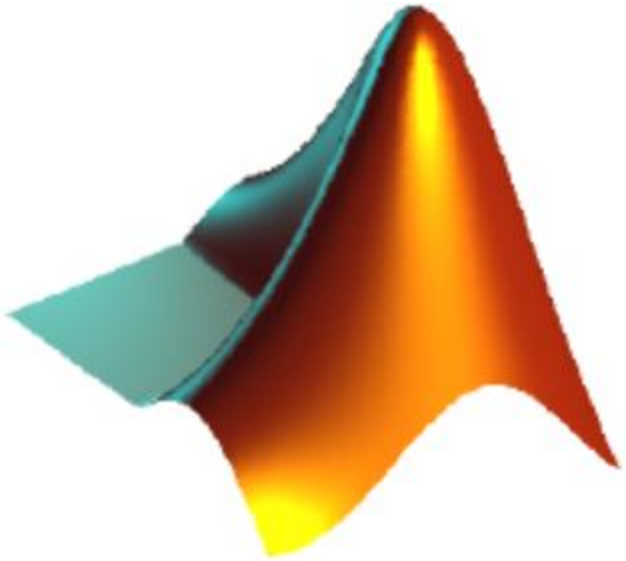
MatLab Programming
Decision Making
(1)



Decision Making

Statement	Description
if ... end statement	An if ... end statement consists of a boolean expression followed by one or more statements.
if...else...end statement	An if statement can be followed by an optional else statement, which executes when the boolean expression is false.
If... elseif...elseif...else...end statements	An if statement can be followed by one (or more) optional elseif... and an else statement, which is very useful to test various conditions.
nested if statements	You can use one if or elseif statement inside another if or elseif statement(s).
switch statement	A switch statement allows a variable to be tested for equality against a list of values.
nested switch statements	You can use one switch statement inside another switch statement(s).

if... end Statement

- An **if ... end** statement **consists** of an **if statement** and a **boolean expression** that is **followed** by **one or more statements**.
- An **if ... end** statement is **delimited** by the **end statement**.
- If the **expression** evaluates to **true**, then the block of **code inside** the **if** statement will be **executed**.
- If the **expression** evaluates to **false**, then the first set of **code after** the **end** statement will be **executed**.
- **Syntax:**

```
if <expression>
        % statement(s) will execute if the boolean expression is true
<statements>
end
```

if... end Statement Example

- Create a script file sp1.m and type the following code:

```
a = 10;
% check the condition using if statement
if a < 20
% if condition is true then print the following
fprintf('a is less than 20\n' );    %print formatted text
end
fprintf('value of a is : %d\n', a);
```

- Run the file, the result will be:

```
>>sp1.m
    a is less than 20
    value of a is : 10
```

Relational and logical operators

Operator	Description
>	Greater than
<	Less than
>=	Greater than or equal to
<=	Less than or equal to
==	Equal to
~=	Not equal to
&	AND operator
|	OR operator
~	NOT operator

Quiz 1

- Create a script file sp2.m that define a variable ‘a’ with value -3. Then it prints negative number if the value of a is negative.

if... end Statement: Quiz Solution

- Create a script file sp2.m that define a variable a with value -3. Then it prints negative number if the value of a is negative.

```
a =-3;
% check the condition using if statement
if a <0
% if condition is true then print the following
fprintf('negative number' );
end
```

- Run the file, the result will be:

```
>>sp2.m
    negative number
```

if... else... end Statement

- **If** statement can be followed by an **optional else** statement, which **executes when** the **expression** is **false**.
- If the boolean **expression** evaluates to **true**, then the **if** block of **code** will be **executed**, **otherwise else** block of **code** will be **executed**.
- **Syntax**

```
if <expression>
% statement(s) will execute if the boolean expression is true
<statement(s)>
else
<statement(s)>
% statement(s) will execute if the boolean expression is false
end
```

if... else... end Statement Example

- Create a script file sp2.m and type the following code:

```
a = 100;
% check the boolean condition
if a < 20
% if condition is true then print the following
fprintf('a is less than 20\n' );
else
% if condition is false then print the following
fprintf('a is not less than 20\n' );
end
fprintf('value of a is : %d\n', a);
```

- Run the file, the result will be:

```
>>sp2.m
   a is not less than 20
   value of a is : 100
```

if...elseif...elseif...else...end Statements

- **If statement** can be followed by **one (or more) optional elseif... and an else** statement, which is very **useful** to **test various conditions.**

- When using if... elseif...else statements, there are few points to keep in mind:
  - An **if** can **have zero or one else's** and it **must** come **after** any **elseif's**.
  - An **if** can **have zero** to **many elseif's** and they **must** come **before** the **else**.
  - Once an **else if succeeds**, **none of** the **remaining elseif's or else's** will be **tested**.

if...elseif...elseif...else...end Statements: syntax

```
if <expression 1>
% Executes when the expression 1 is true
<statement(s)>
elseif <expression 2>
% Executes when the boolean expression 2 is true
<statement(s)>
elseif <expression 3>
% Executes when the boolean expression 3 is true
<statement(s)>
else
% executes when the none of the above condition is true
<statement(s)>
end
```

if... else... end Statement Example

- Create a script file sp3.m and type the following code:

```
a = 100;
%check the boolean condition
if a == 10
% if condition is true then print the following
fprintf('Value of a is 10\n' );
elseif( a == 20 )
% if else if condition is true
fprintf('Value of a is 20\n' );
elseif a == 30
% if else if condition is true
fprintf('Value of a is 30\n' );
else % if none of the conditions is true
fprintf('None of the values are matching\n');
fprintf('Exact value of a is: %d\n', a );
end
```

- Run the file, the result will be:

```
>>sp3.m
   None of the values are matching
   Exact value of a is: 100
```

Quiz 2

calculate the discriminant using the following equation: **b^2 - 4*a*c**

If the resulted value is <0, print(roots are imaginary)

If the resulted value=0,print(roots are repeated)

If the resulted value is>0, print(roots are real)

The Nested if Statements

- **Nested if Statement**: use **one or more if or if-else** statements **inside** another **if or elseif** statement(s).
- **Syntax**

```
if <expression 1>
% Executes when the boolean expression 1 is true
if <expression 2>
% Executes when the boolean expression 2 is true
end
end
```

The Nested if Statements : Example

- Create a script file sp4.m and type the following code:

```
a = 100;  b = 200;
% check the boolean condition
if( a == 100 )
% if condition is true then check the following
if( b == 200 )
% if condition is true then print the following
fprintf('Value of a is 100 and b is 200\n' );
end
end
fprintf('Exact value of a is : %d\n', a );
fprintf('Exact value of b is : %d\n', b );
```

- Run the file, the result will be:

```
>>sp4.m
    Value of a is 100 and b is 200
    Exact value of a is : 100
    Exact value of b is : 200
```

The switch Statement

- **Switch** statement is **equivalent** to **nested-if statements** **but** with **conditions** that **only test the equality** of a variable or an expression against a scalar or a string
- A **switch** block conditionally **executes one set of statements from** several **choices**. Each **choice** is **covered** by a **case** statement.
- An **evaluated switch_expression** is a **scalar** or **string**.
- An **evaluated case_expression** is a **scalar**, a **string** or a **cell** array **of scalars or strings**.
- The **switch** block **tests each case until one** of the cases is **true**. A **case** is **true** when:
  - For **numbers**, ***eq(case_expression,switch_expression)***.
  - For **strings**, ***strcmp(case_expression,switch_expression)***.
  - For a **cell array** **case_expression**, **at least one of the elements** **of** the **cell array** matches **switch_expression**, as defined above for numbers, strings and objects.
- When a **case** is **true**, MATLAB **executes** the **corresponding statements** and **then exits** the **switch** block.
- The **otherwise block** is **optional** and **executes only** when **no case is true**.

The switch Statement: Syntax

```
switch <switch_expression>
case <case_expression>
<statements>
case <case_expression>
<statements>
...
...
otherwise
<statements>
end
```

The switch Statement: Syntax

- Create a script file sp4.m and type the following code:

```
grade = 'B';
switch(grade)
case 'A'
fprintf('Excellent!\n' );
case 'B'
fprintf('Well done\n' );
case 'C'
fprintf('Well done\n' );
case 'D'
fprintf('You passed\n' );
case 'F'
fprintf('Better try again\n' );
otherwise
fprintf('Invalid grade\n' );
end
```

- Run the file, the result will be:

```
>>sp4.m
    Well done
    Your grade is B
```

Quiz 3

- Rewrite the following code using **nested if-else:**

```
switch num
   case 1
      disp(one')
   case 2
      disp('two')
   case 3
      disp('three');
   case 4
      disp('four')
   otherwise
      disp('other')
end
```

The Nested switch Statement

- **Nested Switch statement**: It is possible to have a **switch** as **part** of the statement sequence of an **outer switch**.

```
switch(ch1)
case 'A'
fprintf('This A is part of outer switch');
switch(ch2)
case 'A'
fprintf('This A is part of inner switch' );
case 'B'
fprintf('This B is part of inner switch' );
end
case 'B'
fprintf('This B is part of outer switch' );
end
```

The switch Statement: Example

- Create a script file sp5.m and type the following code:

```
a= 100; b = 200;
switch(a)
case 100
fprintf('This is part of outer switch %d\n', a );
switch(b)
case 200
fprintf('This is part of inner switch %d\n', a );
end
end
fprintf('Exact value of a is : %d\n', a );
fprintf('Exact value of b is : %d\n', b );
```

- Run the file, the result will be:

```
>>sp5.m
    This is part of outer switch 100
    This is part of inner switch 100
    Exact value of a is : 100
    Exact value of b is : 200
```

Quiz 4

Solve the following problems:

- **Problem 1:**

Using switch statement, display the number of days in each month in 2019.

- **Problem 2:**

Using nested switch, develop a simple calculator (e.g., division by zero is not correct).

Problem1:Solution

```
y = 2016; m = 1;
switch(m)
case {1, 3, 5, 7, 8, 10, 12}  %ORing of cases
disp('31 days');
case {4,6, 9, 11} %ORing of cases
disp('30 days');
case 2
if mod(y,4) == 0
disp('29 days');
else
disp('28 days');
end                    % end of if
otherwise
disp('invalid month');
end  % end of switch
```

Quiz 5

- Convert the previous script to 2 nested switches

- Convert the previous script to nested if

Quiz 5 (a): Solution

```
y = 2016;
for m=1:12
 switch(m)
   case {1, 3, 5, 7, 8, 10, 12}  %ORing of cases
      disp('31 days');
   case {4,6, 9, 11}
      disp('30 days');
   case 2
     % t= mod(y,4);  switch(t)
      switch(mod(y, 4))
          case 0
             disp('29 days');
          otherwise
            disp('28 days');
      end                % end of inner switch
   otherwise
      disp('invalid month');
 end       % end of outer switch
end         % end of for loop
```