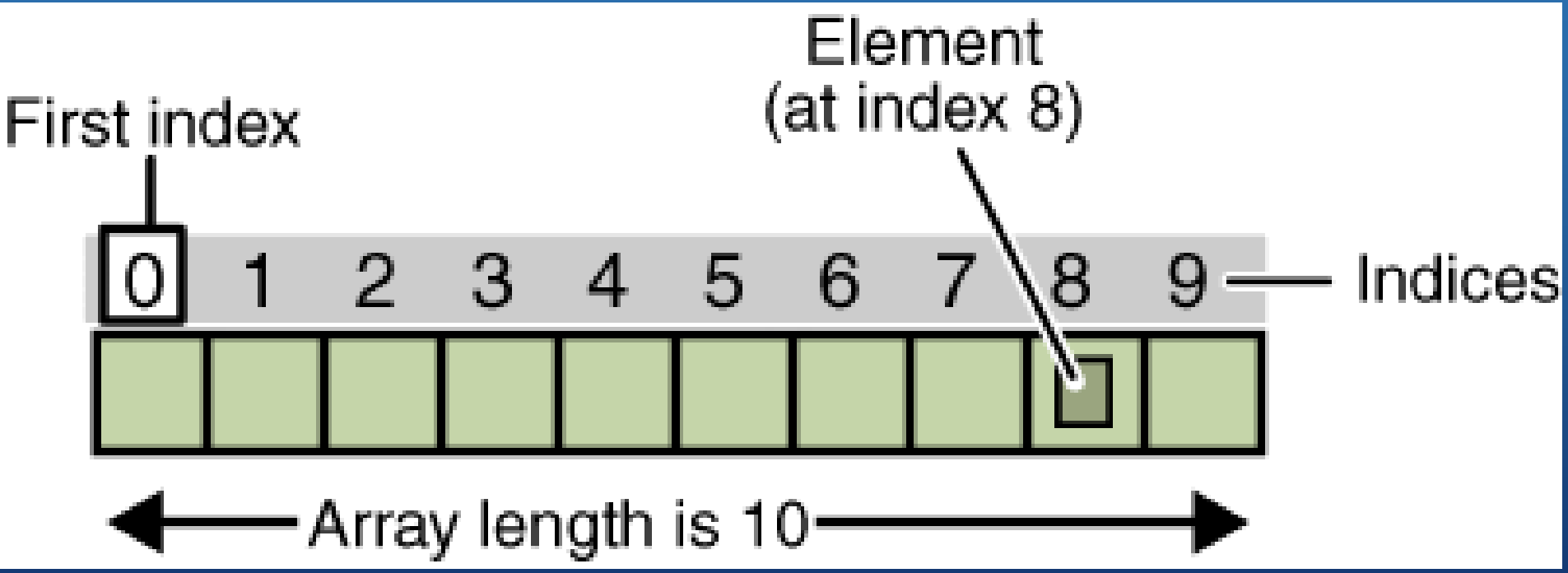
Is there a more efficient way?

Yes. We can use an array to improve the management and processing of data items.

Different visual representation of Arrays:

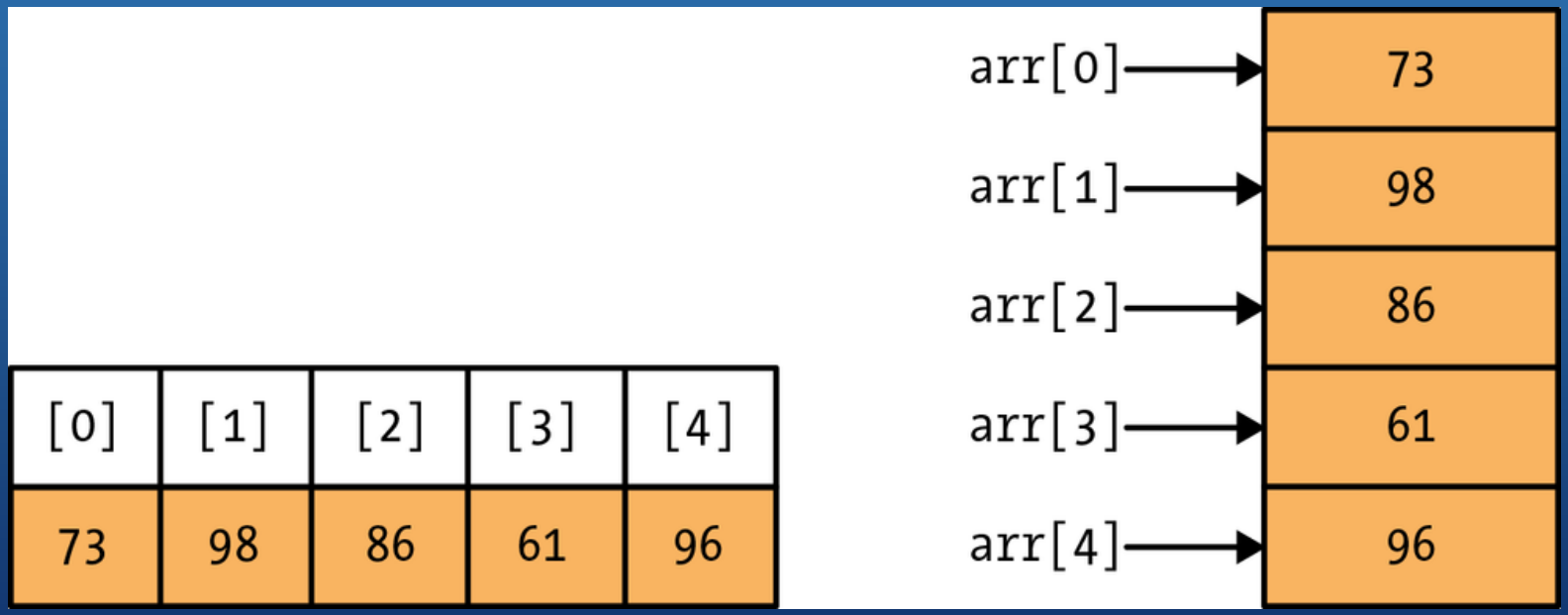


Different visual representation of Arrays:



Different representation → horizontal or vertical (same)

It doesn't matter….

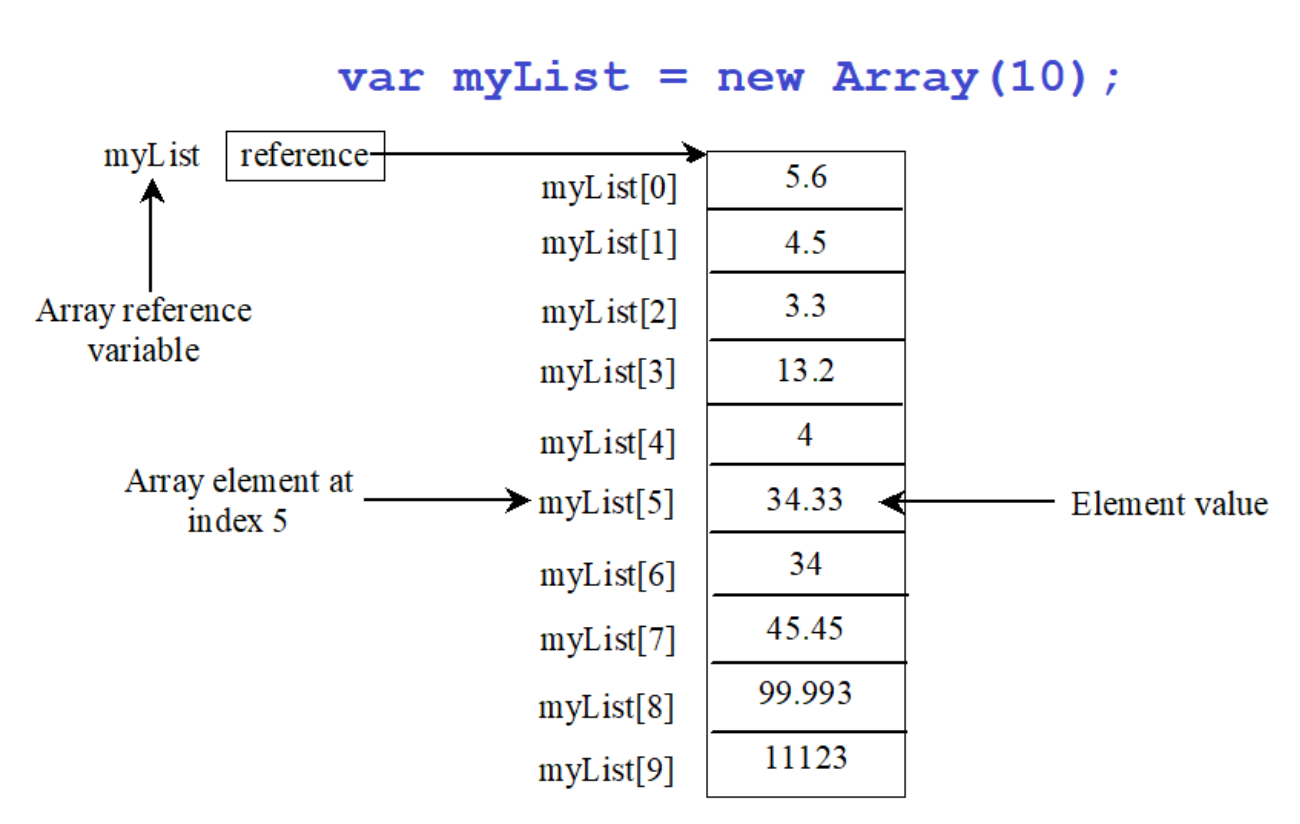


What's the array size?
What does the 3rd element contain?

Can you think of daily living, office setting, schools, etc……where your can use arrays to store data?

Introducing Arrays

An array is a data structure that represents a collection of the same or different type of data.



Creating an Array

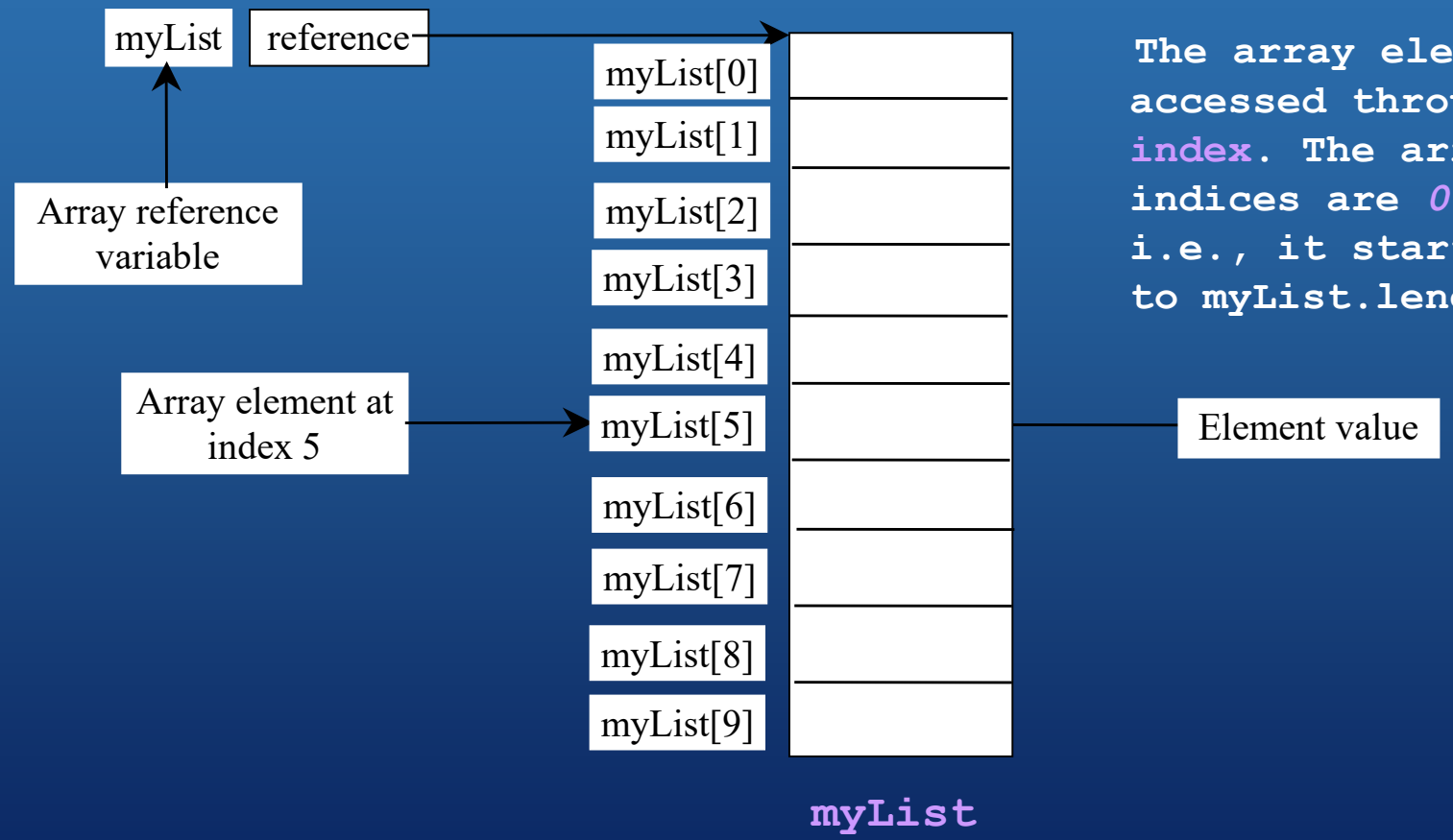
```
var arrayRefVar = new Array(size);
Example:
   var myList = new Array(10);
```

The above statement does two things:

1) Creates an array object with 10 elements.
2) Assigns the reference of the newly created array to the variable arrayRefVar.

Creating an Array

```
var myList = new Array(10);
```



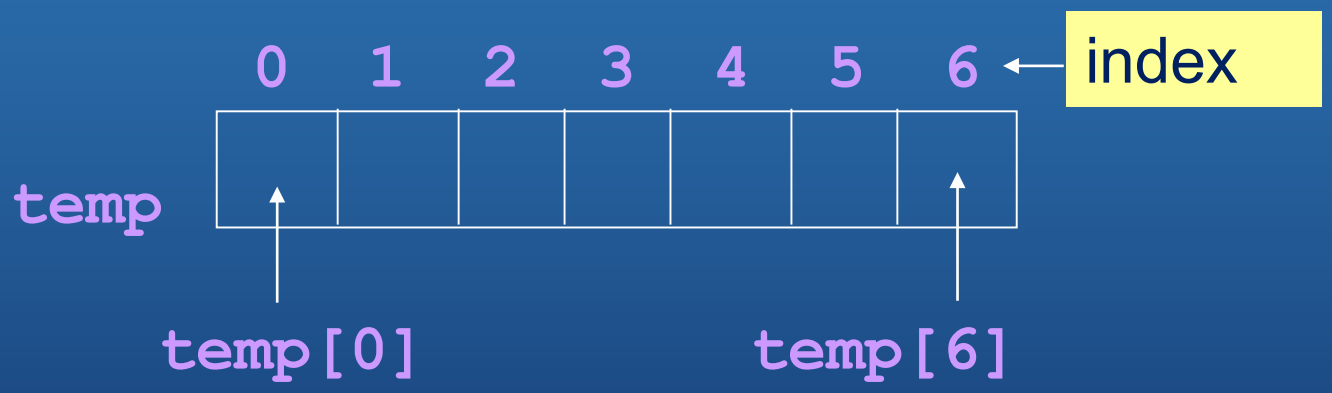
The array elements are accessed through the *index*. The array indices are *0-based*, i.e., it starts from 0 to myList.length-1.

0	first
9	last

Example

- Declare an array to store the temperature of 7 days:

```
var temp = new Array(7);
```



- The length of the array is determined by the number of elements in the array
- Length is a property of an array object.

  E.g. `console.log(temp.length);` //prints 7
- The index number always starts with 0
- Last element index is 1 less than array size

Creating an Array

An array can be created empty, with a length or with elements.

```
//Empty array without a length
var a1 = new Array();

//Empty array with a length
var a2 = new Array(5);

//Another way to create an empty array without a length
var a3 = [];

//Array with elements
var a4 = ["Cat", "Dog", "Hamster", "Chinchilla"];

//Another way to create an array with elements
var a4 = new Array("Cat", "Dog", "Hamster", "Chinchilla");
```

Array Element Data Type

- An element inside an array can be of any type, and *different elements of the same array can be of different types*: string, boolean, even objects or other arrays.

- This means it is possible to create an array that has a string in the first position, a number in the second and a Boolean in the third as follows:

```
var myArray = ["Hello World", 88.88, false];
var omg = ["wow!",  myArray];
```

Accessing Array Element

- Array elements are accessed through the *index*.
- The array index starts from 0 to arrayRefVar.length - 1.
- E.g. myList holds 10 elements and the indices are from 0 to 9.
- Each element in the array is represented using the following syntax, known as an *indexed variable*:

```
arrayRefVar[index];
```

- E.g.

```
x = [44, 55, 88, 99];
console.log(x[2]);    //prints 88
```

Accessing Array Element

- After an array is created, an indexed variable can be used in the same way as a regular variable.

- For example, the following code adds the value in myList[0] and myList[1] to myList[2].

```
myList[2] = myList[0] + myList[1];
```

Let's Try It!

Which of the following statement(s) can be used to create a new array?

Multiple Choice

A. var animalList = Array new(4);
B. var animalList = new array();
C. var animalList = ["Leopard", 12, "Puma", 15];
D. var animalList = new ("Lion", "Tiger");
E. var animalList = new array(2);

Let's Try It!

Which of the following code retrieve the size for array animalList?

Multiple Choice

A. animalList.length ( )
B. animalList.size ( )
C. animalList.length
D. animalList.len

Let's Try It!

Given the following array, which statement will print out "Lion"?

var animalList = ["Tiger", "Leopard", "Puma", "Lion"];

Multiple Choice

A. console.log (animalList [4]);
B. console.log (animalList [animalList.length – 1]);
C. console.log (animalList [animalList.length]);
D. console.log (animalList [animalList.length – 2]);

Default Values

- When an array is created with a length, its elements are assigned the default value of `undefined`.
- In JavaScript, the undefined value indicates that a variable has not been initialized (has not been assigned a value).

For example:

```
var a1 = new Array(3);
var x;
console.log(a1[0]); //prints undefined
console.log(a1[1]); //prints undefined
console.log(a1[2]); //prints undefined

console.log(x); //also prints undefined
```

Let's Try It!

What will the following code return?

```
var numArr = [ "1", 2, 9,  "4", 16 ];
console.log( numArr["4"] );
```

a) "4"
b) 16
c) undefined
d) None of the above.

Multiple Choice

Let's Try It!

What will the following code return?

```
var numArr = [ "1",   2,   9,  "4",  16 ];
console.log( numArr[1]  + (numArr[2]  + numArr[3]) );
```

a) 15
b) 129
c) 213
d) 294

Multiple Choice

Let's Try It!

What is the output for the following codes?

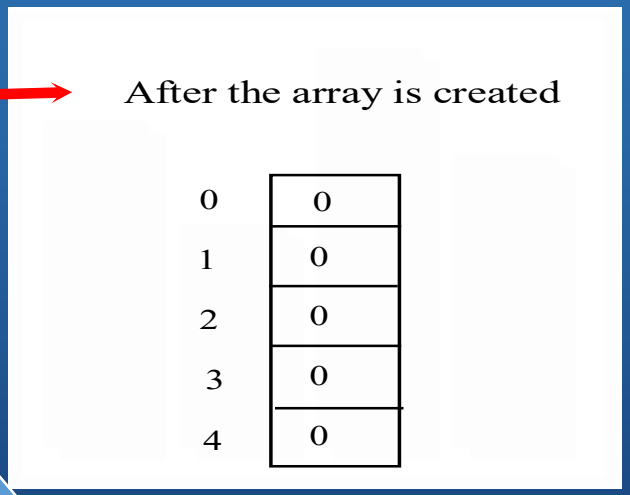
```
var myArr = new Array(4);
console.log (myArr[0] + "myArr[2]");
```

a) 02
b) myArr[0]myArr[2]
c) undefinedmyArr[2]
d) undefinedundefined

Multiple Choice

Array Example

```
var values = [0,0,0,0,0];
for (var i = 1; i < 5; i++) {
  values[i] = i + values[i-1];
}
values[0] = values[1] + values[4];
```



Declare array variable values, create an array, and assign its reference to values

Array Example

```
var values = [0,0,0,0,0];
for (var i = 1; i < 5; i++) {
  values[i] = i + values[i-1];
}
values[0] = values[1] + values[4];
```

After the array is created

0	0
1	0
2	0
3	0
4	0

i becomes 1

`i = 1`

Array Example

```
var values = [0,0,0,0,0];
for (var i = 1; i < 5; i++) {
  values[i] = i + values[i-1];
}
values[0] = values[1] + values[4];
```

After the array is created

0	0
1	0
2	0
3	0
4	0

i (which is 1) is less than 5

Array Example

```
var values = [0,0,0,0,0];
for (var i = 1; i < 5; i++) {
  values[i] = i + values[i-1];
}
values[0] = values[1] + values[1];
```

After the array is created

0	0
1	1
2	0
3	0
4	0

After this line is executed, value[1] is 1

Array Example

```
var values = [0,0,0,0,0];
for (var i = 1; i < 5; i++) {
  values[i] = i + values[i-1];
}
values[0] = values[1] + values[1];
```

After the array is created

0	0
1	1
2	0
3	0
4	0

After i++, i becomes 2

Array Example

```
var values = [0,0,0,0,0];
for (var i = 1; i < 5; i++) {
  values[i] = i + values[i-1];
}
values[0] = values[1] + values[1];
```

After the array is created

0	0
1	1
2	0
3	0
4	0

i (which is 2) is less than 5

Array Example

```
var values = [0,0,0,0,0];
for (var i = 1; i < 5; i++) {
  values[i] = i + values[i-1];
}
values[0] = values[1] + values[1];
```

After the array is created

0	0
1	1
2	3
3	0
4	0

After this line is executed,
values[2] is 3 (2 + 1)

Array Example

```
var values = [0,0,0,0,0];
for (var i = 1; i < 5; i++) {
  values[i] = i + values[i-1];
}
values[0] = values[1] + values[1];
```

After the array is created

0	0
1	1
2	3
3	0
4	0

After this, i becomes 3.

Array Example

```
var values = [0,0,0,0,0];
for (var i = 1; i < 5; i++) {
  values[i] = i + values[i-1];
}
values[0] = values[1] + values[4];
```

After the array is created

0	0
1	1
2	3
3	0
4	0

i (which is 3) is less than 5

Array Example

```
var values = [0,0,0,0,0];
for (var i = 1; i < 5; i++) {
  values[i] = i + values[i-1];
}
values[0] = values[1] + values[1];
```

After the array is created

0	0
1	1
2	3
3	6
4	0

After this line is executed, value[3] is 6

Array Example

```
var values = [0,0,0,0,0];
for (var i = 1; i < 5; i++) {
  values[i] = i + values[i-1];
}
values[0] = values[1] + values[1];
```

After the array is created

0	0
1	1
2	3
3	6
4	0

After i++, i becomes 4

Array Example

```
var values = [0,0,0,0,0];
for (var i = 1; i < 5; i++) {
  values[i] = i + values[i-1];
}
values[0] = values[1] + values[4];
```

After the array is created

0	0
1	1
2	3
3	6
4	0

i (which is 4) is less than 5

Array Example

```
var values = [0,0,0,0,0];
for (var i = 1; i < 5; i++) {
  values[i] = i + values[i-1];
}
values[0] = values[1] + values[1];
```

After the array is created

0	0
1	1
2	3
3	6
4	10

After this line is executed, value[4] is 10

Array Example

```
var values = [0,0,0,0,0];
for (var i = 1; i < 5; i++) {
  values[i] = i + values[i-1];
}
values[0] = values[1] + values[1];
```

After the array is created

0	0
1	1
2	3
3	6
4	10

After i++, i becomes 5

Array Example

```
var values = [0,0,0,0,0];
for (var i = 1; i < 5; i++) {
  values[i] = i + values[i-1];
}
values[0] = values[1] + values[4];
```

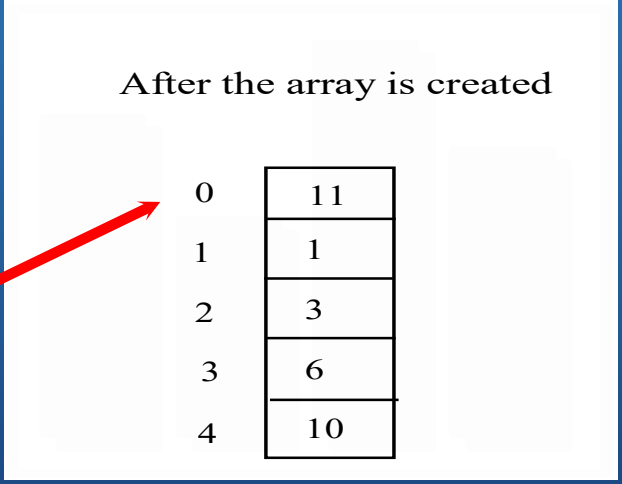
After the array is created

0	0
1	1
2	3
3	6
4	10

i (which is 5) is NOT less than 5, hence is false and exit the for loop

Array Example

```
var values = [0,0,0,0,0];
for (var i = 1; i < 5; i++) {
  values[i] = i + values[i-1];
}
values[0] = values[1] + values[4];
```



After this line, values[0] is 11 (1 + 10)

Google the meaning of ‘Dynamic’

Dictionary

Definitions from Oxford Languages · Learn more

Search for a word

dynamic

/dʌɪˈnamɪk/

See definitions in:

All | Philosophy | Physics | Linguistics | Technology | Music

adjective

1. (of a process or system) characterized by constant change, activity, or progress.
   "a dynamic economy"
2. (of a person) positive in attitude and full of energy and new ideas.
   "a dynamic young advertising executive"

   Similar: energetic | spirited | active | lively | zestful | vital | vigorous

noun

1. a force that stimulates change or progress within a system or process.
   "evaluation is part of the basic dynamic of the project"
2. MUSIC
   another term for dynamics (sense 3).

https://www.google.com/search?q=dynamic&rlz=1C1GCEB_enSG943SG943&oq=dynamic&aqs=chrome..69i57j0i433i512j0i131i433i512l2j0i512l6.1481j0j7&sourceid=chrome&ie=UTF-8

Dynamic Array

In fact, arrays in JavaScript are all dynamic in size. This means you can create an empty array without specifying its length. Whenever you assign a new value to the array, its size then will grow dynamically.

```
var list1 = new Array();  //array without length
for (var i=0; i < 7 ; i++ ) {
    list1[i] = "Hello" + i;    //or. . .
  //list1.push("Hello" + i);  //push = add last
}
```

Hello0	Hello1	Hello2	Hello3	Hello4	Hello5	Hello6
0	1	2	3	4	5	6

Let’s see how the .push works !

```
var animal = new Array("Cat", "Dog", "Hamster", “Rat");

console.log("Length of animal array BEFORE push : " + animal.length);
console.log("Element animal[0] : " + animal[0]);
console.log("Element animal[1] : " + animal[1]);
console.log("Element animal[2] : " + animal[2]);
console.log("Element animal[3] : " + animal[3]);

animal.push("Rabbit");

console.log("\nLength of animal array AFTER push : " + animal.length);
console.log("Element animal[0] : " + animal[0]);
console.log("Element animal[1] : " + animal[1]);
console.log("Element animal[2] : " + animal[2]);
console.log("Element animal[3] : " + animal[3]);
console.log("Element animal[4] : " + animal[4]);

console.log();
console.log(animal);
```

Results of the .push !

```
Length of animal array BEFORE push : 4
Element animal[0] : Cat
Element animal[1] : Dog
Element animal[2] : Hamster
Element animal[3] : Rat

Length of animal array AFTER push : 5
Element animal[0] : Cat
Element animal[1] : Dog
Element animal[2] : Hamster
Element animal[3] : Rat
Element animal[4] : Rabbit

[ 'Cat', 'Dog', 'Hamster', 'Rat', 'Rabbit' ]
```

Exercise : for loop

```
var animal = new Array("Cat", "Dog", "Hamster", “Rat");

console.log("Length of animal array BEFORE push : " + animal.length);
console.log("Element animal[0] : " + animal[0]);
console.log("Element animal[1] : " + animal[1]);
console.log("Element animal[2] : " + animal[2]);
console.log("Element animal[3] : " + animal[3]);

animal.push("Rabbit");

console.log(“\nLength of animal array AFTER push : " + animal.length);
console.log("Element animal[0] : " + animal[0]);
console.log("Element animal[1] : " + animal[1]);
console.log("Element animal[2] : " + animal[2]);
console.log("Element animal[3] : " + animal[3]);
console.log("Element animal[4] : " + animal[4]);

console.log();
console.log(animal);
```

Write a for loop to print the elements

Let's Try It!

What is the output of the following statements?

```
var animalList = ["Leopard", "Puma", "Lion"];
animalList.push( "Cheetah");
console.log( animalList );
```

a) ["Cheetah", "Leopard", "Puma", "Lion"];
b) ["Lion", "Leopard", "Puma", "Cheetah"];
c) ["Leopard", "Puma", "Lion", "Cheetah"];
d) ["Puma", "Leopard", "Lion", "Cheetah"];

Multiple Choice

Let's Try It!

What is the output of the following statements?

```
var animalList = ["Tiger", "Puma", "Lion"];
animalList.pop( );
console.log( animalList );
```

a) ["Lion", "Puma", "Tiger"];
b) ["Tiger", "Lion"];
c) ["Puma", "Lion"];
d) ["Tiger", "Puma"];

Multiple Choice

Processing Arrays

When processing array elements, we often use a for loop.

The following loop initializes the array myList with random values between 1 and 100.

```
var myList = new Array();
for (var i=0; i <= 5; i++ ) {
    myList.push(Math.floor(Math.random() * 100) + 1);
}
```

Processing Arrays – Printing arrays

To display array content, you can use for-loop as follows:

```
for (var i=0; i < myList.length; i++ ) {
   console.log( myList[i]);
}
```

```
79
84
31
26
58
```

Or simple execute…

```
console.log(myList);
```

```
[79, 84, 31, 26, 58]
```

Processing Arrays – Computing sum

To sum all element in an array, use a variable named total to store the sum. Initially total is 0. Add each element in the array to total, using for loop.

```
var total =0;
for (var  i=0; i < myList.length; i++ ) {
   total += myList[i];
}
```

```
var avg = total / myList.length;
```

To calculate the average of the array, use a variable named avg.

Let's Try It!

What is the output of the following codes?

```
var numArr = new Array(10);
for(var i = 0 ; i < numArr.length; i+=2) {
        numArr[i] = i + 1;
}
console.log(numArr[2], numArr[9]);
```

a) [2,9]
b) [2, undefined]
c) [3,9]
d) [3, undefined]

Multiple Choice

Let's Try It!

What is the output of the following codes?

```
var numArr = new Array(5, 1, 3, -5, 8, -4);
for (var k = 1; k < numArr.length; k += 2) {
        numArr[k] +=numArr[k];
}
console.log(numArr);
```

a) [5, 2, 3, -10, 8, -8]
b) [10, 1, 6, -5, 16, -4]
c) [5, 4, 6, -10, 16, -8]
d) [10, 4, 6, -10, 16, -8]

Multiple Choice

Processing Arrays – Finding the largest element

To find the largest element, use a variable named max to store the largest element. Initially max to myList[0].

Compare each element in myList with max, update max if the element is greater than max.

```
var max = myList[0];
for (var  i=1; i < myList.length; i++ ) {
  if ( myList[i] > max)
     max = myList[i];
}
```

Processing Arrays - Finding the smallest index of the largest element

To find the index of the largest element of an array, add some code from the previous slide:

```
var max = myList[0];
var indexOfMax;
for (var  i=1; i < myList.length; i++ ) {
   if ( myList[i] > max){
      max = myList[i];
      indexOfMax = i;
   }
}
```

Can you replace `if ( myList[i] > max )` with `if ( myList[i] >= max )`?

Example Testing Arrays

Objective: The program receives 6 numbers from the keyboard, finds the largest number and counts the occurrence of the largest number entered from the keyboard.

Suppose you entered 3, 5, 2, 5, 5, and 5, the largest number is 5 and its occurrence count is 4.

Example Testing Arrays

```
var input = require('readline-sync');

const TOTAL_NUMBERS = 6;
var numbers = new Array(TOTAL_NUMBERS);

//read all numbers
for(var i = 0 ; i < numbers.length ; i++) {
    numbers[i] = input.question("Enter a number: ");
}
```

Example Testing Arrays

```
//find largest
var max = numbers[0];
for(var i = 1 ; i < numbers.length ; i++) {
    if(max < numbers[i])
        max = numbers[i];
}
```

```
//find the occurrence of the largest number
var count = 0;
for(var i = 0 ; i < numbers.length ; i++) {
    if(numbers[i] == max)
        count++;
}
```

Example Testing Arrays

```
//find largest
var max = numbers[0];
for(var i = 1 ; i < numbers.length ; i++) {
    if(max < numbers[i])
        max = numbers[i];
}
```

```
//find the occurrence of the largest number
var count = 0;
for(var i = 0 ; i < numbers.length ; i++) {
    if(numbers[i] == max)
        count++;
}
```

Let's recap Arrays!

var numArr = [12, 32, 5, 9, 0, 10, 4];

Write the value for the following given the array above. Treat each part independently.

a) numArr[0]
b) numArr.length
c) numArr[3*3]
d) numArr[4]
e) numArr[7]
f) numArr[2]+numArr[5]
g) NumArr[6]
h) numArr[2] +=numArr[3]
i) numArr[6] = numArr[1] – numArr[3]

ST0502 Fundamentals of Programming

Topic 5 Arrays

Summary
Topic 5 Arrays

- Purpose of using array in programming
- Declaring array reference variables and creating arrays
- Initialize values in an array
- Store and process an array
- Using methods with array arguments and return type

SP SCHOOL OF Computing