# Review of Lecture 6

- **Create a matrix in MATLAB**

```
% Create a 2x3 matrix
>> M = [7, 2, 3; 9, 5, 1];
>> M=[7 2 3; 9 5 1];
% Create a 3x3 matrix using enter key
>> M = [7 2 3
9 5 1
7 4 9]
% arranged rows
>> M = [1:3; 4:6; 7:2:11]
```

  - **Some usual matrices**

`zeros(), ones(), eye()`

  - **`rand()` function**

`rand, rand(n), rand(n,m)`
`randi, randi(i), rand(i,n),randi(i,n,m)`

- **Operations on matrices**

`M(x), V(i,j), M(i:ii,j:jj)`
`M(i,:); M(:,j),`
`M(i,:) = [], M(:,j) = [], M(i,j) = a`
`M(i,:) = V, M(:,j) = V,`
`M = [M, v], M = [M; v]`

- **Important functions for matrices**

`size(), numel(), det(), trace(), inv()`

- **Algebraic operations**

`A+B, A-B, A.*B, A./B, A*B, A/B`

Info 3

# Introduction to MATLAB®

**M. Bouzenita**
2nd year Engineer - University of Jijel

## Lecture 7

# Control instructions (if and switch statements)

# 1. Control instructions `if` and `switch`

In MATLAB environments, **`if-else`** and **`switch`** are important **control tasks and conditional statements** that help the users to take decisions based on various conditions.

# 2. `if-else` statement

The **`if-else`** statement is employed to execute various parts of a program based on whether a specified condition is **true** or **false**.

1. **`if`** statement
2. **`if-else`** statement
3. **`if-elseif-else`** statement

# 2.1 `if` statemer

The **`if`** statement is utilized to test and examine a condition.

If the condition is **true**, a program or a set of instructions is executed.

The general form of the **`if`** statement is as follows.

```
if   condition
 <Instructions>
end
```

Calculate the **root square** of a positive number.

**SquareRoot.m**

```
x = input('Please enter a number: ');

if x >= 0

 fprintf('SQRT of %.2f is : %.2f\n', x, sqrt(x));

end
```

```
>> SquareRoot
Please enter a number: 4
SQRT of 4.00 is: 2.00

>> SquareRoot
Please enter a number: -4
>>
```

# 2.2 `if-else` statement

Generally, the **`if`** statement is followed by another statement called **`else`** that allows to run another code or program when the condition in the **`if`** statement is **false**.

```
if  condition
 <Instruction 1>
else
 <Instruction 2>
end
```

Calculate the **absolute** value of a number.

$$|x| = \begin{cases} -x & \text{if } x < 0 \\ x & \text{if } x >= 0 \end{cases}$$

**absolute.m**

```
x = input('Please enter a number: ');
if x < 0
   Absx = -x;
else
   Absx = x;
end
fprintf('Absolute value of %.2f is %.2f \n',x, Absx);
```

# 2.2 `if-elseif-else` statement

In some cases, **`if`** statement can be followed by one or more optional **`elseif`** and an **`else`** statement, which is very useful to test various conditions.

```
if condition 1
   <Instructions 1>
elseif condition 2
   <Instructions 2>
else
   <Instructions 3>
end
```

Determine if a number is positive, negative or null.

**numberType.m**

```
x = input('Please enter a number: ');
if x > 0
    type = 'positive';
elseif x < 0
    type = 'negative';
else
    type = 'null';
end
fprintf('%.2f is a %s number.\n',x, type);
```

# 2.2 if-elseif-else statem

In some cases, **if** statement can be followed by one or more optional **elseif** and an **else** statement, which is very useful to test various conditions.

```
if condition 1
   <Instructions 1>
elseif condition 2
   <Instructions 2>
else
   <Instructions 3>
end
```

Determine if a number i

**numberType.m**

```
x = input('Please enter a num
if x > 0
    type = 'positive';
elseif x < 0
    type = 'negative';
else
    type = 'null';
end
fprintf('%.2f is a %s number.
```

```
>> numberType
Please enter a number: 4
4.00 is a positive number.
>> numberType
Please enter a number: -8
-8.00 is a negative number.
>> numberType
Please enter a number: 0
0.00 is a null number.
```

# 3. `Switch` statement

The mechanism of the **`switch`** statement is based on **selecting one among various blocks to be executed**, this is done according to the value of a specific expression.

```
switch expression
case caseValue1
        < code 1 >
case caseValue2
        < code 2 >
    ...
otherwise
         < code n >    % if expression doesn't match any case
end
```

# 3. Switch s

The mechanism of th
**various blocks to be**
specific expression. In

```
switch expression
case caseValue1
        < code 1 >
case caseValue2
        < code 2 >
    ...
otherwise
         < code n >
end
```

Displays the grade of a student based on their numerical score.

| 0 | Very weak | 5 | Weak | 10 | Middle | 15 | Good | 18 | Excellent | 20 |
|---|---|---|---|---|---|---|---|---|---|---|

grade.m

```
numGrade = input('Enter the numerical grade (0-20): ');
switch true
case numGrade >= 18
        Grade = 'Excellent';
case numGrade >= 15
        Grade = 'Good';
case numGrade >= 10
        Grade = 'Middle';
case numGrade >= 5
        Grade = 'Weak';
case numGrade >= 0
        Grade = 'Very weak';
otherwise
        Grade = 'not a valid numerical grade';
end
fprintf('The letter grade for %.2f is: %s\n', numGrade, Grade);
```

# 3. Switch s

The mechanism of th
**various blocks to be**
specific expression. In

```
switch expression
case caseValue1
        < code 1 >
case caseValue2
        < code 2 >
    ...
otherwise
         < code n >
end
```

Determine the type of a vehicle based on a category

| Road vehicle | Water vehicle | Air vehicle |
|---|---|---|
| • Car<br>• Truck<br>• Bus | • Boat<br>• Ship | • Plane<br>• Helicopter |

**vehicle.m**

```
vehicleType = input('Enter the vehicle type : ', 's');
switch vehicleType
case {'car' , 'truck', 'bus'}
        disp('This is a road vehicle.');
case {'boat' , 'ship'}
        disp('This is a water vehicle.');
case {'plane' , 'helicopter'}
        disp('This is an air vehicle.');
otherwise
       disp('Unknown vehicle type.')
end
```

```
>> vehicle
Enter the vehicle type : plane
This is an air vehicle.
```

# Practice