

# Lab #3

## Contents

<b>1</b>	<b>Vectors</b>	<b>1</b>
1.1	Array . . . . .	1
1.2	Vector . . . . .	2
1.3	Motivation . . . . .	2
1.4	Row vectors . . . . .	2
1.5	Column vectors . . . . .	2
1.6	Operations with vectors . . . . .	3
1.7	Other ways to create vectors . . . . .	4
<b>2</b>	<b>Task #3</b>	<b>4</b>
<b>3</b>	<b>For loops</b>	<b>5</b>
3.1	Hello world 20 times . . . . .	5
3.2	Hello world with numbers . . . . .	6
3.3	Sum of numbers from 1 to 10 . . . . .	6
3.4	Sum of all elements in a vector . . . . .	6
<b>4</b>	<b>Task #4</b>	<b>6</b>
<b>5</b>	<b>Task #5</b>	<b>7</b>
<b>6</b>	<b>Task #6</b>	<b>7</b>

Created latex link.

## 1 Vectors

### 1.1 Array

An array, is a data structure consisting of a collection of elements (values or variables), each identified by at least one array index or key.

## 1.2 Vector

Is one-dimensional array.

## 1.3 Motivation

What if we have a collection of numbers, for example parsed from a column of an Excel table. How we can work with them in Matlab? Store in separate variables?

```
1 driverWithLicenseNoExperienceSalary = 2000;  
2 driverWithLicenseOneYearOfExperienceSalary = 3000;  
3 driverWithLicenseTwoYearsOfExperienceSalary = 4000;  
4 driverWithLicenseThreeYearsOfExperienceSalary = 5000;  
5 driverWithLicenseFourYearsOfExperienceSalary = 6000;  
6 ...
```

Too much time, not convenient, too much space, etc.

What if the size of the collection is variable? In one Excel file we have 100 rows, in second file we have 200 rows, etc.

## 1.4 Row vectors

```
1 driverSalary = [1000, 2000, 3000, 4000];  
2  
3 driverSalary  
4 driverSalary(1)  
5 driverSalary(2)  
6 driverSalary(end)  
7 driverSalary(1:3)  
8 driverSalary(:)  
9  
10 driverSalary(2) = driverSalary(2) - 200;  
11 driverSalary  
12  
13 length(driverSalary)  
14 size(driverSalary)
```

## 1.5 Column vectors

The same as row vectors but column vectors

```

1 driverSalary = [1000; 2000; 3000; 4000];
2
3 driverSalary
4 driverSalary(1)
5 driverSalary(2)
6 driverSalary(end)
7 driverSalary(1:3)
8 driverSalary(:)
9
10 length(driverSalary)
11 size(driverSalary)

```

## 1.6 Operations with vectors

```

1 % change all elements of a vector
2 driverSalary = driverSalary + 1000
3 driverSalary = driverSalary - 1000
4 driverSalary = driverSalary * 1000
5 driverSalary = driverSalary / 1000
6
7 array1 = [10, 20, 30];
8 array2 = [30, 20, 10];
9
10 % element-wise operations
11 array1 + array2
12 array1 - array2
13 array1 .* array2
14 array1 ./ array2
15
16 % vectors concatenation
17 [array1, array2]
18
19 % transpose
20 array1'
21 array1' '
22
23 % concatenation again
24 [array1'; array2]

```

## 1.7 Other ways to create vectors

```
1 1:1:10
2 % or
3 1:10
4
5 1:2:10
6
7 -1:-1:-10
8
9 1:-1:10 % empty vector -- we cannot create a vector
          from 1 to 10
10         % with step -1
11
12 linspace(1, 10, 5)
13
14 zeros(1, 10)
15 ones(1, 10)
16 rand(1, 10)
17
18 % and many other
```

## 2 Task #3

Create a new script.

- Create a vector `managerSalaries` with numbers from 1000 to 9000 with step 1000 ([1000, 2000, 3000, ..., 9000]).
- Display the value of the 4th element of the vector `managerSalaries`.
- Decrease the value of the 5th element of the vector `managerSalaries` by 300.
- Display the elements of the vector `managerSalaries` on positions 3 to 7.
- Create a vector `socialismSalaries` with 10 numbers all of which are 2000. ([2000, 2000, 2000, ..., 2000]).

Results should be:

```

1 Value of the 4th element of managerSalaries is:
2     4000
3
4 Value of the 5th element of managerSalaries before
5     change is:
6     5000
7
8 Value of the 5th element of managerSalaries after
9     change is:
10    4700
11
12 Elements of managerSalaries with indices 3-7 are:
13    3000    4000    4700    6000
14          7000
15
16 Vector socialismSalaries is:
17 Columns 1 through 5
18    2000    2000    2000    2000
19          2000
20
21 Columns 6 through 10
22    2000    2000    2000    2000
23          2000

```

## 3 For loops

See MIT slides #2.

### 3.1 Hello world 20 times

```

1 clear;
2
3 for i=1:20
4     disp('Hello, world!');
5 end

```

### 3.2 Hello world with numbers

```
1 clear;
2
3 for i=1:20
4     disp('Hello, world!');
5     disp(i);
6 end
```

### 3.3 Sum of numbers from 1 to 10

```
1 clear;
2
3 to = 10;
4 sum = 0;
5
6 for i = 1:to
7     sum = sum + i;
8 end
```

### 3.4 Sum of all elements in a vector

```
1 clear;
2
3 vector = [1, 20, -3, 5, 6];
4 vector_length = length(vector);
5
6 sum = 0;
7
8 for i = 1:vector_length
9     element = vector(i);
10    sum = sum + element;
11 end
```

## 4 Task #4

Write a program which prints lyrics of a Swedish version of the song "99 bottles of beer".

The lyrics of the song are as follows:

```
99 bottles of mjölk on the wall, 99 bottles of mjölk.  
Take one down, pass it around, 98 bottles of mjölk on the wall
```

```
98 bottles of mjölk on the wall, 98 bottles of mjölk.  
Take one down, pass it around, 97 bottles of mjölk on the wall
```

...

```
No more bottles of mjölk on the wall, no more bottles of mjölk.  
We've taken them down and passed them around; now we're healthy and strong!
```

**Hint:** to print a line of text with a number you can use `num2str()` function as follows.

```
bottles = 99;
```

```
disp([num2str(bottles) ' bottles of mjölk,' num2str(bottles) ' bottles of mjölk.']);
```

This code will print a string "99 bottles of mjölk on the wall, 99 bottles of mjölk.". Do not forget to replace variable `bottles` with your variable name. Print the second line using the same idea.

## 5 Task #5

Write a script which computes the product of all elements of the vector `v` for any given vector `v`.

For example, for a vector `v = [2, 3, 5, 6]` you should get 180.

## 6 Task #6

Write a script which prints all pairs of elements of a given vector `v`

For example, for a vector `v = [1, 10, 20]`, print

```
1 [1, 1]  
2 [1, 10]  
3 [1, 20]  
4 [10, 1]  
5 [10, 10]
```

```
6 [10, 20]
7 [20, 1]
8 [20, 10]
9 [20, 20]
```

**Hint:** if you want to print two numbers **a** and **b** on the same line, you can try to use `disp([a, b]);`.

**Hint 2:** You need to use one for loop inside of the other.